

Learning for Control and Decision Making toward Medical Autonomy

by

Qitong Gao

Department of Electrical and Computer Engineering
Duke University

Defense Date: March 22, 2024

Approved:

Miroslav Pajic, Supervisor

Michael M. Zalvanos, Co-Chair

Guillermo Sapiro

Warren M. Grill

Majda Hadziahmetovic

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Electrical and Computer Engineering
in the Graduate School of Duke University

2024

ABSTRACT

Learning for Control and Decision Making toward Medical Autonomy

by

Qitong Gao

Department of Electrical and Computer Engineering
Duke University

Defense Date: March 22, 2024

Approved:

Miroslav Pajic, Supervisor

Michael M. Zalvanos, Co-Chair

Guillermo Sapiro

Warren M. Grill

Majda Hadziahmetovic

An abstract of a dissertation submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in the Department of Electrical and Computer
Engineering
in the Graduate School of Duke University
2024

Copyright © 2024 by
Qitong Gao

All rights reserved except the rights granted by the Creative Commons
Attribution-Noncommercial Licence

Abstract

Artificial intelligence (AI) and deep learning (DL) have recently shown success in domains related to healthcare and its decision-making systems. However, most of the existing methods are developed upon benchmark environments which are often defined with simplistic dynamics and allow access to data that are well-structured, pre-processed, and with substantial amount. It is intractable to leverage such methods to facilitate real-world applications; as *limited access* to the real-world healthcare environments leads to significantly reduced *sample efficiency* during training. Moreover, strict *safety* protocols are usually enforced in practice upon deployment to human participants, while the policy selection criteria weighs human feedback (HF) more than environmental returns; both of which could be intractable to be captured in simulations. From data logging perspectives, *data irregularities* are often encountered in healthcare facilities, *e.g.*, missingness due to malfunctioned devices.

This dissertation aims to introduce AI/ML methods that can overcome limitations including insufficient and imperfect data as well as complying with safety protocols, which are applicable to real-world decision-making processes in healthcare systems, with focuses on (i) sample-efficient reinforcement learning (RL) based frameworks that can synthesize control policies of medical devices to maximize both environmental returns and HF in *offline* manners, with off-policy evaluation (OPE) facilitating the evaluation of RL policies without online interactions, *i.e.*, for improved safety and efficiency upon deployment of the RL policies. (ii) DL-based analyses of multivariate healthcare data constituted by multiple modalities to facilitate clinical decision making systems, by tackling data irregularities and capturing underlying factors important to automated disease diagnoses and prognoses.

To tackle (i), we introduce an algorithmic OPE framework, variational latent branching model (VLBM), which can be integrated into most existing offline RL methods for efficient and safe policy evaluation and selection upon deployment. Specifically, it leverages variational inference to learn the transition function of MDPs by formulating the environmental dynamics as a compact latent space, from which the next states and rewards

are then sampled. It’s efficacy is validated by benchmark environments including Mujoco and Adroit. Then, an OPE for human feedback (OPEHF) method is developed on top of VLBM’s framework to capture the HF participants could have provided once the policies are deployed, further ensuring the satisfaction of human participants who received procedures or medical devices guided by RL agents. At last, we design a *full-stack* offline RL policy optimization pipeline, into which both OPE methods are integrated, toward training control policies of a implantable deep brain stimulation (DBS) device for treatment of Parkinson’s disease (PD), by adjusting the stimulation amplitude in real time. The goal is to reduce the energy used for generating the stimulus, while maintain the same level of treatment (*i.e.*, control) efficacy as continuous DBS (cDBS) (*i.e.*, constantly stimulating with the highest amplitude possible, determined by clinicians). The efficacy is validated a cohort of 5 human participants, where results show that, and OPE components are able to pinpoint high-performing policies among the policy candidates trained using different offline RL algorithms or hyper-parameter sets.

In terms of *(ii)*, we introduce three frameworks to address the following challenges, respectively. *(ii.a)* Data missingness, *e.g.*, due to the non-periodical logging of patient vitals or lab results. *(ii.b)* High dimensionality and multi-modality within healthcare data, given that substantial amount of lab/vital results need to be recorded, and such information could come in the format of images (*e.g.*, CT scan), tabular (*e.g.*, demographic information) etc. The efficacy of each framework is validated by retrospective studies pertaining to the identification, prognoses and treatment of ophthalmic diseases. The results show that our frameworks are able to accurately identify the presence of diseases and automatically design treatment plans.

Contents

Abstract	iv
List of Tables	xii
List of Figures	xiv
Acknowledgements	xvi
1 Introduction	1
1.1 Challenges and Contributions	2
1.1.1 Reliable Off-Policy Evaluation (OPE)	2
1.1.2 OPE to Capture Human Feedback (HF)	5
1.1.3 Applications of Offline Reinforcement Learning (RL) and OPE in Real-World Medical Systems	7
1.1.4 Data Missingness	11
1.1.5 A Real-World Case Study – Automated Identification of Referable Retinal Pathology in Teleophthalmology Setting	13
2 Preliminaries	18
2.1 Variational Inference	18
2.1.1 Reparameterization.	18
2.2 Offline RL	19
2.2.1 Actor-Critic RL	20
2.3 OPE	21
2.3.1 Objective	21
2.3.2 Importance Sampling (IS) and per-decision IS (PDIS)	22
2.3.3 Doubly Robust (DR)	22
2.3.4 DIstributional Correction Estimation (DICE)	23
2.3.5 Fitted Q-Evaluation (FQE)	23
2.3.6 Definition of the OPE Metrics	23
3 VLBM	25

3.1	Related Work	25
3.1.1	Latent Modeling in RL	25
3.1.2	OPE	26
3.2	Problem Formulation	27
3.3	Variational Latent Model	27
3.3.1	Latent Prior $p(z_0)$	27
3.3.2	Variational Encoder for Inference $q_\psi(z_t z_{t-1}, a_{t-1}, s_t)$	27
3.3.3	Generative Decoder for Sampling $p_\phi(z_t, s_t, r_{t-1} z_{t-1}, a_{t-1})$	29
3.4	Recurrent State Alignment	30
3.5	Branching for Generative Decoder	32
3.5.1	Branching Architecture.	33
3.6	Experiments	35
3.6.1	Environmental and Training Setup.	36
3.6.2	Baselines and Evaluation Metrics.	36
3.6.3	Ablation.	37
3.6.4	Results.	37
3.6.5	Branching versus Classic Ensembles.	39
3.6.6	t -SNE Visualization of the Latent Space.	41
4	Off-Policy Evaluation for Human Feedback (OPEHF)	50
4.1	Related Work	50
4.1.1	Reinforcement Learning from Human Feedback (RLHF)	50
4.1.2	Reward Shaping	51
4.2	Problem Formulation	51
4.3	Reconstruction of Immediate Human Rewards (IHRs) for OPEHF	52
4.3.1	Reconstruction of IHRs.	54
4.4	Reconstruction of IHRs over Latent Representations (RILR) for OPEHF	55

4.4.1	Trajectory Inference (Encoding).	56
4.4.2	Trajectory Generation (Decoding).	56
4.4.3	Derivation of the evidence lower bound (ELBO) Above	58
4.4.4	Regularizing the Reconstruction of IHRs.	58
4.4.5	Overall Objective of RILR for OPEHF.	59
4.4.6	Move from RILR to OPEHF.	59
4.5	Real-World Experiments with Human Participants	60
4.5.1	Baselines and Ablations.	60
4.5.2	Adaptive Neurostimulation: Deep Brain Stimulation	61
4.5.3	Intelligent Tutoring	65
4.6	A Challenging Simulation Environment: Visual Q&A Dialogue	69
4.6.1	Overall Setup.	70
4.6.2	MDP Formulation.	71
4.6.3	Offline Trajectories and Target Policies.	71
4.6.4	Results and Discussion.	72
5	A Real-World Case Study in Healthcare: Deep Brain Stimulation (DBS)	73
5.1	Related Work and Motivation	73
5.1.1	The Need for Closed-Loop DBS	74
5.2	DBS Setup Used in Clinical Trials	76
5.3	Offline RL Design of DBS Controllers	78
5.3.1	Modeling the Basal Ganglia (BG) as an Markov Decision Process (MDP)	79
5.3.2	Policy Distillation	81
5.4	OPE of DBS Controllers Including Patient Feedback and Tremor Data	82
5.4.1	Algorithm to Train Deep Latent Sequential Model (DLSM)	87
5.5	Clinical Evaluations	88
5.5.1	Therapy Efficacy and Energy-Efficiency of the RL Control Policies	90

5.5.2	Evaluation of the OPE Methodology	93
5.5.3	Limitations	98
6	Data Missingness	99
6.1	Related Work	99
6.1.1	Missing Data Imputation.	99
6.1.2	Attention.	100
6.2	Gradient Importance Learning (GIL)	101
6.2.1	Problem Formulation	102
6.2.2	Gradient Importance	102
6.2.3	RL to Generate Importance Matrix	104
6.2.4	State Space \mathcal{S}	105
6.2.5	Action Space \mathcal{A}	105
6.2.6	Transitions $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$	105
6.2.7	Reward Function R	106
6.2.8	Extensions of the Framework	107
6.3	Experiments	108
6.3.1	Variants of GIL and Baselines	109
6.3.2	MIMIC-III	109
6.3.3	Ophthalmic Data	111
6.3.4	MNIST	112
6.3.5	Correlation between Imputation and Prediction Performance	114
7	A Real-World Case Study – Automated Identification of Referable Retinal Pathology in Teleophthalmology Setting	115
7.1	Related Work	115
7.2	Materials and Methods	116
7.2.1	Retinal Imaging	116
7.2.2	Dataset Formulation	116

7.2.3	Study Design and Outcomes Measures	118
7.3	Results	124
7.4	Discussion	128
8	Conclusion	129
8.1	Summary	129
8.2	Future Work	131
8.2.1	Meta-Learning-Based Reward Shaping to Improve the Robustness of OPE and OPEHF	131
8.2.2	A More Comprehensive Automation of DBS Therapy	132
8.2.3	Capturing Temporal Correlations within Healthcare Data for Disease Progression Analyses	132
Appendix A	Proof of Proposition 1	133
Appendix B	Gradient Importance Learning	135
B.1	Extending GIL to LSTMs	135
B.2	Description of Algorithm 4	140
B.3	Importance vs Attentions	141
B.4	Experimental Details and Additional Experiments	143
B.4.1	MIMIC-III	144
B.4.2	Ophthalmic Data	147
B.4.3	MNIST	149
B.4.4	Physionet	150
B.5	Proof for Proposition 2	152
B.6	Additional Ablation Study	155
B.6.1	Additional rationales for using RL to obtain gradient importance in GIL.	156
B.6.2	More on importance versus attentions.	156
Appendix C	Automated Identification of Referable Retinal Pathology in Teleoph- thalmology Setting	157

C.1 Label Consensus Mechanism (LCM)	157
C.2 Dataset Augmentation and Pre-Processing	157
C.3 Mathematical Illustration of the AGD Algorithm and Transfer Learning . .	158
C.3.1 The AGD Algorithm	158
C.3.2 Transfer Learning	159
C.4 Intuition of Designing Baseline A and B and Their Implementation Details .	159
Bibliography	161
Biography	182

List of Tables

3.1	Rank correlation for Gym-Mujoco tasks.	42
3.2	Rank correlation for Adriot tasks.	43
3.3	Regret@1 for Gym-Mujoco tasks.	44
3.4	Regret@1 for Adroit tasks.	45
3.5	MAE for Gym-Mujoco tasks.	46
3.6	MAE for Adriot tasks.	47
3.7	Summary of the Gym-Mujoco environments and datasets used to train VLBM and baselines.	48
3.8	Summary of the Adroit environments and datasets used to train VLBM and baselines.	49
4.1	Correlations between the <i>environmental</i> and <i>human</i> returns of the 6 target DBS policies associated with each PD patient.	63
4.2	Raw results of the adaptive neurostimulation experiment from each patient (Patient #0).	65
4.3	Raw results of the adaptive neurostimulation experiment from each patient (Patient #1).	66
4.4	Raw results of the adaptive neurostimulation experiment from each patient (Patient #2).	67
4.5	Raw results of the adaptive neurostimulation experiment from each patient (Patient #3).	68
4.6	Results from the intelligent tutoring experiment.	68
4.7	Correlations between the <i>environmental</i> and <i>human</i> returns from data collected over each academic year.	68
4.8	Results from the visual Q&A environment.	69
4.9	Environmental and synthetic human returns of the target policies.	72
5.1	Characteristic of each participant.	90
5.2	Overall time spent toward testing each type of controller in clinical trials.	91
5.3	Computation time of the original RL versus the distilled RL policy.	91

5.4	Overall battery runtime of the DBS system when the RL, distilled RL or random controllers were used.	94
5.5	Mean and standard error of the trajectory lengths.	96
6.1	Accuracy and AUC obtained from the MIMIC-III dataset.	110
6.2	Average Accuracy and AUC obtained from the Ophthalmic dataset over 3 different random masks. Subscripts are standard deviations.	111
6.3	Average Accuracy reported for the MNIST dataset over different missing rates.	113
6.4	Correlation between imputation MSE and prediction accuracy for different MRs.	114
7.1	Distribution of the original dataset, augmented training set, and testing dataset by modality	118
7.2	Distribution of the original dataset, augmented training Set, and testing dataset by eye	118
7.3	Performance comparison between our approach and baselines.	124
7.4	Performance comparison between our approach and baselines over only interpretable images.	126
B.1	Accuracy for GIL-D on the MCAR version of MNIST dataset. Standard deviations are in subscripts ($\times 10^{-3}$).	150
B.2	Accuracy, AUC and average precision (AP) obtained from the Physionet dataset	150
B.3	Performance of the ablation model over the ophthalmic and MNIST datasets.	155
C.1	Label consensus between outcomes of two modalities.	158

List of Figures

1.1	MNIST digits imputed by state-of-the-art imputation methods.	12
3.1	Architecture of variational latent model (VLM) we consider.	28
3.2	Recurrent state alignment (RSA) and single-step forward pass of the variational latent branching model (VLBM).	31
3.3	Mean rank correlation, regret@1 and MAE over all the 32 Gym-Mujoco and Adroit tasks, showing VLBM achieves state-of-the-art performance overall.	35
3.4	Mean rank correlation, regret@1 and MAE over all datasets, for each Mujoco environment.	38
3.5	Mean rank correlation, regret@1 and MAE over all datasets, for each Adroit environment.	38
3.6	Distribution of all branching weights, w_b 's, over all VLBM's trained on the 32 tasks.	38
3.7	Correlation between the estimated (y-axis) and true returns (x-axis), across different model-based OPE methods and environments.	40
3.8	t -SNE visualization.	41
4.1	Methodology of off-policy evaluation for human feedback (OPEHF).	55
4.2	Setup of the neurostimulation experiments.	61
4.3	Results from the adaptive neurostimulation experiment.	62
4.4	t -SNE visualizing the VLM-H encodings.	64
5.1	An implantable deep brain stimulation (DBS) device.	74
5.2	The overall architecture of the Summit RC+S DBS system.	75
5.3	Timeline for training RL-based DBS controllers in clinical studies.	82
5.4	Architecture of the new deep latent sequential model (DLSM).	83
5.5	Quality of control (QoC) results from all clinical trials across participants.	89
5.6	DLSM results.	93
5.7	Inspection of DLSM-reconstructed rewards.	96
5.8	Results from DLSM's Scalability study. The means and standard errors of rank correlations, regrets and MAEs were averaged over the entire cohort.	97

6.1	Overview of the GIL framework.	101
6.2	t -SNE visualization of the feature space learned by (a) GIL, (b) MIWAE and (c) GAIN on the MNIST dataset with 90% missing rate.	112
7.1	Methodology for diabetic retinopathy detection.	117
7.2	Architecture of the network for detection of diabetic retinopathy.	121
7.3	ACC-FNR and ROC Curves.	125
B.1	Graphical depiction of the 2-hour observation window and 4-hour early prediction window (EPW) considered in this case study.	144
B.2	Histogram of the sequences lengths of the MIMIC-III dataset.	145
B.3	Example of a 3D OCT volume scan and a 2D OCT image slice from the volume scan.	147
B.4	Diagram showing the pipeline of this experiment.	148
C.1	Overview of Baseline A.	159
C.2	Overview of Baseline B.	160

Acknowledgements

I would like to extend my heartfelt thanks to everyone who has supported me on this journey of completing the Ph.D. program. Foremost, I express my deepest gratitude to Professor Miroslav Pajic, my advisor, for his invaluable guidance, unwavering encouragement, and steadfast support throughout this endeavor. His profound expertise and mentorship have played a pivotal role in my successful completion of this dissertation. Furthermore, his support extends beyond academic guidance, as I have also greatly benefited from his invaluable advice regarding professional developments, career paths, personal support, etc.

I am also extremely grateful for the guidance I have received from other esteemed faculty members at Duke University, including Professors Dennis A. Turner, Warren M. Grill, Ricardo Henao, Michael M. Zavlanos, Guillermo Sapiro, and Majda Hadziahmetovic. Additionally, my sincere appreciation goes out to my research collaborators at Duke, particularly Dr. Steve Schmidt, Afsana Chowdhury, Juncheng Dong, David Kuo, Vuk Lesi, Amir Khazraei, and Hao-Lun Hsu, for their incredible feedback and support in enhancing my work.

Special thanks are also due to Jennifer Peters, Katherine Genty, and Rocio Rodriguez for their diligent involvement in clinical sessions with patients undergoing reinforcement learning-based deep brain stimulation (DBS) therapy, overseeing testing procedures, and collecting data and feedback. I am equally thankful to Joshua Amason, Amanda Del Risco, Terry Lee, Praruj Pant, Jay Rathinavelu, and Aditya Kotla, exceptional medical students, fellows, and residents at Duke University School of Medicine, whose assistance was invaluable in collecting, organizing, and screening data for my research on automated ophthalmic disease diagnosis and prognosis.

Furthermore, I extend my sincere appreciation to Professor Min Chi of North Carolina State University, Raleigh, NC, USA, for her remarkable support, feedback, and guidance as both an exceptional external mentor and collaborator.

I also wish to express my appreciation to the faculty and staff of Duke University's Electrical and Computer Engineering Department for their invaluable assistance and support

with administrative tasks. I am particularly grateful to the PhD program coordinators, Angela Chanh and Kevyn Light, for their exceptional dedication.

I am deeply honored to have had the opportunity to collaborate with Professors Neil Gong and Helen (Hai) Li during my two teaching assistant positions for their respective courses. Through these experiences, I gained invaluable insights into mentorship, teaching methodologies, and much more.

I am also incredibly fortunate to have received unwavering support from my family throughout this journey. My parents have generously gone above and beyond to ensure I could maintain a comfortable standard of living, and have been instrumental in guiding me through significant life decisions along the way. Additionally, my wife, Ge Gao, not only delicately manages our personal life but also offers invaluable academic involvement and support, drawing from her expertise in cross-functional domains. Our two beloved furry companions, Winston and Teddy, have brought phenomenal vibrancy and joy to our daily lives.

Last but not least, I extend my gratitude to the funding agencies, including the National Science Foundation (NSF), National Institutes of Health (NIH), Air Force Office of Scientific Research (AFOSR), and Office of Naval Research (ONR), for their financial support in covering my tuition, stipend, and conference travel expenses. The specific grants include NSF CNS-1837499 award and the National AI Institute for Edge Computing Leveraging Next Generation Wireless Networks, NSF Grant CNS-2112562, CNS-1652544, CNS-1837499, AFOSR under award number FA9550-19-1-0169, ONR under agreements N00014-17-1-2504 and N00014-20-1-2745, NIH UH3 NS103468, NIH R37 NS040894, NIH/NIDDK R01-DK123062, NIH/NINDS 1R61NS120246. I'm grateful that Medtronic PLC, and Rune Labs, respectively provided the Investigational Summit RC+S systems, and Apple Watches, used in DBS therapy. Additionally, I would like to express appreciation to the organizers of the conferences I attended, as well as the anonymous reviewers, for providing a platform for presenting my work and offering eye-opening opportunities to engage with outstanding scholars around the globe.

1. Introduction

AI and ML have recently shown success in domains related to medical and robotics systems, such as RL for automated control of robots and autonomous vehicles (Gu et al., 2017; Mnih et al., 2015), as well as deep learning models that facilitate comprehensive analysis of multivariate and multi-modal healthcare data (Bashir & Wei, 2018; Fortuin et al., 2020).

However, most of the existing methods are developed upon benchmark environments which are often defined with simplistic dynamics and allow access to data that are well-structured, pre-processed, and with substantial amount. It is worth noting that data obtained from environments in the real-world rarely suffice these conditions. For example, Mujoco (Todorov et al., 2012) is a simulator that can model sophisticated dynamics and emulate robots' behavior in response to different control inputs, which is usually used as a test base to evaluate performance of RL methods (Lillicrap et al., 2016). It is capable of generating trajectories $4\times$ faster than real robots (Körber et al., 2021) and allow access to oracle information (*i.e.*, ground-truth state over the dynamics); hence providing data, with more-than-realistic amount and quality, to train and evaluate the algorithms. As a result, it may not be straightforward to adopt such methods to systems in real-world healthcare systems, given *limited access* to the environments (*i.e.*, *insufficient data*) and *imperfect data* could usually be obtained (*e.g.*, missing packets due to connection issues). Moreover, strict *safety* protocols are usually be enforced in healthcare practice, and the fact that the HF provided by human participants usually weigh more than generic environmental returns – for example, new controllers of medical devices need to be thoroughly examined by healthcare professionals before deploying in clinical trials (Parvinian et al., 2018) – making it intractable for such information to be modeled and captured in simulations.

On the other hand, ML models developed for retrospective study of healthcare information could as well be hindered by limitations highlighted above. Specifically, the UCI/UEA repositories (Bagnall et al., 2017) provide standardized datasets, which are commonly used to test the performance of multivariate sequential models (Fortuin et al., 2020). However,

it does not model *data missingness* as encountered in healthcare facilities (*e.g.*, caused by malfunctioned devices or missing responses from patients), nor captures patient-specific correlations as found in electronic health records (EHRs) (Cao et al., 2018), such as disease progression over time; thus, the ML methods achieving compelling results over such datasets are not necessarily expected to attain similar performance using data collected in clinics.

As a result, this dissertation aims to develop AI/ML methods that can overcome limitations including insufficient and imperfect data as well as complying with safety protocols, which are applicable to real-world healthcare systems; with special focuses on *(i)* sample-efficient RL based frameworks that can synthesize control policies of medical devices to maximize both environmental returns and HF in *offline* manners, with OPE facilitating the evaluation of RL policies without online interactions, *i.e.*, for improved safety and efficiency upon deployment of the RL policies, and *(ii)* DL-based analyses of multivariate healthcare data constituted by multiple modalities to facilitate clinical decision making systems, by tackling data irregularities and capturing underlying factors important to automated disease diagnoses and prognoses.

1.1 Challenges and Contributions

In this section, we first introduce the specific challenges to be tackled in each chapter, as well as the main contributions, which are oriented toward attaining the overall goals introduced above.

1.1.1 Reliable Off-Policy Evaluation (OPE)

Offline RL has demonstrated significant potential in controlling complex systems (G. Gao, Gao, Yang, Pajic, & Chi, 2022a; Q. Gao, Wang, et al., 2022b; Gu et al., 2017; Mnih et al., 2015). However, it becomes challenging to determine which specific policy can possibly lead to the best outcome once get deployed in the real world system, out of a set of candidate policies trained following various hyper-parameter sets or offline RL algorithms. This is especially important in healthcare systems, as the high bar for ensuring

safety testing (Parvinian et al., 2018), as well as the limited interactions allowed between the policy and environment due to the availability of human participants.

OPE allows for evaluation of RL policies without online interactions, which is crucial in ensuring the safety and efficiency of the online deployment and testing processes. It is applicable to many domains where on-policy data collection could be prevented due to efficiency and safety concerns, *e.g.*, healthcare (G. Gao, Gao, Yang, Pajic, & Chi, 2022b; G. Gao, Gao, Yang, et al., 2023; Q. Gao, Wang, et al., 2022a; S. Tang & Wiens, 2021; X. Yang et al., 2023b), recommendation systems (L. Li et al., 2011; R. Mehrotra et al., 2018), education (G. Gao et al., 2024; Mandel et al., 2014), social science (Segal et al., 2018) and optimal control (Q. Gao, Hajinezhad, Zhang, Kantaros, & Zavlanos, 2019; Q. Gao, Naumann, et al., 2020; Q. Gao, Pajic, & Zavlanos, 2020; Silver et al., 2016; Vinyals et al., 2019). Recently, as reported in the deep OPE (DOPE) benchmark (Fu, Norouzi, et al., 2020), model-based OPE methods, leveraging feed-forward (Fu, Norouzi, et al., 2020) and auto-regressive (AR) (M. R. Zhang et al., 2020) architectures, have shown promising results toward estimating the return of target policies over other existing OPE methods including importance sampling (IS), doubly robust (DR) etc., by fitting transition functions of MDPs. However, model-based OPE methods remain challenged as they can only be trained using offline trajectory data, which often offers limited coverage of state and action space. Thus, they may perform sub-optimally on tasks where parts of the dynamics are not fully explored (Fu, Norouzi, et al., 2020). Moreover, different initialization of the model weights could lead to varied evaluation performance (Hanin & Rolnick, 2018; Rossi et al., 2019), reducing the robustness of downstream OPE estimations. Some approaches in RL policy optimization literature use latent models trained to capture a compact space from which the dynamics underlying MDPs are extrapolated; this allows learning expressive representations over the state-action space. However, such approaches usually require *online* data collections as the focus is on quickly navigating to the high-reward regions (Rybkin et al., 2021), as well as on improving coverage of the explored state and action space (Hafner, Lillicrap, et al., 2020; Hafner et al., 2019; M. Zhang et al., 2019) or sample efficiency (Lee

et al., 2020).

1.1.1.1 Contributions

In Chapter 3, we propose the variational latent branching model (VLBM), aiming to learn a compact and disentangled latent representation space from offline trajectories, which can better capture the dynamics underlying environments. VLBM enriches the architectures and optimization objectives for existing latent modeling frameworks, allowing them to learn from a *fixed* set of *offline* trajectories. Specifically, VLBM considers learning variational (encoding) and generative (decoding) distributions, both represented by long short-term memories (LSTMs) with reparameterization (Kingma & Welling, 2013), to encode the state-action pairs and enforce the transitions over the latent space, respectively. To train such models, we optimize over the evidence lower bound (ELBO) jointly with a *recurrent state alignment* (RSA) term defined over the LSTM states; this ensures that the information encoded into the latent space can be effectively teased out by the decoder. Then, we introduce the *branching architecture* that allows for multiple decoders to jointly infer from the latent space and reach a consensus, from which the next state and reward are generated. This is designed to mitigate the side effects of model-based methods where different weight initializations could lead to varied performance (Fu, Norouzi, et al., 2020; Hanin & Rolnick, 2018; Rossi et al., 2019).

The key contributions of this chapter are summarized as follows: (i) to the best of our knowledge, the VLBM is the first method that leverages variational inference for OPE. It can be trained using offline trajectories and capture environment dynamics over latent space, as well as estimate returns of target (evaluation) policies accurately. (ii) The design of the RSA loss term and branching architecture can effectively smooth the information flow in the latent space shared by the encoder and decoder, increasing the expressiveness and robustness of the model. This is empirically shown in experiments by comparing with ablation baselines. (iii) Our method generally outperforms existing model-based and model-free OPE methods, for evaluating policies over various D4RL environments (Fu, Kumar, et

al., 2020). Specifically, we follow guidelines provided by the DOPE benchmark (Fu, Norouzi, et al., 2020), which contains challenging OPE tasks where the training trajectories include varying levels of coverage of the state-action space, and target policies are designed toward resulting in state-action distributions different from the ones induced by behavioral policies.

1.1.2 OPE to Capture Human Feedback (HF)

In real-world healthcare systems, feedback provided by participants, *i.e.*, the HF, are critical for evaluating the efficacy of the control policies, and can be used to determine if the policies can be deployed for long-term use (Parvinian et al., 2018). However, the majority of existing OPE methods focus on evaluating the policies' performance defined over the *environmental* reward functions which are mainly designed for use in policy optimization (training). However, as an increasing number of offline RL frameworks are developed for human-involved systems (Abeyruwan et al., 2023; G. Gao, Gao, Yang, Pajic, & Chi, 2022b; E. Liu et al., 2022; Mandel et al., 2017; Ruan et al., 2023), existing OPE methods lack the ability to estimate how human users would evaluate the policies, *e.g.*, ratings provided by patients (on a scale of 1-10) over the procedure facilitated by automated surgical robots; as human feedback (HF) can be noisy and conditioned over various confounders that could be difficult to be captured explicitly (Chesnaye et al., 2022; Lis et al., 2015; Namkoong et al., 2020). For example, patient satisfaction over a specific diabetes therapy may vary across the cohort, depending on many subjective factors, such as personal preferences and activity level of the day, while participating in the therapy, in addition to the physiological signals (*e.g.*, blood sugar level, body weight) that are more commonly used as the sources for determining environmental rewards toward policy optimization (Q. Gao, Hajinezhad, Zhang, Kantaros, & Zavlanos, 2019; Q. Gao, Pajic, & Zavlanos, 2020; Jia et al., n.d.; Tejedor et al., 2020). Moreover, the environmental rewards are sometimes discrete to ensure optimality of the learned policies (Sutton & Barto, 2018), which further reduces its correlation against HF signals.

1.1.2.1 Contributions

In Chapter 4, we introduce the OPE for human feedback (OPEHF) framework that revives existing OPE approaches in the context of evaluating HF from offline data. Specifically, we consider the challenging scenario where the HF signal is only provided at the end of each episode – *i.e.*, no per-step HF signals, referred to as *immediate human rewards* (IHRs) below, are provided – benchmarking the common real-world situations where the participants are allowed to rate the procedures only at the end of the study. The goal is set to estimate the end-of-episode HF signals, also referred to as *human returns*, over the target (evaluation) policies, using a fixed set of offline trajectories collected over some behavioral policies. To facilitate OPEHF, we introduce an approach that first maps the human return back to the sequence of IHRs, over the horizon, for each trajectory. Specifically, this follows from optimizing over an objective that consists of a necessary condition where the cumulative discounted sum of IHRs should equal the human return, as well as a regularization term that limits the discrepancy of the reconstructed IHRs over state-action pairs that are determined similar over a latent representation space into which environmental transitions and rewards are encoded. At last, this allows for the use of any existing OPE methods to process the offline trajectories with reconstructed IHRs and estimate human returns under target policies.

The key contributions of this chapter are tri-fold. (i) We introduce a novel OPEHF framework that revives existing OPE methods toward accurately estimating highly sparse HF signals (provided only at the end of each episode) from offline trajectories, through IHRs reconstruction. (ii) Our approach does not require the environmental rewards and the HF signals to be strongly correlated, benefiting from the design where both signals are encoded to a latent space regularizing the objective for reconstructions of IHRs, which is justified empirically over real-world experiments. (iii) Two *real-world experiments*, *i.e.*, adaptive *in-vivo* neurostimulation for the treatment of Parkinson’s disease and intelligent tutoring for computer science students in colleges, as well as one simulation environment (*i.e.*, visual

Q&A), facilitated the thorough evaluation of our approach; various degrees of correlations between the environment rewards and HF signals existed across the environments, as well as the varied coverage of the state-action space provided by offline data over sub-optimal behavioral policies, imposing different levels of challenges for OPEHF.

1.1.3 Applications of Offline Reinforcement Learning (RL) and OPE in Real-World Medical Systems

Presently, approximately 1.05 million people in the United States have Parkinson’s disease (PD) (Marras et al., 2018). Deep brain stimulation (DBS) has emerged as an effective treatment to alleviate PD motor symptoms like tremor and bradykinesia (Benabid, 2003; Deuschl et al., 2006; Follett et al., 2010; Hsu et al., 2024; Okun, 2012a; Schmidt et al., 2024). The DBS system consists of electrodes placed in the basal ganglia (BG) region of the brain and a pulse generator implanted in the chest, generating short electrical pulses. Currently only continuous DBS (cDBS), that is DBS with fixed stimulation parameters, is approved by the FDA. In this approach stimulation parameters are determined through trial-and-error (Pineau et al., 2009). However, such stimulation methods consume a significant amount of energy, leading to a reduction in the device’s battery lifespan. Moreover, treatment needs vary over the course of hours and even intermittent over-stimulation can result in adverse effects like dyskinesia and speech impairment (Beudel & Brown, 2016). As a result, there is considerable interest among clinicians and patients in developing closed-loop DBS controllers that can respond to the patient’s activity and state (*i.e.*, context). These advances aim to enhance the overall effectiveness and safety of DBS treatments.

Reinforcement learning (RL) has demonstrated significant potential in controlling complex systems (G. Gao, Gao, Yang, Pajic, & Chi, 2022a; Q. Gao, Wang, et al., 2022b; Gu et al., 2017; Mnih et al., 2015). Various RL-based approaches have been proposed to facilitate closed-loop DBS (Q. Gao, Naumann, et al., 2020; Guez et al., 2008; Nagaraj et al., 2017; Pineau et al., 2009). In particular, some of these approaches (Guez et al., 2008; Nagaraj et al., 2017; Pineau et al., 2009) utilize EEG and LFP as the state space of the RL environment and employ temporal difference learning or fitted Q-iteration to design

control policies that adapt stimulation amplitudes/frequencies to conserve energy. Another approach presented in (Q. Gao, Naumann, et al., 2020) employs a deep actor-critic method, allowing the temporal pattern of stimuli to be adjusted over time. This benefits from the use of deep RL techniques capable of searching in larger state and action spaces. Although these methods achieve satisfactory control of efficacy and energy savings in simulation environments using computational BG models (Jovanov et al., 2018; So et al., 2012), it is essential to note that *they have not been evaluated in real-world scenarios*. The crucial difference lies in the availability of training data. In simulations, unlimited training data can be obtained from the computational models, which is intractable in real-world cases; since the device programming occurs in clinics, and the patient’s participation is sparse over time. As a result, the real-world evaluation poses unique challenges and considerations that need to be addressed for successful application of RL-based closed-loop DBS controllers in practical clinical settings.

One limitation of using deep RL methods directly for real-time DBS control is the computational complexity involved in evaluating RL policies *in vivo*. RL policies are often represented by deep neural networks (DNNs), which can require millions of multiplications in a single forward pass. This poses a challenge for resource-constrained implantable devices, as they may not be able to facilitate such extensive computations. Therefore, **challenge (ii)** in closed-loop DBS is to design controllers that can be trained with limited samples and executed without significant computing resources. Additionally, when deploying RL controllers directly on patients, safety and control efficacy must be thoroughly evaluated before each test condition starts. Unlike simulated or robotic environments, where most RL policies can be directly deployed for performance evaluation, controllers used on patients need to undergo rigorous evaluation due to safety concerns (Parvinian et al., 2018). Hence, **challenge (iii)** in enabling closed-loop DBS in patients is to provide accurate estimations of the expected performance of the controllers in a prospective manner.

1.1.3.1 Contributions

To address the challenges above, Chapter 5 introduces an offline RL framework for closed-loop DBS that is both *effective (in terms of therapy)* and *energy-efficient*. The framework models the BG regions of the brain as a Markov decision process (MDP), capturing the neuronal activity in response to stimuli. Then, the deep actor-critic algorithm (Lillicrap et al., 2016) is adapted to adjust the stimulation amplitude based on changes in LFPs. Data from five patients implanted with Medtronic Summit RC+S DBS devices (Stanslaski et al., 2018) are collected and used for training and analyses. Specifically, to address challenge (i), the offline RL approach leverages historically collected trajectories through experience replay, enabling optimization of the control policy under varying patient conditions, including the level of activities, medications etc. of the patients before and during the trials. Additionally, experience collected from non-RL controllers can also be incorporated to update the policy. For instance, in the early stages of learning, a controller that generates uniformly random amplitudes within a specific range can be used. This random exploration of the state and action space helps the RL agent gather information about the system dynamics and patient response. As more data are collected, the RL controller refines its policy and gradually improves its performance based on the observed outcomes from both random exploration and historical experience. To address challenge (ii), we introduce model distillation techniques (Hinton et al., n.d.) specifically for DBS applications, reducing the size of the DNNs used to represent RL policies and ensuring efficient execution within the required control rates.

To overcome challenge (iii), we propose a model-based offline policy evaluation (OPE) method. This method captures the underlying dynamics of the considered MDP and allows estimation of the expected returns of the control policy without deploying the policy directly to the patient. In each DBS session, the control efficacy is evaluated from various sources, including LFP biomarkers recorded from the implantable DBS device, patient responses to bradykinesia tests, patient-reported satisfaction levels, and overall tremor severity quan-

tified from accelerometry data collected by external wearable devices (i.e., smartwatches). However, the latter three efficacy metrics are only evaluated once at the end of each trial, making them sparsely available compared to the LFPs that can be sensed in each time step. As a result, it is intractable to leverage existing OPE methods like importance sampling (IS) (G. Gao, Ju, et al., 2023; Precup, 2000), distributional correction estimations (DICE)(Nachum et al., 2019), and model-based OPE (Q. Gao, Schmidt, et al., 2022), as they do not explicitly capture or model such end-of-session rewards. Our proposed OPE method addresses this limitation by utilizing a specially designed architecture and training objective that can capture such end-of-session reward behaviors. This enables the method to estimate effectively the expected performance of the control policy using the sparsely available efficacy metrics, resulting in improved performance compared to existing OPE methods. The effectiveness of our OPE method is demonstrated in clinical experiments, demonstrating its superiority in evaluating the control efficacy of closed-loop DBS controllers for practical application on patients.

The contributions of this chapter can be summarized as follows. (i) This work presents the first *‘full-stack’* offline RL methodology that enables both optimization and evaluation of RL-based DBS control policies using historical data. (ii) A novel RL-based DBS controller is developed and its performance is validated through clinical trials with PD patients. The controller demonstrates reduced energy consumption while maintaining non-inferior control efficacy compared to traditional cDBS methods. Notably, this is **the first effective closed-loop DBS control approach that goes beyond simple ON/OFF switching or proportional scaling, and it has been extensively tested in clinical settings on real patients.** (iii) The proposed OPE method effectively captures the end-of-session rewards, leading to accurate estimations of control efficacy using data collected during clinic visits. This enables proactive testing and prioritization of policies, optimizing performance within the limited amount of testing time available.

1.1.4 Data Missingness

Components of the data could be missing for various reasons, for instance, missing responses from participants, data loss, and restricted access issues. This phenomenon is prevalent in the healthcare domain, mostly in the context of electronic health records (EHRs), which are structured as patient-specific irregular timelines with attributes, *e.g.*, diagnosis, laboratory tests, vitals, thus resulting in high missingness across patients for any arbitrary time point. Such missingness introduces difficulties when developing models and performing inference in real-world applications such as Kam and Kim, 2017; Scherpf et al., 2019. Existing works tackle this problem by either proposing imputation algorithms (IAs) to explicitly produce estimates of the missing data, or by imposing imputation objectives during inference, *e.g.*, withholding observed values as being the ground-truth and learning to impute them. However, some of these either require additional modeling assumptions for the underlying distributions (Bashir & Wei, 2018; Fortuin et al., 2020), or formatting of the data (Schnabel et al., 2016; H.-F. Yu et al., 2016). Other applications depend on domain knowledge for pre-processing and modeling such as Kam and Kim, 2017; X. Yang et al., 2018, or introduce additional information-based losses (Cao et al., 2018; Lipton et al., 2016), which are usually intractable with real-world data. Moreover, generative methods (Mattei & Frellsen, 2019; Yoon et al., 2018) could result in imputations with high variation (*e.g.*, low confidence of the output distribution), when data have high missingness rates or small sample sizes. Figure 1.1 illustrates imputations generated by two state-of-the-art deep generative models on the MNIST dataset with 50%, 70% and 90% of the pixels set as missing (*i.e.*, masked out). It is observed that both approaches suffer from inaccurate reconstruction of the original digits as the missingness rate increases, which is manifested as some of the imputed images not being recognizable as digits. Importantly, the error introduced by the imputation process can be further propagated into downstream inference and prediction stages.

Imputing the missing data (either explicitly or implicitly) is, in most cases, not nec-

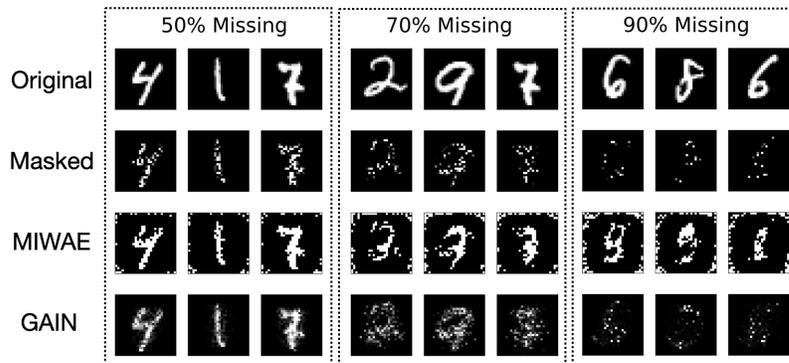


FIGURE 1.1: MNIST digits imputed by state-of-the-art imputation methods MIWAE (Mattei & Frellsen, 2019), GAIN (Yoon et al., 2018), illustrating the challenging task of missing data imputation.

essary, more so, considering that sometimes *where* and *when* the data are missing can be intrinsically informative. Consider a scenario in which two patients, A and B, admitted to the intensive care unit (ICU) suffer from bacterial and viral infections, respectively, and assume that the healthcare provider monitors the status of the patients by ordering two (slightly) different blood tests periodically, namely, a culture test/panel specific to bacterial infections and a RT-PCR test/panel specific to viral infections. Hence, patient A is likely to have much fewer orders (and results) for viral tests. In both cases, the *missingness patterns* are indicative of the underlying condition of both patients. Moreover, such patterns are more commonly found in incomplete data caused by missing at random (MAR) or missing not at random (MNAR) mechanisms, which usually introduce additional difficulties for inference (R. J. Little & Rubin, 2019). Inspired by this, we propose the gradient importance learning (GIL) method, which facilitates an *imputation-free* learning framework for incomplete data by simultaneously leveraging the observed data and the information underlying missingness patterns.

1.1.4.1 Contributions

In Chapter 6, we propose to use an *importance* matrix to weight the gradients that are used to update the model parameters in the back-propagation process, *after* the computational graph of the model is settled, which allows the model to exploit the information

underlying missingness without imputation. However, these gradients cannot be tracked by automated differentiation tools such as Tensorflow (Abadi et al., 2016). So motivated, we propose to generate the importance matrix using a reinforcement learning (RL) policy *on-the-fly*. This is done by conditioning on the status of training procedure characterized by the model parameters, inputs and model performance at the current training step. Concurrently, RL algorithms are used to update the policy by first modeling the back-propagation process as an RL environment, or a Markov decision process (MDP). Then, by interacting with the environment, the RL algorithm can learn the optimal policy so the importance matrix can aid the training of the prediction models to attain desirable performance. Moreover, we also show that our framework can be augmented with feature learning techniques, *e.g.*, contrastive learning as in T. Chen et al., 2020, which can further improve the performance of the proposed imputation-free models.

The technical contributions of Chapter 6 can be summarized as follows: (i) to the best of our knowledge, this is the first work that trains deep learning (DL) models to perform accurate *imputation-free* predictions with missing inputs. This allows models to effectively handle high missing rates, while significantly reducing the prediction error compared to existing imputation-prediction frameworks. (ii) Unlike existing approaches that require additional modeling assumptions or that rely on methods (or pre-existing knowledge) that intrinsically have advantages over specific types of data, our method *does not require any* assumptions or domain expertise. Consequently, it can consistently achieve top performance over a variety of data and under different conditions. (iii) The proposed framework also facilitates feature learning from incomplete data, as the importance matrix can guide the hidden layers of the model to capture information underlying missingness patterns during training, which results in more expressive features.

1.1.5 A Real-World Case Study – Automated Identification of Referable Retinal Pathology in Teleophthalmology Setting

COVID19 pandemic has brought teleophthalmology into the spotlight and highlighted the need for a well-run remote retinal imaging model that, besides good image quality,

provides an accurate and fast image interpretation.

Several groups have attempted to address this issue by proposing automated solutions that are either human-in-the-loop systems or operated semi-autonomously. However, developing fully automated approach was challenging as a significant percentage of uninterpretable images were present in training and testing datasets (Bennett, 2009; Christopher et al., 2018; G. Gao & Chi, 2023; G. Gao, Khoshnevisan, & Chi, 2022; Gargeya & Leng, 2017; Gulshan et al., 2016; Kermany, Goldbaum, Cai, Valentim, & et al., 2018; S. Liu et al., 2009; W. Lu et al., 2018; Medeiros et al., 2019; Mookiah et al., 2013; Muhammad et al., 2017; Raman et al., 2019; Shibata et al., 2018; Singh & Gorantla, 2020; Treder et al., 2018a, 2018b; Usher et al., 2004; Vaghefi et al., 2020; W. Wang et al., 2019a; Y. Wang et al., 2016; Winder et al., 2009; Z. Xu et al., 2020; X. Yang et al., 2023a; Yoo et al., 2019). Uninterpretable images exist due to inappropriate focus, exposure, or illumination settings used during the image-capturing process and do not contain sufficient image biomarkers for the reviewer to conclude the absence or presence of retinal pathology (i.e., ungradable) (Hadziahmetovic et al., 2019). Specifically, computer-aided diagnosis tools developed by Usher et al., 2004 were able to identify retinal pathology in a semi-automated manner using color fundus photography (CFP) images. However, human interaction was necessary for the image pre-processing or feature extraction steps. Christopher et al., 2018; Gargeya and Leng, 2017; Gulshan et al., 2016; Raman et al., 2019; Shibata et al., 2018; Singh and Gorantla, 2020 improved the automation degree, but for interpretable images only, by proposing a one-fit-for-all pre-processing method for CFP images, with the resulting images being processed and classified by convolutional neural networks (CNNs). Additionally, Kermany, Goldbaum, Cai, Valentim, and et al., 2018; W. Lu et al., 2018; Muhammad et al., 2017; Treder et al., 2018a; Y. Wang et al., 2016 devised CNN-based models that can identify ophthalmic pathologies from optical coherence tomography (OCT) scans. To further improve the prediction performance and capture the image features jointly across different modalities, Vaghefi et al., 2020; W. Wang et al., 2019a; Z. Xu et al., 2020; Yoo et al., 2019 proposed multi-stream CNN models for automated diagnosis using multi-model

inputs (e.g., OCT and CFP). However, to the best of our knowledge, no existing work can be deployed for fully automated retinal pathology diagnosis, mostly because uninterpretable images are excluded from training and testing (Christopher et al., 2018; Gargeya & Leng, 2017; Gulshan et al., 2016; Kermany, Goldbaum, Cai, Valentim, & et al., 2018; W. Lu et al., 2018; Medeiros et al., 2019; Mookiah et al., 2013; Muhammad et al., 2017; Raman et al., 2019; Shibata et al., 2018; Singh & Gorantla, 2020; Treder et al., 2018a, 2018b; Usher et al., 2004; Vaghefi et al., 2020; W. Wang et al., 2019a; Y. Wang et al., 2016; Winder et al., 2009; Z. Xu et al., 2020; Yoo et al., 2019). This process requires an expert’s input to determine ungradable images and exclude them from the dataset. The presence of images with substandard quality is universal and inevitable to encounter in clinical practice (Bennett, 2009; S. Liu et al., 2009). This problem might become more accentuated in the future with broader acceptance of automated image capture systems with integrated AI-based diagnosis algorithms. In such instances, no clinician would be present on-site to fine-tune the scanner for each patient or re-take images if the outputs were unsatisfactory. Consequently, a substantial number of ophthalmic screenings on un-dilated pupils will likely contain uninterpretable images, and it is essential to include those while designing such DL models to allow for their integration into a fully automated diagnosis system for instant and accurate diagnoses.

The purpose of this study was to create such an accurate DL approach for retinal image classification and identification of referable retinal pathology. Our main goal was to develop a CNN model that can automatically handle imperfect, including uninterpretable images, and provide high validation accuracy and low false-negative rate to identify retinal pathology.

1.1.5.1 Contributions

There is an unmet need for automated imaging and diagnosis systems for identifying retinal pathology. This limitation of the current healthcare model has been emphasized during the COVID19 pandemic, especially since ophthalmology has been one of the hardest-

hit specialties (A. Mehrotra et al., 2017). Additionally, early recognition of sight-threatening retinal diseases might offer timely treatment, potentially improve visual outcomes, and reduce healthcare costs. Moreover, with improved triage, clinician effort, and clinic time might be better spent on other activities providing improved referral accuracy and more efficient use of ophthalmic resources (B. Li et al., 2015; Michalak & Hadziahmetovic, 2019; Rathi et al., 2017; Sreelatha & Ramesh, 2016).

Chapter 7 introduces a CNN-based approach that enables fully automated retinal image classification into present or absent retinal pathology. Similar existing methods cannot be applied autonomously as they have not been developed while considering uninterpretable images, which are frequently encountered during eye screening (Christopher et al., 2018; Gargeya & Leng, 2017; Gulshan et al., 2016; Kermany, Goldbaum, Cai, Valentim, & et al., 2018; W. Lu et al., 2018; Raman et al., 2019; Shibata et al., 2018; Singh & Gorantla, 2020; Treder et al., 2018a, 2018b; Vaghefi et al., 2020; W. Wang et al., 2019a; Y. Wang et al., 2016; Z. Xu et al., 2020; Yoo et al., 2019), and thus cannot handle them well. By addressing these limitations, our approach facilitates the development of automated retinal diagnosis systems, where a healthcare worker does not need to evaluate the quality of the images (in order for some to be retaken) before they are submitted for the analysis. This system can be deployed either in the clinics for triage or during remote screening (e.g., teleophthalmology) without involving physical interactions between patients and physicians.

Herein, we presented a CNN model that takes OCT and CFP images as dual-modal inputs and predicts if the corresponding eye has retinal pathology (e.g., DR, DME, and AMD). Our model was able to process imperfect/uninterpretable images resulting from the patient’s poor positioning during the screening or inappropriate parameters (Bennett, 2009; S. Liu et al., 2009) (e.g., focus, exposure, and illumination). Inputs obtained from uninterpretable images were utilized during the training through a novel backpropagation algorithm that can minimize the impact from images that do not contain sufficient image biomarkers to be determined as RPN/RPP during the training process. We created a fully-automated retinal pathology diagnosis system (i.e., that requires no human interaction).

To train and validate our model, we collected 1148 pairs of CFP and OCT images from 674 patients, where each pair pertains to a single eye of a patient. We used a 9:1 ratio to split the training and testing dataset. Finally, we attained a validation accuracy of 88.6%, Recall/Sensitivity of 87.7%, Specificity of 89.5%, and area under the curve (AUC) for receiver operating characteristic (ROC) of 0.93. We presented the case, which only considers the dual-modal inputs (OCT and CFP); regardless, the proposed approach can be further extended to include other imaging modalities (e.g., fundus autofluorescence). Moreover, we observed that the performance of baseline methods could be negatively impacted when uninterpretable images are used for testing. On the other hand, the performance of our approach was not affected when evaluated with either full testing dataset or interpretable images only.

2. Preliminaries

In this chapter we review some basics of variational inference, RL and OPE. The definitions introduced in this chapter will be used consistently throughout the remaining chapters.

2.1 Variational Inference

Classic variational auto-encoders (VAEs) are designed to generate synthetic data that share similar characteristics than the ones used for training (Kingma & Welling, 2013). Specifically, VAEs learn an approximated posterior $q_\psi(z|x)$ and a generative model $p_\phi(x|z)$, over the prior $p(z)$, with x being the data and z the latent variable. It's true posterior $p_\phi(z|x)$ is intractable, *i.e.*,

$$p_\phi(z|x) = \frac{p_\phi(x|z)p(z)}{p_\phi(x)}; \quad (2.1)$$

since the marginal likelihood in the denominator, $p_\phi(x) = \int_z p_\phi(x|z)p(z)dz$, requires integration over the unknown latent space. For the same reason, VAEs cannot be trained to directly maximize the marginal log-likelihood, $\max \log p_\phi(x)$. To resolve this, one could maximize a lower bound of $p_\phi(x)$, *i.e.*,

$$\max_{\psi, \phi} -KL(q_\psi(z|x)||p(z)) + \mathbb{E}_{q_\psi}[\log p_\phi(x|z)], \quad (2.2)$$

which is the evidence lower bound (ELBO).

2.1.1 Reparameterization.

During training, it is required to sample from $q_\psi(z|x)$ and $p_\phi(x|z)$ constantly. The reparameterization technique is introduced in (Kingma & Welling, 2013), to ensure that the gradients can flow through such sampling process during back-propagation. For example, if both distributions ($q_\psi(z|x)$ and $p_\phi(x|z)$) follow diagonal Gaussians, with mean and diagonal covariance determined by MLPs, *i.e.*,

$$z \sim q_\psi(z|x) = \mathcal{N}\left(\boldsymbol{\mu} = \psi_\mu^{MLP}(x), \quad \boldsymbol{\Sigma} = \psi_\Sigma^{MLP}(x)\right), \quad (2.3)$$

$$x \sim p_\phi(x|z) = \mathcal{N}\left(\boldsymbol{\mu} = \phi_\mu^{MLP}(z), \quad \boldsymbol{\Sigma} = \phi_\Sigma^{MLP}(z)\right); \quad (2.4)$$

here, $\psi_\mu^{MLP}, \psi_\Sigma^{MLP}, \phi_\mu^{MLP}, \phi_\Sigma^{MLP}$ are the MLPs that generate the means and covariances. The sampling processes above can be captured by reparameterization, *i.e.*,

$$z = \psi_\mu^{MLP}(x) + \psi_\Sigma^{MLP}(x) \cdot \epsilon, \quad (2.5)$$

$$x = \phi_\mu^{MLP}(z) + \phi_\Sigma^{MLP}(z) \cdot \epsilon, \quad (2.6)$$

with $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. Consequently, the gradients over ψ and ϕ can be calculated following the chain rule, and used for back-propagation during training. We direct readers to (Kingma & Welling, 2013) for a comprehensive review of reparameterization.

2.2 Offline RL

Offline RL has proven useful in many domains, including robotics (Q. Gao, Hajinezhad, Zhang, Kantaros, & Zavlanos, 2019; Gu et al., 2017), healthcare (Q. Gao, Wang, et al., 2022b), etc., since it can optimize the control policies without requiring the environment to be presented, which guarantees the safety of the learning process. Further, it does not require the training data to be exclusively collected by the control policy being updated, leading to improved sample efficiency. To facilitate offline RL, the underlying dynamical environments are firstly modeled as Markov decision processes (MDPs).

Definition 1 (MDP). *An MDP is a tuple $\mathcal{M} = (\mathcal{S}, s_0, \mathcal{A}, \mathcal{P}, R, \gamma)$, where \mathcal{S} is a finite set of states; s_0 is the initial state; \mathcal{A} is a finite set of actions; \mathcal{P} is the transition function defined as $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$; $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is a discount factor.*

Then, the RL policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ determines the action $a = \pi(s)$ to be taken at a given state s . The accumulated return under a policy π can be defined as follows.

Definition 2 (Accumulated Return). *Given an MDP \mathcal{M} and a policy π , the accumulated return over a finite horizon starting from the stage t and ending at stage T , for $T > t$, is defined as*

$$G_t^\pi = \sum_{k=0}^{T-t} \gamma^{t+k} r_{t+k}, \quad (2.7)$$

where r_{t+k} is the return at the stage $t+k$.

The goal of offline RL can now be defined as follows.

Definition 3 (Objective of Offline Reinforcement Learning). *Given an MDP \mathcal{M} with unknown transition dynamics \mathcal{P} , a pre-defined reward function R , and a experience replay buffer $\mathcal{E}^\mu = \{[(s_0, a_0, r_0, s_1), \dots, (s_{T-1}, a_{T-1}, r_{T-1}, s_T)]^{(0)}, [(s_0, a_0, r_0, s_1), \dots]^{(1)}, \dots | a_t \sim \mu(a_t | s_t)\}$ containing trajectories collected over an unknown behavioral policy μ , find the target policy π^* such that the expected accumulative return starting from the initial stage over the entire horizon is maximized, i.e.,*

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{s, a \sim \rho^\pi, r \sim R} [G_0^\pi]; \quad (2.8)$$

here, ρ^π is the state-action visitation distribution under policy π .

The deep actor-critic RL framework (Lillicrap et al., 2016) can be leveraged to solve (2.8). Other value-based RL methods such as conservative Q-learning (Kumar et al., 2020) and implicit Q-learning (Kostrikov et al., 2022) could also be considered; however, actor-critic methods can in general reduce the variance of gradient estimations and result in faster convergence (Q. Gao, Naumann, et al., 2020; Mnih et al., 2016; Y. Wu et al., 2017). Here, we specifically introduce the deterministic version of actor-critic (Lillicrap et al., 2016), instead of one producing stochastic policies (Haarnoja et al., 2018), as it would be easier to demonstrate the effectiveness of deterministic policies in the real world, as well as via OPE methods introduced below. Details in regards to the actor-critic framework are introduced in the sub-section below.

2.2.1 Actor-Critic RL

We now briefly introduce the deep actor-critic algorithm (Lillicrap et al., 2016) and refer the readers to (Q. Gao, Naumann, et al., 2020; Q. Gao, Schmidt, et al., 2022; Lillicrap et al., 2016) for more details. First, the state-action value functions can be defined as follows.

Definition 4 (State-Action Value Function). *Given an MDP \mathcal{M} and policy π , the state-action value function $Q^\pi(s, a)$, where $s \in \mathcal{S}$ and $a \in \mathcal{A}$, is defined as the expected return for taking action*

a when at state s following policy π at stage t , i.e.,

$$Q^\pi(s, a) = \mathbb{E}_{s \sim \mathcal{S}, a \sim \mathcal{A}}[G_t | s_t = s, a_t = a]. \quad (2.9)$$

Two neural networks, with weights θ_a and θ_c , can be used to parameterize the policy (actor) $\pi_{\theta_a}(s) : \mathcal{S} \rightarrow \mathcal{A}$ and the Q-functions (critic) $Q_{\theta_c}(s, a) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, respectively. Finally, the target policy $\pi^* = \pi_{\theta_a^*}$ can be obtained by optimizing over

$$\max_{\theta_a, \theta_c} \mathbb{E}_{s, a, r, s' \sim \mathcal{E}^\mu} [Q_{\theta_c}(s, \pi_{\theta_a}(s))]; \quad (2.10)$$

this can be achieved using gradient descent, over all the training samples in the experience replay buffer \mathcal{E}^μ (Lillicrap et al., 2016).

2.3 OPE

In this section, we first introduce the general objective for OPE problems. Then, four types of basic OPE techniques are reviewed. At last, the metrics commonly used to evaluate OPE’s performance are introduced.

2.3.1 Objective

We first introduce the MDP used to characterize the environment. Specifically, an MDP can be defined as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, R, s_0, \gamma)$, where \mathcal{S} is the set of states, \mathcal{A} the set of actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the transition distribution usually captured by probabilities $p(s_t | s_{t-1}, a_{t-1})$, $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, s_0 is the initial state sampled from the initial state distribution $p(s_0)$, $\gamma \in [0, 1)$ is the discounting factor. Finally, the agent interacts with the MDP following some policy $\pi(a|s)$ which defines the probabilities of taking action a at state s . Then, the goal of OPE can be formulated as follows.

Definition 5 (Objective of OPE). *Given trajectories collected by a behavioral policy β , $\rho^\beta = \{[(s_0, a_0, r_0, s_1), \dots, (s_{T-1}, a_{T-1}, r_{T-1}, s_T)]^{(0)}, [(s_0, a_0, r_0, s_1), \dots]^{(1)}, \dots | a_t \sim \beta(a_t | s_t)\}^1$, estimate the expected total return over the unknown state-action visitation distribution ρ^π of the*

¹ We slightly abuse the notation ρ^β , to represent either the trajectories or state-action visitation distribution under the behavioral policy, depending on the context.

target (evaluation) policy π – *i.e.*, for T being the horizon,

$$\mathbb{E}_{(s,a) \sim \rho^\pi, r \sim R} \left[\sum_{t=0}^T \gamma^t R(s_t, a_t) \right]. \quad (2.11)$$

2.3.2 Importance Sampling (IS) and per-decision IS (PDIS)

IS refers to a statistical technique that can calculate the expectation of a function $f(x)$ w.r.t. an unknown distribution $p(x)$ using a given distribution $q(x)$ through re-weighting, *i.e.*,

$$\mathbb{E}_p[f(x)] = \mathbb{E}_q \left[\frac{f(x)p(x)}{q(x)} \right]. \quad (2.12)$$

This technique can be applied in the context of OPE by setting $f(x)$ as the accumulated return $G_{0:T}$, p as the trajectory distribution ρ^π over the target policy π , and q as the trajectory distribution ρ^β over the behavioral policy β .

According to (Precup, 2000), the vanilla IS estimator follows

$$\hat{G}_{IS}^\pi = \frac{1}{N} \sum_{i=1}^{N-1} \omega_{0:T-1}^{(i)} G_{0:T}^{(i)}, \quad (2.13)$$

where $\omega_{0:T-1}^{(i)} = \prod_{t=0}^{T-1} \frac{\pi(a_t|s_t)}{\beta(a_t|s_t)}$ refers to the IS weight, and $G_{0:T}^{(i)}$ is the return for the i -th (out of N) offline trajectory. On the other hand, the PDIS estimator follows

$$\hat{G}_{PDIS}^\pi = \frac{1}{N} \sum_{i=1}^{N-1} \sum_{t=0}^{T-1} \gamma^t \omega_{0:t}^{(i)} r_t^{(i)}, \quad (2.14)$$

with $\omega_{0:t}^{(i)} = \prod_{k=0}^t \frac{\pi(a_k|s_k)}{\beta(a_k|s_k)}$ being the PDIS weight, $r_t^{(i)}$ is the environmental reward obtained at the t -th step of the i -th offline trajectory, and γ is the discounting factor.

2.3.3 Doubly Robust (DR)

DR attempts to reduce the variance of IS estimation by introducing the value function approximation, which trades off estimation variance by making the estimation biased. The non-recursive definition of DR estimation is provided in (P. Thomas & Brunskill, 2016), *i.e.*,

$$\hat{G}_{DR}^\pi = \frac{1}{N} \sum_{i=1}^{N-1} \sum_{t=0}^{T-1} \gamma^t \omega_{0:t}^{(i)} r_t^{(i)} - \frac{1}{N} \sum_{i=1}^{N-1} \sum_{t=0}^{T-1} \left(\gamma^t \omega_{0:t}^{(i)} r_t^{(i)} Q^\pi(s_t^{(i)}, a_t^{(i)}) - \omega_{0:t-1}^{(i)} V^\pi(s_t^{(i)}) \right); \quad (2.15)$$

here, $Q^\pi(\cdot, \cdot)$ and $V^\pi(\cdot)$ are the Q-function and value function, respectively, over the target policy, while $s_t^{(i)}$ and $a_t^{(i)}$ are the state and action taken at the t -th step of the i -th offline trajectory.

2.3.4 Distributional Correction Estimation (DICE)

DICE aims to estimate the propensity of the target policy to visit particular state-action pairs relative to their likelihood of appearing in the offline trajectories, *i.e.*,

$$\hat{G}_{DICE}^\pi = \mathbb{E}_{(s,a,r) \sim \rho^\beta} \left[\frac{d^\pi(s,a)}{d^\beta(s,a)} \cdot r \right], \quad (2.16)$$

where $\frac{d^\pi(s,a)}{d^\beta(s,a)}$ is the distribution correction ratio. Existing DICE variants (Dai et al., 2020; Nachum et al., 2019; M. Yang et al., 2020; R. Zhang et al., 2020; S. Zhang et al., 2020) seek to approximate the ratio without knowledge of d^π or d^β , while a recent work has summarized that all existing variants can be formulated as regularized Lagrangians of the same linear program (M. Yang et al., 2020), unifying the choice of regularizations and constraints used in their original objectives.

2.3.5 Fitted Q-Evaluation (FQE)

According to (Le et al., 2019), FQE approximates the Q-function over the target policy by minimizing the loss

$$\min_{\kappa} \mathbb{E}_{(s_t, a_t, r_t) \sim \rho^\beta} \left[(Q(s_t, a_t; \kappa) - y_t)^2 \right], \quad (2.17)$$

$$\text{s.t. } y_t = r_t + \gamma Q(s_{t+1}, \pi(s_{t+1}); \kappa); \quad (2.18)$$

here, the approximated Q-function is parameterized by κ . Note that it is different from the objective for classic Q-learning, or fitted Q-iteration, by using $\pi(s_{t+1})$ instead of $\max_a Q(s_{t+1}, a; \kappa)$ in the target y_t .

2.3.6 Definition of the OPE Metrics

The three commonly used metrics in OPE are introduced below (Fu, Norouzi, et al., 2020; Q. Gao, Gao, Chi, & Pajic, 2023a; Q. Gao, Schmidt, et al., 2023; Q. Gao et al., 2024).

2.3.6.1 Mean Absolute error (MAE).

MAE is defined as the absolute difference between the actual return and estimated return of a policy: $MAE = |V^\pi - \hat{V}^\pi|$; here, V^π is the actual value of the policy π , and \hat{V}^π is the estimated value of π .

2.3.6.2 Rank correlation.

Rank correlation measures the Spearman's rank correlation coefficient between the ordinal rankings of the estimated returns and actual returns across policies, *i.e.*,

$\rho = \frac{Cov(\text{rank}(V_{1:P}^\pi), \text{rank}(\hat{V}_{1:P}^\pi))}{\sigma(\text{rank}(V_{1:P}^\pi))\sigma(\text{rank}(\hat{V}_{1:P}^\pi))}$, where $\text{rank}(V_{1:P}^\pi)$ is the ordinal rankings of the actual returns, and $\text{rank}(\hat{V}_{1:P}^\pi)$ is the ordinal rankings of the OPE-estimated returns.

2.3.6.3 Regret@1.

Regret@1 is the (normalized) difference between the value of the actual best policy, against the value of the policy associated with the best OPE-estimated return, which is defined as $(\max_{i \in 1:P} V_i^\pi - \max_{j \in \text{best}(1:P)} V_j^\pi) / \max_{i \in 1:P} V_i^\pi$, where $\text{best}(1 : P)$ denotes the index of the best policy over the set of P policies as measured by estimated values \hat{V}^π .

3. VLBM

In this chapter, we introduce VLBM that aims to learn a compact and disentangled latent representation space from offline trajectories, toward better capturing the dynamics underlying environments. Specifically, we first introduce related works and the objective of OPE and the variational latent model (VLM) we consider. Then, we propose the recurrent state alignment (RSA) term as well as the branching architecture that constitute the variational latent branching model (VLBM). Finally, numerical experiments are used to validate the efficacy of the approach.

3.1 Related Work

This section introduces the works related to the topic being considered, including the use of latent modeling techniques in RL, as well as OPE in general.

3.1.1 Latent Modeling in RL

Though variational inference has rarely been explored to facilitate model-based OPE methods so far, there exist several latent models designed for RL policy optimization that are related to our work, such as SLAC (Lee et al., 2020), SOLAR (M. Zhang et al., 2019), LatCo (Rybkin et al., 2021), PlaNet (Hafner et al., 2019), Dreamer (Hafner, Lillicrap, et al., 2020; Hafner, Lillicrap, et al., 2020). Below we discuss the connections and distinctions between VLBM and the latent models leveraged by them. Specifically, SLAC and SOLAR learn latent representations of the dynamics jointly with optimization of the target policies, using the latent information to improve sample efficiency. Similarly, LatCo performs trajectory optimization over the latent space to allow for temporarily bypassing dynamic constraints. As a result, latent models used in such methods are not designed toward rolling out trajectories independently, as opposed to the use of VLBM in this paper. PlaNet and Dreamer train the recurrent state space model (RSSM) using a *growing* experience dataset collected by the target policy that is being concurrently updated (with exploration noise added), which requires *online* data collection. In contrast, under the OPE setup, VLBM is trained over a *fixed* set of offline trajectories collected over unknown behavioral poli-

cies. Moreover, note that the VLM baseline is somewhat reminiscent of the RSSM and similar ones as in (Lee et al., 2020; C. Lu et al., 2022), however, experiments above show that directly using VLM for OPE could lead to subpar performance. On the other hand, though MOPO (T. Yu et al., 2020), LOMPO (Rafailov et al., 2021) and COMBO (T. Yu et al., 2021) can learn from offline data, they focus on quantifying the uncertainty of model’s predictions toward next states and rewards, followed by incorporating them into policy optimization objectives to penalize for visiting regions where transitions are not fully captured; thus, such works are also orthogonal to the use case of OPE.

3.1.2 OPE

Classic OPE methods adopt IS to estimate expectations over the unknown visitation distribution over the target policy, resulting in weighted IS, step-wise IS and weighted step-wise IS (Precup, 2000). IS can lead to estimations with low (or zero) bias, but with high variance (Jiang & Li, 2016; Kostrikov & Nachum, 2020), which sparks a long line of research to address this challenge. DR methods propose to reduce variance by coupling IS with a value function approximator (Farajtabar et al., 2018; Jiang & Li, 2016; P. Thomas & Brunskill, 2016). However, the introduction of such approximations may increase bias, so the method proposed in (Z. Tang et al., 2019) attempts to balance the scale of bias and variance for DR. Unlike IS and DR methods that require the behavioral policies to be fully known, DICE family of estimators (Dai et al., 2020; Nachum et al., 2019; M. Yang et al., 2020, 2021; R. Zhang et al., 2020; S. Zhang et al., 2020) and VPM (Wen et al., 2020) can be behavioral-agnostic; they directly capture marginalized IS weights as the ratio between the propensity of the target policy to visit particular state-action pairs, relative to their likelihood of appearing in the logged data. There also exist FQE methods which extrapolate policy returns from approximated Q-functions (Hao et al., 2021; Kostrikov & Nachum, 2020; Le et al., 2019). Existing model-based OPE methods are designed to directly fit MDP transitions using feed-forward (Fu, Norouzi, et al., 2020) or auto-regressive (M. R. Zhang et al., 2020) models, and has shown promising results over model-free methods as

reported in a recent benchmark (Fu, Norouzi, et al., 2020). However, such model-based approaches could be sensitive to the initialization of weights (Hanin & Rolnick, 2018; Rossi et al., 2019) and produce biased predictions, due to the limited coverage over state and action space provided by offline trajectories (Fu, Norouzi, et al., 2020). Instead, VLBM mitigates such effects by capturing the dynamics over the latent space, such that states and rewards are evolved from a compact feature space over time. Moreover, RSA and the branching can lead to increased expressiveness and robustness, such that future states and rewards are predicted accurately. There also exist OPE methods proposed toward specific applications (M. Chen et al., 2022; G. Gao, Ju, et al., 2023; Q. Gao, Schmidt, et al., 2022; Saito et al., 2021).

3.2 Problem Formulation

The objective of this chapter directly follows from Section 2.3.1.

3.3 Variational Latent Model

We consider the VLM consisting of a prior $p(z)$ over the latent variables $z \in \mathcal{Z} \subset \mathbb{R}^l$, with \mathcal{Z} representing the latent space and l the dimension, along with a variational encoder $q_\psi(z_t|z_{t-1}, a_{t-1}, s_t)$ and a generative decoder $p_\phi(z_t, s_t, r_{t-1}|z_{t-1}, a_{t-1})$, parameterized by ψ and ϕ respectively. Basics of variational inference are introduced in Section 2.1.

3.3.1 Latent Prior $p(z_0)$.

The prior specifies the distribution from which the latent variable of the *initial* stage, z_0 , is sampled. We configure $p(z_0)$ to follow a Gaussian with zero mean and identity covariance matrix, which is a common choice under the variational inference framework (Kingma & Welling, 2013; Lee et al., 2020).

3.3.2 Variational Encoder for Inference $q_\psi(z_t|z_{t-1}, a_{t-1}, s_t)$.

The encoder is used to approximate the intractable posterior, $p(z_t|z_{t-1}, a_{t-1}, s_t) = \frac{p(z_{t-1}, a_{t-1}, z_t, s_t)}{\int_{z_t \in \mathcal{Z}} p(z_{t-1}, a_{t-1}, z_t, s_t) dz_t}$, where the denominator requires integrating over the unknown latent

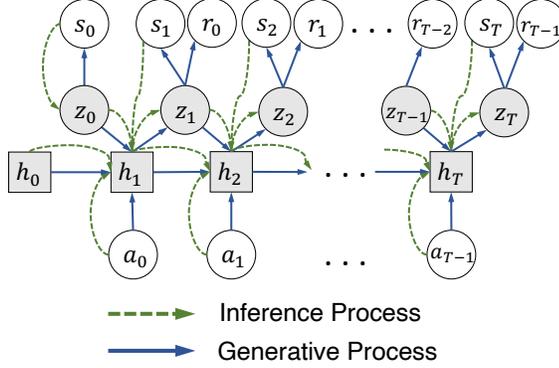


FIGURE 3.1: Architecture of variational latent model (VLM) we consider.

space. Specifically, the encoder can be decomposed into two parts, given that

$$\begin{aligned}
 & q_{\psi}(z_{0:T}|s_{0:T}, a_{0:T-1}) \\
 &= q_{\psi}(z_0|s_0) \prod_{t=1}^T q_{\psi}(z_t|z_{t-1}, a_{t-1}, s_t);
 \end{aligned} \tag{3.1}$$

here, $q_{\psi}(z_0|s_0)$ encodes the initial state s_0 into the corresponding latent variable z_0 , then, $q_{\psi}(z_t|z_{t-1}, a_{t-1}, s_t)$ enforces the transition from z_{t-1} to z_t conditioned on a_{t-1} and s_t . Both distributions are *diagonal* Gaussians¹, with means and diagonal of covariance matrices determined by multi-layered perceptron (MLP) (Bishop, 2006) and long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) respectively. The weights for both neural networks are referred to as ψ in general.

Consequently, the *inference* process for z_t can be summarized as

$$z_0^{\psi} \sim q_{\psi}(z_0|s_0), \quad h_t^{\psi} = f_{\psi}(h_{t-1}^{\psi}, z_{t-1}^{\psi}, a_{t-1}, s_t), \quad z_t^{\psi} \sim q_{\psi}(z_t|h_t^{\psi}), \tag{3.2}$$

where f_{ψ} represents the LSTM layer and h_t^{ψ} the LSTM recurrent (hidden) state. Note that we use ψ in superscripts to distinguish the variables involved in this *inference* process, against the *generative* process introduced below. Moreover, reparameterization can be used to sample z_0^{ψ} and z_t^{ψ} , such that gradients of sampling can be back-propagated, as introduced in (Kingma & Welling, 2013). Overview of the inference and generative processes are illustrated in Figure 3.1.

¹ Assume that different dimensions of the states are non-correlated with each other. Otherwise, the states can be projected to orthogonal basis, such that non-diagonal elements of the covariance matrix will be zeros.

3.3.3 Generative Decoder for Sampling $p_\phi(z_t, s_t, r_{t-1}|z_{t-1}, a_{t-1})$.

The decoder is used to interact with the target policies and acts as a synthetic environment during policy evaluation, from which the expected returns can be estimated as the mean return of simulated trajectories. The decoder can be represented by the multiplication of three diagonal Gaussian distributions, given that

$$p_\phi(z_{1:T}, s_{0:T}, r_{0:T-1}|z_0, \pi) = \prod_{t=0}^T p_\phi(s_t|z_t) \prod_{t=1}^T p_\phi(z_t|z_{t-1}, a_{t-1}) p_\phi(r_{t-1}|z_t), \quad (3.3)$$

with $a_t \sim \pi(a_t|s_t)$ at each time step. Specifically, $p_\phi(z_t|z_{t-1}, a_{t-1})$ has its mean and covariance determined by an LSTM, enforcing the transition from z_{t-1} to z_t in the latent space given action a_{t-1} . In what follows, $p_\phi(s_t|z_t)$ and $p_\phi(r_{t-1}|z_t)$ generate the current state s_t and reward r_{t-1} given z_t , whose mean and covariance are determined by MLPs. As a result, the *generative* process starts with sampling the initial latent variable from the latent prior, *i.e.*, $z_0^\phi \sim p(z_0)$. Then, the initial state $s_0^\phi \sim p_\phi(s_0|z_0^\phi)$ and action $a_0 \sim \pi(a_0|s_0^\phi)$ are obtained from p_ϕ and target policy π , respectively; the rest of *generative* process can be summarized as

$$\begin{aligned} h_t^\phi &= f_\phi(h_{t-1}^\phi, z_{t-1}^\phi, a_{t-1}), & \tilde{h}_t^\phi &= g_\phi(h_t^\phi), & z_t^\phi &\sim p_\phi(\tilde{h}_t^\phi), \\ s_t^\phi &\sim p_\phi(s_t|z_t^\phi), & r_{t-1}^\phi &\sim p_\phi(r_{t-1}|z_t^\phi), & a_t &\sim \pi(a_t|s_t^\phi), \end{aligned} \quad (3.4)$$

where f_ϕ is the LSTM layer producing recurrent state h_t^ϕ . Then, an MLP g_ϕ is used to generate mapping between h_t^ϕ and \tilde{h}_t^ϕ that will be used for recurrent state alignment (RSA) introduced below, to augment the information flow between the inference and generative process.

Furthermore, to train the elements in the encoder (3.2) and decoder (3.4), one can maximize the evidence lower bound (ELBO), a lower bound of the joint log-likelihood $p(s_{0:T}, r_{0:T-1})$, following

$$\begin{aligned} \mathcal{L}_{ELBO}(\psi, \phi) &= \mathbb{E}_{q_\psi} \left[\sum_{t=0}^T \log p_\phi(s_t|z_t) + \sum_{t=1}^T \log p_\phi(r_{t-1}|z_t) - \text{KL}(q_\psi(z_0|s_0)||p(z_0)) \right. \\ &\quad \left. - \sum_{t=1}^T \text{KL}(q_\psi(z_t|z_{t-1}, a_{t-1}, s_t)||p_\phi(z_t|z_{t-1}, a_{t-1})) \right]; \end{aligned} \quad (3.5)$$

here, the first two terms represent the log-likelihood of reconstructing the states and rewards, and the last two terms regularize the approximated posterior. The proof can be found below.

3.3.3.1 Proof of (3.5)

We now derive the evidence lower bound (ELBO) for the joint log-likelihood distribution, *i.e.*,

$$\log p_\phi(s_{0:T}, r_{0:T-1}) \tag{3.6}$$

$$= \log \int_{z_{1:T} \in \mathcal{Z}} p_\phi(s_{0:T}, z_{1:T}, r_{0:T-1}) dz \tag{3.7}$$

$$= \log \int_{z_{1:T} \in \mathcal{Z}} \frac{p_\phi(s_{0:T}, z_{1:T}, r_{0:T-1})}{q_\psi(z_{0:T}|s_{0:T}, a_{0:T-1})} q_\psi(z_{0:T}|s_{0:T}, a_{0:T-1}) dz \tag{3.8}$$

$$\geq \mathbb{E}_{q_\psi} [\log p(z_0) + \log p_\phi(s_{0:T}, z_{1:T}, r_{0:T-1}|z_0) - \log q_\psi(z_{0:T}|s_{0:T}, a_{0:T-1})] \tag{3.9}$$

$$\begin{aligned} &= \mathbb{E}_{q_\psi} \left[\log p(z_0) + \log p_\phi(s_0|z_0) + \sum_{t=1}^T \log p_\phi(s_t, z_t, r_{t-1}|z_{t-1}, a_{t-1}) \right. \\ &\quad \left. - \log q_\psi(z_0|s_0) - \sum_{t=1}^T \log q_\psi(z_t|z_{t-1}, a_{t-1}, s_t) \right] \end{aligned} \tag{3.10}$$

$$\begin{aligned} &= \mathbb{E}_{q_\psi} \left[\log p(z_0) - \log q_\psi(z_0|s_0) + \log p_\phi(s_0|z_0) + \sum_{t=1}^T \log (p_\phi(s_t|z_t) p_\phi(r_{t-1}|z_t) p_\phi(z_t|z_{t-1}, a_{t-1})) \right. \\ &\quad \left. - \sum_{t=1}^T \log q_\psi(z_t|z_{t-1}, a_{t-1}, s_t) \right] \end{aligned} \tag{3.11}$$

$$\begin{aligned} &= \mathbb{E}_{q_\psi} \left[\sum_{t=0}^T \log p_\phi(s_t|z_t) + \sum_{t=1}^T \log p_\phi(r_{t-1}|z_t) \right. \\ &\quad \left. - KL(q_\psi(z_0|s_0)||p(z_0)) - \sum_{t=1}^T KL(q_\psi(z_t|z_{t-1}, a_{t-1}, s_t)||p_\phi(z_t|z_{t-1}, a_{t-1})) \right]. \end{aligned} \tag{3.12}$$

Note that the transition from (3.8) to (3.9) follows Jensen's inequality.

3.4 Recurrent State Alignment

The latent model discussed above is somewhat reminiscent of the ones used in model-based RL policy training methods, *e.g.*, recurrent state space model (RSSM) used in PlaNet (Hafner et al., 2019) and Dreamer (Hafner, Lillicrap, et al., 2020; Hafner, Lilli-

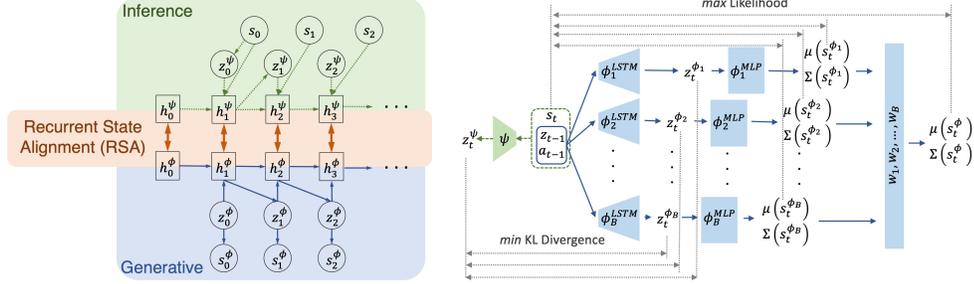


FIGURE 3.2: (Left) Recurrent state alignment (RSA) applied over the recurrent hidden states between inference and generative process illustrated separately. (Right) Single-step forward pass of the variational latent branching model (VLBM), the training objectives for each branch and final predictions.

crap, et al., 2020), as well as similar ones in (Lee et al., 2020; C. Lu et al., 2022). Such methods rely on a *growing* experience buffer for training, which is collected *online* by the target policy that is being concurrently updated (with exploration noise added); however, OPE aims to extrapolate returns from a fixed set of *offline* trajectories which may result in limited coverage of the state and action space. Consequently, directly applying VLM for OPE can lead to subpar performance empirically; see results in Section 3.6. Moreover, the encoder above plays a key role of capturing the temporal transitions between latent variables, *i.e.*, $p_\psi(z_t|z_{t-1}, a_{t-1}, s_t)$ from (3.1). However, it is *absent* in the generative process, as the decoder leverages a separate network to determine the latent transitions, *i.e.*, $p_\phi(z_t|z_{t-1}, a_{t-1})$. Moreover, from the ELBO (3.5) above it can be seen that only the KL-divergence terms are used to regularize these two parts, which may not be sufficient for OPE as limited offline trajectories are provided. As a result, we introduce the RSA term as part of the training objective, to further regularize $p_\psi(z_t|z_{t-1}, a_{t-1}, s_t)$ and $p_\phi(z_t|z_{t-1}, a_{t-1})$. A graphical illustration of RSA can be found in Figure 3.2.²

Specifically, RSA is defined as the mean *pairwise* squared error between h_t^ψ from the

² Rewards and actions are omitted for conciseness of the presentation.

encoder (3.2) and \tilde{h}_t^ϕ from the decoder (3.4), *i.e.*,

$$\mathcal{L}_{RSA}(\tilde{h}_t^\phi, h_t^\psi; \psi, \phi) = \frac{1}{N} \sum_{i=1}^N \sum_{t=0}^T \frac{M(M-1)}{2} \left[\sum_{j=1}^{M-1} \sum_{k=j+1}^M ((\tilde{h}_t^\phi[j] - \tilde{h}_t^\phi[k]) - (h_t^\psi[j] - h_t^\psi[k]))^2 \right]; \quad (3.13)$$

here, we assume that both LSTM recurrent states have the same dimension $\tilde{h}_t^\phi, h_t^\psi \in \mathbb{R}^M$, with $h_t^{(\cdot)}[j]$ referring to the j -th element of the recurrent state, and N the number of training trajectories.

Here, we choose the pairwise squared loss over the classic mean squared error (MSE), because MSE could be too strong to regularize h_t^ψ and \tilde{h}_t^ϕ which support the inference and generative processes respectively and are not supposed to be exactly the same. In contrast, the pairwise loss (3.13) can promote structural similarity between the LSTM recurrent states of the encoder and decoder, without strictly enforcing them to become the same. Note that this design choice has been justified in Section 3.6 through an ablation study by comparing against models trained with MSE. In general, the pairwise loss has also been adopted in many domains for similar purposes, *e.g.*, object detection (Gould et al., 2009; Rocco et al., 2018), ranking systems (Doughty et al., 2018; Saquil et al., 2021) and contrastive learning (T. Chen et al., 2020; X. Wang et al., 2021). Similarly, we apply the pairwise loss over h_t^ψ and \tilde{h}_t^ϕ , instead of directly over h_t^ψ and h_t^ϕ , as the mapping g_ϕ (from (3.4)) could serve as a regularization layer to ensure optimality over \mathcal{L}_{RSA} without changing h_t^ψ, h_t^ϕ significantly.

As a result, the objective for training the VLM, following architectures specified in (3.2) and (3.4), can be formulated as

$$\max_{\psi, \phi} \mathcal{L}_{VLM}(\psi, \phi) = \max_{\psi, \phi} \left(\mathcal{L}_{ELBO}(\psi, \phi) - C \cdot \mathcal{L}_{RSA}(\tilde{h}_t^\phi, h_t^\psi; \psi, \phi) \right), \quad (3.14)$$

with $C > 0$ and $C \in \mathbb{R}$ being the constant balancing the scale of the ELBO and RSA terms.

3.5 Branching for Generative Decoder

The performance of model-based methods can vary upon different design factors (Fu, Norouzi, et al., 2020; Hanin & Rolnick, 2018). Specifically, (Rossi et al., 2019) has

found that the convergence speed and optimality of variational models are sensitive to the choice of weight initialization techniques. Moreover, under the typical variational inference setup followed by the VLM above, the latent transitions reconstructed by the decoder, $p_\phi(z_t|z_{t-1}, a_{t-1})$, are only trained through regularization losses in (3.5) and (3.13), but are fully responsible for rolling out trajectories during evaluation. Consequently, in this subsection we introduce the branching architecture for decoder, with the goal of minimizing the impact brought by random weight initialization of the networks, and allowing the decoder to best reconstruct the latent transitions $p_\phi(z_t|z_{t-1}, a_{t-1})$ as well as s_t 's and r_{t-1} 's correctly. Specifically, the branching architecture leverages an ensemble of $B \in \mathbb{Z}^+$ decoders to tease out information from the latent space formulated by the encoder, with final predictions sampled from a mixture of the Gaussian output distributions from (3.4). Note that the classic setup of ensembles is not considered, *i.e.*, train and average over B VLMs end-to-end; because in this case B different latent space exist, each of which is still associated with a single decoder, leaving the challenges above unresolved. This design choice is justified by ablations studies in Section 3.6, by comparing VLBM against a (classic) ensemble of VLMs.

3.5.1 Branching Architecture.

Consider the generative process involving B branches of the decoders parameterized by $\{\phi_1, \dots, \phi_B\}$. The forward architecture over a single step is illustrated in Figure 3.2.³ Specifically, the procedure of sampling $z_t^{\phi_b}$ and $s_t^{\phi_b}$ for each $b \in [1, B]$ follows from (3.4). Recall that by definition $p_{\phi_b}(s_t|z_t^{\phi_b})$ follows multivariate Gaussian with mean and diagonal of covariance matrix determined by the corresponding MLPs, *i.e.*, $\mu(s_t^{\phi_b}) = \phi_{b,\mu}^{MLP}(z_t^{\phi_b})$ and $\Sigma_{diag}(s_t^{\phi_b}) = \phi_{b,\Sigma}^{MLP}(z_t^{\phi_b})$. In what follows, the final outcome s_t^ϕ can be sampled following diagonal Gaussian with mean and variance determined by weighted averaging across all branches using weights w_b 's, *i.e.*,

$$s_t^\phi \sim p_\phi(s_t|z_t^{\phi_1}, \dots, z_t^{\phi_B}) = \mathcal{N}\left(\boldsymbol{\mu} = \sum_b w_b \cdot \mu(s_t^{\phi_b}), \boldsymbol{\Sigma}_{diag} = \sum_b w_b^2 \cdot \Sigma_{diag}(s_t^{\phi_b})\right). \quad (3.15)$$

³ For simplicity, the parts generating rewards are omitted without loss of generality.

The objective below can be used to jointly update, w_b 's, ψ and ϕ_b 's, *i.e.*,

$$\begin{aligned} & \max_{\psi, \phi, w} \mathcal{L}_{VLBM}(\psi, \phi_1, \dots, \phi_B, w_1, \dots, w_B) \\ &= \max_{\psi, \phi, w} \left(\sum_{t=0}^T \log p_\phi(s_t^\phi | z_t^{\phi_1}, \dots, z_t^{\phi_B}) - C_1 \cdot \sum_b \mathcal{L}_{RSA}(\tilde{h}_t^{\phi_b}, h_t^\psi; \psi, \phi_b) + C_2 \sum_b \mathcal{L}_{ELBO}(\psi, \phi_b) \right), \\ & \text{s.t. } w_1, \dots, w_B > 0, \sum_b w_b = 1 \text{ and constants } C_1, C_2 > 0. \end{aligned} \quad (3.16)$$

Though the first term above already propagates through all w_b 's and ϕ_b 's, the third term and constraints over w_b 's regularize ϕ_b in each individual branch such that they are all trained toward maximizing the likelihood $p_{\phi_b}(s_t^{\phi_b} | z_t^{\phi_b})$. Pseudo-code for training and evaluating the VLBM can be found in Algorithms 1 and 2. Further, in practice, one can define $w_b = \frac{v_b^2}{\epsilon + \sum_b v_b^2}$, with $v_b \in \mathbb{R}$ the learnable variables and $0 < \epsilon \ll 1$, $\epsilon \in \mathbb{R}$, the constant ensuring denominator to be greater than zero, to convert (3.16) into unconstrained optimization and solve it using gradient descent. Lastly, note that complementary latent modeling methods, *e.g.*, latent overshooting from (Hafner et al., 2019), could be adopted in (3.16). However, we keep the objective straightforward, so that the source of performance improvements can be isolated.

Algorithm 1 Train VLBM.

Require: Model weights $\psi, \phi_1, \dots, \phi_B, w_1, \dots, w_B$, offline trajectories ρ^β , and learning rate α .

Ensure:

- 1: Initialize $\psi, \phi_1, \dots, \phi_B, w_1, \dots, w_B$
 - 2: **for** $iter$ in $1 : max_iter$ **do**
 - 3: Sample a trajectory $[(s_0, a_0, r_0, s_1), \dots, (s_{T-1}, a_{T-1}, r_{T-1}, s_T)] \sim \rho^\beta$
 - 4: $z_0^\psi \sim q_\psi(z_0 | s_0)$
 - 5: $z_0^{\phi_b} \sim p(z_0)$, for all $b \in [1, B]$
 - 6: Run forward pass of VLBM following (3.2), (3.4) and (3.15) for $t = 1 : T$, and collect all variables needed to evaluate \mathcal{L}_{VLBM} as specified in (3.16).
 - 7: $\psi \leftarrow \psi + \alpha \nabla_\psi \mathcal{L}_{VLBM}$
 - 8: **for** b in $1 : B$ **do**
 - 9: $\phi_b \leftarrow \phi_b + \alpha \nabla_{\phi_b} \mathcal{L}_{VLBM}$
 - 10: $w_b \leftarrow w_b + \alpha \nabla_{w_b} \mathcal{L}_{VLBM}$
 - 11: **end for**
 - 12: **end for**
-

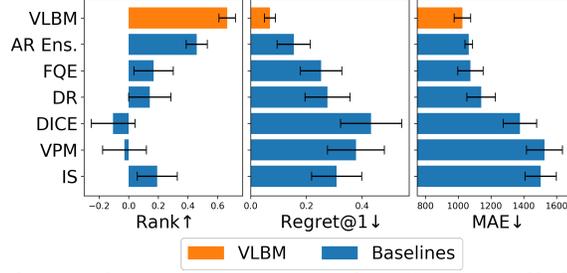


FIGURE 3.3: Mean rank correlation, regret@1 and MAE over all the 32 Gym-Mujoco and Adroit tasks, showing VLBM achieves state-of-the-art performance overall.

Algorithm 2 Evaluate VLBM.

Require: Trained model weights $\psi, \phi_1, \dots, \phi_B, w_1, \dots, w_B$

Ensure:

- 1: Initialize the list that stores the accumulated returns over all episodes $\mathcal{R} = []$
 - 2: **for** epi in $1 : max_epi$ **do**
 - 3: Initialize the variable $r = 0$ that tracks the accumulated return for the current episode
 - 4: Initialize latent states from the prior, *i.e.*, $z_0^{\phi_b} \sim p(z_0)$ for all $b \in [1, B]$
 - 5: Initialize LSTM hidden states $h_0^{\phi_b} = 0$ for all $b \in [1, B]$
 - 6: Sample $s_0^{\phi_b} \sim p_\phi(s_0 | z_0^{\phi_b})$ for all $b \in [1, B]$ and generate initial MDP state s_0^ϕ following (3.15)
 - 7: **for** t in $1 : T$ **do**
 - 8: Determine the action following the target policy π , *i.e.*, $a_{t-1} \sim \pi(a_{t-1} | s_{t-1}^\phi)$
 - 9: **for** b in $1 : B$ **do**
 - 10: Update $h_t^{\phi_b}, \tilde{h}_t^{\phi_b}, z_t^{\phi_b}, s_t^{\phi_b}, r_{t-1}^{\phi_b}$ following (3.4).
 - 11: **end for**
 - 12: Generate the next state s_t^ϕ following (3.15), as well as the reward $r_{t-1}^\phi \sim p_\phi(r_{t-1} | z_t^{\phi_1}, \dots, z_t^{\phi_B}) = \mathcal{N}(\mu = \sum_b w_b \cdot \mu(r_{t-1}^{\phi_b}), \Sigma_{diag} = \sum_b w_b^2 \cdot \Sigma_{diag}(r_{t-1}^{\phi_b}))$
 - 13: Update $r \leftarrow r + \gamma^{t-1} r_{t-1}^\phi$, with γ being the discounting factor
 - 14: **end for**
 - 15: Append r into \mathcal{R}
 - 16: **end for**
 - 17: Average over all elements in \mathcal{R} , which serves as the estimated return over π
-

3.6 Experiments

To evaluate the VLBM, we follow the guidelines from the deep OPE (DOPE) benchmark (Fu, Norouzi, et al., 2020). Specifically, we follow the D4RL branch in DOPE and use the Gym-Mujoco and Adroit suites as the test base (Fu, Kumar, et al., 2020). Such environments have long horizons and high-dimensional state and action space, which are

usually challenging for model-based methods. The provided offline trajectories for training are collected using behavioral policies at varied scale, including limited exploration, human teleoperation etc., which can result in different levels of coverage over the state-action space. Also, the target (evaluation) policies are generated using online RL training, aiming to reduce the similarity between behavioral and target policies; it introduces another challenge that during evaluation the agent may visit states unseen from training trajectories.

3.6.1 Environmental and Training Setup.

A total of 8 environments are provided by Gym-Mujoco and Adroit suites (Fu, Kumar, et al., 2020; Fu, Norouzi, et al., 2020). Moreover, each environment is provided with 5 (for Gym-Mujoco) or 3 (for Adroit) training datasets collected using different behavioral policies, resulting in a total of 32 sets of `env-dataset` tasks⁴. DOPE also provides 11 target policies for each environment, whose performance are to be evaluated by the OPE methods. They in general result in varied scales of returns, as shown in the x-axes of Figure 3.7. Moreover, we consider the decoder to have $B = 10$ branches, *i.e.*, $\{p_{\phi_1}, \dots, p_{\phi_{10}}\}$. The dimension of latent space is set to be 16, *i.e.*, $z \in \mathcal{Z} \subset \mathbb{R}^{16}$. The statistics pertaining to each environment are summarized in Tables 3.7 and 3.8.

3.6.2 Baselines and Evaluation Metrics.

In addition to the five baselines reported from DOPE, *i.e.*, importance sampling (IS) (Precup, 2000), doubly robust (DR) (P. Thomas & Brunskill, 2016), variational power method (VPM) (Wen et al., 2020), distribution correction estimation (DICE) (M. Yang et al., 2020), and fitted Q-evaluation (FQE) (Le et al., 2019), the effectiveness of VLBM is also compared against the state-of-the-art model-based OPE method leveraging the auto-regressive (AR) architecture (M. R. Zhang et al., 2020). Specifically, for each task we train an ensemble of 10 AR models, for fair comparisons against VLBM which leverages the branching architecture. Following the DOPE benchmark (Fu, Norouzi, et al., 2020), our evaluation

⁴ From now on the dataset names are abbreviated by their initials, *e.g.*, Ant-M-R refers to Ant-Medium-Replay.

metrics includes rank correlation, regret@1, and mean absolute error (MAE). VLBM and all baselines are trained using 3 different random seeds over each task, leading to the results reported below.

3.6.3 Ablation.

Four ablation baselines are also considered, *i.e.*, VLM, VLM+RSA, VLM+RSA(MSE) and VLM+RSA Ensemble. Specifically, VLM refers to the model introduced in Section 3.3, trained toward maximizing only the ELBO, *i.e.*, (3.5). Note that, arguably, VLM could be seen as the generalization of directly applying latent-models proposed in existing RL policy optimization literature (Hafner, Lillicrap, et al., 2020; Hafner, Lillicrap, et al., 2020; Hafner et al., 2019; Lee et al., 2020; C. Lu et al., 2022). The VLM+RSA ablation baseline follows the same model architecture as VLM, but is trained to optimize over both ELBO and recurrent state alignment (RSA) as introduced in (3.14), *i.e.*, branching is not used comparing to VLBM. The design of these two baselines can help analyze the effectiveness of the RSA loss term and branching architecture introduced in Section 3.4 and 3.5. Moreover, VLM+RSA(MSE) uses mean squared error to replace the pairwise loss introduced in (3.13), and the VLM+RSA Ensemble applies classic ensembles by averaging over B VLM+RSA models end-to-end, instead of branching from decoder as in VLBM. These two ablation baselines can help justify the use of pairwise loss for RSA, and the benefit of using branching architecture over classic ensembles.

3.6.4 Results.

Figure 3.3 shows the mean overall performance attained by VLBM and baselines over all the 32 Gym-Mujoco and Adroit tasks. In general VLBM leads to significantly increased rank correlations and decreased regret@1’s over existing methods, with MAEs maintained at the state-of-the-art level. Specifically, VLBM achieves state-of-the-art performance in 31, 29, and 15 (out of 32) tasks in terms of rank correlation, regret@1 and MAE, respectively. Performance for each task can be found in Tables 3.1- 3.6. Note that results for IS, VPM, DICE, DR, and FQE are obtained directly from DOPE benchmark (Fu, Norouzi, et al.,

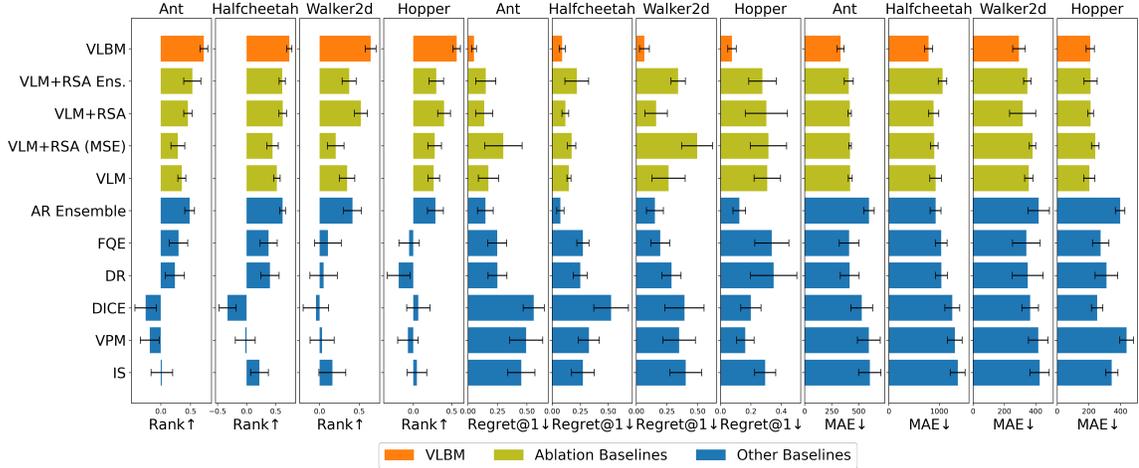


FIGURE 3.4: Mean rank correlation, regret@1 and MAE over all datasets, for each Mujoco environment.

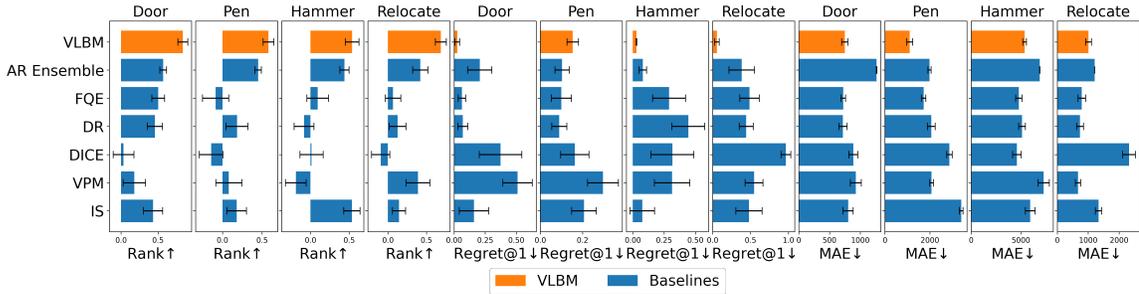


FIGURE 3.5: Mean rank correlation, regret@1 and MAE over all datasets, for each Adroit environment.

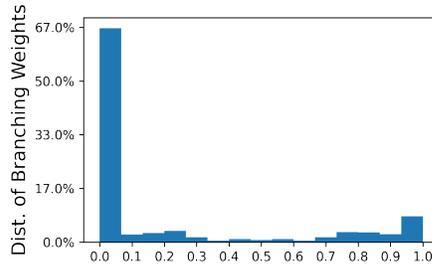


FIGURE 3.6: Distribution of all branching weights, w_b 's, over all VLBM models trained on the 32 tasks.

2020), since the same experimental setup is considered. Figure 3.4 and 3.5 visualize the mean performance for each Gym-Mujoco and Adroit environment respectively, over all the associated datasets. It can be also observed that the model-based and FQE baselines

generally perform better than the other baselines, which is consistent with findings from DOPE.

The fact that VLM+RSA outperforming the VLM ablation baseline, as shown in Figure 3.4, illustrates the need of the RSA loss term to smooth the flow of information between the encoder and decoder, in the latent space. Moreover, one can observe that VLM+RSA(MSE) sometimes performs worse than VLM, and significantly worse than VLM+RSA in general. Specifically, it has been found that, compared to VLM and VLM+RSA respectively, VLM+RSA(MSE) significantly worsen at least two metrics in 7 and 12 (out of 20) Gym-Mujoco tasks; detailed performance over these tasks can be found in Tables 3.1- 3.6 at the end of Appendices. Such a finding backs up the design choice of using pairwise loss for RSA instead of MSE, as MSE could be overly strong to regularize the LSTM recurrent states of the encoder and decoder, while pairwise loss only enforces structural similarities. Moreover, VLBM significantly improves rank correlations and regrets greatly compared to VLM+RSA, illustrating the importance of the branching architecture. In the paragraph below, we show empirically the benefits brought in by branching over classic ensembles.

3.6.5 Branching versus Classic Ensembles.

Figure 3.4 shows that the VLM+RSA Ensemble does not improve performance over the VLM+RSA in general, and even leads to worse overall rank correlations and regrets in Walker2d and Hopper environments. This supports the rationale provided in Section 3.5 that each decoder still samples from different latent space exclusively, and averaging over the output distributions may not help reduce the disturbance brought in by the modeling artifacts under the variational inference framework, *e.g.*, random weight initializations (Hanin & Rolnick, 2018; Rossi et al., 2019). In contrast, the VLBM leverages the branching architecture, allowing all the branches to sample from the same latent space formulated by the encoder. Empirically, we find that the branching weights, w_b 's in (3.15), allows VLBM to kill branches that are not helpful toward reconstructing the trajectories accurately, to possibly overcome bad initializations etc. Over all the the 32 tasks we consider, most of

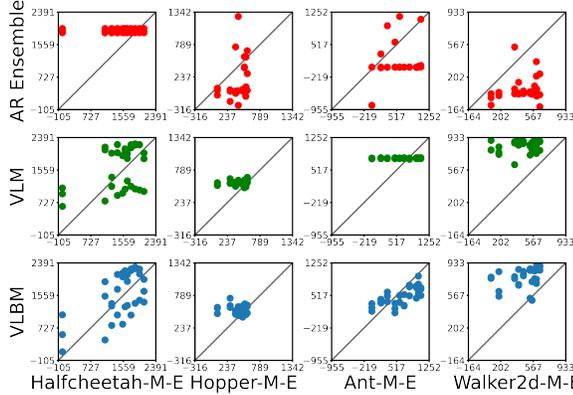


FIGURE 3.7: Correlation between the estimated (y-axis) and true returns (x-axis), across different model-based OPE methods and environments.

VLBMs only keep 1-3 branches (out of 10), *i.e.*, $w_b < 10^{-5}$ for all other branches. The distribution of all w_b 's, from VLBM's trained on the 32 tasks, are shown in Figure 3.6; one can observe that most of the w_b 's are close to zero, while the others generally fall in the range of $(0, 0.25]$ and $[0.75, 1)$.

AR ensembles also lead to compelling rank correlations and regrets, but attains much smaller margins in MAEs over other baselines in general; see Figure 3.3. From Figure 3.7, one can observe that it tends to significantly under-estimate most of the high-performing policies. The reason could be that its model architecture and training objectives are designed to directly learn the transitions of the MDP; thus, may produce biased predictions when the target policies lead to visitation of the states that are not substantially presented in training data, since such data are obtained using behavioral policies that are sub-optimal. In contrast, the VLBM can leverage RSA and branching against such situations, thus outperforming AR ensembles in most of the OPE tasks in terms of all metrics we considered. Interestingly, Figure 3.7 also shows that latent models could sometimes over-estimate the returns. For example, in Hopper-M-E and Walker2d-M-E, VLM tends to over-estimate most policies. The VLBM performs consistently well in Hopper-M-E, but is mildly affected by such an effect in Walker2d-M-E, though over fewer policies and smaller margins. It has been found that variational inference may fall short in approximating true distributions

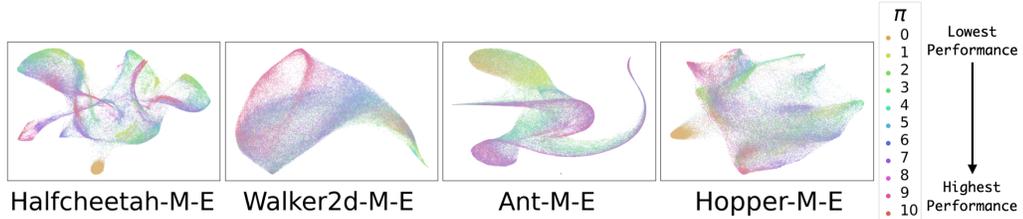


FIGURE 3.8: t -SNE visualization over the latent space, capturing encoded state-action visitations induced from all target policies. Each point is colored by the corresponding policy from which it is generated. Policies in the legend are sorted in the order of increasing performance.

that are asymmetric, and produce biased estimations (Yao et al., 2018). So the hypothesis would be that the dynamics used to define certain environments may lead to asymmetry in the true posterior $p(z_t|z_{t-1}, a_{t-1}, s_t)$, which could be hard to be captured by the latent modeling framework we consider. More comprehensive understanding of such behavior can be explored in future work. However, the VLBM still significantly outperforms VLM overall, and achieves top-performing rank correlations and regrets; such results illustrate the VLBM’s improved robustness as a result of its architectural design and choices over training objectives.

3.6.6 t -SNE Visualization of the Latent Space.

Figure 3.8 illustrates t -SNE visualization of the latent space by rolling out trajectories using all target policies respectively, followed by feeding the state-action pairs into the encoder of VLBM which maps them into the latent space. It shows the encoded state-action pairs induced from policies with similar performance are in general swirled and clustered together, illustrating that VLBM can learn expressive and disentangled representations of its inputs.

Table 3.1: Rank correlation between estimated and ground-truth returns for all Gym-Mujoco tasks. Results are obtained by averaging over 3 random seeds used for training, with standard deviations shown in subscripts.

Rank Corr.	Ant -E	Ant -M-E	Ant -M	Ant -M-R	Ant -R
IS	.14 _{.41}	-.21 _{.35}	-.17 _{.32}	.07 _{.39}	.26 _{.34}
VPM	-.42 _{.38}	-.28 _{.28}	-.2 _{.31}	-.26 _{.29}	.24 _{.31}
DICE	-.13 _{.37}	-.33 _{.4}	-.36 _{.28}	-.24 _{.39}	-.21 _{.35}
DR	-.28 _{.32}	.35 _{.35}	.66 _{.26}	.45 _{.32}	.01 _{.33}
FQE	-.13 _{.32}	.37 _{.35}	.65 _{.25}	.57 _{.28}	.04 _{.33}
AR Ensemble	.40 _{.12}	.44 _{.25}	.56 _{.01}	.54 _{.16}	.48 _{.17}
VLM	.28 _{.14}	.39 _{.16}	.37 _{.03}	.37 _{.19}	.36 _{.07}
VLM+RSA (MSE)	.33 _{.11}	.29 _{.13}	.35 _{.22}	.30 _{.42}	.17 _{.14}
VLM+RSA	.40 _{.03}	.53 _{.19}	.42 _{.12}	.53 _{.19}	.40 _{.11}
VLM+RSA Ens.	.62 _{.16}	.76 _{.02}	.65 _{.07}	.62 _{.13}	0. _{.60}
VLBM	.79 _{.01}	.81 _{.05}	.65 _{.06}	.59 _{.14}	.78 _{.24}
Rank Corr.	Halfcheetah -E	Halfcheetah -M-E	Halfcheetah -M	Halfcheetah -M-R	Halfcheetah -R
IS	.01 _{.35}	-.06 _{.37}	.80 _{.11}	.59 _{.26}	-.24 _{.36}
VPM	.18 _{.35}	-.47 _{.29}	-	-.07 _{.36}	.27 _{.36}
DICE	-.44 _{.30}	-.08 _{.35}	-.26 _{.07}	-.15 _{.41}	-.70 _{.22}
DR	.77 _{.17}	.62 _{.27}	.32 _{.32}	.32 _{.37}	-.02 _{.38}
FQE	.78 _{.15}	.62 _{.27}	.34 _{.17}	.26 _{.37}	-.11 _{.41}
AR Ensemble	.65 _{.11}	.65 _{.07}	.60 _{.09}	.59 _{.14}	.60 _{.06}
VLM	.75 _{.19}	.45 _{.06}	.33 _{.1}	.64 _{.06}	.43 _{.09}
VLM+RSA (MSE)	.54 _{.31}	.49 _{.03}	.6 _{.08}	.47 _{.11}	.13 _{.27}
VLM+RSA	.80 _{.17}	.54 _{.08}	.65 _{.21}	.61 _{.03}	.51 _{.08}
VLM+RSA Ens.	.71 _{.14}	.66 _{.08}	.64 _{.02}	.60 _{.05}	.45 _{.17}
VLBM	.88 _{.01}	.74 _{.13}	.81 _{.13}	.64 _{.04}	.60 _{.06}
Rank Corr.	Walker2d -E	Walker2d -M-E	Walker2d -M	Walker2d -M-R	Walker2d -R
IS	.22 _{.37}	.24 _{.33}	-.25 _{.35}	.65 _{.24}	-.05 _{.38}
VPM	.17 _{.32}	.49 _{.37}	.44 _{.21}	-.52 _{.25}	-.42 _{.34}
DICE	-.37 _{.27}	-.34 _{.34}	.12 _{.38}	.55 _{.23}	-.19 _{.36}
DR	.26 _{.34}	.19 _{.33}	.02 _{.37}	-.37 _{.39}	.16 _{.29}
FQE	.35 _{.33}	.25 _{.32}	-.09 _{.36}	-.19 _{.36}	.21 _{.31}
AR Ensemble	.54 _{.11}	.25 _{.33}	.55 _{.14}	.38 _{.17}	.36 _{.29}
VLM	.57 _{.13}	.16 _{.13}	.18 _{.30}	.39 _{.18}	.44 _{.18}
VLM+RSA (MSE)	.27 _{.28}	.20 _{.25}	.09 _{.18}	.10 _{.11}	.36 _{.19}
VLM+RSA	.56 _{.11}	.57 _{.11}	.46 _{.08}	.43 _{.14}	.59 _{.29}
VLM+RSA Ens.	.62 _{.17}	.57 _{.25}	.43 _{.20}	-.14 _{.09}	.39 _{.14}
VLBM	.70 _{.13}	.55 _{.17}	.66 _{.15}	.60 _{.07}	.72 _{.14}
Rank Corr.	Hopper -E	Hopper -M-E	Hopper -M	Hopper -M-R	Hopper -R
IS	.37 _{.27}	.35 _{.26}	-.55 _{.26}	-.16 _{.03}	.23 _{.34}
VPM	.21 _{.32}	-	.13 _{.37}	-.16 _{.03}	-.46 _{.20}
DICE	-.08 _{.32}	.08 _{.14}	.19 _{.33}	.27 _{.28}	-.13 _{.39}
DR	-.41 _{.27}	-.08 _{.30}	-.31 _{.34}	.05 _{.17}	-.19 _{.36}
FQE	-.33 _{.30}	.01 _{.08}	-.29 _{.33}	.45 _{.13}	-.11 _{.36}
AR Ensemble	.23 _{.30}	.14 _{.29}	.53 _{.03}	.28 _{.18}	.26 _{.10}
VLM	-.05 _{.22}	.22 _{.11}	.34 _{.08}	.46 _{.21}	.36 _{.03}
VLM+RSA (MSE)	-.18 _{.24}	.05 _{.09}	.51 _{.20}	.43 _{.18}	.58 _{.14}
VLM+RSA	.15 _{.28}	.26 _{.10}	.51 _{.11}	.53 _{.06}	.55 _{.19}
VLM+RSA Ens.	.09 _{.21}	.13 _{.12}	-.01 _{.3}	.66 _{.07}	.63 _{.16}
VLBM	.28 _{.16}	.32 _{.10}	.70 _{.03}	.75 _{.07}	.77 _{.04}

Table 3.2: Rank correlation between estimated and ground-truth returns for all Adroit tasks. Results are obtained by averaging over 3 random seeds used for training, with standard deviations shown in subscripts.

Rank Corr.	Door human	Door cloned	Door expert	Pen human	Pen cloned	Pen expert
IS	-.12 _{.35}	.66 _{.22}	.76 _{.17}	.28 _{.28}	.71 _{.08}	-.45 _{.31}
VPM	-	-.29 _{.36}	.65 _{.23}	-	-	.08 _{.33}
DICE	-.02 _{.20}	.18 _{.31}	-.06 _{.32}	.17 _{.33}	-.07 _{.26}	-.53 _{.30}
DR	.01 _{.18}	.60 _{.28}	.76 _{.13}	-.36 _{.29}	.39 _{.25}	.52 _{.28}
FQE	.07 _{.09}	.55 _{.27}	.89_{.09}	-.31 _{.21}	.06 _{.42}	-.01 _{.33}
AR Ens.	.58 _{.06}	.52 _{.13}	.61 _{.07}	.33_{.07}	.42 _{.08}	.60_{.09}
VLBM	.80_{.14}	.78_{.18}	.93_{.03}	.34_{.17}	.82_{.07}	.58_{.15}
Rank Corr.	Hammer human	Hammer cloned	Hammer expert	Relocate human	Relocate cloned	Relocate expert
IS	.39_{.07}	.58_{.27}	.64_{.24}	-.23 _{.07}	-.22 _{.18}	.52_{.23}
VPM	-	-.77 _{.22}	.39 _{.31}	-	-	.39 _{.31}
DICE	.11 _{.18}	.35 _{.38}	-.42 _{.31}	-.23 _{.16}	.22 _{.16}	-.27 _{.34}
DR	-.04 _{.25}	-.70 _{.20}	.49 _{.31}	.65_{.19}	.10 _{.16}	-.40 _{.24}
FQE	.14 _{.10}	-.15 _{.33}	.29 _{.34}	.62_{.11}	.15 _{.17}	-.57 _{.28}
AR Ens.	.44_{.12}	.40 _{.20}	.53 _{.11}	.42 _{.23}	.30 _{.10}	.54_{.23}
VLBM	.34 _{.14}	.58_{.18}	.70_{.20}	.68_{.17}	.80_{.04}	.58_{.17}

Table 3.3: Regret@1 for all Gym-Mujoco tasks. Results are obtained by averaging over 3 random seeds used for training, with standard deviations shown in subscripts.

Regret@1	Ant -E	Ant -M-E	Ant -M	Ant -M-R	Ant -R
IS	.47 _{.32}	.46 _{.18}	.61 _{.18}	.16 _{.23}	.56 _{.22}
VPM	.88 _{.22}	.32 _{.24}	.4 _{.21}	.72 _{.43}	.15 _{.24}
DICE	.62 _{.15}	.60 _{.16}	.43 _{.1}	.64 _{.13}	.50 _{.29}
DR	.43 _{.22}	.37 _{.13}	.12 _{.18}	.05 _{.09}	.28 _{.15}
FQE	.43 _{.22}	.36 _{.14}	.12 _{.18}	.05 _{.09}	.28 _{.15}
AR Ensemble	.18 _{.09}	.17 _{.20}	.05 ₀	.31 _{.20}	.03 _{.02}
VLM	.38 _{.24}	.07 _{.02}	.20 _{.25}	.08 _{.02}	.14 _{.16}
VLM+RSA (MSE)	.05 ₀	.26 _{.21}	.28 _{.4}	.48 _{.33}	.43 _{.44}
VLM+RSA	.18 _{.09}	.13 _{.12}	.14 _{.16}	.17 _{.24}	.07 _{.02}
VLM+RSA Ens.	.13 _{.08}	.05 ₀	.03 _{.02}	.03 _{.02}	.52 _{.37}
VLBM	.05 ₀	.05 ₀	.05 ₀	.11 _{.09}	0 _{.0}
Regret@1	Halfcheetah -E	Halfcheetah -M-E	Halfcheetah -M	Halfcheetah -M-R	Halfcheetah -R
IS	.15 _{.08}	.73 _{.42}	.05 _{.05}	.13 _{.10}	.31 _{.11}
VPM	.14 _{.09}	.80 _{.34}	.33 _{.19}	.25 _{.09}	.12 _{.07}
DICE	.32 _{.40}	.38 _{.37}	.82 _{.29}	.30 _{.07}	.81 _{.30}
DR	.11 _{.08}	.14 _{.07}	.37 _{.15}	.33 _{.18}	.31 _{.10}
FQE	.12 _{.07}	.14 _{.07}	.38 _{.13}	.36 _{.16}	.37 _{.08}
AR Ensemble	.02 _{.03}	.11 _{.07}	.13 _{.10}	.07 _{.05}	.04 _{.05}
VLM	.11 _{.04}	.12 _{.06}	.25 _{.01}	.04 _{.03}	.23 ₀
VLM+RSA (MSE)	.09 _{.08}	.22 _{.09}	.20 _{.06}	.09 _{.08}	.27 _{.05}
VLM+RSA	.08 _{.02}	.17 _{.05}	.09 _{.12}	.02 _{.03}	.23 ₀
VLM+RSA Ens.	.13 _{.05}	.19 _{.13}	.07 _{.09}	.02 _{.03}	.69 _{.44}
VLBM	.14 _{.04}	.09 _{.02}	0 _{.0}	.07 _{.09}	.15 _{.07}
Regret@1	Walker2d -E	Walker2d -M-E	Walker2d -M	Walker2d -M-R	Walker2d -R
IS	.43 _{.26}	.13 _{.07}	.70 _{.39}	.02 _{.05}	.74 _{.33}
VPM	.09 _{.19}	.24 _{.42}	.08 _{.06}	.46 _{.31}	.88 _{.20}
DICE	.35 _{.36}	.78 _{.27}	.27 _{.43}	.18 _{.12}	.39 _{.33}
DR	.06 _{.07}	.30 _{.12}	.25 _{.09}	.68 _{.23}	.15 _{.20}
FQE	.06 _{.07}	.22 _{.14}	.31 _{.10}	.24 _{.20}	.15 _{.21}
AR Ensemble	.13 _{.11}	.17 _{.19}	.16 _{.15}	.14 _{.16}	.16 _{.02}
VLM	.10 _{.05}	.51 _{.25}	.30 _{.39}	.33 _{.38}	.08 _{.07}
VLM+RSA (MSE)	.49 _{.16}	.39 _{.30}	.43 _{.35}	.86 ₀	.31 _{.29}
VLM+RSA	.10 _{.07}	.11 _{.02}	.18 _{.15}	.34 _{.37}	.08 _{.04}
VLM+RSA Ens.	.11 _{.04}	.14 _{.16}	.02 _{.02}	.86 ₀	.58 _{.20}
VLBM	.05 _{.04}	.05 _{.01}	.03 _{.04}	.14 _{.16}	.06 _{.06}
Regret@1	Hopper -E	Hopper -M-E	Hopper -M	Hopper -M-R	Hopper -R
IS	.06 _{.03}	.10 _{.12}	.38 _{.28}	.88 ₀	.05 _{.05}
VPM	.13 _{.10}	-	.10 _{.14}	-	.26 _{.10}
DICE	.20 _{.08}	.16 _{.08}	.18 _{.19}	.16 _{.13}	.30 _{.15}
DR	.34 _{.35}	.34 _{.39}	.32 _{.32}	.34 _{.24}	.41 _{.17}
FQE	.41 _{.20}	.42 _{.08}	.32 _{.32}	.18 _{.23}	.36 _{.22}
AR Ensemble	.07 _{.05}	.23 _{.11}	.14 _{.09}	.06 _{.02}	.12 _{.11}
VLM	.76 _{.18}	.35 _{.22}	.22 _{.22}	.14 _{.15}	.07 _{.02}
VLM+RSA (MSE)	.42 _{.34}	.51 ₀	.33 _{.39}	.26 _{.13}	.06 _{.04}
VLM+RSA	.62 _{.38}	.18 _{.23}	.13 _{.12}	.25 _{.15}	.33 _{.39}
VLM+RSA Ens.	.31 _{.18}	.51 ₀	.47 _{.36}	.03 _{.02}	.06 _{.04}
VLBM	.10 _{.03}	.10 _{.03}	.11 _{.11}	.04 ₀	.03 _{.04}

Table 3.4: Regret@1 for all Adroit tasks. Results are obtained by averaging over 3 random seeds used for training, with standard deviations shown in subscripts.

Regret@1	Door human	Door cloned	Door expert	Pen human	Pen cloned	Pen expert
IS	.45 _{.40}	.02 _{.07}	.01 _{.04}	.17 _{.15}	.14 _{.09}	.31 _{.10}
VPM	.69 _{.24}	.81 _{.33}	.03 _{.03}	.28 _{.12}	.36 _{.18}	.25 _{.13}
DICE	.10 _{.27}	.65 _{.45}	.37 _{.27}	.04 _{.09}	.12 _{.08}	.33 _{.20}
DR	.05 _{.09}	.11 _{.08}	.05 _{.07}	.09 _{.08}	.13 _{.06}	.05 _{.07}
FQE	.05 _{.08}	.11 _{.06}	.03 _{.03}	.07 _{.05}	.12 _{.07}	.11 _{.14}
AR Ens.	.08 _{.10}	.44 _{.31}	.10 _{.09}	.09 _{.08}	.14 _{.05}	.08 _{.07}
VLBM	.03 _{.04}	.03 _{.04}	.02 _{.03}	.29 _{.07}	.08 _{.06}	.09 _{.02}
Regret@1	Hammer human	Hammer cloned	Hammer expert	Relocate human	Relocate cloned	Relocate expert
IS	.19 _{.30}	.03 _{.15}	.01 _{.04}	.63 _{.41}	.63 _{.41}	.18 _{.14}
VPM	.18 _{.29}	.72 _{.39}	.04 _{.07}	.77 _{.18}	.11 _{.29}	.76 _{.23}
DICE	.04 _{.08}	.67 _{.48}	.24 _{.34}	.97 _{.11}	.96 _{.18}	.97 _{.07}
DR	.46 _{.23}	.78 _{.38}	.09 _{.09}	.17 _{.15}	.18 _{.27}	.98 _{.08}
FQE	.46 _{.23}	.36 _{.39}	.05 _{.04}	.17 _{.14}	.29 _{.42}	1.00 _{.06}
AR Ens.	.08 _{.06}	.05 _{.05}	0 _{.0}	.26 _{.33}	.63 _{.35}	.26 _{.33}
VLBM	.08 _{.0}	0 _{.0}	.01 _{.01}	.08 _{.08}	.02 _{.02}	.07 _{.07}

Table 3.5: MAE between estimated and ground-truth returns for all Gym-Mujoco tasks. Results are obtained by averaging over 3 random seeds used for training, with standard deviations shown in subscripts.

MAE	Ant -E	Ant -M-E	Ant -M	Ant -M-R	Ant -R
IS	605 ₁₀₄	604 ₁₀₂	594 ₁₀₄	603 ₁₀₁	606 ₁₀₃
VPM	607 ₁₀₈	604 ₁₀₆	570 ₁₀₉	612 ₁₀₅	570 ₉₉
DICE	558 ₁₀₈	471 ₁₀₀	495 ₉₀	583 ₁₁₀	530 ₉₂
DR	584 ₁₁₄	326 ₆₆	345 ₆₆	421 ₇₂	404 ₁₀₆
FQE	583 ₁₂₂	319 ₆₇	345 ₆₄	410 ₇₉	398 ₁₁₁
AR Ensemble	551 ₈₁	629 ₁₄	574 ₃₅	642 ₁	575 ₆₁
VLM	331 ₁₅	315 ₂₀	310 ₃₁	486 ₆	663 ₂
VLM+RSA (MSE)	343 ₁₃	324 ₄	306 ₃	463 ₂₁	661 ₈
VLM+RSA	351 ₇	314 ₂₃	305 ₂₅	448 ₃	665 ₄
VLM+RSA Ens.	242 ₂₀	312 ₃₇	345 ₈₀	464 ₆	667 ₂₀
VLBM	202 ₄	269 ₅₅	331 ₄₃	265 ₂	598 ₁₁
MAE	Halfcheetah -E	Halfcheetah -M-E	Halfcheetah -M	Halfcheetah -M-R	Halfcheetah -R
IS	1404 ₁₅₂	1400 ₁₄₆	1217 ₁₂₃	1409 ₁₅₄	1405 ₁₅₅
VPM	945 ₁₆₄	1427 ₁₁₁	1374 ₁₅₃	1384 ₁₄₈	1411 ₁₅₄
DICE	944 ₁₆₁	1078 ₁₃₂	1382 ₁₃₀	1440 ₁₅₈	1446 ₁₅₆
DR	1025 ₉₅	1015 ₁₀₃	1222 ₁₃₄	1001 ₁₂₉	949 ₁₂₆
FQE	1031 ₉₅	1014 ₁₀₁	1211 ₁₃₀	1003 ₁₃₂	938 ₁₂₅
AR Ensemble	1226 ₂₂₂	480 ₂₄	553 ₆₄	846 ₆₄	1537 ₁₆
VLM	520 ₂₄₂	526 ₄₉	624 ₅₃	1478 ₂₇	1490 ₁
VLM+RSA (MSE)	469 ₁₅₉	426 ₄₉	689 ₃₉	1432 ₁₀	1489 ₀
VLM+RSA	414 ₁₅₅	446 ₅₀	622 ₁₅₃	1473 ₂₀	1492 ₆
VLM+RSA Ens.	253 ₂₀	773 ₁₃₉	1306 ₁₁₃	1468 ₄₁	1525 ₂₂
VLBM	201 ₂₂	456 ₃₀	517 ₅₀	1281 ₁₇₀	1495 ₂
MAE	Walker2d -E	Walker2d -M-E	Walker2d -M	Walker2d -M-R	Walker2d -R
IS	405 ₆₂	436 ₆₂	428 ₆₀	427 ₆₀	430 ₆₁
VPM	367 ₆₈	425 ₆₁	426 ₆₀	424 ₆₄	440 ₅₈
DICE	437 ₆₀	322 ₆₀	273 ₃₁	374 ₅₁	419 ₅₇
DR	519 ₁₇₉	217 ₄₆	368 ₇₄	296 ₅₄	347 ₇₄
FQE	453 ₁₄₂	233 ₄₂	350 ₇₉	313 ₇₃	354 ₇₃
AR Ensemble	530 ₁₀₂	408 ₄	444 ₆	327 ₁₀₆	383 ₄₂
VLM	538 ₃₀	380 ₁₂	250 ₁₇	160 ₄₆	452 ₈
VLM+RSA (MSE)	521 ₃₀	340 ₂₀	361 ₁₉	236 ₁₄	443 ₁₅
VLM+RSA	522 ₄₁	358 ₈₆	253 ₉	125 ₇	326 ₁₆₁
VLM+RSA Ens.	538 ₉	386 ₂₃	201 ₃₈	168 ₁₁	441 ₂₄
VLBM	517 ₂₄	288 ₇₂	244 ₃₃	156 ₂₈	262 ₂₂
MAE	Hopper -E	Hopper -M-E	Hopper -M	Hopper -M-R	Hopper -R
IS	106 ₂₉	360 ₄₇	405 ₄₈	438 ₁₁	412 ₄₅
VPM	442 ₄₃	-	433 ₄₄	-	438 ₄₄
DICE	259 ₅₄	266 ₄₀	215 ₄₁	398 ₂	122 ₁₆
DR	426 ₉₉	234 ₇₇	307 ₈₅	298 ₁₄	289 ₅₀
FQE	282 ₇₆	252 ₂₈	283 ₇₃	295 ₇	261 ₄₂
AR Ensemble	369 ₁₆	292 ₁₁	393 ₄₂	477 ₃₄	454 ₃₄
VLM	148 ₃₁	136 ₁₉	210 ₂₂	138 ₉	382 ₆₆
VLM+RSA (MSE)	246 ₄₀	186 ₁₀	232 ₂₉	124 ₁₂	415 ₁₅
VLM+RSA	270 ₂	140 ₁₅	117 ₂₈	117 ₁₆	412 ₂₀
VLM+RSA Ens.	253 ₂₃	149 ₄₂	233 ₆₂	115 ₁₇	306 ₄₇
VLBM	266 ₈	140 ₄	126 ₄₇	124 ₂₁	385 ₂₇

Table 3.6: MAE between estimated and ground-truth returns for all Adroit tasks. Results are obtained by averaging over 3 random seeds used for training.

MAE	Door human	Door cloned	Door expert	Pen human	Pen cloned	Pen expert
IS	870 ₁₇₃	891 ₁₈₈	648 ₁₂₂	3926 ₁₂₈	1707 ₁₂₈	4547 ₂₂₂
VPM	862 ₁₆₃	1040 ₁₈₈	879 ₁₈₂	1569 ₂₁₅	2324 ₁₂₉	2325 ₁₃₆
DICE	1108 ₁₉₉	697 ₇₉	856 ₁₃₄	4193 ₂₄₄	1454 ₂₁₉	2963 ₂₇₉
DR	379 ₆₅	424 ₇₃	1353 ₂₁₈	2846 ₂₀₀	1323 ₉₈	2013 ₅₆₄
FQE	389 ₆₀	438 ₈₁	1343 ₈₄	2872 ₁₇₀	1232 ₁₀₅	1057 ₂₈₁
AR Ens.	734 ₃	826 ₇	2236 ₁₆	2161 ₁₂	1981 ₁₀₆	1803 ₂₂₆
VLBM	710 ₁₅₂	933 ₁	600 ₈₄	1637 ₂₈₆	669 ₂₇₀	1002 ₂₆₂
MAE	Hammer human	Hammer cloned	Hammer expert	Relocate human	Relocate cloned	Relocate expert
IS	7352 ₁₁₁₈	7403 ₁₁₂₆	3052 ₆₀₈	638 ₂₁₇	632 ₂₁₅	2731 ₁₄₇
VPM	7105 ₁₁₀₇	7459 ₁₁₁₄	7312 ₁₁₁₇	806 ₁₆₆	586 ₁₃₅	620 ₂₁₄
DICE	5677 ₉₃₆	4169 ₈₃₉	3963 ₇₅₈	4526 ₄₇₄	1347 ₄₈₅	1095 ₂₂₁
DR	5768 ₇₅₁	6101 ₆₇₉	3485 ₅₉₀	606 ₁₁₆	412 ₁₂₄	1193 ₃₅₀
FQE	6000 ₆₁₂	5415 ₅₅₈	2950 ₇₂₈	593 ₁₁₃	439 ₁₂₅	1351 ₃₉₃
AR Ens.	6897 ₂₇	7240 ₁₂	3057 ₈	823 ₇	662 ₆	2138 ₄
VLBM	6184 ₄₇₉	7267 ₄₀₂	2682 ₁₄₆	624 ₂₅	388 ₁₈₃	2021 ₂₇₀

Table 3.7: Summary of the Gym-Mujoco environments and datasets used to train VLBM and baselines.

	State Dim.	Action Dim.	Early Term.	Continuous Ctrl.	Dataset	Dataset Size
Ant	27	8	Yes	Yes	random	999,427
					medium-replay	301,698
					medium	999,175
					medium-expert	1,998,158
					expert	999,036
Halfcheetah	17	6	No	Yes	random	999,000
					medium-replay	201,798
					medium	999,000
					medium-expert	1,998,000
					expert	999,000
Hopper	11	3	Yes	Yes	random	999,999
					medium-replay	401,598
					medium	999,998
					medium-expert	1,998,966
					expert	999,061
Walker2d	17	6	Yes	Yes	random	999,999
					medium-replay	301,698
					medium	999,322
					medium-expert	1,998,318
					expert	999,000

Table 3.8: Summary of the Adroit environments and datasets used to train VLBM and baselines.

	State Dim.	Action Dim.	Early Term.	Continuous Ctrl.	Dataset	Dataset Size
Pen	45	24	Yes	Yes	human	4,975
					cloned	496,264
					expert	494,248
Door	39	28	No	Yes	human	6,704
					cloned	995,642
					expert	995,000
Hammer	46	26	No	Yes	human	11,285
					cloned	996,394
					expert	995,000
Relocate	39	30	No	Yes	human	9,917
					cloned	996,242
					expert	995,000

4. Off-Policy Evaluation for Human Feedback (OPEHF)

In this chapter, we introduce an OPE for HF (OPEHF) framework that revives existing OPE methods in order to accurately evaluate the HF signals. Specifically, we develop an immediate human reward (IHR) reconstruction approach, regularized by environmental knowledge distilled in a latent space that captures the underlying dynamics of state transitions as well as issuing HF signals. Our approach has been tested over *two real-world experiments*, adaptive *in-vivo* neurostimulation and intelligent tutoring, as well as in a simulation environment (visual Q&A). Results show that our approach significantly improves the performance toward estimating HF signals accurately, compared to directly applying (variants of) existing OPE methods.

4.1 Related Work

This section introduces the works related to the topic being considered, including reinforcement learning from human feedback (RLHF) and reward shaping.

4.1.1 Reinforcement Learning from Human Feedback (RLHF)

Recently, the concept of RLHF has been widely used in guiding RL policy optimization with the HF signals deemed more informative than the environmental rewards (Christiano et al., 2017; MacGlashan et al., 2017; Ziegler et al., 2019). Specifically, they leverage the *ranked preference* provided by labelers to train a reward model, captured by feed-forward neural networks, that is fused with the environmental rewards to guide policy optimization. However, in this work, we focus on estimating the HF signals that serve as *direct evaluation* of the RL policies used in human-involved experiments, such as the level of satisfaction (*e.g.*, on a scale 1-10) and the treatment outcome. The reason is that in many scenarios the participants cannot revisit the same procedure multiple times, *e.g.*, patients may not undergo the same surgeries several times and rank the experiences. More importantly, OPEHF’s setup is critical when online testing of RL policies may be even prohibited, without sufficient justifications over safety and efficacy upfront, as illustrated by the experiments above.

4.1.2 Reward Shaping

Although reward shaping methods (Arjona-Medina et al., 2019; Han et al., 2022; Patil et al., 2020) pursue similar ideas of decomposing the delayed and/or sparse rewards (*e.g.*, the human return) into immediate rewards, they fundamentally rely on transforming the MDP to such that the value functions can be smoothly captured and high-return state-action pairs can be quickly identified and frequently re-visited. For example, RUDDER (Arjona-Medina et al., 2019) leverages the transformed MDP that has expected future rewards equal to zero. Though the optimization objective is consistent between pre- and post-transformed MDPs, this approach likely would not converge to an optimal policy in practice. On the other hand, the performance (*i.e.*, returns) of sub-optimal policies is not preserved across the two MDPs. This significantly limits its use cases toward OPE which requires the returns resulted by sub-optimal policies to be estimated accurately. As a result, such methods are not directly applicable to the OPEHF problem we consider.

4.2 Problem Formulation

We first formulate the human-involved MDP (HMDP), which is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, R, R^{\mathcal{H}}, s_0, \gamma)$, where \mathcal{S} is the set of states, \mathcal{A} the set of actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the transition distribution usually captured by probabilities $p(s_t | s_{t-1}, a_{t-1})$, $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the *environmental* reward function, $R^{\mathcal{H}}(r^{\mathcal{H}} | s, a)$ is the *human* reward distribution from which the IHR $r_t^{\mathcal{H}} \sim R^{\mathcal{H}}(\cdot | s_t, a_t)$ are sampled, s_0 is the initial state sampled from the initial state distribution $p(s_0)$, and $\gamma \in [0, 1)$ is the discounting factor. Note that we set the IHRs to be determined probabilistically, as opposed to the environmental rewards $r_t = R(s_t, a_t)$ that are deterministic; this is due to the fact that many underlying factors may affect the feedback provided by humans (Chesnaye et al., 2022; Lis et al., 2015; Namkoong et al., 2020), as we have also observed while performing human-involved experiments. Finally, the agent interacts with the MDP following some policy $\pi(a | s)$ that defines the probabilities of taking action a at state s . We then make the following assumption over R and $R^{\mathcal{H}}$.

Assumption 1 (Unknown IHRs). *We assume that the immediate environmental reward func-*

tion R is known and $R(s, a)$ can be obtained for any state-action pairs in $\mathcal{S} \times \mathcal{A}$. Moreover, the IHR distribution $R^{\mathcal{H}}$ is assumed to be unknown, i.e., $r^{\mathcal{H}} \sim R^{\mathcal{H}}(\cdot|s, a)$ are unobservable, for all $(s, a) \in \mathcal{S} \times \mathcal{A}$. Instead, the cumulative human return $G_{0:T}^{\mathcal{H}}$, defined over $R^{\mathcal{H}}$, is given at the end of each trajectory, i.e., $G_{0:T}^{\mathcal{H}} = \sum_{t=0}^T \gamma^t r_t^{\mathcal{H}}$, with T being the horizon and $r_t^{\mathcal{H}} \sim R^{\mathcal{H}}(\cdot|s_t, a_t)$.

The assumption above follows the fact that human feedback (HF) is not available until the end of each episode, as opposed to immediate rewards that can be defined over the environment and evaluated for any (s_t, a_t) pairs at any time. This is especially true in environments such as healthcare where the clinical treatment outcome is not foreseeable until a therapeutic cycle is completed, or in intelligent tutoring where the overall gain from students over a semester is mostly reflected by the final grades. Note that although the setup can be generalized to the scenario where HF can be sparsely obtained over the horizon, we believe that issuing the HF only at the end of each trajectory leads to a more challenging setup for OPE.

Consequently, the goal of OPEHF can be formulated as follows. Given offline trajectories collected by some *behavioral* policy β , $\rho^\beta = \{\tau^{(0)}, \tau^{(1)}, \dots, \tau^{(N-1)} | a_t \sim \beta(a_t|s_t)\}$, with $\tau^{(i)} = [(s_0^{(i)}, a_0^{(i)}, r_0^{(i)}, r_0^{\mathcal{H}(i)}, s_1^{(i)}), \dots, (s_{T-1}^{(i)}, a_{T-1}^{(i)}, r_{T-1}^{(i)}, r_{T-1}^{\mathcal{H}(i)}, s_T^{(i)})]$ being a single trajectory, N the total number of offline trajectories, and $r_t^{\mathcal{H}}$'s being **unknown**, the objective is to estimate the *expected total human return* over the unknown state-action visitation distribution ρ^π of the *target* (evaluation) policy π , i.e., $\mathbb{E}_{(s,a) \sim \rho^\pi, r^{\mathcal{H}} \sim R^{\mathcal{H}}} \left[\sum_{t=0}^T \gamma^t r_t^{\mathcal{H}} \right]$.

4.3 Reconstruction of Immediate Human Rewards (IHRs) for OPEHF

We emphasize that the human returns are only issued at the end of each episode, with IHRs remaining unknown. One can set all IHRs from $t = 0$ to $t = T - 2$ to be zeros (i.e., $r_{0:T-2}^{\mathcal{H}} = 0$), and *rescale* the cumulative human return to be the IHR at the last step (i.e., $r_{T-1}^{\mathcal{H}} = G_{0:T}^{\mathcal{H}}/\gamma^{T-1}$), to allow the use of existing OPE methods toward OPEHF. However, the sparsity over $r^{\mathcal{H}}$'s here may impose difficulties for OPE to estimate the human returns accurately over the target policies. For OPEHF, we start by showing

that for the per-decision importance sampling (PDIS) method – a variance-reduction variant of the importance sample (IS) family of OPE methods (Precup, 2000) – if IHRs *were to be available*, they could reduce the variance in the estimation compared to the rescale approach above.

Recall that the PDIS estimator follows $\hat{G}_{PDIS}^\pi = \frac{1}{N} \sum_{i=1}^{N-1} \sum_{t=0}^{T-1} \gamma^t \omega_{0:t}^{(i)} r_t^{\mathcal{H}^{(i)}}$, where $\omega_{0:t}^{(i)} = \prod_{k=0}^t \frac{\pi(a_k^{(i)} | s_k^{(i)})}{\beta(a_k^{(i)} | s_k^{(i)})}$ are the PDIS weights for offline trajectory $\tau^{(i)}$. Moreover, the estimator of the rescale approach¹ above is $\hat{G}_{Rescale}^\pi = \frac{1}{N} \sum_{i=1}^{N-1} \omega_{0:T-1}^{(i)} G_{0:T}^{\mathcal{H}^{(i)}}$, which is equivalent to the vanilla IS estimator (Precup, 2000; P. S. Thomas, 2015). We now show the variance reduction property of \hat{G}_{PDIS}^π in the context of OPEHF.

Proposition 1. *Assume that (i) $\mathbb{E}[r_t^{\mathcal{H}}] \geq 0$, and (ii) given the horizon T , consider any $1 \leq t+1 \leq k \leq T$ of any offline trajectory τ , $\omega_{0:k}$ and $r_t^{\mathcal{H}} \omega_{0:k}$ are positively correlated. Then, $\mathbb{V}(\hat{G}_{PDIS}^\pi) \leq \mathbb{V}(\hat{G}_{Rescale}^\pi)$, with $\mathbb{V}(\cdot)$ representing the variance.*

The proof can be found in Appendix A. Assumption (i) can be easily satisfied in the real world, as HF signals are usually quantified as positive values, *e.g.*, ratings (1-10) provided by participants. Assumption (ii) is most likely to be satisfied when the target policies do not visit low-return regions substantially (Y. Liu et al., 2020), which is a pre-requisite for testing RL policies in human-involved environments as initial screening are usually required to filter the ones that could potentially pose risks to participants (Parvinian et al., 2018).

Besides IS, doubly robust (DR) (Farajtabar et al., 2018; Jiang & Li, 2016; Z. Tang et al., 2019; P. Thomas & Brunskill, 2016) and fitted Q-evaluation (FQE) (Le et al., 2019) methods require learning value functions. Sparsity of rewards (following the rescale approach above) in the offline dataset may lead to poorly learned value functions (Vecerik et al., 2017), considering that the offline data in OPE is usually fixed (*i.e.*, no new samples can be added), and are often generated by behavioral policies that are sub-optimal, which results in limited coverage of the state-action space. Limited availabilities of environment-policy

¹ We call it the *rescale approach* instead of vanilla IS as the idea behind also generalizes to non-IS methods.

interactions (*e.g.*, clinical trials) further reduce the scale of the exploration and therefore limit the information that can be leveraged toward obtaining accurate value function approximations.

4.3.1 Reconstruction of IHRs.

To address this challenge, our approach aims to project the end-of-episode human returns back to each environmental step, *i.e.*, to learn a mapping $f_\theta(\tau, G_{0:T}^{\mathcal{H}}) : (\mathcal{S} \times \mathcal{A})^T \times \mathbb{R} \rightarrow \mathbb{R}^T$, parameterized by θ , that maximizes the sum of log-likelihood of the estimated IHRs, $[\hat{r}_0^{\mathcal{H}}, \dots, \hat{r}_{T-1}^{\mathcal{H}}]^\top \sim f_\theta(\tau, G_{0:T}^{\mathcal{H}})$, following $\max_\theta \frac{1}{N} \sum_{i=0}^{N-1} \sum_{t=0}^{T-1} \log p(\hat{r}_t^{\mathcal{H}} = r_t^{\mathcal{H}(i)} | \theta, \tau^{(i)}, G_{0:T}^{\mathcal{H}(i)})$, where $G_{0:T}^{\mathcal{H}(i)}$ and $r_t^{\mathcal{H}(i)}$'s are respectively the human return and IHRs (unknown) of the i -th trajectory in the offline dataset ρ^β , and N is the total number of trajectories in ρ^β . Given that the objective above is intractable due to unknown $r_t^{\mathcal{H}(i)}$'s, we introduce a surrogate objective

$$\max_\theta \frac{1}{N} \sum_{i=0}^{N-1} \left[\log p \left(\sum_{t=0}^{T-1} \gamma^t \hat{r}_t^{\mathcal{H}} = G_{0:T}^{\mathcal{H}(i)} | \theta, \tau^{(i)}, G_{0:T}^{\mathcal{H}(i)} \right) - C \cdot \mathcal{L}_{regu}(\hat{r}_{0:T-1}^{\mathcal{H}} | \theta, \tau^{(i)}, G_{0:T}^{\mathcal{H}(i)}) \right]. \quad (4.1)$$

Here, the *first term* is a necessary condition for $\hat{r}_t^{\mathcal{H}}$'s to be valid for estimating $r_t^{\mathcal{H}}$'s, as they should sum to $G_{0:T}^{\mathcal{H}}$. Since many solutions may exist if one only optimizes over the first term, the *second term* \mathcal{L}_{regu} serves as a regularization that imposes constraints on $r_t^{\mathcal{H}}$'s to follow the properties specific to their corresponding state-action pairs; *e.g.*, (s, a) pairs that are similar to each other in a representation space, defined over the state-action visitation space, tend to yield similar immediate rewards (Q. Gao, Gao, Chi, & Pajic, 2023b).

The detailed regularization technique is introduced in sub-section below. Practically, we choose f_θ to be a bi-directional long-short term memory (LSTM) (Hochreiter & Schmidhuber, 1997), since the reconstruction of IHRs can leverage information from both previous and subsequent steps as provided in the offline trajectories.

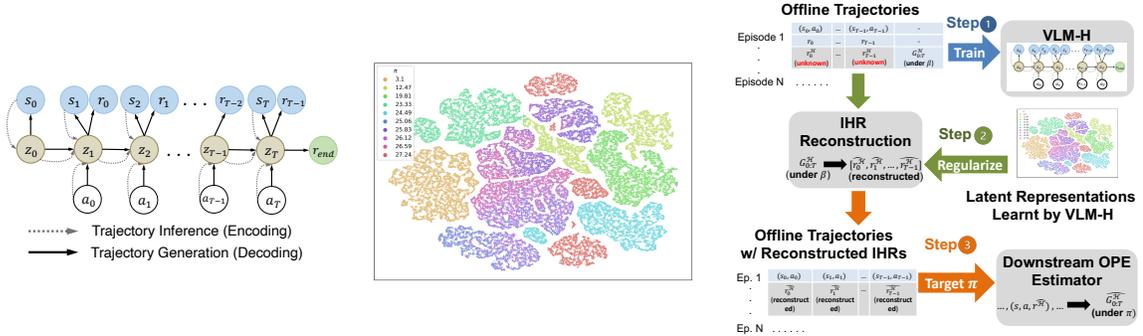


FIGURE 4.1: **(Left)** Architecture of the variational latent model with human returns (VLM-H). **(Mid)** Illustration of the clustering behavior in the latent space using t -SNE visualization (Van der Maaten & Hinton, 2008), where the encoded state-action pairs (output by the encoder of VLM-H) are in general clustered together if they are generated by policies with similar human returns (shown in the legend at the top left). **(Right)** Diagram summarizing the pipeline of the OPEHF framework.

4.4 Reconstruction of IHRs over Latent Representations (RILR) for OPEHF

Now, we introduce the regularization technique for the reconstruction of IHRs, *i.e.*, reconstructing IHRs over latent representations (RILR). Specifically, we leverage the representations captured by variational auto-encoders (VAEs) (Kingma & Welling, 2013), learned over ρ^β , to regularize the reconstructed IHRs, \hat{r}_t^H .

VAEs have been adapted toward learning a compact latent space over offline state-action visitations, facilitating both offline policy optimization (Hafner, Lillicrap, et al., 2020; Hafner, Lillicrap, et al., 2020; Hafner et al., 2019; Lee et al., 2020; Rybkin et al., 2021; M. Zhang et al., 2019) and OPE (Q. Gao, Gao, Chi, & Pajic, 2023b). In this work, we specifically consider building on the variational latent model (VLM) proposed in (Q. Gao, Gao, Chi, & Pajic, 2023b) since it is originally proposed to facilitate OPE, as opposed to others that mainly use knowledge captured in the latent space to improve sample efficiency for policy optimization. Moreover, the VLM has shown to be effective for learning an expressive representation space, where the encoded state-action pairs are clustered well in the latent space, as measured by the difference over the returns of the policies from which the state-action pairs are sampled; see Figure 4.1 (mid) which uses t -

SNE to visualize the encoded state-action pairs in trajectories collected from a visual Q&A environment.

Note that VLM originally does not account for HF signals (neither $r_t^{\mathcal{H}}$'s nor $G_{0:T}^{\mathcal{H}}$'s), so we introduce the variational latent model with human returns (VLM-H) below, building on the architecture introduced in (Q. Gao, Gao, Chi, & Pajic, 2023b). VLM-H consists of a prior $p(z)$ over the latent variables $z \in \mathcal{Z} \subset \mathbb{R}^L$, with \mathcal{Z} representing the latent space and L the dimension, along with a variational encoder $q_\psi(z_t|z_{t-1}, a_{t-1}, s_t)$, a decoder $p_\phi(z_t, s_t, r_{t-1}|z_{t-1}, a_{t-1})$ for generating per-step transitions (over both state-action and latent space), and a separate decoder $p_\phi(G_{0:T}^{\mathcal{H}}|z_T)$ for the reconstruction of the human returns at the end of each episode. Note that encoders and decoders are parameterized by ψ and ϕ respectively. The overall architecture is illustrated in Figure 4.1 (left).

4.4.1 Trajectory Inference (Encoding).

VLM-H's encoder approximates the intractable posterior

$$p(z_t|z_{t-1}, a_{t-1}, s_t) = \frac{p(z_{t-1}, a_{t-1}, z_t, s_t)}{\int_{z_t \in \mathcal{Z}} p(z_{t-1}, a_{t-1}, z_t, s_t) dz_t'} \quad (4.2)$$

by avoiding to integrate over the unknown latent space *a priori*. The inference (or encoding) process can be decomposed as, *i.e.*, $q_\psi(z_{0:T}|s_{0:T}, a_{0:T-1}) = q_\psi(z_0|s_0) \prod_{t=1}^T q_\psi(z_t|z_{t-1}, a_{t-1}, s_t)$; here, $q_\psi(z_0|s_0)$ encodes initial states s_0 into latent variables z_0 , and $q_\psi(z_t|z_{t-1}, a_{t-1}, s_t)$ captures all subsequent environmental transitions in the latent space over z_t 's. In general, both q_ψ 's are represented as diagonal Gaussian distributions² with mean and variance determined by neural network ψ , as in (Q. Gao, Gao, Chi, & Pajic, 2023b; Hafner, Lillicrap, et al., 2020; Hafner, Lillicrap, et al., 2020; Hafner et al., 2019; Lee et al., 2020).

4.4.2 Trajectory Generation (Decoding).

The generative (or decoding) process follows, *i.e.*, $p_\phi(z_{1:T}, s_{0:T}, r_{0:T-1}, G_{0:T}^{\mathcal{H}}|z_0, \pi) = p_\phi(G_{0:T}^{\mathcal{H}}|z_T) \cdot \prod_{t=1}^T p_\phi(z_t|z_{t-1}, a_{t-1}) p_\phi(r_{t-1}|z_t) \cdot \prod_{t=0}^T p_\phi(s_t|z_t)$; here, $p_\phi(z_t|z_{t-1}, a_{t-1})$ enforces

² This helps facilitate an orthogonal basis of the latent space, which would improve the expressiveness of the model.

the transition of latent variables z_t over time, $p_\phi(s_t|z_t)$ and $p_\phi(r_{t-1}|z_t)$ are used to sample the states and immediate *environmental* rewards, while $p_\phi(G_{0:T}^H|z_T)$ generates the *human return* issued at the end of each episode. Note that here we still use the VLM-H to capture environmental rewards, allowing the VLM-H to formulate a latent space that captures as much information about the dynamics underlying the environment as possible. All p_ϕ 's are represented as diagonal Gaussians³ with parameters determined by network ϕ .

To train ϕ and ψ , one can maximize the evidence lower bound (ELBO) of the joint log-likelihood $\log p_\phi(s_{0:T}, r_{0:T-1}, G_{0:T}^H|\phi, \psi, \rho^\beta)$, *i.e.*,

$$\begin{aligned} \max_{\psi, \phi} \quad & \mathbb{E}_{q_\psi} \left[\log p_\phi(G_{0:T}^H|z_T) + \sum_{t=0}^T \log p_\phi(s_t|z_t) + \sum_{t=1}^T \log p_\phi(r_{t-1}|z_t) \right. \\ & \left. - KL(q_\psi(z_0|s_0)||p(z_0)) - \sum_{t=1}^T KL(q_\psi(z_t|z_{t-1}, a_{t-1}, s_t)||p_\phi(z_t|z_{t-1}, a_{t-1})) \right]; \quad (4.3) \end{aligned}$$

the first three terms are the log-likelihoods of reconstructing the human return, states, and environmental rewards, and the two terms that follow are Kullback-Leibler (KL) divergence (Kullback & Leibler, 1951) regularizing the inferred posterior q_ψ . Derivation of the ELBO can be found below. In practice, if ϕ and ψ are chosen to be recurrent networks, one can also regularize the hidden states of ϕ, ψ by including the additional regularization term introduced in (Q. Gao, Gao, Chi, & Pajic, 2023b).

³ If needed, one can project the states over to the orthogonal basis, to ensure that they follow a diagonal covariance.

4.4.3 Derivation of the evidence lower bound (ELBO) Above

Note that unlike the ELBO in (Q. Gao, Gao, Chi, & Pajic, 2023b), the VLM-H includes an additional component that estimates the human return $G_{0:T}^{\mathcal{H}}$ of each trajectory, *i.e.*,

$$\log p_{\phi}(s_{0:T}, r_{0:T-1}, G_{0:T}^{\mathcal{H}}) \quad (4.4)$$

$$= \log \int_{z_{1:T} \in \mathcal{Z}} p_{\phi}(s_{0:T}, z_{1:T}, r_{0:T-1}, G_{0:T}^{\mathcal{H}}) dz \quad (4.5)$$

$$= \log \int_{z_{1:T} \in \mathcal{Z}} \frac{p_{\phi}(s_{0:T}, z_{1:T}, r_{0:T-1}, G_{0:T}^{\mathcal{H}})}{q_{\psi}(z_{0:T}|s_{0:T}, a_{0:T-1})} q_{\psi}(z_{0:T}|s_{0:T}, a_{0:T-1}) dz \quad (4.6)$$

$$\geq \mathbb{E}_{q_{\psi}} [\log p(z_0) + \log p_{\phi}(s_{0:T}, z_{1:T}, r_{0:T-1}|z_0) + \log p_{\phi}(G_{0:T}^{\mathcal{H}}|z_T) - \log q_{\psi}(z_{0:T}|s_{0:T}, a_{0:T-1})] \quad (4.7)$$

$$\begin{aligned} = & \mathbb{E}_{q_{\psi}} \left[\log p(z_0) + \log p_{\phi}(s_0|z_0) + \log p_{\phi}(G_{0:T}^{\mathcal{H}}|z_T) + \sum_{t=1}^T \log p_{\phi}(s_t, z_t, r_{t-1}|z_{t-1}, a_{t-1}) \right. \\ & \left. - \log q_{\psi}(z_0|s_0) - \sum_{t=1}^T \log q_{\psi}(z_t|z_{t-1}, a_{t-1}, s_t) \right] \quad (4.8) \end{aligned}$$

$$= \mathbb{E}_{q_{\psi}} \left[\log p(z_0) - \log q_{\psi}(z_0|s_0) + \log p_{\phi}(s_0|z_0) + \log p_{\phi}(G_{0:T}^{\mathcal{H}}|z_T) \quad (4.9)$$

$$\begin{aligned} & - \sum_{t=1}^T \log q_{\psi}(z_t|z_{t-1}, a_{t-1}, s_t) \\ & + \sum_{t=1}^T \log (p_{\phi}(s_t|z_t) p_{\phi}(r_{t-1}|z_t) p_{\phi}(z_t|z_{t-1}, a_{t-1})) \Big] \quad (4.10) \end{aligned}$$

$$\begin{aligned} = & \mathbb{E}_{q_{\psi}} \left[\log p_{\phi}(G_{0:T}^{\mathcal{H}}|z_T) + \sum_{t=0}^T \log p_{\phi}(s_t|z_t) + \sum_{t=1}^T \log p_{\phi}(r_{t-1}|z_t) \right. \\ & \left. - \text{KL}(q_{\psi}(z_0|s_0)||p(z_0)) - \sum_{t=1}^T \text{KL}(q_{\psi}(z_t|z_{t-1}, a_{t-1}, s_t)||p_{\phi}(z_t|z_{t-1}, a_{t-1})) \right]. \quad (4.11) \end{aligned}$$

Note that to simplify our presentation, we omit ϕ, ψ, ρ^{β} as part of the conditional terms in the joint likelihoods. The transition from (4.6) to (4.7) follows Jensen's inequality.

4.4.4 Regularizing the Reconstruction of IHRs.

Existing works have shown that the latent space not only facilitates the generation of synthetic trajectories but demonstrated that the latent encodings of state-action pairs form clusters, over some measures in the latent space (Van der Maaten & Hinton, 2008), if they

are rolled out from policies that lead to similar returns (Q. Gao, Gao, Chi, & Pajic, 2023b; Lee et al., 2020). As a result, we regularize $\hat{r}_t^{\mathcal{H}}$ following

$$\min_{\theta} \mathcal{L}_{regu}(\hat{r}_t^{\mathcal{H}} | \theta, \psi, s_{0:t}^{(i)}, a_{0:t-1}^{(i)}, G_{0:T}^{\mathcal{H}(i)}) = \sum_{j \in \mathcal{J}} -\log p(\hat{r}_t^{\mathcal{H}} = (1 - \gamma) G_{0:T}^{\mathcal{H}(j)} | \theta, \psi, s_{0:t}^{(j)}, a_{0:t-1}^{(j)}, G_{0:T}^{\mathcal{H}(j)}) \quad (4.12)$$

for each step t ; here, $(s_{0:t}^{(i)}, a_{0:t-1}^{(i)}) \in \tau^{(i)} \sim \rho^\beta$, $\mathcal{J} = \{j_0, \dots, j_{K-1}\}$ are the indices of offline trajectories that correspond to the latent encodings $\{z_{t'}^{(j_k)} \sim q_\psi(\cdot | s_{0:t'}^{(j_k)}, a_{0:t'-1}^{(j_k)}) | j_k \in \mathcal{J}, t' \in [0, T-1]\}$ that are K -neighbours of the latent encoding $z_t^{(i)}$ pertaining to $(s_{0:t}^{(i)}, a_{0:t-1}^{(i)})$, defined over some similarity/distance function $d(\cdot || \cdot)$, following, *i.e.*,

$$\min_{j_k \in \mathcal{J}} \sum_{k=0}^{K-1} d(z_t^{(i)} || z_{t'}^{(j_k)}), \quad \text{s.t. } z_{t'}^{(j_k)} \text{'s corresponding } (s_{0:t'}^{(j_k)}, a_{0:t'-1}^{(j_k)}) \in \tau^{(j_k)} \sim \rho^\beta. \quad (4.13)$$

In practice, we choose $d(\cdot || \cdot)$ to follow stochastic neighbor embedding (SNE) similarities (Van der Maaten & Hinton, 2008), as it has been shown effective for capturing Euclidean distances in high-dimensional space (Wattenberg et al., 2016).

4.4.5 Overall Objective of RILR for OPEHF.

As a result, by following (4.1) and leveraging the \mathcal{L}_{regu} from (4.12) above, the objective for reconstructing the IHRs is set to be, *i.e.*,

$$\max_{\theta} \frac{1}{N} \sum_{i=0}^{N-1} \left[\log p\left(\sum_{t=0}^{T-1} \gamma^t \hat{r}_t^{\mathcal{H}} = G_{0:T}^{\mathcal{H}(i)} | \theta, \tau^{(i)}, G_{0:T}^{\mathcal{H}(i)}\right) - C \cdot \sum_{t=0}^{T-1} \mathcal{L}_{regu}(\hat{r}_t^{\mathcal{H}} | \theta, \psi, s_{0:t}^{(i)}, a_{0:t-1}^{(i)}, G_{0:T}^{\mathcal{H}(i)}) \right]. \quad (4.14)$$

4.4.6 Move from RILR to OPEHF.

In what follows, one can leverage any existing OPE methods to take as inputs the offline trajectories, with the immediate environmental rewards r_t 's replaced by the reconstructed IHRs $\hat{r}_t^{\mathcal{H}}$'s, to achieve the OPEHF's objective. Moreover, our method does not require the IHRs to be correlated with the environmental rewards, as the VLM-H learns to reconstruct both by sampling from two independent distributions, $p_\phi(r_{t-1} | z_t)$ and $p_\phi(G_{0:T}^{\mathcal{H}} | z_T)$ respec-

tively, following (4.3); this is also illustrated empirically over the experiments introduced below, where exceedingly low correlations are found in specific scenarios.

The overall pipeline summarizing our method is shown in Figure 4.1 (right).

4.5 Real-World Experiments with Human Participants

In this section, we validate the OPEHF framework introduced above over two real-world experiments, adaptive neurostimulation, and intelligent tutoring. Specifically, we consider four types of OPE methods to be used as the downstream estimator following the RILR step (Section 4.4), including per-decision importance sampling (IS) with behavioral policy estimation (Hanna et al., 2019), doubly robust (DR) (P. Thomas & Brunskill, 2016), distribution correction estimation (DICE) (M. Yang et al., 2020) and fitted Q-evaluation (FQE) (Le et al., 2019). A brief overview of these methods can be found in Section 2.3. In Section 4.6, we have also tested our method within a visual Q&A environment (Das et al., 2017; Snell et al., 2022), which follows similar mechanisms as in the two real-world experiments, *i.e.*, two types of return signals are considered though no human participants are involved.

4.5.1 Baselines and Ablations.

The baselines include two variants for each of the OPE methods above, *i.e.*, (i) the *rescale* approach discussed in Section 4.3, and (ii) another variant that sets all the IHRs to be equal to the environmental rewards at corresponding steps, $r_t^{\mathcal{H}} = r_t \forall t \in [0, T - 2]$, and then let $r_{T-1}^{\mathcal{H}} = r_{T-1} + (G_{0:T}^{\mathcal{H}} - G_{0:T})/\gamma^{T-1}$ with $G_{0:T} = \sum_t \gamma^t r_t$ being the environmental return, which is referred to as *fusion* below – this baseline may perform better when strong correlations existed between environmental and human rewards, as it intrinsically decomposes the human returns into IHRs. Consequently, in each experiment below, we compare the performance of the OPEHF framework extending all four types of OPE methods above, <IS/DR/DICE/FQE>-OPEHF, against the corresponding baselines, <IS/DR/DICE/FQE>-<Fusion/Rescale>. We also include the VLM-H as an ablation baseline, as if it is a model-based approach standalone; this is achieved by sampling the estimate

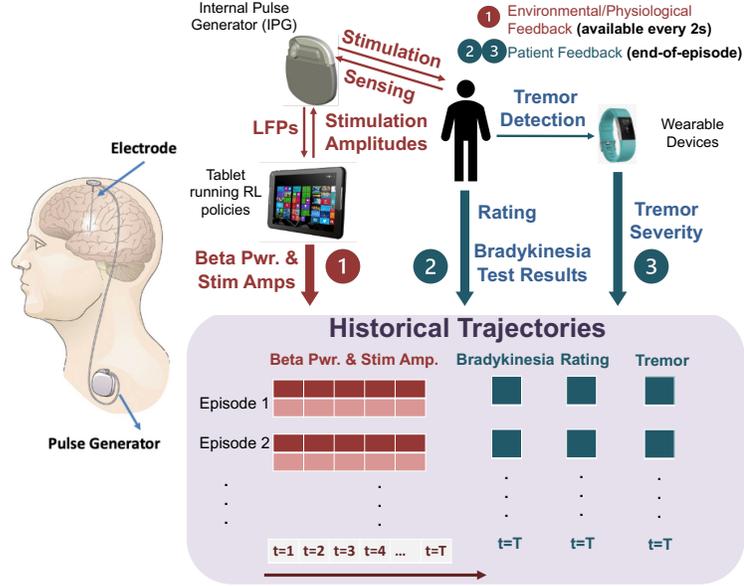


FIGURE 4.2: Setup of the neurostimulation experiments, as well as the formulation of offline trajectories. Environmental rewards and human returns are captured in streams 1 and 2-3 respectively.

returns from the decoder, $\hat{G}_{0:T}^{\mathcal{H}} \sim p_{\phi}(G_{0:T}^{\mathcal{H}}|z_T)$.

4.5.1.1 Metrics.

Following a recent OPE benchmark (Fu, Norouzi, et al., 2020), three metrics are considered to validate the performance of each method, including mean absolute error (MAE), rank correlation, and regret@1. Mathematical definitions can be found in Section 2.3. Also, following (Fu, Norouzi, et al., 2020), each method is evaluated over 3 random seeds, and the mean performance (with standard errors) is reported.

4.5.2 Adaptive Neurostimulation: Deep Brain Stimulation

Adaptive neurostimulation facilitates treatments for a variety of neurological disorders (Benabid, 2003; Deuschl et al., 2006; Follett et al., 2010; Okun, 2012a). Deep brain stimulation (DBS) is a type of neurostimulation used specifically toward Parkinson’s disease (PD), where an internal pulse generator (IPG), implanted under the collarbone, sends electrical stimulus to the basal ganglia (BG) area of the brain through invasive electrodes;

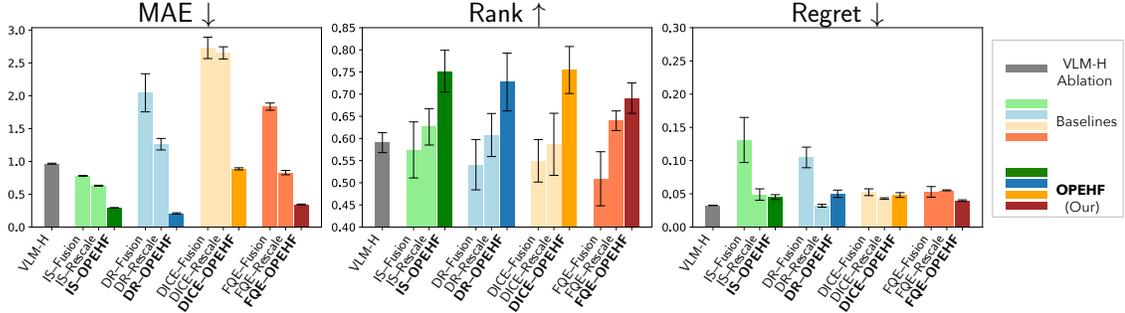


FIGURE 4.3: Results from the adaptive neurostimulation experiment, *i.e.*, deep brain stimulation (DBS). Each method is evaluated over the data collected from each patient, toward corresponding target policies, respectively. The performance shown above are averaged over all 4 human participants affected by Parkinson’s disease (PD). Raw statistics can be found in Tables 4.2-4.5.

Figure 4.2 illustrates the setup.⁴ Adaptive DBS aims to adjust the strength (amplitude) of the stimulus in real-time, to respond to irregular neuronal activities caused by PD, leveraging the local field potentials (LFPs) as the immediate feedback signals, *i.e.*, the environmental rewards. Existing works have leveraged RL for adaptive DBS over *computational* BG models (Q. Gao, Naumann, et al., 2020; Guez et al., 2008; Nagaraj et al., 2017; Pineau et al., 2009), using rewards defined over a physiological signal – beta-band power spectral density of LFPs (*i.e.*, the beta power) since physiologically PD could lead to increased beta power due to the irregular neuronal activations it causes (Kuncel & Grill, 2004). However, in clinical practice, the correlation between beta power and the level of satisfaction reported by the patients varies depending on the specific characteristics of each person, as PD can cause different types of symptoms over a wide range of severity (Brown et al., 2001; Kühn et al., 2006; Okun, 2012b; Wong et al., 2022). Such findings further justify the significance of evaluating HF/human returns in the real world using OPEHF.

In this experiment, we leverage OPEHF to estimate the feedback provided by 4 *PD patients* who participate in monthly clinical testings of RL policies trained to adapt amplitudes of the stimulus toward reducing their PD symptoms, *i.e.*, bradykinesia and tremor. A mixture of behavioral policies is used to collect the offline trajectories ρ^β . Specifically, in ev-

⁴ A more detailed introduction of this DBS system can be found in Chapter 5.

Table 4.1: Correlations between the *environmental* and *human* returns of the 6 target DBS policies associated with each PD patient.

Patient #	0	1	2	3
Pearson’s	-0.396	-0.477	-0.599	-0.275
Spearman’s	-0.2	-0.6	0.086	0.086

ery step, the state s_t is a historical sequence of LFPs capturing neuronal activities, and the action a_t updates the amplitude of the stimulus to be sent⁵. Then, an *environmental* reward $r_t = R(s_t, a_t)$ gives a penalty if the beta power computed from the latest LFPs is greater than some threshold (to promote treatment efficacy) as well as a penalty proportional to the amplitudes of the stimulus being sent (to improve battery life of the IPG). At the end of each episode, the *human returns* $G_{0:T}^H$ are determined from three sources (weighted by 50%, 25%, 25%, respectively), *i.e.*, (i) a satisfaction rating (between 1-10) provided by the patient, (ii) hand grasp speed as a result of the bradykinesia test (Ramaker et al., 2002), and (iii) level of tremor calculated over the data from a wearable accelerometry (W. Chen et al., 2021; Powers et al., 2021). Each session lasts more than 10 minutes, and each discrete step above corresponds to 2 seconds in the real world; thus, the horizon $T \geq 300$. Approval of an Institutional Review Board (IRB) is obtained from Duke University Health System, as well as the exceptional use of the DBS system by the US Food and Drug Administration (FDA).

For each patient, OPEHF and the baselines are used to estimate the human returns of 6 target policies with varied performance. The ground-truth human return for each target policy is obtained as a result of extensive clinical testing following the same schema above, over more than 100 minutes. Table 4.1 shows the Pearson’s and Spearman’s correlation coefficients (Freedman et al., 2007), measuring the linear and rank correlations between the environmental returns $G_{0:T}$ and the human returns $G_{0:T}^H$ over all the target DBS policies considered for each patient. Pearson’s coefficients are all negative since the environmental

⁵ RL policies only adapt the stimulation amplitudes within a safe range as determined by neurologists/neurosurgeons, making sure they will not lead to negative effects to participants.

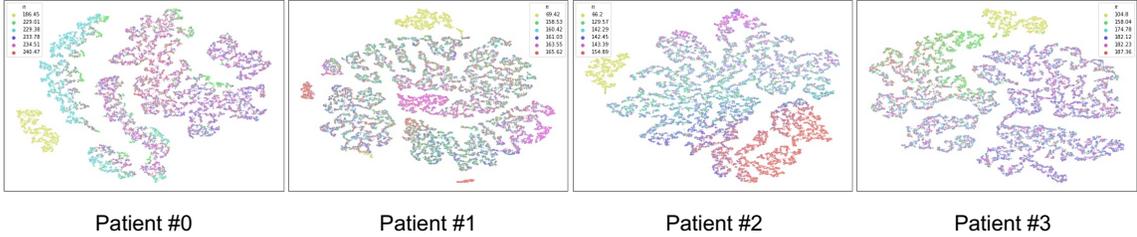


FIGURE 4.4: t -SNE visualizing the VLM-H encodings of the state-action pairs rolled out over DBS policies with different human returns (shown in the legend). It can be observed that distances among the encoded pairs associated with the policies that lead to similar returns are in general smaller, justifying the RILR objective (4.14).

reward function only issues penalties, while human returns are all captured by positive values. It can be observed that only weak-to-moderate degrees of linear correlations exist for all four patients, while ranks between $G_{0:T}$'s and $G_{0:T}^H$'s are not preserved across patients; thus, it highlights the need for leveraging OPEHF to estimate human returns, which is different than the classic OPE that focus on estimating environmental returns.

The overall performance averaged across the 4-patient cohort, is reported in Figure 4.3. Raw performance over every single patient can be found in Tables 4.2-4.5. It can be observed that our OPEHF framework significantly improves MAEs and ranks compared to the two baselines, for all 4 types of downstream OPE methods we considered (IS, DR, DICE, and FQE). Moreover, our method also significantly outperforms the ablation VLM-H in terms of these two metrics, as the VLM-H's performance is mainly determined by how well it could capture the underlying dynamics and returns. In contrast, our OPEHF framework not only leverages the latent representations learnt by the VLM-H (for regularizing RILR), it also inherits the advantages intrinsically associated with the downstream estimators; *e.g.*, low-bias nature of IS, or low-variance provided by DR. Moreover, the fusion baseline in general performs worse than the rescale baseline as expected, since no strong correlations between environmental and human returns are found, as reported in Table 4.1.

Note that the majority of the methods lead to similar (relatively low) regrets, as there exist a few policies that lead to human returns that are close over some patients. The reason is that all the policies to be extensively tested in clinics are subject to initial screening, where

Table 4.2: Raw results of the adaptive neurostimulation experiment from each patient (Patient #0).

	<i>IS</i>			<i>DR</i>			<i>Ablation</i>
	Fusion	Rescale	OPEHF (our)	Fusion	Rescale	OPEHF (our)	<i>VLM-H</i>
MAE	0.96±0.0	0.74±0.0	0.17±0.01	2.21±0.38	1.65±0.07	0.09±0.01	0.97±0.02
Rank	0.65±0.01	0.65±0.01	0.74±0.07	0.43±0.05	0.63±0.08	0.58±0.1	0.73±0.09
Regret@1	0.05 ± 0.0	0.05 ± 0.0	0.04 ± 0.01	0.03 ± 0.38	0.02 ± 0.07	0.03 ± 0.01	0.02±0.02
	<i>DICE</i>			<i>FQE</i>			
	Fusion	Rescale	OPEHF (our)	Fusion	Rescale	OPEHF (our)	
MAE	4.59±0.17	4.0±0.2	0.98±0.01	1.64±0.03	0.63±0.0	0.27±0.0	
Rank	0.52±0.05	0.56±0.08	0.62±0.09	0.48±0.07	0.49±0.02	0.58±0.07	
Regret@1	0.05 ± 0.17	0.02 ± 0.2	0.03 ± 0.01	0.02 ± 0.03	0.05 ± 0.0	0.03 ± 0.0	

clinicians ensure they would not lead to undesired outcomes or pose significant risks to the patients; thus, the performance of some target policies tends to be close. Nonetheless, low MAEs and high ranks achieved by our method show that it can effectively capture the subtle differences in returns resulting from other HF signals, *i.e.*, levels of bradykinesia and tremor. Moreover, Figure 4.4 visualizes the VLM-H encodings over the trajectories collected from the 6 target DBS policies for each participant and shows that encoded pairs associated with the policies that lead to similar returns are in general clustered together, which justifies the importance of leveraging the similarities over latent representations to regularize the reconstruction of IHRs as in the RILR objective (4.14).

4.5.3 Intelligent Tutoring

Intelligent tutoring refers to a system where students can actively interact with an autonomous tutoring agent that can customize the learning content, tests, etc., to improve engagement and learning outcomes (Anderson et al., 1985; E. Liu et al., 2022; Ruan et al., 2023). OPEHF is important in such a setup for directly estimating the potential outcomes that could be obtained by students, as opposed to environmental rewards that are mostly discrete; see detailed setup below. Existing works have explored this topic over classic OPE setting in *simulations* (Mandel et al., 2014; Nie et al., 2022).

Table 4.3: Raw results of the adaptive neurostimulation experiment from each patient (**Patient #1**).

	<i>IS</i>			<i>DR</i>			<i>Ablation</i>
	Fusion	Rescale	OPEHF (our)	Fusion	Rescale	OPEHF (our)	<i>VLM-H</i>
MAE	0.59±0.0	0.44±0.0	0.43±0.01	2.11±0.14	1.03±0.08	0.27±0.03	<i>0.98±0.01</i>
Rank	0.58±0.06	0.52±0.08	0.66±0.08	0.74±0.07	0.6±0.05	0.87±0.02	<i>0.58±0.03</i>
Regret@1	0.21±0.0	0.02±0.0	0.02±0.01	0.01±0.14	0.0±0.08	0.02±0.03	<i>0.03±0.01</i>
	<i>DICE</i>			<i>FQE</i>			
	Fusion	Rescale	OPEHF (our)	Fusion	Rescale	OPEHF (our)	
MAE	0.47±0.01	0.45±0.01	0.87±0.01	2.53±0.07	1.05±0.09	0.53±0.02	
Rank	0.73±0.05	0.72±0.09	0.9±0.03	0.48±0.12	0.76±0.01	0.74±0.06	
Regret@1	0.03±0.01	0.03±0.01	0.03±0.01	0.01±0.07	0.04±0.09	0.02±0.02	

The system is deployed in an undergraduate-level introduction to probability and statistics course over 5 academic years at North Carolina State University, where the interaction logs obtained from 1,288 students who voluntarily opted-in for this experiment are recorded.⁶ Specifically, each episode refers to a student working on a set of 12 problems (*i.e.*, horizon $T = 12$), where the agent suggests the student approach each problem through *independent* work, working with the *hints* provided, or directly providing the *full solution* (for studying purposes) – these options constitute the action space of the agent. The states are characterized by 140 features extracted from the logs, designed by domain experts; they include, for example, the time spent on each problem, and the correctness of the solution provided. In each step, an immediate *environmental* reward of +1 is issued if the answer submitted by students, for the current problem, is at least 80% correct (auto-graded following pre-defined rubrics). A reward of 0 is issued if the grade is less than 80% or the agent chooses the action that directly displays the full solution. Moreover, students are instructed to complete two exams, one before working on any problems and another after finishing all the problems. The normalized difference between the grades of two exams constitutes the

⁶ An IRB approval is obtained from North Carolina State University. The use/test of the intelligent tutoring system is overseen by a departmental committee, ensuring it does not risk the academic performance and privacy of the participants.

Table 4.4: Raw results of the adaptive neurostimulation experiment from each patient (**Patient #2**).

	<i>IS</i>			<i>DR</i>			<i>Ablation</i>
	Fusion	Rescale	OPEHF (our)	Fusion	Rescale	OPEHF (our)	<i>VLM-H</i>
MAE	0.63±0.03	0.7±0.02	0.33±0.01	2.17±0.3	1.11±0.16	0.3±0.01	0.97±0.02
Rank	0.65±0.11	0.71±0.05	0.9±0.03	0.52±0.09	0.59±0.04	0.84±0.03	0.6±0.0
Regret@1	0.05±0.03	0.08±0.02	0.08±0.01	0.03±0.3	0.08±0.16	0.1±0.01	0±0.02
	<i>DICE</i>			<i>FQE</i>			
	Fusion	Rescale	OPEHF (our)	Fusion	Rescale	OPEHF (our)	
MAE	5.44±0.55	3.52±0.12	0.66±0.02	1.6±0.06	0.81±0.02	0.38±0.01	
Rank	0.66±0.09	0.48±0.08	0.85±0.05	0.6±0.05	0.67±0.04	0.93±0.01	
Regret@1	0.11±0.55	0.08±0.12	0.1±0.02	0.08±0.06	0.08±0.02	0.08±0.01	

human return for each episode.

The intelligent tutoring agent follows different policies across academic years, where the data collected from the first 4 years (1148 students total) constitutes the offline trajectories ρ^β (as a result of a mixture of behavioral policies). The 4 policies deployed in the 5th year (140 students total) serve as the target policies, whose ground-truth performance is determined by averaging over the human returns of the episodes that are associated with each policy respectively. Table 4.7 documents the Pearson’s and Spearman’s correlation coefficients between the environmental and human returns from data collected over each academic year, showing weak linear and rank correlations across all 5 years. Such low correlations are due to the fact that the environmental rewards are discrete and do not distinguish among the agent’s choices, *i.e.*, a +1 reward can be obtained either if the student works out a solution independently or by following hints, and a 0 reward is issued every time the agent chooses to display the solution even if the student could have solved the problem. As a result, such a setup makes OPEHF to be more challenging; because human returns are only available at the end of each episode, and the immediate environmental rewards do not carry substantial information toward extrapolating IHRs.

Table 4.6 documents the performance of OPEHF and the baselines toward estimating

Table 4.5: Raw results of the adaptive neurostimulation experiment from each patient (Patient #3).

	<i>IS</i>			<i>DR</i>			<i>Ablation</i>
	Fusion	Rescale	OPEHF (our)	Fusion	Rescale	OPEHF (our)	<i>VLM-H</i>
MAE	0.94±0.01	0.76±0.01	0.24±0.0	1.69±0.51	0.69±0.06	0.17±0.01	0.94±0.01
Rank	0.42±0.11	0.16±0.03	0.71±0.05	0.48±0.05	0.42±0.11	0.62±0.14	0.45±0.06
Regret@1	0.21±0.01	0.01±0.01	0.04±0.0	0.35±0.51	0.1±0.06	0.04±0.01	0.03±0.01
	<i>DICE</i>			<i>FQE</i>			
	Fusion	Rescale	OPEHF (our)	Fusion	Rescale	OPEHF (our)	
MAE	0.41±0.02	1.15±0.07	1.03±0.04	1.56±0.09	0.53±0.02	0.19±0.01	
Rank	0.28±0.05	0.22±0.03	0.66±0.07	0.48±0.05	0.29±0.02	0.52±0.01	
Regret@1	0.02±0.02	0.03±0.07	0.03±0.04	0.1±0.09	0.05±0.02	0.03±0.01	

Table 4.6: Results from the intelligent tutoring experiment, *i.e.*, performance achieved by our OPEHF framework compared to the ablation and baselines over all four types of downstream OPE estimators.

	<i>IS</i>			<i>DR</i>			<i>Ablation</i>
	Fusion	Rescale	OPEHF (our)	Fusion	Rescale	OPEHF (our)	<i>VLM-H</i>
MAE	0.7±0.14	0.77±0.08	0.57±0.09	1.03±0.07	1.03±0.25	0.86±0.04	1.00±0.01
Rank	0.47±0.11	0.4±0.09	0.8±0.09	0.33±0.05	0.4±0.0	0.53±0.2	0.41±0.25
Regret@1	0.36±0.16	0.36±0.16	0.41±0.04	0.41±0.0	0.41±0.0	0.41±0.0	0.28±0.19
	<i>DICE</i>			<i>FQE</i>			
	Fusion	Rescale	OPEHF (our)	Fusion	Rescale	OPEHF (our)	
MAE	3.19±0.57	2.33±0.59	1.01±0.01	0.74±0.07	0.98±0.1	0.59±0.1	
Rank	0.47±0.2	0.33±0.2	0.53±0.22	0.27±0.14	0.4±0.0	0.47±0.05	
Regret@1	0.55±0.06	0.45±0.18	0.37±0.15	0.36±0.16	0.41±0.0	0.41±0.0	

Table 4.7: Correlations between the *environmental* and *human* returns from data collected over each academic year.

Year #	0	1	2	3	4
Pearson’s	0.033	0.176	0.089	0.154	0.183
Spearman’s	0.082	0.156	0.130	0.161	0.103

Table 4.8: Results from the visual Q&A environment.

	<i>IS</i>			<i>DR</i>		
	Fusion	Rescale	OPEHF (our)	Fusion	Rescale	OPEHF (our)
MAE	1.13±0.41	0.40±0.27	0.56±0.12	1.27±0.33	0.72±0.21	0.63±0.08
Rank	-0.21±0.55	0.19±0.15	0.64±0.13	0.11±0.45	0.08±0.27	0.59±0.18
Regret@1	0.76±0.34	0.08±0.09	0±0.05	0.43±0.29	0.25±0.09	0.01±0.0
	<i>DICE</i>			<i>FQE</i>		
	Fusion	Rescale	OPEHF (our)	Fusion	Rescale	OPEHF (our)
MAE	0.99±0.12	0.25±0.03	0.11±0.02	0.97±0.08	1.13±0.14	0.89±0.03
Rank	0.02±0.24	-0.13±0.1	0.06±0.32	0.1±0.05	0.12±0.52	0.55±0.28
Regret@1	0.01±0.01	0.07±0.0	0.02±0.03	0.01±0.0	0.03±0.03	0.01±0.0

the human returns of the target policies. It can be observed that our OPEHF framework achieves state-of-the-art performance, over all types of downstream OPE estimators considered. This result echos the design of the VLM-H where both environmental information (state transitions and rewards) and human returns are encoded into the latent space, which helps formulate a compact and expressive latent space for regularizing the downstream RILR objective (4.14). Moreover, it is important to use the latent information to guide the reconstruction of IHRs (as regularizations in RILR), as opposed to using the VLM-H to predict human returns standalone; since limited convergence guarantees/error bounds can be provided for VAE-based latent models, which is illustrated in both Figure 4.3 and Table 4.6 where OPEHF largely outperforms the VLM-H ablation over MAE and rank.

4.6 A Challenging Simulation Environment: Visual Q&A Dialogue

We also consider the visual Q&A dialogue environment Das et al., 2017, following the recent framework Snell et al., 2022 that generalizes offline RL towards language generation tasks. In this environment, language generation agents are trained to ask questions over the image captions that are given (without showing the actual images), in order to gain as much information from the responses (returned by environment) that are useful to characterize the the key elements contained in the underlying image.

4.6.1 Overall Setup.

Though this environment does not involve actually HF, the policies’ performance intrinsically pertain to two types of returns as per the setup from Snell et al., 2022. Specifically, the immediate rewards, used to train the language generation policy (captured by generative pre-trained transformer version 2, or GPT2) to asks questions, are discrete, *i.e.*, a -1 reward is given at each step until most key elements can be inferred from the dialogue, and an additional -1 reward is given if the questions asked by the agent lead to responses that only contain trivial information towards extrapolating the image characteristics (such as the responses that only contain a ‘yes/no’). In contrast, a separate pre-trained referee model is used to determine when sufficient elements in the underlying images can be captured from the dialogue history and terminates the episode, which maps the dialogue history to embeddings that represent the features pertaining to the image elements that can be extrapolated from the dialogue, followed by scoring the dialogue with relative percentile ranking of the ground truth image’s distance from the predicted embedding among a set of images taken from the evaluation set provided by Das et al., 2017; Snell et al., 2022, *i.e.*, an episode is terminated if the percentile ranking is improved by at least 50% compared to the percentile ranking before the dialogue begins, $(1 - p_T) \geq .5 \cdot (1 - p_0)$, where p_T is the ground truth image’s percentile rank at dialogue’s last turn and p_0 is the ground truth image’s percentile rank at the beginning of the dialogue, when only the image caption is observed. As a result, the former type of rewards are considered as the *environmental rewards* as per the OPEHF setup, and the latter one is closely related to the human returns as it is only provided at the end of each episode and are obtained following a different schema. Specifically, we consider $p_0 - p_T$ as the *synthetic* human return, which quantifies the improvement over the percentile ranking throughout the dialogue. More details on the environmental reward function and the percentile ranking generated by the referee model can be found in Snell et al., 2022.

4.6.2 MDP Formulation.

Consequently, in this experiment we compare OPEHF’s performance against baselines, toward predicting the percentile rankings corresponding to the embeddings predicted from the referee model at the end of episodes. The MDP states are captured by the embeddings output by the attention layer from the encoder of the GPT2, by feeding in the dialogue history up to the current step. The action space is captured as \mathcal{V}^l , where \mathcal{V} is the vocabulary considered and l is the maximum number of words the agent is allowed to place in the questions in each step. The definition of environmental rewards and synthetic human returns are introduced above.

4.6.3 Offline Trajectories and Target Policies.

Five behavioral policies are used to collect offline trajectories, which are obtained from training over 4 types of RL algorithms following the setup introduced in Snell et al., 2022, *i.e.*, 1 policy follows conservative Q-learning (CQL), 2 follow implicit Q-learning (IQL), 1 follows behavioral clone (BC) and the last follows decision transformer (DT). All the behavioral policies are trained to obtain only 10%-50% of the best synthetic human returns that can be achieved, which are considered sub-optimal. Eight different policies (also obtained from 4 types of RL algorithms) constitute the set of target policies whose performance will be estimated by OPEHF and the baselines. Their performance spread roughly equally between 0%-100% of the optimal return that could be achieved; details can be found in Table 4.9. It can be found that synthetic human returns can be negative if the dialogue provides trivial, or sometimes misleading, information that are not helpful for capturing the underlying image. In practice one can shift the returns to become all positive in order to comply with the assumption in Proposition 1. Moderate correlations are found between the environmental and synthetic human returns in this experiment, *i.e.*, Pearson’s correlation coefficient is 0.606 and Spearman’s rank correlation coefficient is 0.69.

Table 4.9: Environmental and synthetic human returns of the target policies, as well as the algorithms used to obtain them, considered in the Q&A dialogue experiment.

Target Policy	Environmental Returns	Synthetic Human Returns	Algorithm
#0	-15.46	-18.15	CQL
#1	-11.77	-3.20	IQL
#2	-10.86	1.09	CQL
#3	-13.39	2.24	CQL
#4	-13.50	2.36	BC
#5	-11.33	3.09	CQL
#6	-6.05	3.23	IQL
#7	-9.77	3.66	DT

4.6.4 Results and Discussion.

The results are summarized in Table 4.8. It can be observed that the OPEHF in general outperforms the baselines, with significantly lower MAEs and higher ranks if IS, DR or FQE are selected as the downstream estimator. We also find that if DICE is selected as the downstream estimator, it in general leads to lower MAEs and regrets for both the OPEHF and rescale baseline, but significantly low rank correlations. Our hypothesis would be that considering DICE directly estimate the discrepancy between target and behavioral policy’s propensity of visiting particular state-action pairs across the offline trajectories, it may benefit from leveraging the information (over the state-action space) captured by a language encoder that place roughly equal attention to each step of the dialogue, as opposed to the GPT2’s encoder which is not fine-tuned specifically for such a case.

5. A Real-World Case Study in Healthcare: Deep Brain Stimulation (DBS)

Deep brain stimulation (DBS) is effective in treating motor symptoms caused by Parkinson’s disease (PD), by delivering electrical pulses to the basal ganglia (BG) region of the brain. However, DBS devices approved by the U.S. Food and Drug Administration (FDA) can only deliver continuous DBS (cDBS) at a fixed amplitude; this energy inefficient operation reduces battery lifetime of the device, cannot adapt treatment dynamically for patient activity, and may cause significant side-effects (*e.g.*, gait impairment). In this chapter, we introduce an offline reinforcement learning (RL) framework, allowing the use of past clinical data to train an RL policy to adjust the stimulation amplitude in real time, with the goal of reducing energy use while maintaining the same level of treatment (*i.e.*, control) efficacy as cDBS. Moreover, clinical protocols require the safety and performance of such RL controllers to be demonstrated ahead of deployments in patients. Thus, we also introduce an offline policy evaluation (OPE) method to estimate the performance of RL policies using historical data, before deploying them on patients. We evaluated our framework on five PD patients implanted with the RC+S DBS system, employing the RL controllers during monthly clinical visits, with the overall *control efficacy* evaluated by severity of symptoms (*i.e.*, bradykinesia and tremor), changes in PD biomarkers (*i.e.*, local field potentials), and patient ratings. The results from clinical experiments show that our RL-based controller maintains the same level of control efficacy as cDBS, but with significantly reduced stimulation energy. Further, the OPE method is shown effective in accurately estimating and ranking the expected returns of RL controllers.

5.1 Related Work and Motivation

Existing DBS control methods primarily involve simple on/off switching of stimulation or adjusting stimulation intensity using a proportional control approach (Hoang et al., 2017). These adjustments are triggered by changes in specific biomarkers, typically when they exceed predetermined thresholds (Arlotti, Rosa, et al., 2016; Arlotti, Rossi, et al., 2016;

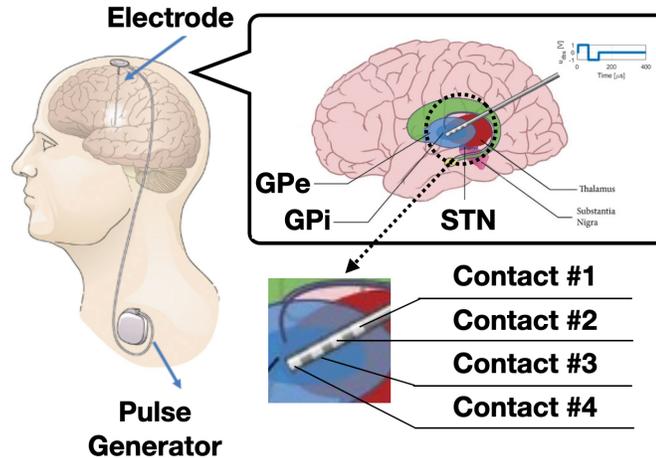


FIGURE 5.1: An implantable deep brain stimulation (DBS) device. The stimuli, generated by the pulse generator at a given amplitude and frequency, are delivered to the basal ganglia (BG) through multi-contact electrodes. Each electrode has four contacts; one stimulates the BG and the two surrounding it sense local field potentials (LFPs) that may be used for control feedback.

Beudel & Brown, 2016; S. Little et al., 2013, 2016; Opri et al., 2020; Swann et al., 2016). These biomarkers include local field potentials (LFPs) from the BG, electroencephalography (EEG) from the cortex, as well as data from wearable devices like accelerometry and electromyography (Opri et al., 2020). Despite enhancing energy efficiency (Habets et al., 2018; S. Little et al., 2016), these methods still require significant effort to empirically fine-tune the thresholds for each individual patient. Additionally, patients may experience sub-optimal DBS settings between clinical visits, resulting in inadequate symptom control due to variations in their condition. Factors such as exercise, fluctuations in medication dosage or timing can influence their PD symptoms and DBS control, leading to biased tuning results. Consequently, the primary **challenge (I)** in developing closed-loop DBS controllers is to ensure consistent performance across diverse and dynamic patient contexts and states.

5.1.1 The Need for Closed-Loop DBS

Parkinson’s disease (PD) is characterized by the progressive death of dopaminergic neurons in the substantia nigra region of the brain. This neuronal loss leads to changes in

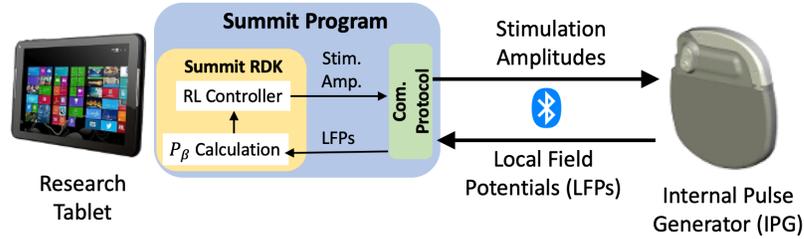


FIGURE 5.2: The overall architecture of the Summit RC+S DBS system. The Summit research and development kit (RDK) can be used to configure the Summit program, allowing us to compute the beta amplitude (P_{β}) and execute the RL controller.

dopaminergic signaling and abnormal activity in the basal ganglia (BG) regions targeted by DBS, namely the globus pallidus pars interna (GPi), globus pallidus pars externa (GPe), and subthalamic nucleus (STN) (see Figure 5.1). The reduced number of neurons and decreased dopamine levels in the BG contribute to various motor symptoms observed in PD patients, such as bradykinesia (slowness of movement) and tremor (Brown et al., 2001; Kühn et al., 2006; Okun, 2012b). The pathophysiological effects of PD can be captured through changes in LFPs recorded from the GPi, GPe, and STN regions. PD can cause abnormal firing patterns of neurons in these regions, leading to an increase in beta-band (13-35 Hz) amplitude, or P_{β} , in the LFPs (Q. Gao, Schmidt, et al., 2022). Increased beta amplitude is a characteristic feature of PD and is associated with the akinetic and rigid symptoms observed in patients.

DBS devices are equipped with multi-contact electrodes that are implanted in the STN and/or GP, as depicted in Figure 5.1. In this work, we utilized 4-contact electrodes placed bilaterally in both the STN and GP regions. Monopolar stimulation was delivered through a single contact on each lead, with the case of the device serving as the counter-electrode. The two contacts surrounding the stimulation contact were used for sensing LFPs, employing a technique known as sandwich sensing. The current open-loop cDBS devices stimulate the brain with pulses at a fixed amplitude, which, in many cases, can effectively correct abnormal neuronal activity (Kuncel & Grill, 2004). However, continuous high-amplitude stimulation significantly reduces the battery lifespan of the DBS device and can lead to side effects, such as speech impairment (Beric et al., 2001; S. Little et al., 2013; Swann et al.,

2016). Therefore, it is crucial to develop DBS controllers that are both effective in terms of therapeutic outcomes and energy-efficient. The proposed closed-loop DBS approach aims to strike a balance between achieving optimal therapeutic benefits and conserving energy, thus addressing the limitations of continuous high-amplitude stimulation.

As mentioned in Section 1, the current aDBS approaches involve significant time and effort to determine appropriate stimulation thresholds through trial-and-error (Wong et al., 2022). Several deep RL based controllers have been proposed for closed-loop DBS, that can dynamically adjust the stimulation pulse amplitude in response to feedback signals such as P_β (Q. Gao, Naumann, et al., 2020; Q. Gao, Schmidt, et al., 2022). However, these frameworks have only been validated through numerical simulations using simplified computational models of the BG, rather than in real clinical studies with human participants. In real-world scenarios, learning an RL policy with suitable control efficacy and patient satisfaction may require a substantial amount of historical experience or trajectories collected from past interactions between the controller and the patient (Lee et al., 2020). Offline RL offers a promising solution to address this challenge. It allows the use of data collected from any type of controller, including cDBS or policies that switch between arbitrary stimulation amplitudes/frequencies, to optimize an RL control policy. Moreover, before deploying a new control policy to a patient, clinicians need to assess its effectiveness and may require justifications based on its estimated control efficacy and performance (Parvinian et al., 2018). OPE plays a crucial role in such an context, as it can estimate the expected return of RL policies using historical trajectories, bridging the gap between offline RL training and evaluations. In the following two subsections, preliminaries for offline RL and OPE are presented, demonstrating how these methods can potentially overcome the challenges of closed-loop DBS and facilitate more effective and patient-specific treatments.

5.2 DBS Setup Used in Clinical Trials

We build on the research-only Medtronic Summit RC+S system (Stanslaski et al., 2018) to enable testing of RL-based controllers in clinical studies. The overall architecture of the

RC+S-based system we developed is illustrated in Figure 5.2. Specifically, Medtronic provides the code and communication APIs (Summit program), which enable the stimulation amplitude of the pulses delivered by the internal pulse generator (IPG) to be adapted over time. The Summit program is developed using the C# language under the .NET framework, which we extended to execute RL policies leveraging the provided Summit research development kit (RDKit), requiring the use of a Windows OS.

Thus, a research tablet is used for the execution of the developed DBS controllers; the desired stimulation amplitude is computed for each control cycle (every 2 seconds) and sent to the IPG over BluetoothTM, using proprietary communication and security protocols. In the other direction, the IPG transmits to the controller the LFPs captured from the BG, from which the beta amplitude of the LFPs (P_β), is calculated and used as a quality of control (QoC) metric as well as potential control feedback signals (*i.e.*, inputs to the RL controller). Each clinical trial session lasts 5-20 minutes depending on the schedule of the visit, and multiple controllers can be tested across different sessions. All the computed P_β and stimulation amplitudes applied over time are logged for future training and evaluation purposes, as summarized in Figure 4.2. Specifically, a total of three data streams are collected: (1) the LFPs and stimulation amplitudes are recorded over time; the logged trajectories are used to evaluate the performance of deployed RL controllers, as well as training data for further fine-tuning; (2) patient feedback including results from bradykinesia tests and a rating on the scale between 1-10; (3) patient tremor severity captured by wearable devices. For the developed system design, we obtained an FDA Investigative Device Exemption (IDE) G180280 and Institutional Review Board (IRB) protocol approval by Duke University Health System.

In addition to P_β , three other QoC metrics are collected from every patient at the end of each session. Specifically, near the end of each session, the patient is asked to perform 10 seconds of hand grasps (rapid and full extension and flexion of all fingers) (Ramaker et al., 2002) to evaluate the severity of bradykinesia. Hand motions are captured by a leap motion sensor by Ultraleap (Butt et al., 2018). Then, the elapsed time between any

two consecutive open fist is captured and recorded by the sensor, after which the grasp frequency can be calculated as

$$QoC_{grasp} = \frac{1}{\frac{1}{N-1} \sum_{i=1}^{N-1} t_{(i,i+1)}}; \quad (5.1)$$

here, N is the total number of open fists throughout the 10 s test, and $t_{(i,i+1)}$ is the time spent between the i -th and $i + 1$ -th grasp. Further, at the end of each session, the patient provides a score between 1-10, with 10 indicating the highest level of satisfaction with the treatment received in the past session, and 1 being the lowest, *i.e.*,

$$QoC_{rate} \in [1, 10] \subset \mathbb{Z}^+. \quad (5.2)$$

The grasp frequency and rating for each session are also recorded, which corresponds to the patient feedback stream in Figure 4.2.

Throughout all sessions, an Apple watch is worn by the patient at their wrist, where the Apple’s movement disorders kit (Powers et al., 2021) is used to analyze the accelerometry, classifying the patient’s tremor severity as no-tremor, slight, mild, moderate and strong every 1 minute, following StrivePD’s implementation (W. Chen et al., 2021). At the end of each session, an overall tremor severity is recorded as the fraction of time the patient experiencing mild (T_{mild}), moderate ($T_{moderate}$) or strong (T_{strong}) tremor over the entire session with length $T_{session}$, *i.e.*,

$$QoC_{tremor} = \frac{T_{mild} + T_{moderate} + T_{strong}}{T_{session}} \times 100\%. \quad (5.3)$$

The three data streams are collected from all trial sessions after each clinical visit. Moreover, each time a patient may come into the clinic with slightly different PD conditions (*e.g.*, pathology progression over time), medication prescriptions, activity levels etc.; thus, our goal is to capture the impact of such changes via the data collection process, in order to facilitate the training and testing the offline RL and OPE frameworks for DBS.

5.3 Offline RL Design of DBS Controllers

In this section, we employ offline RL for learning control policies for DBS clinical studies, starting from the formulation of an MDP \mathcal{M} capturing the underlying neurological dynam-

ics in the BG, and the policy distillation technique that allows reducing the computational time and resource needed to evaluate the RL policies (represented by DNNs).

5.3.1 Modeling the Basal Ganglia (BG) as an Markov Decision Process (MDP)

We now define the elements of an MDP $\mathcal{M} = (\mathcal{S}, s_0, \mathcal{A}, \mathcal{P}, R, \gamma)$.

5.3.1.1 State Space \mathcal{S} and the Initial State s_0

As discussed in Section 5.2, our DBS controller supports calculation of P_β from LFPs, and the changes in P_β can be used as a biomarker for PD-levels for some patients. Thus, we consider the MDP state, at a *discrete* time step t , as a historical sequence of P_β sampled at a fixed intervals, captured by $m \in \mathbb{Z}^+$, over a sliding queue of size $W \in \mathbb{Z}^+$, *i.e.*,

$$s_t = [\beta_{(\tilde{t}-(W-1)m)}, \beta_{(\tilde{t}-(W-2)m)}, \dots, \beta_{(\tilde{t}-2m)}, \beta_{(\tilde{t}-m)}, \beta_{(\tilde{t})}]. \quad (5.4)$$

Here, $\beta_{(\tilde{t})}$'s are the P_β evaluated at the elapsed time \tilde{t} since the clinical trial starts, m is configurable in our system design (Figure 5.2), and we used $m = 2$ corresponding to calculating P_β every 2 s, resulting in 20 s time-windows for $W = 10$ elements in the queue; finally, $s_t \in \mathbb{R}^W$ is the state at t -th (discrete) step of the MDP. The initial state s_0 is considered to be the β sequence collected right before the clinical trial starts, *i.e.*, from $\tilde{t} = -(W-1)m$ to $\tilde{t} = 0$.

5.3.1.2 Action Space \mathcal{A}

The amplitude of DBS stimulation pulses can be changed in pre-defined (discrete) time steps, *i.e.*, every 2 seconds for the developed controllers. We consider the actions a_t as the percentage of the cDBS amplitude determined by clinicians; *i.e.*, $a_t \in [0, 1] \subset \mathbb{R}$, where $a_t = 0$ and $a_t = 1$ correspond to no-DBS and stimulation with the same amplitude as in cDBS, respectively.

5.3.1.3 Transition Dynamics $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$

Every time after the stimulation amplitude is adjusted following a_t , the system computes the latest $\beta_{(\tilde{t}+m)}$ using the LFPs sent back from the IPG; this leads to the MDP state at

the $(t+1)$ -th (discrete) step as

$$s_{t+1} = [\beta_{(\tilde{t}-(W-2)m)}, \beta_{(\tilde{t}-(W-3)m)}, \dots, \beta_{(\tilde{t})}, \beta_{(\tilde{t}+m)}], \quad (5.5)$$

i.e., the left-most element in (5.4) is pushed out, with $\beta_{(\tilde{t}+m)}$ appended to the right-end. Note that we define the MDP states s_t and actions a_t over discrete time steps, t 's, instead the elapsed time \tilde{t} , for the conciseness of equations and presentations below. Now, the MDP transitions are captured to directly follow $s_{t+1} \sim \mathcal{P}(s_t, a_t)$.

5.3.1.4 Reward Function $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$

Following from the setup of the DBS system (Section 5.2), we define the rewards as

$$R(s_t, a_t, s_{t+1}) = \begin{cases} r_a - C_1 \cdot a_t, & \text{if } \bar{\beta}_{(\tilde{t}+m)} < \bar{\zeta}_\beta; \\ r_b - C_1 \cdot a_t, & \text{if } \bar{\beta}_{(\tilde{t}+m)} \geq \bar{\zeta}_\beta; \end{cases} \quad (5.6)$$

specifically, if the beta amplitude received at the $(t+1)$ -th step, $\beta_{(\tilde{t}+m)}$, is less than some threshold $\bar{\zeta}_\beta$, then a non-negative reward r_a is issued along with the term $-C_1 \cdot a_t$ ($C_1 > 0$, $C_1 \in \mathbb{R}$) penalizing over-usage of large stimulation amplitudes (for better energy efficiency). On the other hand, if $\beta_{(\tilde{t}+m)}$ is greater than the threshold $\bar{\zeta}_\beta$, a negative reward r_b will be used to replace r_a above.

Remark 1. *The reward functions used for RL training do not consider the QoC metrics that are available not at every step of the control execution (i.e., every 2 s) but only at the end of each clinical session, i.e., QoC_{grasp} , QoC_{rate} , QoC_{tremor} from (5.1), (5.2), (5.3). The reason is that the horizon T is usually large and their coverage can be very sparse. Instead, these QoC metrics serve as great measurements quantifying how well the policies perform, which are thus leveraged by the OPE techniques introduced in Section 5.4.*

For the introduced MDP \mathcal{M} , we leverage the offline RL framework introduced in Section 2.2 to search for the target policy π^* . Following from Definition 3, it requires an experience replay buffer \mathcal{E}^μ that consists of historical trajectories collected over some behavioral policy μ . At the beginning of offline RL training, exploration of the environment is deemed more important than exploitation (Ishii et al., 2002). Hence, a controller that

generates random actions uniformly from $[B, 1]$ is used to constitute \mathcal{E}^μ at earlier stage of clinical trials, where B is the lower bound from which the random a_t can be generated, for the sake of patient’s safety and acceptance.

Once the RL policies attain satisfactory overall performance, *i.e.*, quantified as achieving significantly improved QoCs (introduced in Section 5.2) compared to the random controller above, we consider including into \mathcal{E}^μ the trajectories obtained from such RL policies. From this point onward, the replay buffer \mathcal{E}^μ will be iteratively updated and enriched with the RL-induced trajectories after each trial. Consequently, the behavioral policy μ can be considered as a mixture of random control policy and several RL policies deployed in past trials in general. With \mathcal{E}^μ being defined, the objective for training RL policies, (2.10), can be optimized using gradient descent (Q. Gao, Naumann, et al., 2020; Q. Gao, Schmidt, et al., 2022; Lillicrap et al., 2016).

5.3.2 Policy Distillation

Our system design (Figure 5.2) is set to process various tasks in each 2 s stimulation (*i.e.*, control) period, facilitating communication between the research tablet and IPG, computing P_β from LFPs, evaluating the RL controller, data logging, and other basic functions that ensure the safety and efficacy of DBS. Hence, it was critical to reduce the overall computation requirements, such that each task meets the required timing, as well as prolong the battery lifetime. As introduced in Section 2.2, the RL policies are parameterized as DNNs; although a forward pass of a DNN would not require as much computational resources as for training (through back-propagation), it may still involve hundreds of thousands of multiplication operations. For example, consider the recommended DNN size as in (Lillicrap et al., 2016), it takes at least 120,000 multiplications to evaluate a two-layer NN with 400 and 300 nodes each. Hence, we integrate into our system the model/policy distillation techniques (Hinton et al., n.d.), allowing smaller sized NNs to parameterize RL policies.

We build on a similar approach as in (Rusu et al., 2016), originally proposed to reduce

Appointment	Patient not present	Patient not present	Patient not present	Next Appointment
-------------	---------------------	---------------------	---------------------	------------------

Phase I -- Testing	Phase II -- Offline RL	Phase III -- OPE	Phase IV -- Testing
Data collection using the latest controller.	Use the updated experience dataset to fine-tune existing controllers or train new ones.	Estimate the controller candidates <i>offline</i> and select the one with best performance.	Run the best performing controller selected, and collect new trajectories.

FIGURE 5.3: Timeline for training RL-based DBS controllers in clinical studies. Since only limited data can be collected during each clinical visit, offline RL can be used to fine-tune existing or train new controllers using all the historical data. Then, offline policy evaluation (OPE) facilitates choosing the possible top-performing ones to be tested in the next visit.

the size of DNNs used in deep Q-learning (Mnih et al., 2015), which only works for a discrete action space. In particular, our extension allows for the use in the deterministic actor-critic cases considered in this work. Consider the original policy (*teacher*) π_{θ_a} parameterized by a DNN with weights θ_a . We train a smaller-sized DNN (*student*) with weights $\tilde{\theta}_a$ to learn θ_a 's behavior, by minimizing the mean squared error

$$\min_{\tilde{\theta}_a} \|\pi_{\theta_a}(s_t) - \pi_{\tilde{\theta}_a}(s_t)\|^2, \tag{5.7}$$

for all state samples contained in the experience replay $s_t \in \mathcal{E}^\mu$. We also consider augmenting the data used to optimize (5.7) to smooth out the learning process. We introduce synthetic states, \tilde{s}_t 's, where each \tilde{s}_t is generated by adding noise to each dimension of a state sample s_t that is originally in \mathcal{E}^μ ; the noise is sampled from a zero-mean Gaussian distribution, $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ with σ being a hyper-parameter.

5.4 OPE of DBS Controllers Including Patient Feedback and Tremor Data

Figure 5.3 RL illustrated the need of OPE in obtaining RL-based DBS controllers that can be tested with human participants. As discussed in Remark 1, besides the reward function introduced in Section 5.3.1, for OPE we employ QoC metrics QoC_{grasp} , QoC_{rate} , and QoC_{tremor} defined in (5.1), (5.2), (5.3), respectively, which are only available at the end of each session. As these well-capture performance (i.e., therapy effectiveness) of the

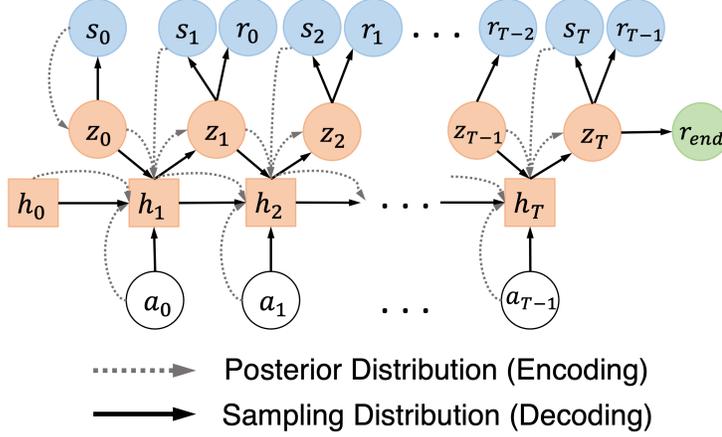


FIGURE 5.4: Architecture of the new deep latent sequential model (DLSM). The conditional dependencies between the variables from the posterior and sampling distributions are shown in dashed and solid lines, respectively.

considered policy, for OPE we additionally consider the end-of-session rewards defined as

$$\begin{aligned}
 r_{end} &= R_{end}(s_0, a_0, s_1, a_1, \dots, s_{T-1}, a_{T-1}, s_T) \\
 &= C_2 \cdot QoC_{grasp} + C_3 \cdot QoC_{rate} - C_4 \cdot QoC_{tremor},
 \end{aligned} \tag{5.8}$$

with $C_2, C_3, C_4 > 0$ real constants.

Without loss of generality, we slightly modify the total return under policy π (for the objective of OPE 2.11) as

$$G_0^\pi = r_{end} + \sum_{t=0}^T \gamma^t r_t, \tag{5.9}$$

where r_t and r_{end} follow from (5.6) and (5.8), respectively.

To better capture the end-of-episode rewards, we slightly modify the VLBM introduced in Chapter 3 by introducing the *deep latent sequential model* (DLSM). The overall model architecture is shown in Figure 5.4. First, the latent prior $p_\psi(z_0)$ is defined only over the initial latent variable at step $t = 0$, z_0 , which follows a multivariate Gaussian distribution with zero mean and identity covariance matrix.

Then, the encoder (approximated posterior) is defined over each trajectory (from $t = 0$ to T) as

$$q_\phi(z_{0:T}|s_{0:T}, a_{0:T-1}) = q_\phi(z_0|s_0) \prod_{t=1}^T q_\phi(z_t|z_{t-1}, a_{t-1}, s_t). \tag{5.10}$$

Further, the second term $q_\phi(z_t|z_{t-1}, a_{t-1}, s_t)$, which enforces the transitions between z_{t-1} and z_t conditioned on (a_{t-1}, s_t) and enables the encoder to capture the dynamical transitions in the LVS, can be obtained iteratively following

$$z_0^\phi \sim q_\phi(z_0|s_0), h_t^\phi = f_\phi(h_{t-1}^\phi, z_{t-1}^\phi, a_{t-1}, s_t), z_t^\phi \sim q_\phi(z_t|h_t^\phi); \quad (5.11)$$

here, $q_\phi(z_0|s_0)$ and $q_\phi(z_t|h_t^\phi)$ are parameterized by multivariate diagonal Gaussian distributions, each with mean and covariance determined by a feedforward DNN (Bishop, 2006); moreover, h_t^ϕ is the hidden state of a recurrent DNN, such as long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997), capturing the historical transitions among s_t , a_t and z_t^ϕ for all past steps up until $t - 1$ within each trajectory.

The decoder (sampling distribution) is responsible for interacting with the target policies to be evaluated, from which the expected returns can be estimated as the mean return obtained by the simulated trajectories. Specifically, the decoder is defined as follows, *i.e.*,

$$p_\psi(z_{1:T}, s_{0:T}, r_{0:T-1}, r_{end}|z_0) = p_\psi(r_{end}|z_T) \cdot \prod_{t=0}^T p_\psi(s_t|z_t) \prod_{t=1}^T p_\psi(z_t|z_{t-1}, a_{t-1}) p_\psi(r_{t-1}|z_t); \quad (5.12)$$

here, $p_\psi(r_{end}|z_T)$ estimates the end-of-session rewards given the latent variable at $t = T$, z_T ; $p_\psi(s_t|z_t)$, $p_\psi(r_{t-1}|z_t)$ reconstruct the states and rewards; $p_\psi(z_t|z_{t-1}, a_{t-1})$ enforces the transitions over the latent variables, z_t 's, conditioned on the actions; and $z_0 \sim p_\psi(z_0)$ is sampled from the prior. As a result, each simulated trajectory can be generated by the decoder following

$$h_t^\psi = f_\psi(h_{t-1}^\psi, z_{t-1}^\psi, a_{t-1}), z_t^\psi \sim p_\psi(z_t|h_t^\psi), s_t^\psi \sim p_\psi(s_t|z_t^\psi), \\ r_{t-1}^\psi \sim p_\psi(r_{t-1}|z_t^\psi), a_{t-1} \sim \pi(a_{t-1}|s_{t-1}^\psi), r_{end}^\psi \sim p_\psi(r_{end}|z_T); \quad (5.13)$$

here, h_t^ψ is the hidden state of a recurrent DNN; $p_\psi(z_t|h_t^\psi)$, $p_\psi(s_t|z_t^\psi)$, $p_\psi(r_{t-1}|z_t^\psi)$ and $p_\psi(r_{end}|z_T)$ are multivariate diagonal Gaussians with means and covariances determined by four feedforward DNNs separately. Hence, s_t^ψ 's and r_{t-1}^ψ 's can be sampled iteratively following the process above, using the actions obtained from the target policy $a_{t-1} \sim$

$\pi(a_{t-1}|s_{t-1}^\psi)$ accordingly, which constitute the simulated trajectories; and r_{end}^ψ is sampled at the end of each simulated trajectory.

The theorem below derives an ELBO for the joint log-likelihood $\log p_\psi(s_{0:T}, r_{0:T-1}, r_{end})$, following the above DLSM architecture.

Theorem 1 (ELBO for DLSM). *An ELBO of the joint log-likelihood $\log p_\psi(s_{0:T}, r_{0:T-1}, r_{end})$ can be obtained as*

$$\begin{aligned} \mathcal{L}_{ELBO}(\psi, \phi) = & \mathbb{E}_{z_t \sim q_\phi} \left[\sum_{t=0}^T \log p_\psi(s_t | z_t) + \sum_{t=1}^T \log p_\psi(r_{t-1} | z_t) \right. \\ & + \log p_\psi(r_{end} | z_T) - KL(q_\phi(z_0 | s_0) || p(z_0)) \\ & \left. - \sum_{t=1}^T KL(q_\phi(z_t | z_{t-1}, a_{t-1}, s_t) || p_\psi(z_t | z_{t-1}, a_{t-1})) \right] \end{aligned} \quad (5.14)$$

$$\leq \log p_\psi(s_{0:T}, r_{0:T-1}, r_{end}); \quad (5.15)$$

here, the first three terms are the log-likelihood of the decoder to reconstruct s_t , r_{t-1} and r_{end} correctly, and the two terms that follow regularize the transitions captured by the encoder over the LVS, with $KL(\cdot || \cdot)$ being the Kullback–Leibler (KL) divergence (Kullback & Leibler, 1951). The proof can be found below.

Proof. We now derive the evidence lower bound (ELBO) for the joint log-likelihood dis-

tribution, *i.e.*,

$$\log p_\psi(s_{0:T}, r_{0:T-1}, r_{end}) \quad (5.16)$$

$$= \log \int_{z_{1:T} \in \mathcal{Z}} p_\psi(s_{0:T}, z_{1:T}, r_{0:T-1}, r_{end}) dz \quad (5.17)$$

$$= \log \int_{z_{1:T} \in \mathcal{Z}} \frac{p_\psi(s_{0:T}, z_{1:T}, r_{0:T-1}, r_{end})}{q_\phi(z_{0:T}|s_{0:T}, a_{0:T-1})} q_\phi(z_{0:T}|s_{0:T}, a_{0:T-1}) dz \quad (5.18)$$

$$\begin{aligned} &\geq \mathbb{E}_{z_t \sim q_\phi} [\log p(z_0) + \log p_\psi(s_{0:T}, z_{1:T}, r_{0:T-1}|z_0) + \log p_\psi(r_{end}|z_T) \\ &\quad - \log q_\phi(z_{0:T}|s_{0:T}, a_{0:T-1})] \end{aligned} \quad (5.19)$$

$$\begin{aligned} &= \mathbb{E}_{z_t \sim q_\phi} \left[\log p(z_0) + \log p_\psi(s_0|z_0) + \log p_\psi(r_{end}|z_T) \right. \\ &\quad \left. + \sum_{t=1}^T \log p_\psi(s_t, z_t, r_{t-1}|z_{t-1}, a_{t-1}) \right. \\ &\quad \left. - \log q_\phi(z_0|s_0) - \sum_{t=1}^T \log q_\phi(z_t|z_{t-1}, a_{t-1}, s_t) \right] \end{aligned} \quad (5.20)$$

$$\begin{aligned} &= \mathbb{E}_{z_t \sim q_\phi} \left[\log p(z_0) - \log q_\phi(z_0|s_0) + \log p_\psi(s_0|z_0) + \log p_\psi(r_{end}|z_T) \right. \\ &\quad \left. + \sum_{t=1}^T \log (p_\psi(s_t|z_t) p_\psi(r_{t-1}|z_t) p_\psi(z_t|z_{t-1}, a_{t-1})) \right. \\ &\quad \left. - \sum_{t=1}^T \log q_\phi(z_t|z_{t-1}, a_{t-1}, s_t) \right] \end{aligned} \quad (5.21)$$

$$\begin{aligned} &= \mathbb{E}_{z_t \sim q_\phi} \left[\sum_{t=0}^T \log p_\psi(s_t|z_t) + \log p_\psi(r_{end}|z_T) \right. \\ &\quad \left. + \sum_{t=1}^T \log p_\psi(r_{t-1}|z_t) - \text{KL}(q_\phi(z_0|s_0) || p(z_0)) \right. \\ &\quad \left. - \sum_{t=1}^T \text{KL}(q_\phi(z_t|z_{t-1}, a_{t-1}, s_t) || p_\psi(z_t|z_{t-1}, a_{t-1})) \right]. \end{aligned} \quad (5.22)$$

Note that the transition from (5.18) to (5.19) follows Jensen's inequality. \square

Empirically, the ELBO can be evaluated using the trajectories from the experience replay \mathcal{E}^μ , by replacing the expectation as the mean over all trajectories, after which the objective $\max_{\psi, \phi} \mathcal{L}(\psi, \phi)$ can be achieved using gradient descent (Kingma & Ba, 2014) following the Algorithm 3 introduced in Section 5.4.1 below. Moreover, the reparameterization trick (Kingma & Welling, 2013) is used, which allows for the gradients to be

Algorithm 3 Train DLSM.

Require: Model weights ψ, ϕ , experience replay buffer \mathcal{E}^μ , and learning rate α .

Ensure:

- 1: Initialize ψ, ϕ
 - 2: **for** $iter$ in $1 : max_iter$ **do**
 - 3: Sample a trajectory $[(s_0, a_0, r_0, s_1), \dots, (s_{T-1}, a_{T-1}, r_{T-1}, s_T)] \sim \mathcal{E}^\mu$
 - 4: $z_0^\phi \sim q_\phi(z_0|s_0)$
 - 5: $z_0^\psi \sim p_\psi(z_0)$
 - 6: Run forward pass of DLSM following (5.11) and (5.13) for $t = 1 : T$, and collect all variables needed to evaluate the all terms within the expectation in \mathcal{L}_{ELBO} , which is denoted as $\tilde{\mathcal{L}}_{ELBO}$.
 - 7: $\psi \leftarrow \psi + \alpha \nabla_\psi \tilde{\mathcal{L}}_{ELBO}$
 - 8: $\phi \leftarrow \phi + \alpha \nabla_\phi \tilde{\mathcal{L}}_{ELBO}$
 - 9: **end for**
-

back-propagated when sampling from Gaussian distributions with means and covariances determined by DNNs. Details on reparameterization can be found in (Q. Gao, Schmidt, et al., 2022; Kingma & Welling, 2013).

5.4.1 Algorithm to Train Deep Latent Sequential Model (DLSM)

Here we introduce how to use gradient descent to maximize the ELBO (5.14), resulting in Algorithm 3. For simplicity, we first illustrate with the case where the training batch only contains a single trajectory, and then extend to the cases where each batch contain n trajectories. In each iteration, a trajectory is sampled from the experience replay buffer \mathcal{E}^μ . Then, the initial latent state in the encoder is obtained following $z_0^\phi \sim q_\phi(z_0|s_0)$, while the initial latent state for the sampling distribution is generated following the latent prior $z_0^\psi \sim p_\psi(z_0)$. The processes introduced in (5.11) can be used to generate z_t^ϕ 's iteratively given z_0^ϕ . Similarly, $s_t^\psi, r_t^\psi, z_t^\psi$ can be generated iteratively following (5.13). As a result, the log-likelihoods and KL-divergence terms within the expectation in \mathcal{L}_{ELBO} , defined in (5.14), can be evaluated using the variables above, after which ψ, ϕ can be updated using the gradients $\nabla_\psi \tilde{\mathcal{L}}_{ELBO}, \nabla_\phi \tilde{\mathcal{L}}_{ELBO}$, respectively, where $\tilde{\mathcal{L}}_{ELBO}$ refers to all the terms within the expectation in \mathcal{L}_{ELBO} . This algorithm is summarized in Alg. 3.

To extend to batch gradient descent, in line 3 of Algorithm 3, a batch of n trajectories,

$\mathcal{B}(n)$, will be sampled, *i.e.*,

$$\begin{aligned} \mathcal{B}(n) = & \left[[(s_0^{(1)}, a_0^{(1)}, r_0^{(1)}, s_1^{(1)}), \dots, (s_{T-1}^{(1)}, a_{T-1}^{(1)}, r_{T-1}^{(1)}, s_T^{(1)})], \dots, \right. \\ & \left. [(s_0^{(n)}, a_0^{(n)}, r_0^{(n)}, s_1^{(n)}), \dots, (s_{T-1}^{(n)}, a_{T-1}^{(n)}, r_{T-1}^{(n)}, s_T^{(n)})] \right] \sim \mathcal{E}^\mu. \end{aligned} \quad (5.23)$$

Then, the processes illustrated in lines 4-6 in Algorithm 3 can be executed in n parallel threads, with each corresponding to a unique trajectory in $\mathcal{B}(n)$. Further, then, $\tilde{\mathcal{L}}_{ELBO}$ can be evaluated as

$$\begin{aligned} \tilde{\mathcal{L}}_{ELBO}(\psi, \phi) = & \frac{1}{n} \sum_{i=1}^n \left[\sum_{t=0}^T \log p_\psi(s_t^{(i)} | z_t^{(i)}) \right. \\ & + \sum_{t=1}^T \log p_\psi(r_{t-1}^{(i)} | z_t^{(i)}) \\ & + \log p_\psi(r_{end}^{(i)} | z_T^{(i)}) - KL(q_\phi(z_0^{(i)} | s_0^{(i)}) || p(z_0^{(i)})) \\ & \left. - \sum_{t=1}^T KL(q_\phi(z_t^{(i)} | z_{t-1}^{(i)}, a_{t-1}^{(i)}, s_t^{(i)}) || p_\psi(z_t^{(i)} | z_{t-1}^{(i)}, a_{t-1}^{(i)})) \right], \end{aligned} \quad (5.24)$$

with $s_t^{(i)}, a_t^{(i)}, z_t^{(i)}, r_t^{(i)}, r_{end}^{(i)}$ being the variables involved in one of the threads above processing the i -th trajectory in $\mathcal{B}(n)$.

5.5 Clinical Evaluations

Using our closed-loop DBS system presented in Section 5.2, we evaluated the RL-based control framework in clinical studies on five PD patients at Duke University Health System. In particular, we evaluated and compared four different types of controllers: cDBS, RL, RL with policy distillation (*i.e.*, distilled RL), and no-DBS (*i.e.*, without stimulation). The electrodes of the DBS device were placed in STN and GP for all five participants; LFPs were sensed from STN and stimuli were delivered to both STN and GP.

Each participant also has had *different PD symptoms and severity*; their characteristics are summarized in Table 5.1. All trials were conducted under close supervision of clinical experts, strictly following the process approved by the Duke University Health System IRB protocol complying with the obtained FDA IDE (G180280). Further, all participants provided informed written consent. Study staff were on-site to monitor all testing sessions, and were ready to terminate the trial if participants wished to return to clinical

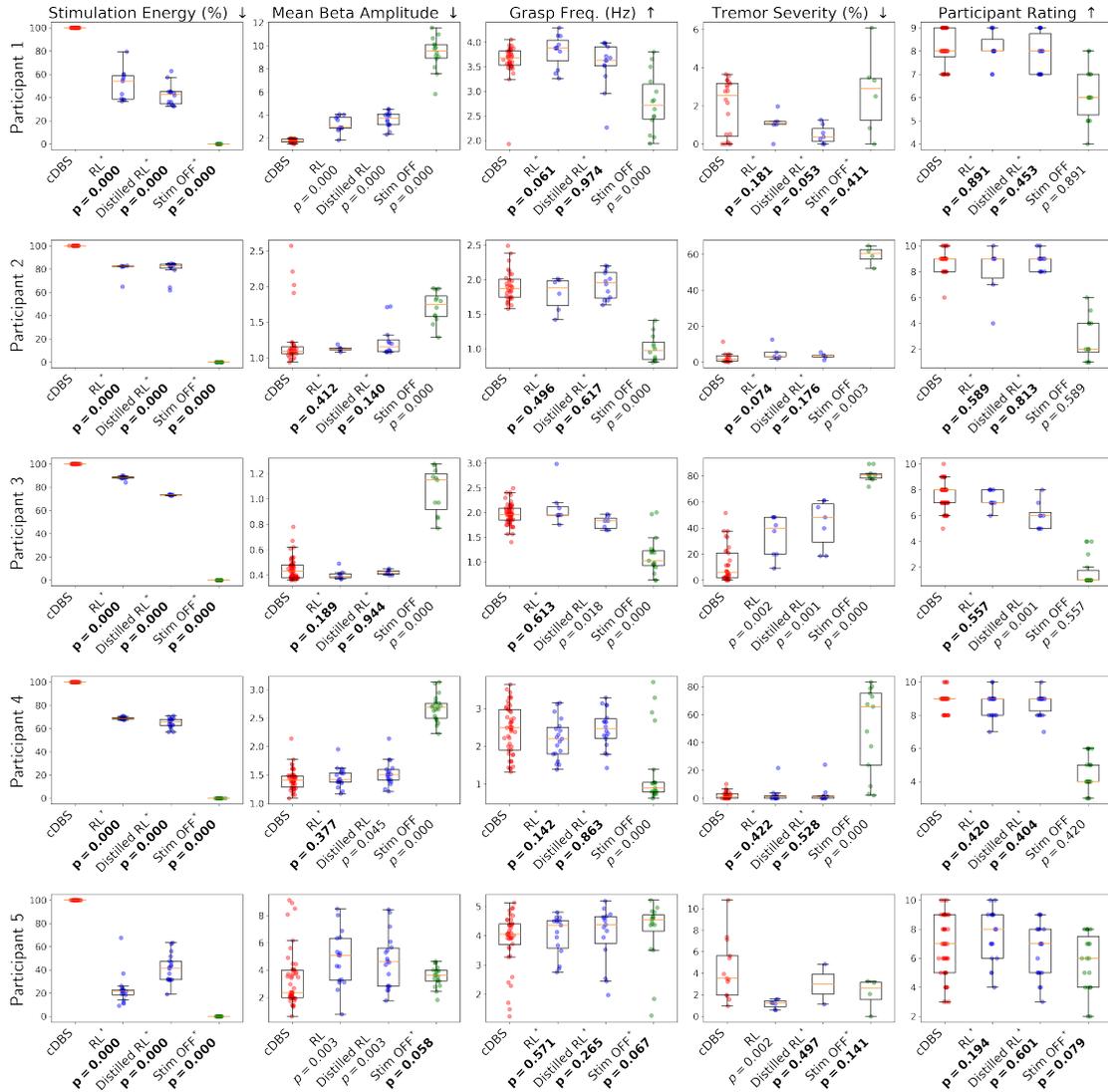


FIGURE 5.5: Quality of control (QoC) results from all clinical trials across participants. Wilcoxon rank-sum tests (Mann & Whitney, 1947) between cDBS and each of the other controllers are used to test the null hypothesis that two sets of measurements are drawn from the same distribution, resulting in the p -values reported above. The null hypothesis is rejected when consider the stimulation energy consumed by both RL controllers, illustrating that they lead to significant energy reduction compared to cDBS. For all other QoCs, the null hypothesis is accepted in majority cases, showing that both RL controllers can in general attain similar control efficacy to cDBS. The controllers that lead to the acceptance/rejection of the null hypothesis in the desired direction are highlighted with asterisks and bold p -values.

settings. Moreover, in clinic we employed two additional mechanism that minimizes the risk to participants. 1) We did not exceed clinical amplitudes as determined by the partic-

Table 5.1: Characteristic of each participant, including the level of tremor when receive insufficient stimulation, if affected by bradykinesia as a symptom of PD, as well as the PD medication dosage quantified as Levodopa equivalent daily dosage (LEDD) (Julien et al., 2021).

	Tremor Level	Bradykinesia	LEDD (mg)
Participant 1	Sporadic	Yes	57
Participant 2	Constant	Yes	113
Participant 3	Constant	Yes	200
Participant 4	Constant	Yes	100
Participant 5	None	Yes	713

ipants’ neurologists, and 2) there was a minimum amplitude that will always be delivered to participants for minimal symptom management needed.

5.5.1 Therapy Efficacy and Energy-Efficiency of the RL Control Policies

We followed the offline RL and policy distillation methodology introduced in Section 4 to train and update (distilled) RL policies iteratively over time. Specifically, each participant returned to clinic monthly, where during each day of trials a total of 2-4 RL policies would be tested. A cDBS session was placed between any two RL sessions as a control group. A small number of no-DBS sessions, with DBS stimulation fully off, were also tested, to validate our choice of the employed QoCs metrics – *i.e.*, whether they change when the participants are not stimulated.

After each trial day was completed, the trajectories collected from all the sessions were added to the experience replay buffer \mathcal{E}^{μ} unique to each participant. Between two consecutive visits of each participant, their \mathcal{E}^{μ} was used to fine-tune the top-performing policies determined from the last trial (using smaller learning rates between $[10^{-7}, 10^{-5}]$) or to train new policies from scratch (with learning rates between $[10^{-5}, 10^{-3}]$); such policies were then tested in the next visit. We followed (Lillicrap et al., 2016) and used two-layer NNs with 400 and 300 nodes each to parameterize the RL policies; moreover, a distilled version (student) of each corresponding full-sized RL policy (teacher) were trained as introduced in Section 5.3.2, with each represented as a two-layer NN with 20 and 10 nodes.

Table 5.2: Overall time, in minutes, spent toward testing each type of controller in clinical trials. Each testing session lasted 5-20 minutes, and no-DBS sessions were usually 5-min long to minimize the discomfort participants may experience.

	cDBS	RL	Distilled RL	No-DBS
Participant 1	183	202	197	66
Participant 2	175	227	222	67
Participant 3	285	129	115	84
Participant 4	234	269	248	98
Participant 5	194	188	250	95

Table 5.3: Computation time of the original RL versus the distilled RL policy.

	RL Policy (400×300 NN)	Distilled RL Policy (20×10 NN)
Mean of Computation Time	4.78 ms	2.98 ms
Std of Computation Time	32.26 ms	1.72 ms

The constants in (5.6) were set to $r_a = 0, r_b = -1, C_1 = 0.3$ for all participants.

In each testing session, to evaluate the overall performance of the employed control policy, a total of 5 metrics were considered: the energy used by the IPG for stimulation, the mean beta amplitude over the session, and the 3 QoCs introduced in Section 5.2; for QoC_{grasp} , we captured the grasp frequencies of the hand that best correlates with the PD symptom for the participant, following the patient characteristics documented in Table 5.1 above.

5.5.1.1 Results

Figure 5.5 summarizes the obtained results, and Table 5.2 documents the total amount of time each controller was tested in clinic. Wilcoxon rank-sum tests (Mann & Whitney, 1947) between cDBS and each of the other controllers were used to test the null hypothesis – *if two sets of measurements were drawn from the same distribution* (i.e., that the controllers perform similarly over the considered metrics); from this, p -values can be calculated. The p -values accepting/rejecting the null hypothesis in the desired direction are highlighted in Figure 5.5. Specifically, it can be observed that, compared to cDBS, the RL

policies and their distilled version can save substantial (15%-80%) and significant stimulation energy across participants; as $p < .05$ achieved for all participants, which rejected the null hypothesis.

When considering the other 4 metrics, there exist a great majority of results with $p \geq .05$, accepting the null hypothesis and indicating that both RL controllers attain control (i.e., therapy) efficacy similar to cDBS. In contrast, for the no-DBS sessions, the null hypothesis is rejected in most cases. Specifically, $p < .05$ attained by no-DBS over the mean beta amplitude, for participants 1-4, show that beta amplitudes can change significantly when sufficient DBS is delivered or not, which justified our choice of using the beta amplitudes to constitute MDP states. This also shows that the RL policies can follow the reward function (from Section 5.3.1) to optimize effectively the control strategies, with beta amplitudes also playing an important role. Participant 3 appears to benefit from relatively extensive stimulation intensity to reduce tremor. It can be observed that the RL controller (with only 15% energy saving) achieved the same level of participant performance in terms of grasp frequency and rating. However, the tremor measurements still differed from cDBS. Moreover, participant 5 appears to be an outlier, whose grasp speed, tremor severity and rating were in general maintained at the same level as cDBS regardless of the controller being used (including turn OFF DBS); such observations were consistent with the clinical findings documented in Table 5.1, where this participant did not have tremor nor bradykinesia, and no correlations were found among beta amplitude, DBS amplitude, grasp speed and rating. Consequently, the results show that both full and distilled RL policies can significantly reduce the stimulation energy in general, while achieving non-inferior control efficacy compared to cDBS.

5.5.1.2 Computational Complexity and Overall Energy Consumption

We also quantified the additional computation time and battery consumption of the DBS system due the use of full-sized RL policies or their distilled version. A Surface Go with an Intel Pentium Gold 4415Y CPU and 4GB RAM was used as the research tablet

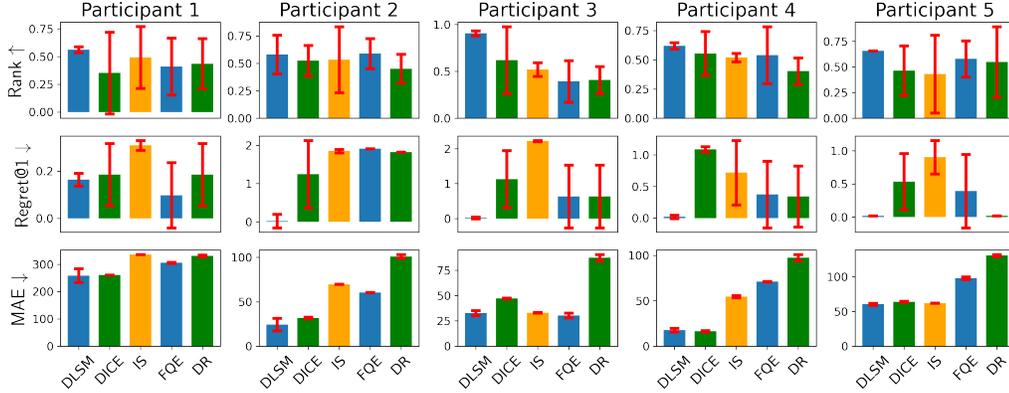


FIGURE 5.6: DLSM in general achieves higher ranks, lower regret@1’s and lower MAEs, compared to DICE and IS. Each method is trained and evaluated with 3 different random seeds, with the standard deviations shown by the error bars.

in Figure 4.2. The computation time was quantified as the time needed to run a single forward pass of the NN that represents the RL policy. We evaluate the forward passes for both types of RL policies 200 times; Table 5.3 summarizes the mean and standard deviation of the obtained computation times. As can be seen, the distilled RL policy can be evaluated significantly faster than its counterpart.

Moreover, we quantify the overall battery consumption of the entire DBS system as the time for which the tablet or the IPG battery drains from 100% to 10% (whichever comes first). We compare the battery runtime among the full RL and distilled RL, as well as a random controller that sets the IPG to stimulate with an arbitrary amplitude in each control cycle. Each experiment was repeated 3 times, resulting in the statistics in Table 5.4 showing that the two RL-based controllers do not dramatically shorten the runtime of the DBS system; *i.e.*, the energy used for RL-based control does not dominate the overall energy used by the DBS system.

5.5.2 Evaluation of the OPE Methodology

For each participant, a DLSM was trained following the methodology introduced in Section 5.4, and then used as a synthetic environment to interact with 6 policies trained using the deep actor-critic method (Section 2.2.1) with different hyper-parameters, over the

Table 5.4: Overall battery runtime of the DBS system when the RL, distilled RL or random controllers were used.

	RL	Distilled RL	Random Controller
Battery Runtime (m)	227 ± 5	220 ± 6	247 ± 4

buffer \mathcal{E}^H specific to the patient; these policies can in general lead to varying performance. Then, for each policy, the mean of total returns (5.9) over all simulated trajectories can be calculated, and was used to estimate the policy’s expected return from Definition 5. The constants in (5.8), balancing the scale of the QoCs (*i.e.*, grasp frequency, rating and tremor severity) were set to $C_2 = C_3 = C_4 = 10$ for participants 2-4 who experience bradykinesia and pronounced tremor with insufficient DBS; in contrast, the symptoms of participant 1 and 5 are considered subtle, and we set $C_2 = C_3 = C_4 = 25$ to distinguish better if sufficient DBS is provided; see Table 5.1 for details on patient characteristics as well as the dosage of PD medications.

DLSM’s performance was compared against the classic IS (Precup, 2000), as well as other state-of-the-art OPE methods including dual-DICE (Nachum et al., 2019), fitted Q-evaluation (FQE) (Le et al., 2019) and doubly robust (DR) (Nie et al., 2022). Three metrics were considered to evaluate the performance of OPE, including mean absolute error (MAE), rank correlation, and regret@1, following from (Fu, Norouzi, et al., 2020). MAE evaluates the absolute error between the total return estimated by OPE, versus the *actual* returns, *i.e.*, mean total return recorded from clinical measurements. Rank correlation quantifies the alignment between the rank of policies over OPE-estimated returns and the actual returns. Regret@1 quantifies the percentage loss, over the total actual returns, one would get by picking the policy with maximum OPE-estimated return, against the actual best-performing policy, showing if the OPE methods can identify the best-performing policy correctly. Their mathematical definitions can be found in Section 2.3. Moreover, 20% of historical trajectories from dataset were withheld and used to inspect and illustrate how the DLSM reconstructed the rewards defined following (5.6).

5.5.2.1 Results

The obtained results are summarized in Figure 5.6. As shown, the DLSTM in general achieved significantly higher ranks and lower regrets, with smaller error bars across different seeds, compared to all four baselines. Both metrics are important as they evaluate the effectiveness toward selecting top-performing policies to be deployed in future participant visits, following the schema introduced in Figure 5.3. By accurately identifying the policies that lead to relatively higher returns, the agent could quickly reach and exploit high-reward regions, which could dramatically reduce the number of visits needed for testing in long term. DLSTM also outperformed baselines in terms of the MAE metric, though with smaller margins. This is expected as a major part of the total return (5.9) is constituted by feedback provided by the participants, which may not differ by a very large scale across different control policies, even for the ones that are statistically different – this is illustrated by the last 3 columns of Figure 5.5.

5.5.2.2 Visual Interpretation of the DLSTM-reconstructed Rewards

Figure 5.7 was used to interpret how DLSTM captured and reconstructed environmental rewards as defined in (5.6) – each point in the heatmap represented a *group of* state-action pairs (*i.e.*, a single transition in an MDP trajectory) that are associated with similar averaged beta amplitude over the W -window captured in states as well as the stimulation intensity captured in actions; the specific definitions of states and actions can be found in Section 5.3.1. Then, the reconstructed rewards were averaged over each group, whose scales were distinguished by the colors shown in the color bar. In general, higher rewards were reconstructed toward lower beta amplitude and lower stimulation intensity, and vice versa, which aligned with the reward function definition (5.6).

5.5.2.3 On the Scalability of DLSTM

We additionally studied how the lengths of offline trajectories and total number of trajectories used during training affected the performance of the OPE approach introduced in Section 5.4, *i.e.*, the scalability of DLSTM. Specifically, we repeated the OPE experiment

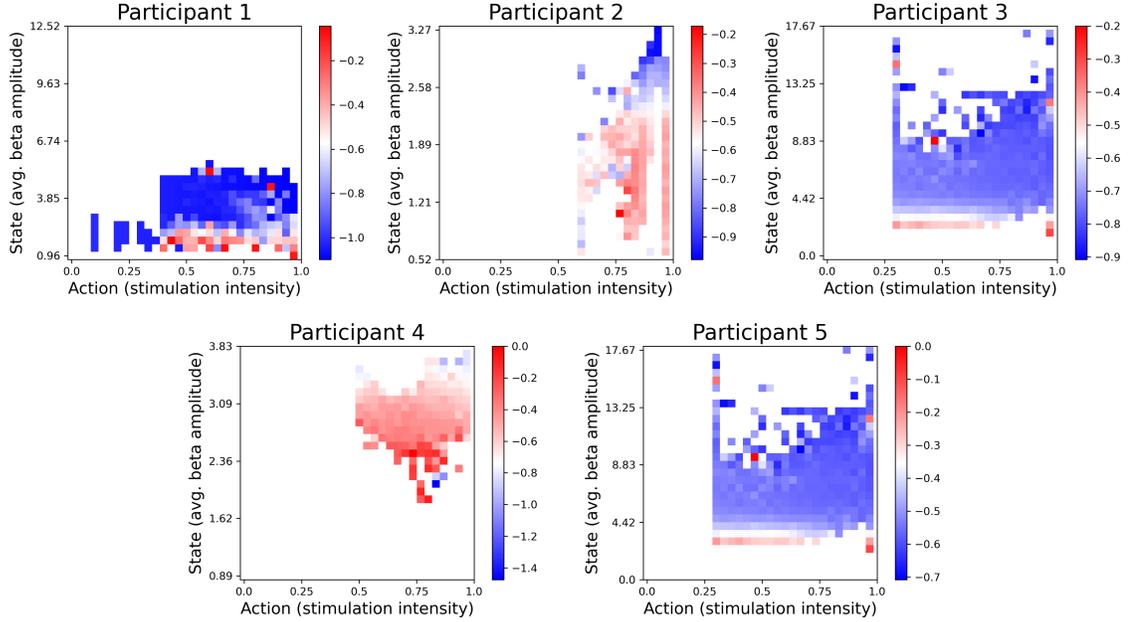


FIGURE 5.7: Inspection of DLSM-reconstructed rewards from a 20% hold-out set of historical trajectories. In general, it can be observed that higher rewards were reconstructed toward lower beta amplitude and lower stimulation intensity, and vice versa, which aligned with the reward function definition (5.6).

Table 5.5: Mean and standard error of the trajectory lengths, in terms of the number of MDP transitions, pertaining to the offline dataset, as well as the statistics of 3 sub-datasets, constituted of long, medium and short trajectories respectively, used in scalability study. The statistics for the overall cohort were reported at the last row.

	Overall	Long Traj. ($\geq 66\%$ quantile)	Medium Traj. (33%-66% quantile)	Short Traj. (<33% quantile)
Participant 1	209 \pm 15	442 \pm 11	155 \pm 5	62 \pm 5
Participant 2	227 \pm 14	402 \pm 11	144 \pm 3	70 \pm 5
Participant 3	200 \pm 11	356 \pm 11	155 \pm 4	71 \pm 4
Participant 4	232 \pm 13	426 \pm 14	164 \pm 4	96 \pm 5
Participant 5	217 \pm 11	369 \pm 13	158 \pm 3	105 \pm 6
Cohort	218 \pm 6	400 \pm 6	156 \pm 2	82 \pm 3

above over three sub-datasets, containing long, medium and short trajectories respectively. Long trajectories referred to the ones in the dataset whose lengths were longer or equal to the 66% quantile of lengths over all the lengths pertaining to the same participant. Similarly, medium and short trajectories corresponded to those with lengths between 33-66% and less

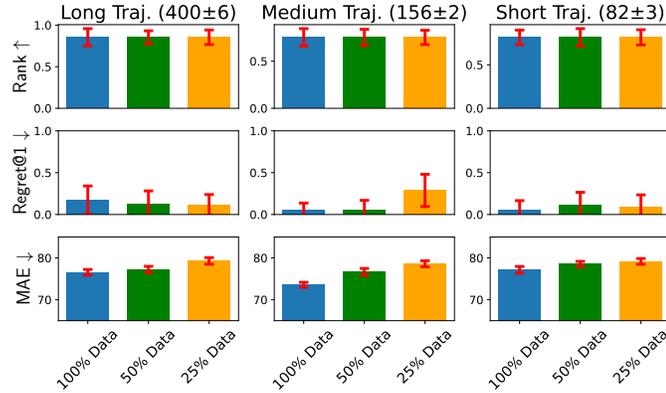


FIGURE 5.8: Results from DLSM’s Scalability study. The means and standard errors of rank correlations, regrets and MAEs were averaged over the entire cohort.

than 33-66% quantiles, respectively. The mean and standard error of the original data as well as the 3 sub-datasets were reported in Table 5.5. Furthermore, for each sub-dataset, we studied the effects of training DLSM with 100%, 50% and 25% of all trajectories provided by the sub-dataset. The results were shown in Figure 5.8, where the columns pertain to the 3 sub-datasets correspondingly, and the results from training on varied number of trajectories were reported within each sub-plot.¹ Neither rank correlations nor regrets were significantly impacted by the lengths or the number of trajectories used to train the DLSM. Even though the MAEs were mildly impacted by the number of trajectories used for training, MAEs were not significantly affected by the trajectory lengths, *i.e.*, MAEs achieved across sub-datasets (representing varied trajectory lengths) were similar to the ones pertaining to the same portion (from 100%, 50% and 25%) of data used for training, as distinguished by the bar colors in Figure 5.8. Such results further justified the effectiveness and importance of accomplishing OPE’s goal by virtue of learning a latent space that encodes environmental transitions and rewards, against existing frameworks that were mainly bounded by the high variance in their estimations (Fu, Norouzi, et al., 2020; Q. Gao, Gao, Chi, & Pajic, 2023a; Y. Liu et al., 2020).

¹ For conciseness and readability, we averaged the individual results obtained over all participants.

5.5.3 Limitations

Note that although the DLSM model has the potential to facilitate RL policy optimization by directly interacting with the agent stand-alone, we leave further investigations to this end for future work. Specifically, in this work, the controllers were supposed to be deployed to humans, and it is important that the control approaches used to interact with participants should have solid demonstrations of safety and efficacy *a priori*. Consequently, we separated the policy optimization (facilitated by the widely adopted actor-critic approach introduced in Section 2.2.1) and evaluation pipelines, where DLSM only facilitated OPE as its validity of such a framework had been shown effective in prior research over computational brain models (Q. Gao, Schmidt, et al., 2022).

Moreover, the number of participants of this research was limited due to the availability of RC+S platforms, as well as constraints imposed by our funding sources for risk and cost management purposes. This may limit the statistical power of the Wilcoxon's tests used in results analyses. The trials are deemed as "first in human" by the FDA, which is scoped to demonstrate preliminary feasibility of developing multidisciplinary techniques to improve an existing medical system that could be carried onto mass production and deployment in the near future. To this end, we were given the flexibility of adopting and modifying existing commercial platforms that can serve the purpose of the research, but the trial cohort is limited by 5 participants. Multiple customizations were made to each individual device in terms of sensing, communication, device configuration etc. to facilitate the success and safe deployment of RL controllers, as summarized in Figure 5.2, Figure 4.2, and Section 5.2. Although approved for research use by the FDA (via IDE) and IRB, such a pipeline may not be directly applicable for mass production and adoption in larger trials.

6. Data Missingness

Though recent works have developed methods that can generate estimates (or imputations) of the missing entries in a dataset to facilitate downstream analysis, most depend on assumptions that may not align with real-world applications and could suffer from poor performance in subsequent tasks such as classification. This is particularly true if the data have large missingness rates or a small sample size. More importantly, the imputation error could be propagated into the prediction step that follows, which may constrain the capabilities of the prediction model. In this chapter, we introduce the gradient importance learning (GIL) method to train multilayer perceptrons (MLPs) and long short-term memories (LSTMs) to *directly* perform inference from inputs containing missing values *without imputation*. Specifically, we employ reinforcement learning (RL) to adjust the gradients used to train these models via back-propagation. This allows the model to exploit the underlying information behind *missingness patterns*. We test the approach on real-world time-series (*i.e.*, MIMIC-III), tabular data obtained from an eye clinic, and a standard dataset (*i.e.*, MNIST), where our *imputation-free* predictions outperform the traditional *two-step* imputation-based predictions using state-of-the-art imputation methods.

6.1 Related Work

This section introduces the works related to the topic being considered, including missing data imputation and attention techniques.

6.1.1 Missing Data Imputation.

Traditional mean/median-filling and carrying-forward imputation methods are still used widely, as they are straightforward to implement and interpret (Honaker & King, 2010). Recently, there have been state-of-the-art imputation algorithms (IAs) proposed to produce smooth imputations with interpretable uncertainty estimates. Specifically, some adopt Bayesian methods where the observed data are fit to data-generating models, including Gaussian processes (Fortuin & Rätsch, 2019; Fortuin et al., 2018; Wilson et al., 2016), multivariate imputation by chained equations (MICE) (Azur et al., 2011), random

forests (Stekhoven & Bühlmann, 2012), *etc.*, or statistical optimization methods such as expectation-maximization (EM) (Bashir & Wei, 2018; García-Laencina et al., 2010). However, they often suffer from limited scalability, require assumptions over the underlying distribution, or fail to generalize well when used with datasets containing mixed types of variables, *i.e.*, when discrete and continuous values exist simultaneously (Fortuin et al., 2020; Yoon et al., 2018). There also exist matrix-completion methods that usually assume the data are static, *i.e.*, information does not change over time, and often rely on low-rank assumptions (Mazumder et al., 2010; Schnabel et al., 2016; J. Wang et al., 2006; H.-F. Yu et al., 2016). More recently, several DL-based IAs have been proposed following advancements in deep generative models such as deep latent variable models (DLVMs) and generative adversarial networks (GANs) (Fortuin et al., 2020; S. C.-X. Li et al., 2019; Ma et al., 2018; Mattei & Frelsen, 2019; Yoon et al., 2018). DLVMs are usually trained to capture the underlying distribution of the data, by maximizing an evidence lower bound (ELBO), which could introduce high variations to the output distributions (Kingma & Welling, 2013) and cause poor performance in practice, if the data have high missingness (as for the example shown in Figure 1.1). There also exist end-to-end methods that withhold observed values from inputs and impose reconstruction losses during training of prediction models (Cao et al., 2018; Ipsen et al., 2020). In addition, data-driven IAs are developed specifically toward medical time series by incorporating prior domain knowledge (Calvert et al., 2016; Q. Gao et al., 2021; Scherpf et al., 2019; X. Yang et al., 2018). On the other hand, a few recent works attempt to address missing inputs through an imputation-free manner (Morvan et al., 2020; Sportisse et al., 2020), however, they are limited to linear regression.

6.1.2 Attention.

The importance matrix used in the GIL is somewhat reminiscent of visual attention mechanisms in the context of image captioning or multi-object recognition (Ba et al., 2014; Mnih et al., 2014; K. Xu et al., 2015) as they both require training of the prediction models with RL. We briefly discuss the connections and distinctions between them, with full details

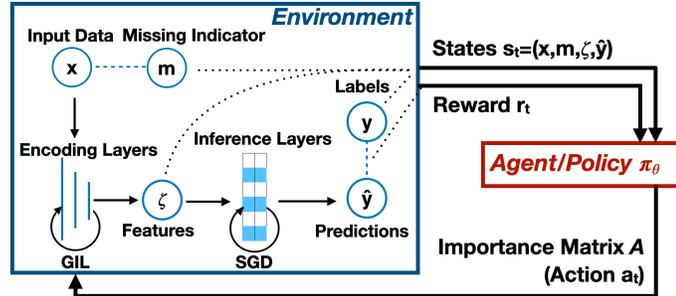


FIGURE 6.1: Overview of the GIL framework.

provided in Appendix B.3. Visual attentions are commonly used to train CNNs to focus on specific portions of the inputs that are most helpful for making predictions. They are determined by maximizing an evidence lower bound (ELBO), which is later proved to be equivalent to the REINFORCE algorithm in the RL literature (Mnih et al., 2014; Sutton & Barto, 2018). However, these methods cannot be applied directly to our problem, as they require features to be exclusively associated *spatially* with specific parts of the inputs, which is attainable by using convolutional encoders with image inputs but intractable with the type of data considered in our work. Instead, our approach overcomes this issue by directly applying importance weights, generated by RL, into the *gradient space* during back-propagation. Such issues motivate use of the term *importance* instead of *attention*. Lastly, our method does not require one to formulate the learning objective as an ELBO, and as a result GIL can adopt any state-of-the-art RL algorithm, without being limited to REINFORCE as in (Ba et al., 2014; Mnih et al., 2014; K. Xu et al., 2015). Besides, other types of attention mechanisms are also proposed toward applications in sequence modeling and natural language processing (NLP) such as (Cheng et al., 2016; Vaswani et al., 2017).

6.2 Gradient Importance Learning (GIL)

In this section, we introduce GIL; a method that facilitates training of imputation-free prediction models with incomplete data. In Section 6.2.2, we show how the gradient descent directions can be adjusted using the *importance matrix* \mathbf{A} . In Section 6.2.3, we introduce an RL framework to generate the elements of \mathbf{A} which can leverage the information underlying

missingness during training, as illustrated in Figure 6.1.

6.2.1 Problem Formulation

We consider a dataset containing N observations $\mathcal{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N)$ where each \mathbf{X}_j can be represented by a vector $\mathbf{x}_j \in \mathbb{R}^d$ (for tabular data) or a matrix $\mathbf{X}_j \in \mathbb{R}^{T_j \times d}$ (for time-series) with T_j denoting the time horizon of the sequence \mathbf{X}_j . Since the focus of this work is principally on time-series data, recurrent neural networks will receive significant attention. We also define the set of missing indicators $\mathcal{M} = (\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_N)$, where $\mathbf{M}_j = \mathbf{m}_j \in \{0, 1\}^d$ or $\mathbf{M}_j \in \{0, 1\}^{T_j \times d}$ depending on the dimension of \mathbf{X}_j , and 0's (or 1's) correspond to the entries in \mathbf{X}_j that are missing (or not missing), respectively. Finally, we assume that each \mathbf{X}_j is associated with some label $\mathbf{y}_j \in \mathcal{Y}$; thus $\mathcal{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N)$ denotes the set of labels. We define the problem as learning a model parameterized by \mathbf{W} to directly predict \mathbf{y}_j by generating $\hat{\mathbf{y}}_j$ that maximizes the log-likelihood $\sum_{j=1}^N \log p(\hat{\mathbf{y}}_j | \mathbf{X}_j, \mathbf{M}_j, \mathbf{W})$, without imputing the missing values in \mathbf{X}_j .

6.2.2 Gradient Importance

Missingness is a common issue found in tabular data and time series, where multi-layered perceptron (MLP) (Bishop, 2006) and long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) models, respectively, are often considered for predictions. Below we illustrate how the *importance matrix* can be applied to gradients, which are produced by taking regular stochastic gradient descent (SGD) steps, toward training MLP models with incomplete tabular inputs. Note that this idea can be easily extended to sequential inputs with LSTM models and the corresponding details can be found in Appendix B.1. First, the definitions of the dataset and missing indicators can be reduced to $\mathcal{X}_{tab} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ and $\mathcal{M}_{tab} = (\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_N)$, with $\mathbf{x}_j \in \mathbb{R}^d$ and $\mathbf{m}_j \in \mathbb{R}^d$, respectively. Then we define an MLP with k hidden layers, for $k \geq 2$, as follows

$$\hat{\mathbf{y}} = \phi_{out}(\mathbf{W}_{out} \phi_k(\mathbf{W}_k \dots \phi_2(\mathbf{W}_2 \phi_1(\mathbf{W}_1 \mathbf{x})))), \quad (6.1)$$

where $\mathbf{x} \in \mathbb{R}^d$ is the input, \mathbf{W}_i and ϕ_i are the weight matrix and activation functions for the i -th hidden layer, respectively, and we have omitted the bias terms for notational convenience. The first layer can be interpreted as the *encoding* layer $\mathbf{W}_{enc} = \mathbf{W}_1$ that maps inputs to features $\zeta = f_{enc}(\mathbf{x}|\mathbf{W}_{enc})$, while the rest are the *inference* layers $\mathbf{W}_{inf} = \{\mathbf{W}_2, \dots, \mathbf{W}_k, \mathbf{W}_{out}\}$ that map the features to prediction $\hat{\mathbf{y}} = f_{inf}(\zeta|\mathbf{W}_{inf})$. In the following proposition, we show that the gradients of some commonly used loss functions, $E(\hat{\mathbf{y}}, \mathbf{y})$, such as cross entropy or mean squared errors, w.r.t. \mathbf{W}_1 , can be formulated in the form of an outer product. The proof and details can be found in Appendix B.5.

Proposition 2. *Given a MLP and a smooth loss function $E(\hat{\mathbf{y}}, \mathbf{y})$, the gradients of E w.r.t. the encoding layer can be formulated as $\partial E / \partial \mathbf{W}_1 = \Delta_1 \mathbf{x}^\top$, where Δ_1 contains the gradients propagated from all the inference layers, $\mathbf{x} \in \mathcal{X}_{tab}$ and $\mathbf{y} \in \mathcal{Y}$.*

From this Proposition it can be seen that the gradients that are used to train the encoding layers, following regular SGD solvers, can be formulated as the *outer product* between the gradients Δ propagated from the inference layers and the input \mathbf{x} as

$$(\partial E / \partial \mathbf{W}_{enc})_{SGD} = \Delta \cdot \mathbf{x}^\top, \quad (6.2)$$

where $\mathbf{W}_{enc} \in \mathbb{R}^{e \times d}$, $\Delta \in \mathbb{R}^e$, $\mathbf{x} \in \mathbb{R}^d$, e is the dimension of the features, and d is the dimension of the input. To simplify notation, note that henceforth we use \mathbf{x} to refer to an individual tabular data $\mathbf{x}_j \in \mathcal{X}_{tab}$, regardless of its index.

The corresponding SGD updates for training \mathbf{W}_{enc} , using learning rate α , are $\mathbf{W}_{enc} \leftarrow \mathbf{W}_{enc} - \alpha \Delta \cdot \mathbf{x}^\top$. As a result, it is observed from (6.2) that the j -th ($j \in [1, d]$) column of $(\partial E / \partial \mathbf{W}_{enc})_{SGD}$ is *weighted* by the j -th element in \mathbf{x} given Δ . However, given that some entries in \mathbf{x} could be *missing* (according to the missing indicator \mathbf{m}) and their values are usually replaced by a placeholder $\zeta \in \mathbb{R}$, it may not be ideal to directly back-propagate the gradients in the form of (6.2), as the features ζ captured by the encoding layers may not be expressive enough for the inference layers to make accurate predictions. Instead, we consider introducing the *importance matrix* $\mathbf{A} \in [0, 1]^{e \times d}$ to adjust the gradient descent

direction as

$$\mathbf{W}_{enc} \leftarrow \mathbf{W}_{enc} - \alpha(\partial E / \partial \mathbf{W}_{enc})_{SGD} \odot \mathbf{A}, \quad (6.3)$$

which not only trains the model to capture information from the observed inputs that are most useful for making accurate predictions, but also to leverage the information underlying the missingness in the data. The elements of \mathbf{A} can be generated using the RL framework introduced in the next section.

Note that above we have omitted all the bias terms to simplify our presentation and note that: *i*) gradients w.r.t. to biases do not conform to the outer product format, and *ii*) more importantly, these gradients do not depend on the inputs – thus in practice, the importance matrix \mathbf{A} is only applied to the gradients of the weights. Moreover, we do not consider convolutional neural network (CNN) models (LeCun et al., 1999) in this work because of *i*). Though the proposed framework could still be applied to CNNs, it may not be as efficient as for MLPs or LSTMs, where the search space for \mathbf{A} is significantly reduced by taking advantages of the outer product format, as shown in (6.4) below. Importantly, our focus is to address the missingness in tabular data and time-series, in which case MLPs and LSTMs are appropriate, while the missingness in images is usually related to applications in other domains such as compressed sensing (Y. Wu et al., 2019) or image super-resolution (Dong et al., 2015), which we plan to explore in the future.

6.2.3 RL to Generate Importance Matrix

Now we show how to use RL to generate the importance matrix \mathbf{A} . In general, RL aims to learn an optimal policy that maximizes expected rewards, by interacting with an unknown environment (Sutton & Barto, 2018). Formally, the RL environment is characterized by a Markov decision process (MDP), with details introduced below. Each time after the agent chooses an action $a = \pi(s)$ at state s following some policy π , the environment responds with the next state s' , along with an immediate reward $r = R(s, a)$ given a reward function $R(\cdot, \cdot)$. The goal is to learn an optimal policy that maximizes the *expected total reward* defined as $J(\pi) = \mathbb{E}_{\rho_\pi}[\sum_{i=0}^T \gamma^i r(s_i, a_i)]$, where T is the horizon of the

MDP, $\gamma \in [0, 1)$ is the discounting factor, and $\rho_\pi = \{(s_0, a_0), (s_1, a_1), \dots | a_i = \pi(s_i)\}$ is the sequence of states and actions drawn from the trajectory distribution determined by π .

In our case, the elements of \mathbf{A} are determined on-the-fly following an RL policy π_θ , parameterized by θ , conditioned on some states that characterize the status of the back-propagation at the current time step. The policy π_θ is updated, concurrently with the weights \mathbf{W}_{enc} , by an RL agent that interacts with the back-propagation process modeled as the MDP defined as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma)$, with each of its elements introduced below.

6.2.4 State Space \mathcal{S} .

The state space characterizes the agent’s knowledge of the environment at each time step. To constitute the states we consider the 4-tuple $\mathbf{s} = (\mathbf{x}, \mathbf{m}, \zeta, \hat{\mathbf{y}})$ including current training input $\mathbf{x} \in \mathbb{R}^d$, the missing indicator $\mathbf{m} \in \mathbb{R}^d$, the feature ζ , *i.e.*, the outputs of the encoding layer of MLPs or the hidden states \mathbf{h}_t of LSTMs, and the predictions $\hat{\mathbf{y}}$ produced by the inference layers.

6.2.5 Action Space \mathcal{A} .

We consider combining the *importance matrix* $\mathbf{A} \in [0, 1]^{e \times d}$ with the parameter gradients $(\partial E / \partial \mathbf{W}_{enc})_{SGD}$ when updating \mathbf{W}_{enc} following

$$\mathbf{W}'_{enc} \leftarrow \mathbf{W}_{enc} - \alpha (\partial E / \partial \mathbf{W}_{enc})_{SGD} \odot \mathbf{A} = \mathbf{W}_{enc} - \alpha \Delta \cdot (\mathbf{x}^\top \odot \mathbf{a}^\top); \quad (6.4)$$

where here the equality follows from (6.2) and Proposition 2, such that all rows of \mathbf{A} can be set to be equal to the *importance* $\mathbf{a}^\top \in [0, 1]^d$, which is obtained from the policy following $\mathbf{a} = \pi_\theta(\mathbf{s})$.

6.2.6 Transitions $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$.

The transition dynamics determines how to transit from a current state \mathbf{s} to the next state \mathbf{s}' given an action \mathbf{a} in the environment. In our case, the pair (\mathbf{x}, \mathbf{m}) is sampled from the training dataset at each step, so the transitions $\mathbf{x} \rightarrow \mathbf{x}'$ and $\mathbf{m} \rightarrow \mathbf{m}'$ are determined by how training samples are selected from the training batch during back-propagation. The update of $\mathbf{W}_{enc} \rightarrow \mathbf{W}'_{enc}$ is conditioned on the importance \mathbf{a} as captured in (6.4), which results in

the transitions $\zeta = f_{enc}(\mathbf{x}|\mathbf{W}_{enc}) \rightarrow \zeta' = f_{enc}(\mathbf{x}'|\mathbf{W}'_{enc})$. The update of $\mathbf{W}_{inf} \rightarrow \mathbf{W}'_{inf}$ follows from the regular SGD updates. Then, the transition $\hat{\mathbf{y}} = f(\mathbf{x}|\mathbf{W}) \rightarrow \hat{\mathbf{y}}' = f(\mathbf{x}'|\mathbf{W}')$ follows from the other transitions defined above. Finally, we can define the transition between states as $\mathbf{s} = (\mathbf{x}, \mathbf{m}, \zeta, \hat{\mathbf{y}}) \rightarrow \mathbf{s}' = (\mathbf{x}', \mathbf{m}', \zeta', \hat{\mathbf{y}}')$.

6.2.7 Reward Function R .

After the RL agent takes an action \mathbf{a} at the state \mathbf{s} , an immediate reward $r = R(\mathbf{s}, \mathbf{a})$ is returned by the environment which provides essential information to update π_θ (Lillicrap et al., 2016; Silver et al., 2014). We define the reward function as $R(\mathbf{s}, \mathbf{a}) = -E(\hat{\mathbf{y}}, \mathbf{y})$.

We utilize actor-critic RL (Lillicrap et al., 2016; Silver et al., 2014) to update π_θ , which outputs the importance \mathbf{a} that is used to concurrently update \mathbf{W}_{enc} . Specifically, we train the target policy (or actor) π_θ along with the critic $Q_\nu : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, parameterized by ν , by maximizing $J_\beta(\pi_\theta) = \mathbb{E}_{s \sim \rho_\beta} [Q_\nu(s, \pi_\theta(s))]$ which gives an approximation to the expected total reward $J(\pi)$. Specifically, the trajectories $\rho_\beta = \{(s_0, a_0), (s_1, a_1), \dots | a_i = \beta(s_i)\}$ collected under the behavioral policy $\beta : \mathcal{S} \rightarrow \mathcal{A}$ are used to update θ and ν jointly following

$$\nu' \leftarrow \nu + \alpha_\nu \delta \nabla_\nu Q_\nu(\mathbf{s}, \mathbf{a}), \quad \theta' \leftarrow \theta + \alpha_\theta \nabla_\theta \pi_\theta(\mathbf{s}) \nabla_{\mathbf{a}} Q_\nu(\mathbf{s}, \mathbf{a})|_{\mathbf{a}=\pi_\theta(\mathbf{s})}; \quad (6.5)$$

where β is usually obtained by adding noise to the output of π_θ to ensure sufficient exploration of the state and action space, $\delta = r + \gamma Q_\nu(\mathbf{s}', \pi_\theta(\mathbf{s}')) - Q_\nu(\mathbf{s}, \mathbf{a})$ is the temporal difference error (residual) in RL, and $\alpha_\theta, \alpha_\nu$ are the learning rates for θ, ν , respectively.

We summarize the GIL approach in Algorithm 4 and the detailed descriptions can be found in Appendix B.2. Note that the missing indicator $\mathbf{m} \in \mathbb{R}^d$ would be a good heuristic to replace the importance \mathbf{a} as it could prevent the gradients produced by the missing dimensions from propagation. However, it does not train the model to capture the hidden information underlying the missing data, which results in its performance being dominated by GIL as demonstrated in Section 6.3.

Algorithm 4 Gradient Importance Learning (GIL).

Require: $\mathcal{X}, \mathcal{Y}, \mathcal{M}, \mathbf{W}_{enc}, \mathbf{W}_{inf}, \pi_\theta, Q_v, \alpha_\theta, \alpha_v, \alpha, E$ **Ensure:**

- 1: Initialize \mathbf{W}_{enc} and \mathbf{W}_{inf} , actor π_θ and critic Q_v
 - 2: Sample \mathbf{x} from \mathcal{X} and obtain the corresponding label \mathbf{y} from \mathcal{Y}
 - 3: Obtain the feature $\zeta \leftarrow f_{enc}(\mathbf{x}|\mathbf{W}_{enc})$ and prediction $\hat{\mathbf{y}} = f_{inf}(\zeta|\mathbf{W}_{inf})$ from the encoding and inference layers, respectively
 - 4: $\mathbf{s} \leftarrow (\mathbf{x}, \mathbf{m}, \zeta, \hat{\mathbf{y}})$
 - 5: **for** $iter$ in $1 : max_iter$ **do**
 - 6: Obtain importance from a behavioral policy $\mathbf{a} = \beta(\mathbf{s}|\pi_\theta)$
 - 7: Train the encoding layer following $\mathbf{W}'_{enc} \leftarrow \mathbf{W}_{enc} - \alpha \Delta \cdot (\mathbf{x}^\top \odot \mathbf{a}^\top)$ as in (6.4)
 - 8: Train the inference layers following regular gradient descent, *i.e.*,
 $\mathbf{W}'_{inf} \leftarrow \mathbf{W}_{inf} - \alpha (\partial E / \partial \mathbf{W}_{inf})_{SGD}$
 - 9: Obtain the prediction following the updated weights $\hat{\mathbf{y}} \leftarrow f(\mathbf{x}|\mathbf{W}'_{enc}, \mathbf{W}'_{inf})$
 - 10: Obtain the reward $r \leftarrow R(\mathbf{s}, \mathbf{a})$
 - 11: Get a new sample \mathbf{x}' from \mathcal{X} and obtain the corresponding label \mathbf{y}' from \mathcal{Y}
 - 12: Obtain the feature $\zeta' \leftarrow f_{enc}(\mathbf{x}'|\mathbf{W}'_{enc})$ and prediction $\hat{\mathbf{y}}' = f_{inf}(\zeta'|\mathbf{W}'_{inf})$ from the encoding and inference layers, respectively
 - 13: $\mathbf{s}' \leftarrow (\mathbf{x}', \mathbf{m}', \zeta', \hat{\mathbf{y}}')$
 - 14: Update the actor π_θ and critic Q_v using $(\mathbf{s}, \mathbf{a}, r, \mathbf{s}')$ following (6.5)
 - 15: $\mathbf{s} \leftarrow \mathbf{s}', \mathbf{W}_{enc} \leftarrow \mathbf{W}'_{enc}, \mathbf{W}_{inf} \leftarrow \mathbf{W}'_{inf}$
 - 16: **end for**
-

6.2.8 Extensions of the Framework

The proposed GIL framework uses RL agents to guide the model toward minimizing its objective $E(\hat{\mathbf{y}}, \mathbf{y})$, which is usually captured by general prediction losses such as cross entropy or mean squared error. Below we use an example to show how our method can be extended to adapt ideas from related domains, which can augment training of the imputation-free prediction models by altering the reward function. Recent works in contrastive learning, *e.g.*, (T. Chen et al., 2020), can train DL models to generate highly expressive features by penalizing (or promoting) the distributional difference $D(\zeta^+, \zeta^-)$ among the features associated with inputs that are similar to (or different from) each other, where ζ^+ denotes the set of features corresponding to one set of homogeneous inputs \mathcal{X}_{con} , and ζ^- are the features generated by data that are significantly different from the ones in \mathcal{X}_{con} . However, such methods may not be directly applied to the missing data setting considered in this work, as their loss D is usually designed toward unsupervised training and

might need to be carefully re-worked in our case. This idea can be adapted to our framework by defining ζ^+ as the features mapped from the inputs (by the encoding layers \mathbf{W}_{enc}) associated with the same label $y \in \mathcal{Y}$ and ζ^- the features corresponding to some other label $y' \in \mathcal{Y}$. Then we can define the new reward function $R(\mathbf{s}, \mathbf{a}) = -E(\hat{\mathbf{y}}, \mathbf{y}) + c \cdot D(\zeta^+, \zeta^-)$, $c > 0$, which does not require D to be carefully crafted, provided that \mathbf{W}_{enc} is not trained by directly propagating gradients from it. In Section 6.3 it will be shown in case studies that by simply defining D as the mean squared error between ζ^+ and ζ^- can improve the prediction performance.

6.3 Experiments

We evaluate the performance of the *imputation-free* prediction models trained by GIL against the existing *imputation-prediction* paradigm on both benchmark and real-world datasets, where the imputation stage employs both state-of-the-art and classic imputation algorithms (IAs), with the details introduced in Section 6.3.1. We also consider variations of the GIL method proposed in Section 6.2 to illustrate its robustness and flexibility. The datasets we use include *i*) MIMIC-III (Johnson et al., 2016) that consists of real-world EHRs obtained from intensive care units (ICUs), *ii*) a de-identified ophthalmic patient dataset obtained from an eye center in North America, and *iii*) hand-written digits MNIST (LeCun & Cortes, 2010). We also tested on a smaller scaled ICU time-series from 2012 Physionet challenge (Silva et al., 2012) and these results can be found in Appendix B.4.4. Some of the data are augmented with additional missingness to ensure sufficient missing rates and the datasets we use cover all types of missing data, *i.e.*, missing complete at random (MCAR), MAR and MNAR (R. J. Little & Rubin, 2019). We found that the proposed method not only outperforms the baselines on all three datasets under various experimental settings, but also offers better feature representations as shown in Figure 6.2. We start with an overview of the baseline methods, in the following sub-section, and then proceed to present our experimental findings.

6.3.1 Variants of GIL and Baselines

Our method (GIL) trains MLPs or LSTMs (depending on the type of data) to directly output the predictions given incomplete inputs without imputation. Following from Section 6.2.8, we also test on binary classification tasks using GIL-D which includes the distributional difference term $D(\zeta^+, \zeta^-)$, captured by mean squared errors, into the reward function. For baselines, we start with another variant of GIL that uses a simple heuristic – the missing indicator \mathbf{m} – to replace the importance \mathbf{a} in GIL; we denote it as GIL-H. This helps analyze the advantages of training the models using the importance obtained from the RL policy learned by GIL, *versus* a heuristic that simply discards the gradients produced by the subset of input entries that are missing.

For other baselines, following the imputation-prediction framework, the imputation stage employs state-of-the-art IAs including MIWAE (Mattei & Frelsen, 2019) which imputes the missing data by training variational auto-encoders (VAEs) to maximize an ELBO, GP-VAE (Fortuin et al., 2020) that couples VAEs with Gaussian processes (GPs) to consider the temporal correlation in time-series, and the generative adversarial network-based method GAIN (Yoon et al., 2018). Further, some classical imputation methods are also considered, including MICE, missForest (MF), k -nearest neighbor (kNN), expectation-maximization (EM), mean-imputation (Mean), zero-imputation (Zero) and carrying-forward (CF). The imputed data from these baselines are fed into the prediction models that share the same architecture as the ones trained by GIL, for fair comparison. For time-series, we also compare to BRITS (Cao et al., 2018) which trains bidirectional LSTMs end-to-end by masking out additional observed data and jointly imposing imputation and classification objectives.

6.3.2 MIMIC-III

We consider the early prediction of septic shock using the MIMIC-III dataset, following the frameworks provided in recent data-driven works for data pre-processing (Fleuren et al., 2020; Khoshnevisan et al., 2020; Sheetrit et al., 2017). Specifically, we use 14 commonly-

Table 6.1: Accuracy and AUC obtained from the MIMIC-III dataset.

	GIL	-D	-H	GAIN	MIWAE	GP-VAE	BRITS	MICE	Mean	CF	kNN	MF	EM	
Var-l.	Acc.	93.32	93.09	89.17	90.32	88.71	-	-	92.17	88.02	87.32	84.79	75.81	68.20
	AUC	96.10	96.79	92.96	95.57	94.28	-	-	95.97	92.56	91.78	91.86	81.73	75.23
Fix-l.	Acc.	91.47	91.01	88.25	88.48	86.18	76.50	80.24	90.09	86.41	86.87	85.48	78.11	70.51
	AUC	95.29	95.57	92.99	91.94	93.10	81.47	92.13	94.02	91.69	91.98	92.38	84.54	79.97

utilized vital signs and lab results over a 2-hour observation window to predict whether a patient will develop septic shock in the next 4-hour window. The resulting dataset used for training and testing contains 2,166 sequences each of which has length between 3 and 125, and the overall missing rate is 71.63%; the missing data in this dataset can be considered as a mixture of MCAR, MAR and MNAR. More details can be found in Appendix B.4.1.

To evaluate performance, we consider the prediction model constituted by a 1024-unit LSTM layer for encoding followed by a dense output layer for inference. Considering that some of the baseline methods are not designed to capture the temporal correlations within the sequences which do not follow a fixed horizon, besides testing with the varied-length sequences (Var-l.) obtained in above, we also test with a fixed-length version (Fix-l.) where the maximum time step for each sequence is set to be 8 (see Appendix B.4.1 for details). The accuracy¹ and AUC for the two tests are summarized in Table 6.1. When varied-length sequences are considered, GIL(-D) slightly outperforms GAIN and MICE while it significantly dominates the other baseline methods. However, when fixed-length sequences are considered, GAIN and MICE’s performance decreases dramatically and are significantly dominated by that of GIL(-D). This could be caused by omitting the records beyond the 8-th time step as both models are flexible enough to capture such information from that point on-wards, while in contrast, the performance increases for kNN, MF and EM which are in general based on much simpler models. On the other hand, the models trained by GIL(-D) was not significantly affected by the information loss. In fact, it emphasizes that applying importance to the gradients during training can enable the model to capture the information behind missing data. Moreover, the dramatically increased performance from

¹ All accuracy values in this work are obtained using a decision threshold of 0.5.

Table 6.2: Average Accuracy and AUC obtained from the Ophthalmic dataset over 3 different random masks. Subscripts are standard deviations.

	M.R.	GIL	-D	-H	GAIN	GP-VAE	MIWAE	MICE	Zero
Acc.	25%	86.84 _{1.43}	87.13 _{1.09}	83.63 _{0.83}	85.38 _{1.09}	85.67 _{0.83}	80.99 _{1.8}	84.21 _{0.72}	84.50 _{1.8}
AUC		92.40 _{1.33}	92.42 _{1.44}	88.87 _{2.16}	90.13 _{2.09}	91.47 _{1.02}	84.04 _{1.47}	91.35 _{1.35}	90.59 _{0.32}
Acc.	35%	83.33 _{0.72}	85.09 _{2.58}	80.41 _{1.49}	80.41 _{0.83}	80.41 _{0.83}	79.24 _{4.38}	80.41 _{1.49}	80.12 _{1.09}
AUC		88.49 _{0.68}	90.68 _{2.36}	87.02 _{3.03}	88.02 _{2.15}	84.85 _{3.29}	85.51 _{3.47}	85.87 _{3.24}	88.02 _{0.97}

GIL-H to GIL(-D) in both tests underscores the significance of training the models using the importance determined by the RL policy learned by GIL(-D), instead of a pre-defined heuristic. Note that according to a recent survey of septic shock predictions (Fleuren et al., 2020), the overall maximum AUC attained by existing data-driven methods, which uses domain knowledge to design models specifically for ICU time-series, is 0.96 and it is comparable to that attained by our method. Finally, GP-VAE and BRITS require the input sequences to have the same horizon, so we only report for the Fix-l. case where they under-perform. These are possibly due to the high variation in GPVAE’s uncertainty estimation component, and the mechanism of imposing additional missingness in BRITS².

6.3.3 Ophthalmic Data

We consider identifying diabetic retinopathy (DR), an eye disease caused by diabetes, using the de-identified data obtained from Duke Eye Center, constituted by a mixture of image feature vectors and tabular information. Specifically, the data collected from each subject is formulated as a vector with 4,101 dimensions containing two 2,048-length feature vectors from two retinal imaging modalities, optic coherence tomography (OCT) and color fundus photography (CFP), followed by a 5-dimensional data characterizing demographic (*e.g.*, age) and diabetic information (*e.g.*, years of diabetes). Additional details of the dataset and examples of the OCT and CFP images are provided in the Appendix B.4.2. At most one of the two types of retinal images could be missing due to MAR as sometimes the diagnosis can be performed with a single modality (Cui et al., 2021), while the demograph-

² The hyper-parameters used to train their models can be found in Appendix B.4.1

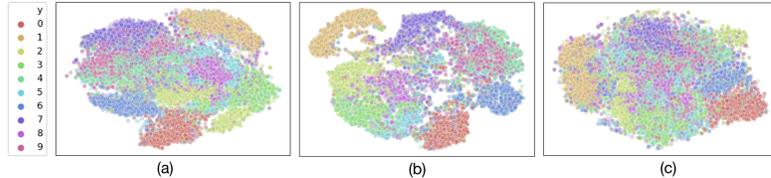


FIGURE 6.2: t -SNE visualization of the feature space learned by (a) GIL, (b) MIWAE and (c) GAIN on the MNIST dataset with 90% missing rate.

ic/diabetic information can be missing due to MCAR. A total of 1,148 subjects are included in this dataset with a 17% missing rate (M.R.)³. We apply additional random masks over both the input image features (by assuming one or two entire image feature vectors are missing) and demographic/diabetic information (by assuming some entries are missing) to increase the missing rate to 25% and 35%, where for each case 3 different random masks are applied.

For this experiment we consider an MLP with 2 hidden layers with 1,000 nodes each for prediction. The results are shown in Table 6.2 where the mean and standard deviations (in subscripts) of accuracy and AUC are reported. Although GIL and GP-VAE result in similar performance in the 25% missing rate case, GIL still achieves higher accuracy and AUC; these are significantly improved, with significantly lower standard deviation over GP-VAE, when the missing rate increases to 35%. In general, GIL-D outperforms all the other methods, with an exception of the higher standard deviation of accuracy in the 35% M.R. case, which benefits from the flexibility of the framework we proposed. With increased missingness, the performance of most baselines drops significantly and are close to the zero-imputation baseline as the image feature vectors occupy most dimensions of the inputs while imputing with zeros would be a reasonable heuristic.

6.3.4 MNIST

This work focuses mostly on tabular inputs or time-series; however, we test on MNIST images given its simple structure, which could be classified using MLPs and more importantly, because results from these data are easier to interpret. Specifically, we test on the

³ When calculating missing rates, each feature vector is only counted as a single entry regardless of size.

Table 6.3: Average Accuracy reported for the MNIST dataset over different missing rates. For each missing rate, 5 random masks were used resulting in standard deviations shown as subscripts ($\times 10^{-3}$).

	M.R.	GIL	-H	GAIN	MIWAE	MICE	Zero	GP-VAE
MCAR	50%	96.29 _{0.7}	96.08 _{1.2}	96.23 _{0.7}	95.46 _{0.8}	94.58 _{0.2}	95.83 _{1.7}	96.57 _{3.1}
	70%	93.35 _{0.9}	93.26 _{0.9}	91.98 _{3.8}	93.20 _{0.8}	90.53 ₃	92.80 _{0.8}	93.49 ₃
	90%	78.47 _{3.9}	78.44 _{4.7}	72.58 ₉	77.91 _{11.3}	73.48 ₅	76.67 _{5.3}	76.25 ₂
MAR	-	93.23 _{0.3}	93.15 _{0.5}	92.97 _{0.8}	93.18 _{0.4}	92.62 _{1.1}	82.50 _{2.9}	92.62 _{1.3}

MCAR version of MNIST where a pre-determined portion of pixels (*i.e.*, out of 50%, 70% and 90%) are masked off uniformly at random from the original images, and the MAR version, which assumes part of the image is always observable and the positions of missing pixels are determined by a distribution conditioned on features extracted from the observable portion following (Mattei & Frellsen, 2019). For each test, the masks are applied with 5 different random seeds, resulting in the standard deviations reported in Table 6.3 which also summarizes obtained results.

We consider using MLPs constituted by two hidden dense layers with 500 nodes each as the prediction models⁴. Note that in the MCAR setting, GP-VAE achieves slightly higher average accuracy than our method when 50% and 70% of the pixels missing, because it relies on convolutional encoding layers, which intrinsically have advantages on this dataset. However, GP-VAE’s performance is associated with higher standard deviation in both cases and is significantly outperformed by our method when 90% pixels are missing. The models trained by GIL also significantly outperforms the baselines following the imputation-prediction framework. This suggests that most of the IAs fail to reconstruct the input data when the missing rate is high (see Figure 1.1). Moreover, the errors produced during the imputation are then propagated into the prediction step which results in the inferior performance. In the MAR setting, the task becomes more challenging as entire rows of pixels could be missing. Finally, the model trained by GIL outperforms most of the

⁴ The performance of MIWAE reported here is different from (Mattei & Frellsen, 2019) because they used a CNN classifier.

Table 6.4: Correlation between imputation MSE and prediction accuracy for different MRs.

	MNIST			Ophthalmic		MIMIC
	50% M.R.	70% M.R.	90% M.R.	25% M.R.	35% M.R.	Ground-truths Unknown
Pearson’s Coeff.	-0.48	-0.88	-0.86	-0.12	-0.54	-
p -value	0.018	<0.01	<0.01	0.581	<0.01	-

baselines depending on IAs with an exception of MIWAE, which only slightly outperforms it. Note that GIL-H’s performance is close to GIL in most of the settings due to the simple structure of the MNIST digits where the missing indicator \mathbf{m} could be a good estimation of the importance \mathbf{a} .

Feature Space Visualization In Figure 6.2, we use t -distributed stochastic neighbor embedding (t -SNE) to visualize the features learned by the prediction model trained by GIL compared to MIWAE and GAIN. We observe that GIL results in more expressive features than the others as it generates clusters with clearer boundaries.

6.3.5 Correlation between Imputation and Prediction Performance

In this section we study if imputation error is correlated with the downstream prediction performance, under the imputation-prediction framework. Each column of Table 6.4 is obtained by calculating the Pearson’s correlation coefficient (Freedman et al., 2007) and its p -value, between the imputation mean squared error (MSE) and prediction accuracy, across different imputation methods under the same dataset. It can be observed that the imputation MSE is negatively correlated with prediction performance for most of the datasets and data with higher M.R. tends to have higher degree of negative correlation, which indicates that imputation error could be propagated to downstream tasks.

7. A Real-World Case Study – Automated Identification of Referable Retinal Pathology in Teleophthalmology Setting

This study aims to meet a growing need for a fully automated, learning-based interpretation tool for retinal images obtained remotely (e.g., teleophthalmology) through different imaging modalities that may include imperfect (uninterpretable) images.

7.1 Related Work

Significant work related to this topic was done by Vaghefi et al., 2020; W. Wang et al., 2019a; Z. Xu et al., 2020; Yoo et al., 2019. Specifically, in Singh and Gorantla, 2020 the pre-trained VGG-1942 was used to convert input OCT and CFP images into feature vectors, which were then classified as AMD and non-AMD by random forests. In this chapter, the pre-trained CNN was applied for feature extraction without fine-tuning and potentially could have led to unsatisfactory performance (Raghu et al., 2019). Precisely, most of the pre-trained models were trained with standard datasets, e.g., ImageNet (Deng et al., 2009), that do not contain ophthalmic images and the resulting models were potentially not optimized for analysis of OCT or CFP inputs. The other mentioned methods proposed CNN models for the multi-modal identification of retinal diseases. W. Wang et al., 2019a; Z. Xu et al., 2020 developed two-stream CNNs to jointly analyze the OCT and CFP images. First, each modality was processed by the corresponding stream through convolutional filters and pooling layers for feature extraction using ResNet-1845 or ResNet-5045 architectures. Then, the two streams' output features were concatenated and fed into a fully connected layer for classification. Slightly different CNN architecture was applied in Vaghefi et al., 2020. Each single-modal stream consisted of a few customized convolutional layers for initial processing, together with the outputs across streams that were combined through max-pooling followed by Inception-ResNet-V246 for further processing and classification. Despite being ground-breaking, these methods neither evaluate nor handle uninterpretable images, making them unsuitable for remote retinal image assessments where uninterpretable and low-quality images regularly occur.

7.2 Materials and Methods

In this section, we introduce the process of data collection and dataset formulation, as well as the model being considered.

7.2.1 Retinal Imaging

This retrospective study analyzed 1148 OCT and CFP retinal images obtained from 647 diabetic patients. Images were captured by Topcon, Maestro 3D-OCT multi-modality OCT/Fundus imaging device (Topcon Inc., Tokyo, Japan). CFP had an angle of $45^{\circ} \pm 5\%$, or 30° , on the non-dilated pupil. B scan horizontal range was $3\text{-}12\text{mm} \pm 5\%$, with a 4x “Moving Average” oversampling performed, with the averaged final image. All eligible patients were invited to participate in the study and verbally consented to participate in the study by their primary care provider. The images were taken by trained CMAs (certified medical assistants). The study was a part of the Duke Quality Assessment/Quality Improvement (QA/QI) project and received institutional review board approval from Duke University Health System. The study complied with the principles of the Declaration of Helsinki.

7.2.2 Dataset Formulation

Retinal images (OCT and CFP) were saved in JPEG compression format with a size of 659×512 and 661×653 pixels. For each OCT volume scan, only the central scan (i.e., the 31st B-Scan out of a total of 60 B-scans in each volume scan) through the fovea was used. The images were resized to 299×299 to comply with the input dimension of the developed CNN architecture (for more details, see sub-section: CNN Design). Images were graded as previously described (Hadziahmetovic et al., 2019) by Duke medical retina fellows and a medical retina faculty, and the final grading of de-identified images was done by consensus. The images were classified as follows: (a) uninterpretable images (if no clear identification of macula was available due to poor positioning or inferior exposure owing to media opacity; containing 2 OCT images and 71 CFP images), (b) retinal pathology

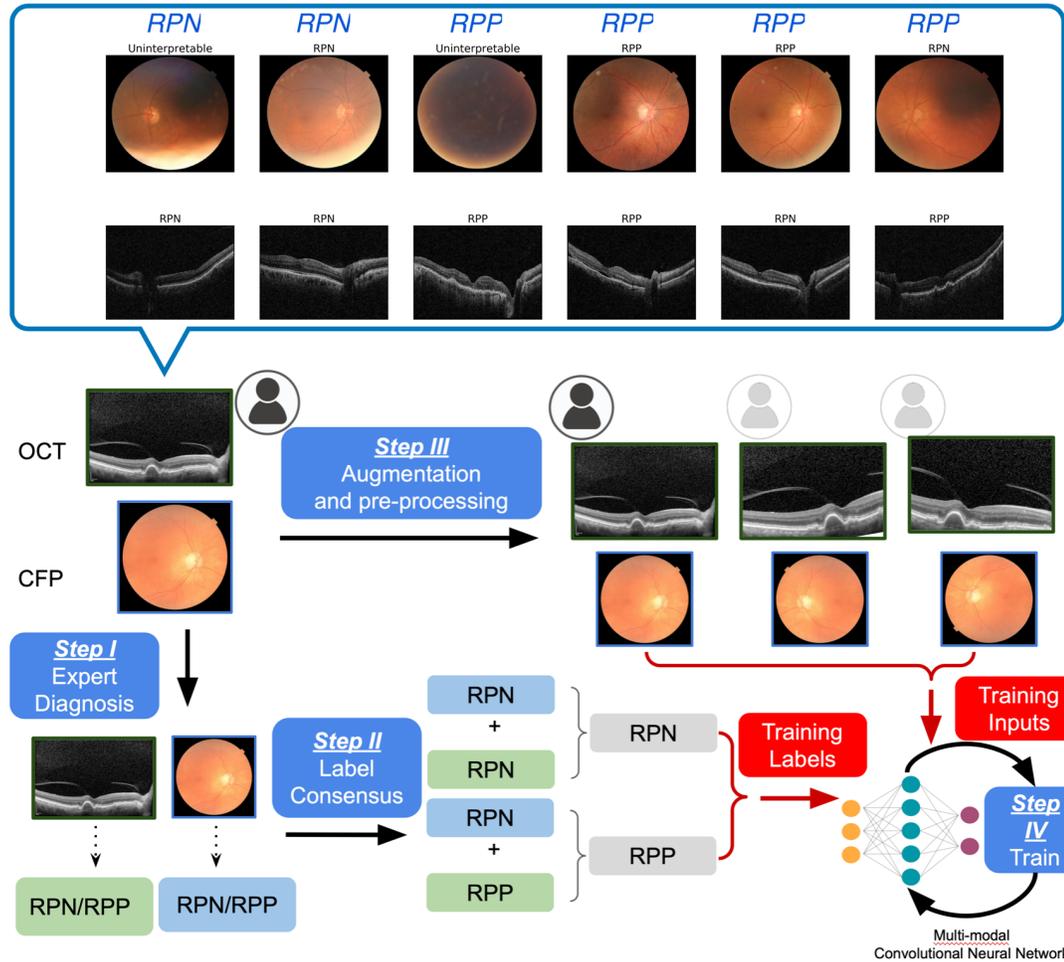


FIGURE 7.1: The OCT and CFP images obtained from the automated screening system were first labelled respectively by experts (Step I), and the individual diagnoses were used to generate training labels according to the Label Consensus Mechanism (Step II). The two types of images were augmented and pre-processed to constitute the inputs to the CNN (Step III), before being used, along with the obtained labels, for the CNN training (Step IV).

negative (RPN; containing 982 OCT and 952 CFP images), and (c) retinal pathology positive (RPP; containing 164 OCT and 125 CFP images, see Tables 7.1 and 7.2). For each patient, there was at least one interpretable image out of all obtained images. The final diagnosis used to train the CNN model was generated using the label consensus mechanism (LCM) presented in Appendix C.1 and Table C.1. As a result, 924 eyes were labeled as normal (i.e., RPN), while 224 eyes were identified as RPP (Tables 7.1 and 7.2 and

Table 7.1: Distribution of the original dataset, augmented training set, and testing dataset by modality

	OCT			CFP		
	RPN	RPP	Uninterpretable	RPN	RPP	Uninterpretable
Original	982	164	2	952	125	71
Augmented Training Set	3189	1736	14	2839	1302	798
Testing Set	73	40	1	68	32	14

Table 7.2: Distribution of the original dataset, augmented training Set, and testing dataset by eye

	RPN	RPP	Total
Original	924	224	1148
Augmented Training Set	2601	2338	4939
Testing Set	57	57	114

Figure 7.1). To form the testing dataset, we randomly selected 57 RPN and 57 RPP eyes from the available data following a uniform distribution. These numbers represented about 10% of the total eyes, roughly 6% and 25% of RPN and RPP eye cohorts, respectively. Uninterpretable images were present in 15 eyes (1 OCT and 14 CFP). The remaining images were used to form the training dataset. We specifically chose this ratio of RPN and RPP samples to assure a well-balanced testing dataset and guarantee that the resulting dataset contained sufficient samples from the minority class (i.e., RPP).

7.2.3 Study Design and Outcomes Measures

We propose a fully automated system that utilizes a multi-modal CNN to identify referable retinal pathology. Additionally, we propose a backpropagation algorithm associated with the CNN model that can train it to minimize the impact of the input images that do not contain sufficient biomarkers to determine diagnoses.

7.2.3.1 Problem Formulation

Pairs of OCT and CFP scans (O_k, C_k) were obtained from each eye of each patient P_k , with some of them being uninterpretable. We designed a CNN model that takes input as (O_k, C_k) and classifies it as “without” (i.e., RPN) and “with” (i.e., RPP) retinal pathology. Precisely, “without” pathology corresponds to the cases with normal OCT and CFP, and “with” retinal pathology refers to cases where retinal pathology can be identified in at least one of the imaging modalities (i.e., OCT or CFP). Moreover, if either O_k or C_k were uninterpretable, the outcome was derived from the interpretable image. Finally, if both O_k and C_k were uninterpretable, we specifically assigned the label as retina pathology potentially present (RPPP); those samples potentially could be selected and removed from the dataset using a separate classification model, as the clinicians would need to perform a further assessment, and potentially re-do the imaging. (A detailed introduction of this labelling mechanism for paired OCT/CFP inputs is in Appendix C.1 and Table C.1.)

7.2.3.2 CNN Design

Design of CNN model was performed in three phases: (1) Expert diagnosis and label consensus (Steps I and II); (2) image augmentation and pre-processing (Step III); and (3) training with the novel backpropagation algorithm that can work with uninterpretable images (Step IV); (illustrated in Figure 7.1).

7.2.3.3 Expert Diagnosis and Label Consensus (Steps I and II)

Each OCT and CFP image was individually labeled by the panel of retina professionals as uninterpretable, RPN, and RPP. Then, to train the CNN model, we determined the final diagnosis as RPN if one imaging modality was deemed uninterpretable and other RPN or both were RPN. Similarly, we labeled a patient RPP if we had at least one modality read as RPP. In the case of both modalities being uninterpretable, we referred to it as RPPP (Appendix C.1 and Table C.1).

7.2.3.4 Image Augmentation and Pre-processing (Step III)

Bearing in mind that our dataset was limited (which is often the case with clinical data), we augmented the dataset by rotation, random cropping, flipping, etc. (Shorten & Khoshgoftaar, 2019) Given that OCT images usually come with extensive background noise, which can prevent the DL-based models from capturing the image biomarkers (Moreno-Barea et al., 2018), we applied Gaussian filters (Gonzalez & Woods, 2007) for noise reduction. No images were augmented for the validation set. However, the OCT images were de-noised using Gaussian blur as in the training set. Details are introduced in Appendix C.2.

7.2.3.5 CNN Model Architecture and the Back Propagation Algorithm (Step IV)

We developed a multi-modal CNN that takes as an input OCT and CFP images jointly and classifies them into RPN and RPP categories (Figure 2). First, the input OCT and CFP images were processed by two sets of convolutional filters to obtain corresponding feature maps. Then the output feature maps were fed into the global average pooling layers for dimension reduction (to derive feature vectors for both imaging modalities), which was then fed into a global, fully connected layer designed to: (1) map feature vectors to logits; and (2) to implicitly reach a consensus between the prediction outcomes (as the results from different imaging modalities could oppose each other – e.g., pathology does exist in one and does not in the other). Finally, Softmax activation was applied to the output layer to map the logits to probabilities of classifying the inputs as RPP. To ensure that the CNN can successfully handle uninterpretable images presented in both training and testing datasets, we developed an alternate gradient descent (AGD) algorithm. This way, we could minimize the impact of uninterpretable images on the prediction performance implicitly without formulating the binary classification problem as a multi-category task (e.g., RPN, RPP, and uninterpretable).

7.2.3.6 The AGD algorithm

We first divided all the weight parameters θ in the CNN into three subsets θ_1 , θ_2 and θ_3 , which represent the weights for the convolutional blocks and global average pooling layer

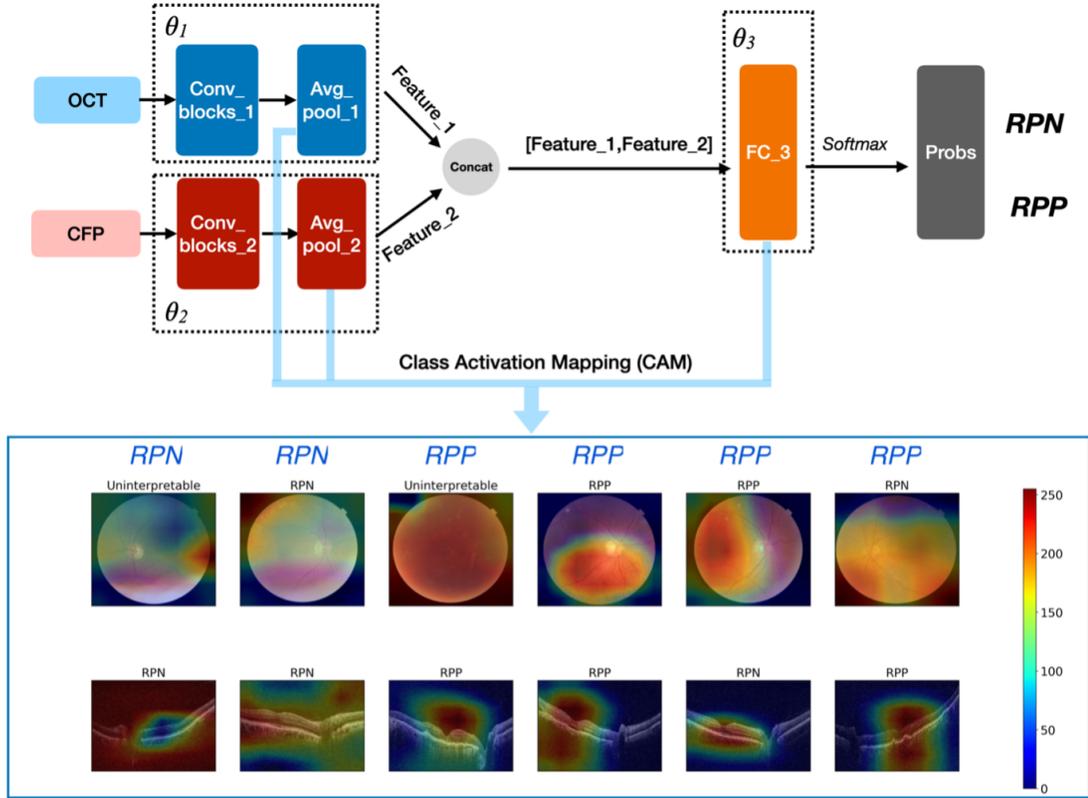


FIGURE 7.2: The OCT and CFP modalities are first processed with two sets of convolutional filters respectively; the resulting features are then concatenated and processed by a fully connected layer (θ_3) for classification. CAMs are generated using the outputs from the two global average pooling layers and weights from the fully connected layer.

that process the OCT inputs (i.e., `Conv_blocks_1` and `Avg_pool_1` in Figure 7.2), the convolutional and global average pooling layers for the CFP modality (i.e., `Conv_blocks_2` and `Avg_pool_2` in Figure 7.2), and the final fully connected layer (i.e., `FC_3` in Figure 7.2) respectively. The following briefly illustrates how the AGD algorithm works during the training of the CNN model. In each training iteration, (I) we first updated θ_1 by minimizing the binary cross-entropy loss (BCEL) between the CNN predictions corresponding to the input (O_k, C_k) samples that contain interpretable OCT images and the labels associated with them (i.e., in this step, the uninterpretable images were not included while calculating the training loss); (II) then similarly, θ_2 was updated by minimizing the BCEL between the CNN predictions corresponding to the input (O_k, C_k) samples with interpretable CFP

images from the training inputs and the labels associated with them; and (III) finally, θ_3 was updated to minimize the BCEL between the CNN predictions given all input (O_k, C_k) samples (i.e., both interpretable and uninterpretable OCT/CFP) and the associated labels. After step (I) and (II), the convolutional filters processing the OCT and CFP modality (i.e., θ_1, θ_2) were trained towards extracting features that can best differentiate RPN/RPP samples if the inputs were interpretable. On the other hand, if one modality (or both modalities) of the inputs was (were) uninterpretable, then the features extracted by the corresponding convolutional filters were considered uninformative, as they were not included during the training of θ_1 and θ_2 . In step (III), the weights of the fully connected layer θ_3 were optimized to capture if the features output from θ_1 and θ_2 implies RPN, RPP or uninformative, as well as learn to infer the correct predictions when the features corresponding to the OCT and CFP modality carry inconsistent information (e.g., one implies RPN whereas the other implies RPP, or the other was uninformative). As a result, the CNN was trained, using the AGD algorithm, to implicitly handle the uninterpretable images contained in the dual-inputs (O_k, C_k) without classifying them as a third class besides RPN and RPP. The illustration of the AGD algorithm from the mathematical perspective is provided in Appendix C.3.

7.2.3.7 Transfer learning

Transfer learning was applied to pre-train the convolutional blocks (i.e., θ_1, θ_2) in the CNN model, as it was shown to be effective in boosting both training efficiency and validation performance³⁰. Specifically, we used the open-source OCT dataset containing 108,312 OCT scans from four different categories – Choroidal Neovascularization (37,206 images), DME (11,349 images), Drusen (8,617 images), and normal (51,140 images), which were provided by Kermany, Goldbaum, Cai, and et al., 2018 We also used the CFP image dataset containing 35,126 CFP images with (25,810 images) and without DR pathology (9,316 images), which are obtained from Kaggle (“Diabetic Retinopathy Detection – Identify signs of diabetic retinopathy in eye images”, n.d.). Then all the CFP images with DR pathology

were flipped over horizontally and vertically to balance the number of images in the two classes, and loose pairing (W. Wang et al., 2019b) was performed to couple the OCT and CFP modality, which then generated 100,000 ‘nominal’ eyes. We further labeled the OCT images that contained any pathology as RPP and took a logical and between the individual OCT and CFP labels to determine the final diagnoses used to pre-train the network. Given that these two datasets did not contain any uninterpretable images, we pre-trained the network, as illustrated in Figure 7.2, by minimizing the cross-entropy loss between the CNN predictions and labels for all images. Appendix C.3 illustrated this optimization problem from a mathematical perspective. Although the open-source OCT dataset did not contain all retinal pathologies that we were interested in, the CNN model was still trained to effectively locate the biomarkers that help distinguish inputs as RPN and RPP, as presented in the results section.

7.2.3.8 Specific convolutional layer architecture and training hyper-parameters

The convolutional blocks in both the OCT and the CFP branches of the network (i.e., the `Conv_blocks_1` and `Conv_blocks_2` from Figure 7.2) employed the inception-v334 architecture. Furthermore, the open-source OCT dataset that we used to pre-train the CNN model had also been shown to attain the highest accuracy with the inception-v3 structure (Keremany, Goldbaum, Cai, & et al., 2018). Specifically, during training, both OCT and CFP images were resized to 299x299 to comply with the design of the convolutional layers before feeding into the network (Szegedy et al., 2016a). After performing global average pooling for the OCT and CFP streams, the image features (i.e., `Feature_1` and `Feature_2` in Figure 7.2) had the size $n \times 1 \times 1 \times 2048$, where n denotes the batch size. The two feature vectors were then concatenated and reshaped to a $n \times 4096$ vector, which was then processed by a fully-connected layer with 4096 nodes to generate prediction logits. Finally, Softmax functions were applied to normalize the logits as probabilities of classifying the inputs as RPN/RPP. During training, Adam optimizer (Saad, 1998) was used to minimize training losses, where the learning rate was set to be $1e-04$ with exponential decay of 0.91

Table 7.3: Performance Comparison between Our Approach (alternate gradient descent with binary output), Baseline A (2 single modal CNNs as 3-output task), Baseline B (interpretability classifiers followed by 2 single modal CNNs as 2-output task), and Baseline C (two-stream CNNs representing state-of-the-art methods for 2-modal image analysis) on the full Testing Dataset.

	Accuracy/No. (% , 95% CI)	FNR/No. (% , 95% CI)	Recall/No. (% , 95% CI)
Our Approach	101 (88.60%, 82.76%-94.43%)	7 (12.28%, 6.26%-18.31%)	50 (87.72%, 81.69%-93.74%)
Baseline A	93 (81.58%, 74.46%-88.70%)	19 (33.33%, 24.68%-41.99%)	38 (66.67%, 58.01%-75.32%)
Baseline B	81 (71.05%, 62.73%-79.38%)	23 (40.35%, 31.34%-49.36%)	34 (59.65%, 50.64%-68.66%)
Baseline C	91 (79.82%, 72.46%-87.19%)	6 (10.53%, 4.89%-16.16%)	51 (89.47%, 83.84%-95.11%)
	Specificity/No. (% , 95% CI)	AUC % (95% CI)	P-values
Our Approach	51 (89.47%, 83.84%-95.11%)	92.74% (87.71%-97.76%)	< 0.001
Baseline A	55 (96.49%, 93.11%-99.87%)	83.58% (76.11%-91.05%)	< 0.001
Baseline B	47 (82.46%, 75.47%-89.44%)	74.05% (64.96%-83.14%)	< 0.001
Baseline C	40 (70.18%, 61.78%-78.57%)	87.32% (80.71%-93.93%)	< 0.001

in 1500 steps.

7.3 Results

To validate our approach, we selected the following three baseline methods to compare with our method: (A) training two CNN models that classifies the OCT and CFP modality respectively into 3 categories (RPN, RPP and uninterpretable), then the final diagnoses are determined following the LCM illustrated in Appendix C.1 and Table C.1; (B) first, two classifiers are trained to classify the interpretability for the OCT and CFP modality separately, followed by two CNN models that identify the presence of retinal pathology for interpretable OCT and CFP images respectively, with the final diagnoses being determined using the LCM; and (C) a two-stream CNN model based on the state-of-the-art multi-modal ophthalmological image analysis methods developed by Vaghefi et al., 2020; W. Wang et al., 2019a; Z. Xu et al., 2020, which uses the CNN architecture that does not consider any uninterpretable images, but is trained to minimize the cross-entropy loss with conventional backpropagation algorithms, instead of the AGD as proposed in our work. In Appendix C.4, we illustrated the intuition of designing Baseline A and B and their implementation details.

Table 7.3 shows the performance comparison between our approach and the baseline methods in terms of accuracy, false-negative rate (FNR), Recall (or true positive rate),

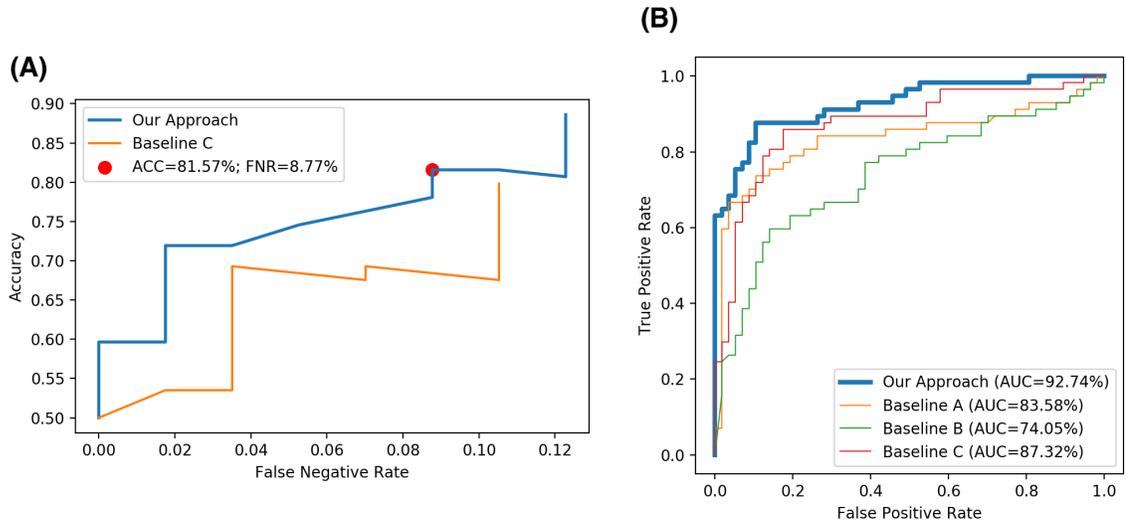


FIGURE 7.3: (A) ACC-FNR Curve for our approach and Baseline C. Baseline C has lower FNR than our approach with a decision threshold of 0.5; however, our method achieves both higher accuracy and lower FNR with a decision threshold providing optimal tradeoff between accuracy and FNR (e.g., the threshold of 0.65 as shown by the red dot in the plot). (B) ROC Curves for our approach and Baseline Methods. Our approach achieves the highest AUC comparing to all the baseline methods.

Specificity (or true negative rate) and Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve; FNR is defined as

$$FNR = \frac{FN}{TP + FN} = 1 - Recall, \quad (7.1)$$

with FN representing the false negatives and TP referring to the true positives. We chose FNR as one of the metrics because it evaluates the portion of the RPP patients who are falsely identified as RPN, or in other words, the patients who have retinal pathology presented but failed to be recognized by the automated diagnosis system due to erroneous classifications. Our approach achieved 88.60% accuracy with 95% CI of [82.76%, 94.43%], which outperforms all three baseline methods, as shown in Table 2. We also attained an FNR of 12.28% with 95% confidence interval (CI) of [6.26%, 18.31%] (or Recall of 87.72% with 95% CI of [81.69%, 93.74%]), which outperforms baseline A and B.

To address the fact that baseline C results in a lower FNR (and thus higher Recall) than our model, we created the Accuracy-FNR plots (Figure 7.3, the blue curve represents our approach, while the orange shows Baseline C) showing how the accuracy and FNR

Table 7.4: Performance Comparison between Our Approach (alternate gradient descent with binary output), Baseline A (2 single modal CNNs as 3-output task), Baseline B (interpretability classifiers followed by 2 single modal CNNs as 2-output task), and Baseline C (two-stream CNNs representing state-of-the-art methods for 2-modal image analysis) on the Dataset containing only Interpretable Images.

	Accuracy/No. (% , 95% CI)	FNR/No. (% , 95% CI)	Recall/No. (% , 95% CI)
Our Approach	69 (88.46%, 81.37%-95.55%)	7 (16.67%, 8.40%-24.94%)	35 (83.33%, 75.06%-91.60%)
Baseline A	59 (75.64%, 66.11%-85.17%)	17 (40.48%, 29.58%-51.37%)	25 (59.52%, 48.63%-70.42%)
Baseline B	65 (83.33%, 75.06%-91.60%)	10 (23.81%, 14.36%-33.26%)	32 (76.19%, 66.74%-85.64%)
Baseline C	68 (87.18%, 79.76%-94.60%)	6 (14.29%, 6.52%-22.05%)	36 (85.71%, 77.95%-93.48%)
	Specificity/No. (% , 95% CI)	AUC % (95% CI)	P-values
Our Approach	34 (94.44%, 89.36%-99.53%)	93.85% (88.39%-99.31%)	0.00766
Baseline A	34 (94.44%, 89.36%-99.53%)	79.89% (71.00%-88.79%)	< 0.001
Baseline B	33 (91.67%, 85.53%-97.80%)	89.48% (81.88%-97.09%)	< 0.001
Baseline C	32 (88.89%, 81.91%-95.86%)	90.21% (83.31%-97.12%)	0.00443

change when different decision thresholds are applied to the probabilities output from the CNN (which can be interpreted as the confidence of classifying the input samples as RPP cases) (Szegedy et al., 2016a). All the thresholds are sampled uniformly between 0.5 and 1, where the top-right endpoints of both curves correspond to the threshold of 0.5 (i.e., the samples that result in prediction probability greater than 0.5 are determined as RPP while the rests are classified as RPN) and the bottom-left points are associated with threshold 1 (i.e., all the inputs are classified as RPN regardless the presence of pathology or not). As can be observed from Figure 7.3, our method is capable of achieving an FNR of 8.77% with 95% CI of [3.58%, 13.96%], with an accuracy of 81.57% with 95% CI of [74.45%, 88.69%], which outperforms Baseline C concerning both metrics given a threshold of 0.65 (shown as the red dot in Figure 3A). Moreover, our method attains higher accuracy than Baseline C for any decision threshold in [0.5, 1]. Our approach achieved a Specificity of 89.47% with 95% CI of [83.84%, 95.11%], which outperforms Baseline B and C. Note that Baseline A gives rise to a higher Specificity due to miss-classifying RPP samples as RPN, which is indicated by the very high FNR (33.33%, 95% CI [24.68%, 41.99%]) and the relatively low AUC (83.58%, 95% CI [76.11%, 91.05%]). Finally, our approach reached an AUC of 92.74% with 95% CI of [87.71%, 97.76%], which is higher than all the baseline methods, as captured

by the ROC curves shown in Figure 7.3. Consequently, our approach achieved satisfactory performance evaluated through the five metrics and was able to balance between accuracy and FNR flexibly by selecting appropriate decision thresholds.

To evaluate the impact of the uninterpretable images on prediction performance, we evaluated our model by excluding them from the testing dataset – i.e., each eye with at least one uninterpretable image was excluded (Table 7.4). Performance of our model did not change when evaluated on interpretable images only. On the other hand, Baseline B and C methods’ performances increased dramatically in this setting, as expected, because both methods were not designed to process the uninterpretable inputs. Finally, Baseline A had slightly decreased Accuracy and Recall, likely due to the higher false negative rate of the Baseline A model. All of this could be observed by comparing the changes in the FNR between Table 7.3 and 7.4, where the number of false negative samples barely decreased when the uninterpretable samples were excluded. In other words, for Baseline B and C, by removing uninterpretable images, the classification performance improved, as those images lead to decreasing Recall (or increasing FNR), while the opposite was true for Baseline A. As presented, the uninterpretable images negatively impacted the baseline methods while having a minimal impact on our approach.

We further validated our model by generating class activation maps (CAMs), which can visualize how much “attention” the CNN model is paying to each pixel of the input images (Figure 7.2). We followed the procedure proposed by Zhou et al., 2016, where the weights of the fully connected layer (i.e., FC_3 in Figure 7.2) and the image features generated from the global average pooling layers (i.e., Avg_pool_1 and Avg_pool_2 in Figure 7.2) were used to generate attention values associated with all pixels in the input images from both imaging modalities. This evaluates to what extent each pixel is weighted while the CNN model generates predictions. The higher values correspond to the stronger attention, while lower to weaker attention (Figure 7.1).

7.4 Discussion

To capture and generalize the ideas behind these four methods and emphasize the importance of uninterpretable image utilization, we combined them in the Baseline C learning approach and compared it to our model. We concluded that although Baseline C achieved slightly lower FNR, it attained 9.3% less accuracy than our method when the presented decision thresholds were used in our model (Table 7.3). However, when the decision thresholds in our model were adjusted, our approach achieved both lower FNR and higher accuracy than Baseline C, but with slightly lower accuracy than with our initial decision threshold – this highlights that by controlling decision thresholds, we were able to make a tradeoff between accuracy and FNR.

Furthermore, this underlined the importance of taking into account uninterpretable images during the training phase and showed that our AGD algorithm and the obtained CNN model could effectively handle uninterpretable images. Also, we designed Baseline A and B models to evaluate the prediction performance when the AGD backpropagation algorithm was not employed, and the input images were classified into three categories (i.e., RPN, RPP, and uninterpretable), as opposed to the two-class problem addressed by our model trained by the AGD algorithm. Comparing these two methods to ours showed that our method had higher accuracy and lower FNR. The improved performance of our method and Baseline C compared to Baseline A and B methods confirm the strength of multi-modal analysis, where the CNN models can effectively capture the correlation among different imaging modalities and make accurate predictions.

Finally, FNR is an important factor to consider while validating different image interpretation models because it is crucial not to miss pathology that can have serious consequences. As shown in the ACC-FNR Curve in Figure 7.3, our CNN based approach allowed the users to balance the tradeoff between accuracy and FNR by customizing the decision thresholds (i.e., a threshold around 0.5 can be applied for attaining higher accuracy, while a threshold greater than 0.5 leads to lower FNR).

8. Conclusion

As a majority of existing AI and DL methods are developed and bench-marked against environments and datasets that are pre-cleaned, well-structured, and provided with substantial amount, it is intractable to acquire data at this scope to support training control policies that can be tested and deployed practically. In this dissertation, we have introduced various frameworks that aimed to facilitate fully automated decision-making and control systems for real-world applications. We have focused on addressing the critical challenges encountered in typical real-world production pipelines that limits the efficiency of training, as well as the safety upon testing and deployment. Specifically, we have developed (i) offline RL frameworks that synthesized control policies in healthcare systems, maximizing both environmental returns and HF, with OPE methods facilitating the evaluation and selection of RL policies without online interactions. We have also introduced frameworks that (ii) tackled high-dimensionality, multi-modality, and irregularities in healthcare data and captured underlying factors crucial to automated disease diagnoses and prognoses.

8.1 Summary

In Chapter 2 we introduced the preliminaries pertaining to variational inference, offline RL and OPE basics.

In Chapter 3 we have developed the VLBM which can accurately capture the dynamics underlying environments from offline training data that provide limited coverage of the state and action space; this is achieved by using the RSA term to smooth out the information flow from the encoders to decoders in the latent space, as well as the branching architecture which improve VLBM’s robustness against random initializations. We have followed evaluation guidelines provided by the DOPE benchmark, and experimental results have shown that the VLBM generally outperforms the state-of-the-art model- based OPE method using AR architectures, as well as other model-free methods. VLBM can also facilitate off-policy optimizations, which can be explored in future works. Specifically, VLBM can serve as a synthetic environment on which optimal controllers (e.g., linear–quadratic regulator) can be

deployed. On the other hand, similar to Dreamer and SLAC, policies can be updated jointly with training of VLBM, but without the need of online interactions with the environment during training.

In Chapter 4 we have introduced the OPEHF framework that revived existing OPE methods for estimating human returns, through RILR. The framework was validated over two real-world experiments and one simulation environment, outperforming the baselines in all setups. Although in the future it could be possible to extend OPEHF to facilitate estimating the HF signals needed for updating the policies similar to RLHF, we focused on policy evaluation which helped to isolate the source of improvements; as policy optimization’s performance may depend on multiple factors, such as the exploration techniques used as well as the objective/optimizer chosen for updating the policy. Moreover, this work mainly focuses on the scenarios where the human returns are directly provided by the participants. So under the condition where the HF signals are provided by 3-rd parties (e.g, clinicians), non-trivial adaptations over this work may be needed to consider special cases such as conflicting HF signals provided by different sources.

In Chapter 5 we have introduced an offline RL and OPE framework to design and evaluate closed-loop DBS controllers using only historical data. Moreover, a policy distillation method was introduced to further reduce the computation requirements for evaluating RL policies. The control efficacy and energy efficiency of the RL controllers were validated with clinical testing over 4 patients. Results showed that RL-based controllers lead to similar control efficacy as cDBS, but with significantly reduced stimulation energy. The computation times for the RL and distilled RL controllers were compared, showing that the distilled version executed significantly faster; future work will focus on further reducing execution times of the distilled RL controllers to match capabilities of implanted devices. Finally, the DLSSM is trained to estimate the expected returns of RL policies, which outperforms existing IS-based OPE methods, in terms of rank correlations, regrets and MAEs.

In Chapter 6 we have have developed the GIL method, training DL models (MLPs and

LSTMs) to directly perform inference with incomplete data, without the need for imputation. Existing methods addressing the problem of missing data mostly follow the two-step (imputation-prediction) framework. However, the error produced during imputation can be propagated into subsequent tasks, especially when the data have high missingness rates or small sample sizes. GIL circumvents these issues by applying importance weighting to the gradients to leverage the information underlying the missingness patterns in the data. We have evaluated our method by comparing it to the state-of-the-art baselines using IAs on two real-world datasets and one benchmark dataset.

In Chapter 7 we have developed a system that facilitated a fully automated, learning-based interpretation tool for retinal images obtained remotely (e.g. teleophthalmology) through different imaging modalities that may include imperfect (uninterpretable) images.

8.2 Future Work

There exist various avenues to which the frameworks introduced in this dissertation can be extended.

8.2.1 Meta-Learning-Based Reward Shaping to Improve the Robustness of OPE and OPEHF

Following from Chapters 3 and 4, the OPEHF framework can be further extended to improve the robustness of OPE and OPEHF during the policy evaluation and selection step, which is critical to guarantee the safety and efficacy of online policy testing/deployment. Since most of the OPE techniques fall short in accurately estimating the returns and ranks of target policies (i.e., the policies to be evaluated or a batch of policy candidates upon which the final deployment is based), once the state-action visitation distribution resulted from the target policies becomes drastically different than the one corresponding to behavioral policies that are used to collect the offline dataset. Such a challenge is also carried onto OPEHF’s setup, and by resolving it, the accuracy and effectiveness of the HF signals extrapolated by OPEHF can be further improved.

8.2.2 A More Comprehensive Automation of DBS Therapy

Following Chapter 5, multiple improvements can be made to further automate the DBS therapy for PD patients. First, generalization to broader cohorts – multi-agent and federated learning techniques can be adapted into our framework, such that one may need collect less patient-specific data to scale up the training efficiency and reduce the number of trials needed before determining the control policy that can be deployed for long term. Second, grouping participants based on disease characteristics – specifically, to lay out foundations of control policies that can be efficiently adapted to participants who experience similar symptoms and have similar characteristics. Such an effort can further improve the efficiency while working with larger cohorts. Finally, improvement in the selection of initial policy – one can develop techniques to best select/train an initial policy that could be deployed to new participants upon entering the cohort, *i.e.*, when limited historical data is available, to quickly explore high-reward regions in early stage of testing.

8.2.3 Capturing Temporal Correlations within Healthcare Data for Disease Progression Analyses

Future work extending the framework introduced in Chapters 6 and 7 include capturing temporal correlations (*e.g.*, pathological progression) existed among visits, which can help determine the initiation and prescription of procedures and medications for diseases that require long-term treatments. Moreover, the GIL framework can be adapted toward handling data missingness in domains with non-tabular modality, or multi-modal inputs, such as image super-resolution and compressive sensing.

Appendix A. Proof of Proposition 1

Recall that the per-decision importance sampling (PDIS) (Precup, 2000) estimator follows $\hat{G}_{PDIS}^\pi = \frac{1}{N} \sum_{i=1}^{N-1} \sum_{t=0}^{T-1} \gamma^t \omega_{0:t}^{(i)} r_t^{\mathcal{H}(i)}$; here, $\omega_t^{(i)} = \frac{\pi(a_t^{(i)} | s_t^{(i)})}{\hat{\beta}(a_t^{(i)} | s_t^{(i)})}$, and henceforth $\omega_{0:t}^{(i)} = \prod_{k=0}^t \frac{\pi(a_k^{(i)} | s_k^{(i)})}{\hat{\beta}(a_k^{(i)} | s_k^{(i)})}$ are the PDIS weights for offline trajectory $\tau^{(i)}$. To simplify our notation, we consider the PDIS estimator defined over a single trajectory, *i.e.*,

$$\hat{G}_{PDIS}^\pi = \sum_{t=0}^{T-1} \gamma^t \omega_{0:t} r_t^{\mathcal{H}}, \quad (\text{A.1})$$

as the results can be carried to N trajectories by multiplying with a factor $1/N$. We omit the superscript-ed (i) 's in the rest of the proof also for conciseness.

Proof. We start with a lemma from (Y. Liu et al., 2020).

Lemma 1 ((Y. Liu et al., 2020)). *Given X_t and Y_t as two sequences of random variables. Then*

$$2 \sum_{t < k} \mathbb{E}[Y_t Y_k] - 2 \sum_{t < k} \mathbb{E}[\mathbb{E}[Y_t | X_t] \mathbb{E}[Y_k | X_k]] \leq \mathbb{V}\left(\sum_t Y_t\right) - \mathbb{V}\left(\sum_t \mathbb{E}[Y_t | X_t]\right), \quad (\text{A.2})$$

where $\mathbb{V}(\cdot)$ refer to the variances.

Now, if we set $Y_t = r_t^{\mathcal{H}} \omega_{0:T}$ and $X_t = \tau_{0:t}$ which is a segment (from $t = 0$ to $t = t$) of an offline trajectory τ , it is sufficient to prove Proposition 1 by proving that for any $1 \leq t+1 \leq k \leq T$,

$$\mathbb{E}[r_t^{\mathcal{H}} r_k^{\mathcal{H}} \omega_{0:t} \omega_{0:k}] \leq \mathbb{E}[r_t^{\mathcal{H}} r_k^{\mathcal{H}} \omega_{0:T-1} \omega_{0:T-1}]; \quad (\text{A.3})$$

since

$$\mathbb{E}[r_t^{\mathcal{H}} r_k^{\mathcal{H}} \omega_{0:t} \omega_{0:k}] = \mathbb{E}[\mathbb{E}[Y_t | X_t] \mathbb{E}[Y_k | X_k]] \quad (\text{A.4})$$

$$\leq \mathbb{E}[Y_t Y_k] \quad (\text{A.5})$$

$$= \mathbb{E}[r_t^{\mathcal{H}} r_k^{\mathcal{H}} \omega_{0:T-1} \omega_{0:T-1}]. \quad (\text{A.6})$$

Applying the law of total expectations to (A.3), *i.e.*,

$$\mathbb{E}[\mathbb{E}[r_t^{\mathcal{H}} r_k^{\mathcal{H}} \omega_{0:t} \omega_{0:k} | \tau_{0:t}]] \leq \mathbb{E}[\mathbb{E}[r_t^{\mathcal{H}} r_k^{\mathcal{H}} \omega_{0:T-1} \omega_{0:T-1} | \tau_{0:t}]]; \quad (\text{A.7})$$

then it is sufficient to show

$$\mathbb{E}[r_t^{\mathcal{H}} r_k^{\mathcal{H}} \omega_{0:t} \omega_{0:k} | \tau_{0:t}] \leq \mathbb{E}[r_t^{\mathcal{H}} r_k^{\mathcal{H}} \omega_{0:T-1} \omega_{0:T-1} | \tau_{0:t}]. \quad (\text{A.8})$$

We start by showing that

$$\mathbb{E}[r_t^{\mathcal{H}} r_k^{\mathcal{H}} \omega_{0:t} \omega_{0:k} | \tau_{0:t}] = \omega_{0:t}^2 \mathbb{E}[r_t^{\mathcal{H}} | \tau_{0:t}] \mathbb{E}[r_k^{\mathcal{H}} \omega_{t+1:k} | \tau_{0:t}] \quad (\text{A.9})$$

$$= \omega_{0:t}^2 \mathbb{E}[r_t^{\mathcal{H}} | \tau_{0:t}] \mathbb{E}[r_k^{\mathcal{H}} \omega_{t+1:T-1} | \tau_{0:t}] \mathbb{E}[\omega_{t+1:T-1} | \tau_{0:t}]. \quad (\text{A.10})$$

The transition above follows from the fact that the likelihood ratio $\omega_{0:t}$ is a martingale as shown in (L'Ecuyer & Tuffin, 2008; Y. Liu et al., 2020), *i.e.*, $\mathbb{E}[\omega_{0:T-1} | \tau_{0:t}] = \omega_{0:t}$, which implies $\mathbb{E}[\omega_{t+1:T-1}] = 1$ and therefore

$$\mathbb{E}[r_k^{\mathcal{H}} \omega_{t+1:k} | \tau_{0:t}] \quad (\text{A.11})$$

$$= \mathbb{E}[r_k^{\mathcal{H}} \omega_{t+1:k} | \tau_{0:t}] \mathbb{E}[\omega_{k+1:T-1} | \tau_{0:t}] \quad (\text{A.12})$$

$$= \mathbb{E}[r_k^{\mathcal{H}} \omega_{t+1:T-1} | \tau_{0:t}]. \quad (\text{A.13})$$

Note that we keep using the notation $\mathbb{E}[r_t^{\mathcal{H}} | \tau_{0:t}]$ in (A.9) and (A.10), due to the setting of HMDP that $r_t^{\mathcal{H}}$ is a variable sampled from the distribution $R^{\mathcal{H}}(\cdot | s_t, a_t)$ given (s_t, a_t) .

Since $\tau_{0:t}$ is given, $r_k^{\mathcal{H}}$ and $\omega_{t+1:T-1}$ are equivalent to $r_{k-t-1}^{(j)}$ and $\omega_{0:T-t-2}^{(j)}$ for some other trajectory $\tau^{(j)} \sim \rho^\beta$. Also given the statement that $\omega_{0:T-t-2}^{(j)}$ and $r_{k-t-1}^{(j)} \omega_{0:T-t-2}^{(j)}$ are positively correlated, it follows from (A.10) that

$$\mathbb{E}[r_t^{\mathcal{H}} r_k^{\mathcal{H}} \omega_{0:t} \omega_{0:k} | \tau_{0:t}] \quad (\text{A.14})$$

$$= \omega_{0:t}^2 \mathbb{E}[r_t^{\mathcal{H}} | \tau_{0:t}] \mathbb{E}[r_k^{\mathcal{H}} \omega_{t+1:T-1} | \tau_{0:t}] \mathbb{E}[\omega_{t+1:T-1} | \tau_{0:t}] \quad (\text{A.15})$$

$$\leq \omega_{0:t}^2 \mathbb{E}[r_t^{\mathcal{H}} | \tau_{0:t}] \mathbb{E}[r_k^{\mathcal{H}} \omega_{t+1:T-1} \omega_{t+1:T-1} | \tau_{0:t}] \quad (\text{A.16})$$

$$= \mathbb{E}[r_t^{\mathcal{H}} r_k^{\mathcal{H}} \omega_{0:T-1} \omega_{0:T-1} | \tau_{0:t}], \quad (\text{A.17})$$

which justifies (A.8) and completes the proof. \square

Appendix B. Gradient Importance Learning

This chapter contains the appendices pertaining to Chapter 6.

B.1 Extending GIL to LSTMs

We also consider the use of an LSTM as the *encoder* for sequential inputs with varied lengths. Specifically, in this case, we can define $\mathbf{X}_j = (\mathbf{x}_{j,1}^\top, \mathbf{x}_{j,2}^\top, \dots, \mathbf{x}_{j,T_j}^\top)^\top \in \mathbb{R}^{T_j \times d}$ along with $\mathbf{M}_j = (\mathbf{m}_{j,1}^\top, \mathbf{m}_{j,2}^\top, \dots, \mathbf{m}_{j,T_j}^\top) \in \{0, 1\}^{T_j \times d}$ such that each $\mathbf{x}_{j,t} \in \mathbb{R}^d$ and $\mathbf{m}_{j,t} \in \{0, 1\}^d$, where $i \in [1, N]$ and $t \in [1, T_j]$. Recall that the forward pass of an LSTM follows

$$\begin{aligned} \mathbf{o}_t &= \sigma(\mathbf{W}_o \mathbf{x}_{j,t} + \mathbf{U}_o \mathbf{h}_{t-1}), & \mathbf{i}_t &= \sigma(\mathbf{W}_i \mathbf{x}_{j,t} + \mathbf{U}_i \mathbf{h}_{t-1}), & \mathbf{f}_t &= \sigma(\mathbf{W}_f \mathbf{x}_{j,t} + \mathbf{U}_f \mathbf{h}_{t-1}), \\ \mathbf{g}_t &= \tanh(\mathbf{W}_g \mathbf{x}_{j,t} + \mathbf{U}_g \mathbf{h}_{t-1}), & \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t, & \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \end{aligned} \quad (\text{B.1})$$

where $\mathbf{x}_t \in \mathbb{R}^d$ is the input at time step t , $\{\mathbf{W}_o, \mathbf{W}_i, \mathbf{W}_g, \mathbf{W}_f, \mathbf{U}_o, \mathbf{U}_i, \mathbf{U}_g, \mathbf{U}_f\}$ are the weights, $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function, and \odot is the element-wise product. The output layer is appended after \mathbf{h}_t to constitute the *inference* layer, *i.e.*,

$$\hat{\mathbf{y}}_t = \phi_{out}(\mathbf{W}_{out} \mathbf{h}_t), \quad (\text{B.2})$$

where ϕ_{out} is the activation function, $\mathbf{W}_{inf} = \mathbf{W}_{out}$ are the weights and $t \in [1, T_j]$. The following proposition shows that given a loss function $E(\hat{\mathbf{y}}, \mathbf{y})$, the gradients of E w.r.t the encoding weights $\mathbf{W}_{enc} = \{\mathbf{W}_o, \mathbf{W}_i, \mathbf{W}_g, \mathbf{W}_f\}$ can be written in the form of outer products. Note that the gradients for (autoregressive) parameters depending on previous hidden states \mathbf{h}_{t-1} , *i.e.*, $\{\mathbf{U}_o, \mathbf{U}_i, \mathbf{U}_g, \mathbf{U}_f\}$, like the bias terms, cannot be written as outer products; thus, these weights are updated following the regular SGD, without importance weighting. The proof is provided following the proposition.

Proposition 3. *Given an LSTM as described in (B.1)-(B.2) and a smooth loss function $E(\hat{\mathbf{y}}, \mathbf{y})$, the gradients of E w.r.t. $\mathbf{W}_{enc} = \{\mathbf{W}_o, \mathbf{W}_i, \mathbf{W}_g, \mathbf{W}_f\}$ at time t can be written in outer product forms – *i.e.*, it holds that $\left. \frac{\partial E}{\partial \mathbf{W}_o} \right|_t = \Delta_t^o \mathbf{x}_t^\top$, $\left. \frac{\partial E}{\partial \mathbf{W}_i} \right|_t = \Delta_t^i \mathbf{x}_t^\top$, $\left. \frac{\partial E}{\partial \mathbf{W}_g} \right|_t = \Delta_t^g \mathbf{x}_t^\top$, $\left. \frac{\partial E}{\partial \mathbf{W}_f} \right|_t = \Delta_t^f \mathbf{x}_t^\top$, where $\Delta_t^o, \Delta_t^i, \Delta_t^g, \Delta_t^f$ are the gradients propagated from the inference layers defined in below, and \mathbf{x}_t represents the observation obtained at time step t from any sequence $\mathbf{X}_j \in \mathcal{X}$ in general.*

Then it follows that if we define \mathbf{x} as the t -th observation from any sequence $\mathbf{X}_j \in \mathbb{R}^{T_j \times d}$, regardless of its index, (6.4) can be directly used for training the LSTM encoding weights $\mathbf{W}_{enc} = \{\mathbf{W}_o, \mathbf{W}_i, \mathbf{W}_g, \mathbf{W}_f\}$.

Now we prove the Proposition above.

We first define

$$\Delta_t^o = \partial E / \partial \mathbf{h}_t \odot \tanh(\mathbf{c}_t) \odot \mathbf{o}_t \odot (1 - \mathbf{o}_t), \quad (\text{B.3})$$

$$\Delta_t^i = \partial E / \partial \mathbf{h}_t \odot \mathbf{o}_t \odot (1 - \tanh^2(\mathbf{c}_t)) \odot \mathbf{g}_t \odot \mathbf{i}_t \odot (1 - \mathbf{i}_t), \quad (\text{B.4})$$

$$\Delta_t^g = \partial E / \partial \mathbf{h}_t \odot \mathbf{o}_t \odot (1 - \tanh^2(\mathbf{c}_t)) \odot \mathbf{i}_t \odot (1 - \mathbf{g}_t^2), \quad (\text{B.5})$$

$$\Delta_t^f = \partial E / \partial \mathbf{h}_t \odot \mathbf{o}_t \odot (1 - \tanh^2(\mathbf{c}_t)) \odot \mathbf{c}_{t-1} \odot \mathbf{f}_t \odot (1 - \mathbf{f}_t). \quad (\text{B.6})$$

Now to prove the proposition we start from the derivative of the smooth loss function E w.r.t. \mathbf{x} , which can be derived as

$$\frac{\partial E}{\partial \mathbf{h}_t} = \left[\frac{\partial (\mathbf{W}_{out} \mathbf{h}_t)}{\partial \mathbf{h}_t} \right]^\top. \quad (\text{B.7})$$

$$\left[\frac{\partial E}{\partial \hat{\mathbf{y}}_t} \odot \phi'(\mathbf{W}_{out} \mathbf{h}_t) \right] \quad (\text{B.8})$$

$$= \mathbf{W}_{out}^\top \left[\frac{\partial E}{\partial \hat{\mathbf{y}}_t} \odot \phi'(\mathbf{W}_{out} \mathbf{h}_t) \right]. \quad (\text{B.9})$$

Then the derivatives of E w.r.t. the \mathbf{o}_t , \mathbf{c}_t , \mathbf{f}_t , \mathbf{c}_{t-1} , \mathbf{i}_t and \mathbf{g}_t are

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{o}_t} &= \frac{\partial E}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \mathbf{o}_t} \\ &= \frac{\partial E}{\partial \mathbf{h}_t} \odot \tanh(\mathbf{c}_t)\end{aligned}\tag{B.10}$$

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{c}_t} &= \frac{\partial E}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \mathbf{c}_t} \\ &= \frac{\partial E}{\partial \mathbf{h}_t} \odot \mathbf{o}_t \odot (1 - \tanh^2(\mathbf{c}_t))\end{aligned}\tag{B.11}$$

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{f}_t} &= \frac{\partial E}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \mathbf{c}_t} \frac{\partial \mathbf{c}_t}{\partial \mathbf{f}_t} \\ &= \frac{\partial E}{\partial \mathbf{h}_t} \odot \mathbf{o}_t \odot (1 - \tanh^2(\mathbf{c}_t)) \odot \mathbf{c}_{t-1}\end{aligned}\tag{B.12}$$

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{c}_{t-1}} &= \frac{\partial E}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \mathbf{c}_t} \frac{\partial \mathbf{c}_t}{\partial \mathbf{c}_{t-1}} \\ &= \frac{\partial E}{\partial \mathbf{h}_t} \odot \mathbf{o}_t \odot (1 - \tanh^2(\mathbf{c}_t)) \odot \mathbf{f}_t\end{aligned}\tag{B.13}$$

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{i}_t} &= \frac{\partial E}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \mathbf{c}_t} \frac{\partial \mathbf{c}_t}{\partial \mathbf{i}_t} \\ &= \frac{\partial E}{\partial \mathbf{h}_t} \odot \mathbf{o}_t \odot (1 - \tanh^2(\mathbf{c}_t)) \odot \mathbf{g}_t\end{aligned}\tag{B.14}$$

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{g}_t} &= \frac{\partial E}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \mathbf{c}_t} \frac{\partial \mathbf{c}_t}{\partial \mathbf{g}_t} \\ &= \frac{\partial E}{\partial \mathbf{h}_t} \odot \mathbf{o}_t \odot (1 - \tanh^2(\mathbf{c}_t)) \odot \mathbf{i}_t.\end{aligned}\tag{B.15}$$

Now, the derivatives of E w.r.t the weights \mathbf{W}_o , \mathbf{W}_i , \mathbf{W}_g and \mathbf{W}_f are

$$\left. \frac{\partial E}{\partial \mathbf{W}_o} \right|_t = \frac{\partial E}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \mathbf{o}_t} \frac{\partial \mathbf{o}_t}{\partial \mathbf{W}_o} \quad (\text{B.16})$$

$$= \left[\frac{\partial E}{\partial \mathbf{h}_t} \odot \tanh(\mathbf{c}_t) \odot \mathbf{o}_t \odot (1 - \mathbf{o}_t) \right] \mathbf{x}_t^\top \quad (\text{B.17})$$

$$= \Delta_t^o \mathbf{x}_t^\top \quad (\text{B.18})$$

$$\left. \frac{\partial E}{\partial \mathbf{W}_i} \right|_t = \frac{\partial E}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \mathbf{c}_t} \frac{\partial \mathbf{c}_t}{\partial \mathbf{i}_t} \frac{\partial \mathbf{i}_t}{\partial \mathbf{W}_i} \quad (\text{B.19})$$

$$= \left[\frac{\partial E}{\partial \mathbf{h}_t} \odot \mathbf{o}_t \odot (1 - \tanh^2(\mathbf{c}_t)) \odot \mathbf{g}_t \odot \mathbf{i}_t \odot (1 - \mathbf{i}_t) \right] \mathbf{x}_t^\top \quad (\text{B.20})$$

$$= \Delta_t^i \mathbf{x}_t^\top \quad (\text{B.21})$$

$$\left. \frac{\partial E}{\partial \mathbf{W}_g} \right|_t = \frac{\partial E}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \mathbf{c}_t} \frac{\partial \mathbf{c}_t}{\partial \mathbf{g}_t} \frac{\partial \mathbf{g}_t}{\partial \mathbf{W}_g} \quad (\text{B.22})$$

$$= \left[\frac{\partial E}{\partial \mathbf{h}_t} \odot \mathbf{o}_t \odot (1 - \tanh^2(\mathbf{c}_t)) \odot \mathbf{i}_t \odot (1 - \mathbf{g}_t^2) \right] \mathbf{x}_t^\top \quad (\text{B.23})$$

$$= \Delta_t^g \mathbf{x}_t^\top \quad (\text{B.24})$$

$$\left. \frac{\partial E}{\partial \mathbf{W}_f} \right|_t = \frac{\partial E}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \mathbf{c}_t} \frac{\partial \mathbf{c}_t}{\partial \mathbf{f}_t} \frac{\partial \mathbf{f}_t}{\partial \mathbf{W}_f} \quad (\text{B.25})$$

$$= \left[\frac{\partial E}{\partial \mathbf{h}_t} \odot \mathbf{o}_t \odot (1 - \tanh^2(\mathbf{c}_t)) \odot \mathbf{c}_{t-1} \odot \mathbf{f}_t \odot (1 - \mathbf{f}_t) \right] \mathbf{x}_t^\top \quad (\text{B.26})$$

$$= \Delta_t^f \mathbf{x}_t^\top. \quad (\text{B.27})$$

Note that since $\sigma'(\cdot) = \sigma(\cdot)(1 - \sigma(\cdot))$, in the transition between (B.16) and (B.17) it follows that

$$\frac{\partial \mathbf{o}_t}{\partial \mathbf{W}_o} = \sigma'(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1}) \mathbf{x}_t^\top \quad (\text{B.28})$$

$$= \left[\sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1}) \odot (1 - \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1})) \right] \mathbf{x}_t^\top \quad (\text{B.29})$$

$$= [\mathbf{o}_t \odot (1 - \mathbf{o}_t)] \mathbf{x}_t^\top. \quad (\text{B.30})$$

Similarly, the transition between (B.19) and (B.20) follows

$$\frac{\partial \mathbf{i}_t}{\partial \mathbf{W}_i} = \mathbf{i}_t \odot (1 - \mathbf{i}_t). \quad (\text{B.31})$$

At last, the transition between (B.25) and (B.26) follows

$$\frac{\partial \mathbf{f}_t}{\partial \mathbf{W}_f} = \mathbf{f}_t \odot (1 - \mathbf{f}_t). \quad (\text{B.32})$$

Furthemore, since $\tanh'(\cdot) = 1 - \tanh^2(\cdot)$, the transition between (B.22) and (B.23) follows

$$\frac{\partial \mathbf{g}_t}{\partial \mathbf{W}_g} = \tanh'(\mathbf{W}_g \mathbf{x}_t + \mathbf{U}_g \mathbf{h}_{t-1}) \mathbf{x}_t^\top \quad (\text{B.33})$$

$$= (1 - \tanh^2(\mathbf{W}_g \mathbf{x}_t + \mathbf{U}_g \mathbf{h}_{t-1})) \mathbf{x}_t^\top \quad (\text{B.34})$$

$$= (1 - \mathbf{g}_t^2) \mathbf{x}_t^\top, \quad (\text{B.35})$$

which concludes the proof.

B.2 Description of Algorithm 4

The algorithm takes as input the training dataset \mathcal{X} , the training targets \mathcal{Y} , the missing indicators \mathcal{M} , the weights of the encoding layers \mathbf{W}_{enc} , the weights for the inference layers \mathbf{W}_{inf} , the actor π_θ , the critic Q_v , learning rates $\{\alpha, \alpha_\theta, \alpha_v\}$ and training loss function E . Our approach starts by initializing all the parameters \mathbf{W}_{enc} , \mathbf{W}_{inf} , π_θ , Q_v , sampling $\mathbf{x} \in \mathcal{X}$ with the corresponding $\mathbf{m} \in \mathcal{M}$ that will be used for training in the first iteration, obtaining the feature ζ and the prediction $\hat{\mathbf{y}}$, which constitute the initial state as $\mathbf{s} = (\mathbf{x}, \mathbf{m}, \zeta, \hat{\mathbf{y}})$. In each iteration, first the importance is generated from a behavioral policy β that is conditioned on the target policy π_θ , such as the noisy exploration policy proposed in (Lillicrap et al., 2016). Then the encoding layer is trained following (6.4), while the inference layers are trained following the regular gradient descent. After training, the new prediction is obtained following the *updated* weights and its value is assigned to $\hat{\mathbf{y}}$, which is then used to generate the reward following the reward function R . Then the training sample \mathbf{x}' for the next iteration is sampled and the corresponding $\mathbf{m}', \zeta', \hat{\mathbf{y}}'$ are obtained to constitute the next state \mathbf{s}' . Finally, the actor π_θ and critic Q_v are updated following (6.5). We refer to (Lillicrap et al., 2016; Silver et al., 2014) for more details on actor-critic RL.

B.3 Importance vs Attentions

We now illustrate the connections and distinctions between the importance in the GIL and visual attentions (Ba et al., 2014; Mnih et al., 2014; K. Xu et al., 2015), which are commonly used to train CNNs to focus on specific dimensions of the inputs that are most helpful for making predictions in the context of image captioning and multi-object recognition. In visual attentions, an input image is first encoded into a set of vectors $\mathcal{I} = \{\mathbf{I}_1, \dots, \mathbf{I}_L\}$ where each \mathbf{I}_i is a feature vector corresponding to *a specific region* in the image as they are retrieved from lower convolutional layers. Then, the attention $\alpha_{t,i}$ for time step t and region $i \in [1, L]$ is generated following (Bahdanau et al., 2014) as $\alpha_{t,i} = \exp(e_{t,i}) / \sum_{j=1}^L \exp(e_{t,j})$, where $e_{t,i} = f_{att}(\mathbf{I}_i, \mathbf{h}_{t-1})$, f_{att} is usually an MLP (or the so-called attention model) and \mathbf{h}_{t-1} is the hidden state of a recurrent neural network (RNN) (Hochreiter & Schmidhuber, 1997) used for prediction. Then, a weighted average of the feature vectors $\sum_i l_{t,i} \mathbf{I}_i$ is fed into the prediction network for further inference where $l_{t,i}$ is a random variable parametrized by $\alpha_{t,i}$ following $p(l_{t,i} = 1 | l_{j < t}, \mathcal{I}) = \alpha_{t,i}$, and $l_{j < t}$ represents the historical values of $l_{t,i}$ for all $i \in [1, L]$.

Given the definition of the multinoulli variable $l_{t,i}$, the model is not smooth and thus cannot be trained following the regular back-propagation. So the prediction network is trained to maximize the evidence lower bound (ELBO) which updates the prediction network parameter θ using

$$\frac{\partial J}{\partial \theta} \approx \frac{1}{n} \sum_{i=1}^n \left[\frac{\partial \log p(\mathbf{y} | \tilde{l}_i, \mathcal{I})}{\partial \theta} + (\log p(\mathbf{y} | \tilde{l}_i, \mathcal{I}) - b) \frac{\partial \log p(\tilde{l}_i | \mathcal{I})}{\partial \theta} \right], \quad (\text{B.36})$$

where \tilde{l}_i are the Monte-Carlo samples drawn from $p(l_{t,i} | l_{j < t}, \mathcal{I})$ and b represents the performance from a baseline $\mathbb{E}[\log p(\mathbf{y} | \theta_{baseline})]$. It is notable that (B.36) is equivalent to the REINFORCE algorithm in RL (Williams, 1992). Specifically, the search space can be interpreted as an MDP which is usually used to model the environment in RL problems – i.e., the state space is constituted by \mathcal{I} , the RNN hidden state \mathbf{h}_t and the historical visitations $l_{j < t}$; the actions are multinoulli variables $l_{t,i}$; the reward function is defined as the marginal

log-likelihood $\log p(\mathbf{y}|l_{t,i}, \mathcal{I})$; and the policy is characterized by $p(l_{t,i}|l_{j<t}, \mathcal{I})$ (Ba et al., 2014; Mnih et al., 2014; K. Xu et al., 2015).

However, these methods cannot be applied directly to our problem as it requires the features \mathcal{I} to exclusively associate with specific parts of the inputs *spatially*, which is attainable by using convolutional encoders with image inputs but intractable with tabular inputs considered in our case. Instead, our approach overcomes this issue by directly applying importance, generated by RL, into the *gradient space* during back-propagation. Moreover, our method does not require to formulate the learning objective as ELBOs and as a result GIL can adopt any state-of-the-art RL algorithm – not limited to REINFORCE only as in (Ba et al., 2014; Mnih et al., 2014; K. Xu et al., 2015).

B.4 Experimental Details and Additional Experiments

The case studies are run on a work station with three Nvidia Quadro RTX 6000 GPUs with 24GB of memory for each. We use Tensorflow to implement the models and training algorithms. To train the imputation-free prediction models using GIL, we perform a grid search for the model learning rate $\alpha \in \{0.001, 0.0007, 0.0005, 0.0003, 0.0001, 0.00005, 0.00001\}$, the exponential decay step for α is selected from $\{1000, 750, 500\}$ and the exponential decay rate for α is selected from $\{0.95, 0.9, 0.85, 0.8\}$. The actor π_θ and critic Q_v in the GIL (i.e., Alg. 1) are trained using deep deterministic policy gradient (DDPG) (Lillicrap et al., 2016) where the discounting factor $\gamma = 0.99$. We choose the behavioural policy β in GIL as

$$\beta(\mathbf{s}) = \begin{cases} \pi_\theta(\mathbf{s}) & \text{with probability } p_1, \\ \text{missing indicator } \mathbf{m} & \text{with probability } p_2, \\ \text{random action} & \text{with probability } p_3. \end{cases} \quad (\text{B.37})$$

The learning rates of the actor α_π and the critic α_Q are selected by performing grid search from $\{0.0005, 0.0001, 0.00005, 0.00001\}$ and $\{0.001, 0.0005, 0.0001\}$ respectively. To train the imputation-free models using GIL-H/GIL-D, we follow the same grid search for α along with its decay steps and decay rates. From the implementation perspective, we replace the missing entries, in the inputs \mathbf{x} , with a placeholder value before feeding them into the model. This can avoid value errors thrown by Tensorflow if the input vectors contain NaN's. However, note these values will not be used to update the parameters. For the state-of-the-art baselines – GAIN¹, MIWAE², GPVAE³ and BRITS⁴ – we use the implementations published on the Github by the authors. Adam optimizer is used to train all the prediction models for baselines, or to compute $(\partial E / \partial \mathbf{W})_{\text{SGD}}$ for GIL, GIL-H and GIL-D. All the models are trained using a batch size of 128. The details of selecting the other hyper-parameters of each case study can be found in the corresponding sub-section below.

¹ <https://github.com/jsyoon0823/GAIN>

² <https://github.com/pamattei/miwae>

³ <https://github.com/ratschlab/GP-VAE>

⁴ <https://github.com/caow13/BRITS>

B.4.1 MIMIC-III

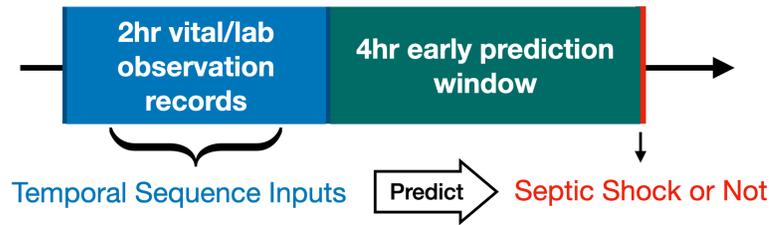


FIGURE B.1: Graphical depiction of the 2-hour observation window and 4-hour early prediction window (EPW) considered in this case study.

B.4.1.1 Dataset Formulation

The MIMIC-III⁵ contains EHRs obtained from roughly 40,000 patients who stayed in the ICUs in the Beth Israel Deaconess Medical Center between 2001 and 2012 (Johnson et al., 2016) and septic shock is a severe type of sepsis that results in above 40% mortality rate. Figure B.1 illustrates the 2-hour observation window, 4-hour early prediction window (EPW) and the relation between inputs and targets for predictions. Note that each work related to septic shock predictions adopts a slightly different strategy in terms of selecting lab/vital attributes and the lengths of the observation and EPW (Darwiche & Mukherjee, 2018; Fleuren et al., 2020; Khoshnevisan et al., 2020; R. Liu et al., 2019; Mao et al., 2018; Sheetrit et al., 2017; Yee et al., 2019). In this work, the 4-hour EPW allows including sufficient number of subjects which can ensure statistical significance of the results, while the smaller observation window keeps the task challenging. To formulate the training and testing dataset, we first selected 14 commonly used attributes for predictions as suggested in previous works (Fleuren et al., 2020; Khoshnevisan et al., 2020; Sheetrit et al., 2017), which are constituted by vital signs including temperature, respiratory rate, heart rate, systolic blood pressure, mean arterial pressure, peripheral oxygen saturation (or SpO₂), fraction of inspired oxygen (or FIO₂), and lab tests including white blood cell count, serum lactate level, platelet count, creatinine, bilirubin, bandemia of white blood cells, blood urea

⁵ Data acquired from <https://mimic.mit.edu>. Access to this dataset follows the MIT-CLP license.

nitrogen. To form the septic shock cohort, we first identify the patients who were diagnosed with sepsis by indexing with the International Classification of Diseases 9 (ICD-9) code 995.91 and 995.92. Then the patients with septic shock history are selected using ICD-9 code 785.52 with their septic shock onset time determined following the third international consensus for sepsis and septic shock (Sepsis-3) standard (Singer et al., 2016). Finally, we remove the patients whose septic shock onset time was less than 6 hours after admission to the ICUs since the data is not enough to be formulated as sequences pertaining to the 2-hour observation and 4-hour prediction window. Consequently, a total of 1,083 septic shock patients are identified. To form the non-shock (or control) cohort, first 1,083 patients are randomly sampled from all admissions who have at least 2 hours of records excluding the ones who are in the septic shock cohort, then a 2-hour time frame is randomly selected to be used as the observation window. All patients selected following the above procedure are split into 8:2 to formulate the training and testing datasets, which are referred to as the varied-length (Var-l.) sequences in Section 6.3.2. In the test set, the number of subjects associated with positive and negative labels, respectively, are selected to be equivalent.

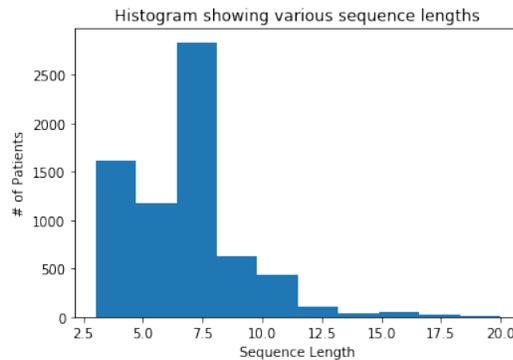


FIGURE B.2: Histogram of the sequences lengths of the MIMIC-III dataset (with maximum length truncated at 20). It can be observed that most of the sequences have length ≤ 8 .

B.4.1.2 Formulation of Fix-l. Sequences

To formulate the fixed-length (Fix-l.) sequences, we follow this protocol – if sequence length is less than 8, we pad it with the latest recording available until the length reaches 8, otherwise we truncate the length to 8 by discarding the data from that point on-wards.

We specifically choose this threshold because most of the sequences have length ≤ 8 , as shown in Figure B.2.

B.4.1.3 Type of Missingness

The missing data in this dataset can be considered as a mixture of MCAR, MAR and MNAR. Specifically, recordings from human mistakes or malfunctioned equipment manifest as MCAR, the dependency across vital signs and lab results give rise to MAR (*e.g.*, specific lab tests are ordered only if abnormalities found in the related vital readings or other lab results), and the mismatch among the sampling frequency of some periodically recorded attributes such as temperature (obtained hourly) and blood test (conducted daily) can be considered as MNAR (R. J. Little & Rubin, 2019; Mamandipoor et al., 2019).

B.4.1.4 Training Details

To train the prediction models for both GIL and baselines, we use maximum training steps of 2,000 and 4,000 for varied-length and fixed-length sequences respectively. For imputation of the missing values, we train the imputation method MIWAE using the learning rates $\{0.001, 0.0001\}$ with other hyper-parameters provided by the authors in the code. To train GAIN, we use the default learning rates provided by the authors to train the generator and discriminator. GAIN contains one hyper-parameter that balances between the two losses \mathcal{L}_G and \mathcal{L}_M defined in (Yoon et al., 2018) which is selected from $\{0.1, 1, 10, 100\}$. To train GP-VAE on the Fix-l. case, the set of hyper-parameters we use contain $latent_dim = \{6, 12, 35\}$, $encoder_size = 256, 256$, $decoder_size = 256, 256, 256$, $window_size = \{3, 4, 6\}$, $beta = \{0.2, 0.5, 0.8\}$, $sigma = 1.005$, $length_scale = \{2, 4, 7\}$. To train BRITS we used the recommended $impute_weight = 0.3$, $label_weight = 1.0$. The term $D(\zeta^+, \zeta^-)$ included in the reward function for GIL-D is defined as in the follows. Assume that in each training epoch $b/2$ inputs with label 0 and $b/2$ inputs with label 1 are sampled from the dataset, where b is the batch size. We use $F^+ = (\zeta_1^+, \dots, \zeta_{b/2}^+) \in \mathbb{R}^{b/2 \times e}$ to denote the features associated with positive labels (*i.e.*, 1) in the current batch and $F^- = (\zeta_1^-, \dots, \zeta_{b/2}^-) \in \mathbb{R}^{b/2 \times e}$

are , where e is the dimension of each individual feature ζ . Then we define D as

$$\begin{aligned} &MSE(F^+[0 : b/4], F^-[0 : b/4]) + MSE(F^+[b/4 : b/2], F^-[b/4 : b/2]) \\ &- MSE(F^+[0 : b/4], F^+[b/4 : b/2]) - MSE(F^-[0 : b/4], F^-[b/4 : b/2]), \end{aligned} \quad (\text{B.38})$$

where the slicing index follows the syntax from Python, *e.g.*, $F^+[0 : b/4]$ corresponds to the 0-th to $(b/4 - 1)$ -th rows of F^+ .

B.4.2 Ophthalmic Data

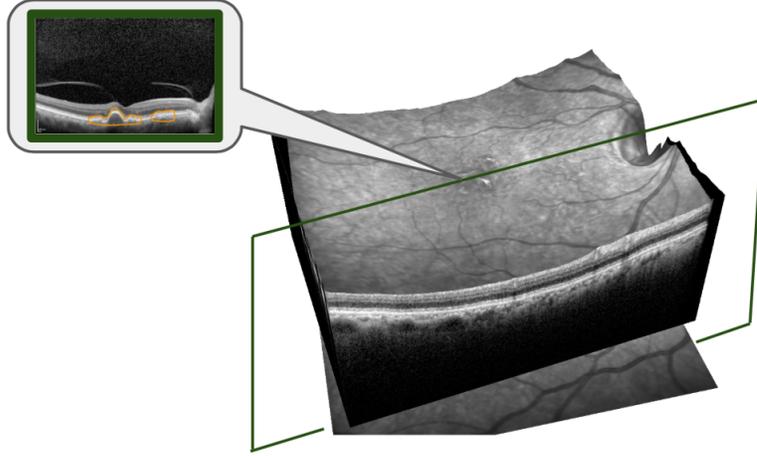


FIGURE B.3: Example of a 3D OCT volume scan and a 2D OCT image slice from the volume scan.

B.4.2.1 Dataset Introduction

This dataset is originally constituted by OCT images, CFP images and patient EHRs including demographic information and test results related to diabetes. A total of 1,148 subjects are included in this dataset. OCT refers to the two- or three-dimensional images capturing the retinal architectures of the eyes that are scanned by low-coherence lights. The 3D OCT image is usually called the volume scan and it is constituted by 61 or 101 2D image slices depending on the spec of the scanner. Examples of 3D and 2D OCT scans are shown in Figure B.3. In this experiment the volume scans we use contain the 61 slices, while we only consider the center slice (31th) 2D image. The CFP images are the color

images captured by a fundus camera showing the condition of the interior surface of the eye, where an example is shown in Figure B.4.

B.4.2.2 Dataset Formulation

To formulate the data into tabular inputs, the 2D OCT and CFP retinal images are first fed in to two inception-v3 (Szegedy et al., 2016b) CNNs, pre-trained with similar images provided by (Kermary, Goldbaum, Cai, Valentim, et al., 2018) and (“Kaggle Diabetic Retinopathy Detection Competition”, n.d.) respectively, and the image feature vectors with dimension of 2,048 each are output by the global average pooling layer. Then the patient EHR information include age (integer), sex (boolean), length of diabetic history (integer), A1C result (integer), which measures blood sugar levels, and if insulin has been used (boolean) constitute a 5-dimensional vector that is concatenated to the end of the OCT and CFP feature vectors. We split all the subjects into a training cohort and a testing cohort following a ratio of 9:1. In the test set, the number of subjects associated with positive and negative labels, respectively, are selected to be equivalent. The raw images from the training cohort are augmented through cropping and rotation before feeding into inception-v3. All the data from the testing cohort are not augmented. Figure B.4 illustrates how the training and testing inputs are formulated, as well as the input-output relation of the prediction models.

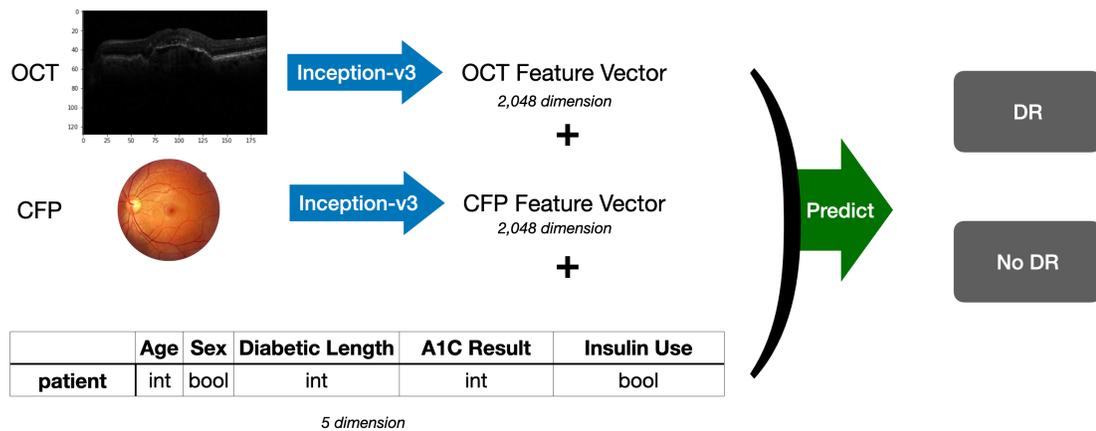


FIGURE B.4: Diagram showing the pipeline of this experiment.

B.4.2.3 Training Details

To train the prediction models for both GIL and baselines, we use maximum training steps of 2,000. For imputation of the missing values, MIWAE is trained using the learning rates $\{0.001, 0.0001\}$ with other hyper-parameters provided by the authors in the code. GAIN is trained using hyper-parameters selected from $\{0.1, 1, 10, 100\}$. GP-VAE is trained by applying the arguments $latent_dim = 256, encoder_size = 256, 256, decoder_size = 256, 256, 256, window_size = 3, beta = 0.8, sigma = 1$ to its code published in Github (<https://github.com/ratschlab/GP-VAE>). MF, EM and kNN are not included in this experiment due to the high dimensionality of the inputs which makes these algorithms very computational inefficient as imputation results cannot be produced within days.

B.4.3 MNIST

B.4.3.1 Training Details

To train the prediction models for both GIL and baselines, we use maximum training steps of 1,0000. For imputation of the missing values, MIWAE is trained using the learning rates $\{0.001, 0.0001\}$ with other hyper-parameters provided by the authors in the code. GAIN is trained using hyper-parameters selected from $\{0.1, 1, 10, 100\}$. GP-VAE is trained by applying the arguments $latent_dim = 256, encoder_size = 256, 256, decoder_size = 256, 256, 256, window_size = 3, beta = 0.8, sigma = 1$ to its code published in Github (<https://github.com/ratschlab/GP-VAE>). MF, EM and kNN are not included in this experiment due to the high dimensionality of the inputs which makes these algorithms very computational inefficient as imputation results cannot be produced within days.

B.4.3.2 Additional Results

We also trained GIL-D on the MCAR version of MNIST dataset and the performance are shown in Table B.1 below.

Table B.1: Accuracy for GIL-D on the MCAR version of MNIST dataset. Standard deviations are in subscripts ($\times 10^{-3}$).

Missing Rate	50%	70%	90%
Acc.	96.29 _{0.09}	93.35 _{0.9}	78.47 _{3.9}

Table B.2: Accuracy, AUC and average precision (AP) obtained from the Physionet dataset

	GIL	GIL-H	GAIN	GP-VAE	MIWAE	BRITS	Mean	Zero
Acc.	87.45	87.38	86.23	86.85	87.12	86.20	86.87	87.15
AUC	82.20	82.00	78.02	82.21	83.47	81.30	82.57	81.14
AP	49.19	48.64	39.89	47.05	49.31	43.57	48.30	47.37

B.4.4 Physionet

Other than MIMIC-III, we also tested on a smaller scaled ICU time-series from 2012 Physionet challenge (Silva et al., 2012) which contain data obtained from 12,000 patients. We use the data pre-processed and open-sourced by (Fortuin et al., 2020).⁶ For each patient the values of 35 different attributes (*e.g.*, blood pressure, temperature) are recorded over a 48-hour window. As a result, the data for each patient can be formulated into a matrix in $\mathbb{R}^{48 \times 35}$, *i.e.*, the sequence length across patients are the same. This dataset has an overall 78.5% missing rate and a binary label is assigned to each patient where 87% of them are 0’s and 13% are 1’s. Therefore we include average precision (AP) which is calculated from the precision-recall curve as an additional metric to evaluate the performance toward imbalanced labels. For this dataset, we consider a 1024-unit LSTM layer for encoding and a dense output layer for inference. It can be observed from Table B.2 that although our method achieves the highest accuracy and 2-nd highest AP, while most methods perform very close to mean- and zero-imputation. This could be caused by the its simple structure, as all the sequences share the same time horizon, which significantly reduces the difficulties for the classification task. Moreover, the highly imbalanced labels and the small population

⁶ <https://github.com/ratschlab/GP-VAE>

result in the performance to be hardly distinguishable across different methods. And this is the reason that we focus on the MIMIC-III dataset to study the strengths and shortcomings of the methods.

B.5 Proof for Proposition 2

Here we prove a generalized version of Proposition 2 as stated in the following proposition, which shows that the gradient of the loss function E w.r.t. any layer in the MLP can be written in the outer product format.

Proposition 4. *Given an MLP as in (6.1) and a smooth loss function $E(\hat{\mathbf{y}}|\mathbf{y})$, the gradients for any hidden layer \mathbf{W}_i , $i \in [1, k]$ can be represented as*

$$\frac{\partial E}{\partial \mathbf{W}_i} = \Delta_i \mathbf{q}_{i-1}^\top, \quad (\text{B.39})$$

where $\mathbf{q}_{i-1} = \phi_{i-1}(\mathbf{W}_{i-1} \mathbf{q}_{i-2})$ is the output from the $i-1$ -th layer with $\mathbf{q}_0 = \mathbf{x}$, and $\Delta_i = \mathbf{W}_{i+1}^\top \Delta_{i+1} \odot \phi'_i(\mathbf{W}_i \mathbf{q}_{i-1})$ with $\Delta_{\text{out}} = \frac{\partial E}{\partial \hat{\mathbf{y}}} \odot \phi'_{\text{out}}(\mathbf{W}_{\text{out}} \mathbf{q}_{2k})$ and \odot denotes the element-wise (Hadamard) product.

Now we prove Proposition 4.

We consider the MLP characterized by

$$\hat{\mathbf{y}} = \phi_{\text{out}}(\mathbf{W}_{\text{out}} \phi_k(\mathbf{W}_k \dots \phi_2(\mathbf{W}_2 \phi_1(\mathbf{W}_1 \mathbf{x})))) \quad (\text{B.40})$$

where $\mathbf{x} \in \mathbb{R}^d$ represents the input to the model, \mathbf{W}_i is the weight matrix for the i -th hidden layer, and ϕ_i is the activation function of the i -th hidden layer.

We start with deriving the derivatives for the output layer

$$\frac{\partial E}{\partial \mathbf{W}_{\text{out}}} = \frac{\partial E}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{W}_{\text{out}}} \quad (\text{B.41})$$

$$= \left[\frac{\partial E}{\partial \hat{\mathbf{y}}} \odot \phi'_{\text{out}}(\mathbf{W}_{\text{out}} \mathbf{q}_k) \right] \cdot \mathbf{q}_k^\top \quad (\text{B.42})$$

$$= \Delta_{\text{out}} \cdot \mathbf{q}_k^\top, \quad (\text{B.43})$$

where \mathbf{q}_k is the output from the k -th hidden layer.

Now we show the derivative of E w.r.t. the k -th hidden layer

$$\frac{\partial E}{\partial \mathbf{W}_k} = \frac{\partial E}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{q}_k} \frac{\partial \mathbf{q}_k}{\partial \mathbf{W}_k} \quad (\text{B.44})$$

$$= \mathbf{W}_{out}^\top \left[\frac{\partial E}{\partial \hat{\mathbf{y}}} \odot \phi'_{out}(\mathbf{W}_{out} \mathbf{q}_k) \right] \cdot \frac{\partial \mathbf{q}_k}{\partial \mathbf{W}_k} \quad (\text{B.45})$$

$$= \mathbf{W}_{out}^\top \Delta_{out} \cdot \frac{\partial \mathbf{q}_k}{\partial \mathbf{W}_k} \quad (\text{B.46})$$

$$= [(\mathbf{W}_{out}^\top \Delta_{out}) \odot \phi'_k(\mathbf{W}_k \mathbf{q}_{k-1})] \cdot \mathbf{q}_{k-1}^\top \quad (\text{B.47})$$

$$= \Delta_k \cdot \mathbf{q}_{k-1}^\top \quad (\text{B.48})$$

We now prove Proposition 4 by induction. First assume that for $j \in [2, k]$, the derivative of E w.r.t. the j -th hidden layer is

$$\frac{\partial E}{\partial \mathbf{W}_j} = [(\mathbf{W}_{j+1}^\top \Delta_{j+1}) \odot \phi'_j(\mathbf{W}_j \mathbf{q}_{j-1})] \cdot \mathbf{q}_{j-1}^\top \quad (\text{B.49})$$

$$= \Delta_j \cdot \mathbf{q}_{j-1}^\top. \quad (\text{B.50})$$

Moreover, let us assume that

$$\frac{\partial E}{\partial \mathbf{q}_j} = \mathbf{W}_{j+1}^\top \left[\frac{\partial E}{\partial \mathbf{q}_{j+1}} \odot \phi'_{j+1}(\mathbf{W}_{j+1} \mathbf{q}_j) \right] \quad (\text{B.51})$$

$$= \mathbf{W}_{j+1}^\top \Delta_{j+1}. \quad (\text{B.52})$$

Now we need to show that for $j-1$, it holds that

$$\frac{\partial E}{\partial \mathbf{W}_{j-1}} = [(\mathbf{W}_j^\top \Delta_j) \odot \phi'_{j-1}(\mathbf{W}_{j-1} \mathbf{q}_{j-2})] \cdot \mathbf{q}_{j-2}^\top, \quad (\text{B.53})$$

$$\frac{\partial E}{\partial \mathbf{q}_{j-1}} = \mathbf{W}_j^\top \left[\frac{\partial E}{\partial \mathbf{q}_j} \odot \phi'_j(\mathbf{W}_j \mathbf{q}_{j-1}) \right] \quad (\text{B.54})$$

$$= \mathbf{W}_j^\top \Delta_j. \quad (\text{B.55})$$

We first prove that (B.55) holds, i.e.,

$$\frac{\partial E}{\partial \mathbf{q}_{j-1}} = \frac{\partial E}{\partial \mathbf{q}_j} \frac{\partial \mathbf{q}_j}{\partial \mathbf{q}_{j-1}} \quad (\text{B.56})$$

$$= \mathbf{W}_{j+1}^\top \Delta_{j+1} \frac{\partial \mathbf{q}_j}{\partial \mathbf{q}_{j-1}} \quad (\text{B.57})$$

$$= \mathbf{W}_j^\top [\mathbf{W}_{j+1}^\top \Delta_{j+1} \odot \phi'_j(\mathbf{W}_j \mathbf{q}_{j-1})] \quad (\text{B.58})$$

$$= \mathbf{W}_j^\top \Delta_j, \quad (\text{B.59})$$

which is equivalent to (B.55).

To prove that (B.53) holds, we start with

$$\frac{\partial E}{\partial \mathbf{W}_{j-1}} = \frac{\partial E}{\partial \mathbf{q}_j} \frac{\partial \mathbf{q}_j}{\partial \mathbf{q}_{j-1}} \frac{\partial \mathbf{q}_{j-1}}{\partial \mathbf{W}_{j-1}} \quad (\text{B.60})$$

$$= \mathbf{W}_{j+1}^\top \Delta_{j+1} \frac{\partial \mathbf{q}_j}{\partial \mathbf{q}_{j-1}} \frac{\partial \mathbf{q}_{j-1}}{\partial \mathbf{W}_{j-1}} \quad (\text{B.61})$$

$$= \mathbf{W}_j^\top [\mathbf{W}_{j+1}^\top \Delta_{j+1} \odot \phi'_j(\mathbf{W}_j \mathbf{q}_{j-1})] \frac{\partial \mathbf{q}_{j-1}}{\partial \mathbf{W}_{j-1}} \quad (\text{B.62})$$

$$= \mathbf{W}_j^\top \Delta_j \frac{\partial \mathbf{q}_{j-1}}{\partial \mathbf{W}_{j-1}} \quad (\text{B.63})$$

$$= [(\mathbf{W}_j^\top \Delta_j) \odot \phi'_{j-1}(\mathbf{W}_{j-1} \mathbf{q}_{j-2})] \cdot \mathbf{q}_{j-2}^\top, \quad (\text{B.64})$$

which is equivalent to (B.53).

B.6 Additional Ablation Study

In this section, we compare GIL against an ablation baseline that applies the importance directly to the inputs \mathbf{x} , instead of the gradient space. Specifically, suppose a differential parametric function $h_\psi(\mathbf{x})$, parameterized by ψ , is multiplied with \mathbf{x} element-wise. Then, as opposed to (6.4), the gradient updates w.r.t. the encoding layers can be formulated as

$$\mathbf{W}'_{enc} \leftarrow \mathbf{W}_{enc} - \alpha \Delta \cdot (\mathbf{x}^\top \odot h_\psi(\mathbf{x})^\top). \quad (\text{B.65})$$

Consequently, all model elements (*i.e.*, h_ψ , \mathbf{W}_{enc} and \mathbf{W}_{inf}) could be trained by gradient descent, without introducing RL policies.

Table B.3: Performance of the ablation model over the ophthalmic and MNIST datasets.

	Ophthalmic		MNIST	
	25% M.R.	35% M.R.	70% M.R.	90% M.R.
Acc.	84.50 _{1.49}	80.99 _{1.09}	93.22 _{0.6e-03}	78.3 _{1.8e-03}
AUC	89.78 _{1.68}	87.26 _{2.53}	-	-

In Table B.3 it shows the performance of the model described above over the ophthalmic dataset, as well as MNIST digits with 70% and 90% missing rate (M.R.). Specifically, $h_\psi(\mathbf{x})$ is captured by a neural network with the same architecture as the policy π_θ in GIL. The training and testing are conducted following the same convention (*e.g.*, random seeds that determine the missing entries and hyper-parameter search) as introduced in Sections 6.3.3, 6.3.4 and Appendix B.4.2, B.4.3. It can be observed that the ablation baseline attained similar performance as to GIL-H and zero imputation. The reason that leads to these results would be that the gradients for training $h_\psi(\mathbf{x})$ still follow the outer product format $\Delta \cdot \mathbf{x}^\top$ which are left to be accounted for. Specifically, our method is built on top of the idea, and the experiments in Section 6.3 also support that if part of the inputs are missing they may not provide sufficient information to train the prediction models, given the outer product format of the gradients. Similarly, the gradients for $h_\psi(\mathbf{x})$ also follow this format and $h_\psi(\mathbf{x})$ may not be properly trained directly on incomplete inputs \mathbf{x} , which could in general limit the overall performance of the prediction model.

B.6.1 Additional rationales for using RL to obtain gradient importance in GIL.

The ablation baseline above is closely related to visual attention (VA) models as discussed in Appendix B.3, and could be seen as a preliminary version of them where $h_\psi(\mathbf{x})$ is used to re-weight elements in the inputs \mathbf{x} . VA techniques benefit from the fact that features learned by CNNs are spatially correlated to the inputs, so the attentions could be directly applied to learned features. To achieve this, VA formulates the objective as an ELBO which is shown equivalent to the objective of a basic policy gradient RL algorithm, REINFORCE (Mnih et al., 2014). However, it was not clear how such VA methods could be adapted to the problem we consider, as the features learned by MLPs or LSTMs are not spatially correlated with inputs.

B.6.2 More on importance versus attentions.

The other baseline, BRITS (Cao et al., 2018), uses bidirectional RNN to process time-series, with attentions applied to the hidden states of RNNs, followed by optimizing over imputation and prediction objectives jointly during training. BRITS is considered as the state-of-the-art method that could predict with incomplete time-series inputs *end-to-end*. However, we showed that BRITS is outperformed by our method on the MIMIC dataset; see Table 6.1. The reason would be that BRITS requires masking off part of the observed inputs to constitute the imputation objective; thus, the information provided for model training is even more limited given the intrinsic $> 70\%$ M.R. of MIMIC.

Appendix C. Automated Identification of Referable Retinal Pathology in Teleophthalmology Setting

This chapter contains the appendices pertaining to Chapter 7.

C.1 Label Consensus Mechanism (LCM)

We specifically designed the LCM as follows: (a) the final outcome is determined as RPN if both modalities are RPN, or one modality is identified as uninterpretable while the other is RPN; (b) the final label is assigned RPP when at least one modality is RPP or both modalities are uninterpretable. Table C.1 listed all the combinations that constitute RPN and RPP cases.

C.2 Dataset Augmentation and Pre-Processing

We augmented the training set by creating nominal eyes. For each eye labelled as normal, according to the LCM illustrated in Appendix C.1, two more nominal eyes were generated by flipping corresponding CFP over x-axis and rotating the associated OCT image randomly between $[-15, 15]$ degrees, and flipping both CFP and OCT images over y-axis. For the eyes that were diagnosed as abnormal, 13 more nominal samples were generated, where CFP samples were flipped over both x- and y-axis and OCT scans were flipped over y-axis, randomly rotated between $[-15, 15]$ degrees 8 times, and randomly cropped 4 times with the resulting size of 400×550 pixels. Then the loose pairing mechanism as introduced by Wang et al.¹¹ was applied, which basically couples the OCT and CFP images randomly selected from the normal and abnormal eye cohorts to constitute nominal eyes even if the two modalities were not associated with the same eye. This method has been proven to successfully augment small datasets (Vaghefi et al., 2020), or in our case the underrepresented group (i.e. the RPP samples). Then, all the OCT images were denoised using a 2D-gaussian filter with kernel size 7×7 and standard deviation $\sigma=1.5$ before they were used to train the CNN model. For the testing dataset, no images were augmented (e.g., by rotation, cropping etc.) to be consistent with the real world screening results, but all the OCT images were processed using Gaussian blur. As a result, the

Table C.1: Label consensus between outcomes of two modalities.

Final Outcome	CFP			OCT		
	RPN	RPP	RPPP	RPN	RPP	RPPP
Retinal Pathology Negative (RPN)	-	-	-	N	U	P
Retinal Pathology Positive (RPP)	U	-	P	N	-	U
Retinal Pathology Potentially Positive (RPPP)	P	U	-	-	-	-

augmented training dataset contained 4939 OCT images (3189 RPN, 1736 RPP and 14 uninterpretable) and 4939 CFP images (2839 RPN, 1302 RPP and 798 uninterpretable), which in total constituted 4939 nominal eyes (2601 RPN and 2338 RPP). The statistics are shown in Tables 7.1 and 7.2.

C.3 Mathematical Illustration of the AGD Algorithm and Transfer Learning

In this section, we introduce the mathematical illustration of the AGD algorithm and transfer learning.

C.3.1 The AGD Algorithm

By employing binary cross-entropy loss functions, we first defined the following loss functions for θ_1 , θ_2 , and θ_3 respectively:

$$J_1(\theta_1) = \frac{1}{N} \sum_{k=1}^N \mathbf{1}(O_k = \text{Interpretable}) [y_k \cdot \log(p_k + (1 - y_k) \cdot \log((1 - p_k))), \quad (\text{C.1})$$

$$J_2(\theta_2) = \frac{1}{N} \sum_{k=1}^N \mathbf{1}(C_k = \text{Interpretable}) [y_k \cdot \log(p_k + (1 - y_k) \cdot \log((1 - p_k))), \quad (\text{C.2})$$

$$J_3(\theta_3) = \frac{1}{N} \sum_{k=1}^N [y_k \cdot \log(p_k + (1 - y_k) \cdot \log((1 - p_k))]; \quad (\text{C.3})$$

here, N is the total number of images in the training batch, y_k is the ground-truth label of the OCT-CFP pair (O_k, C_k) , $\mathbf{1}$ is the indicator function, and p_k is prediction output (i.e., the probability of being RPP) from the CNN model. At last, in each training iteration

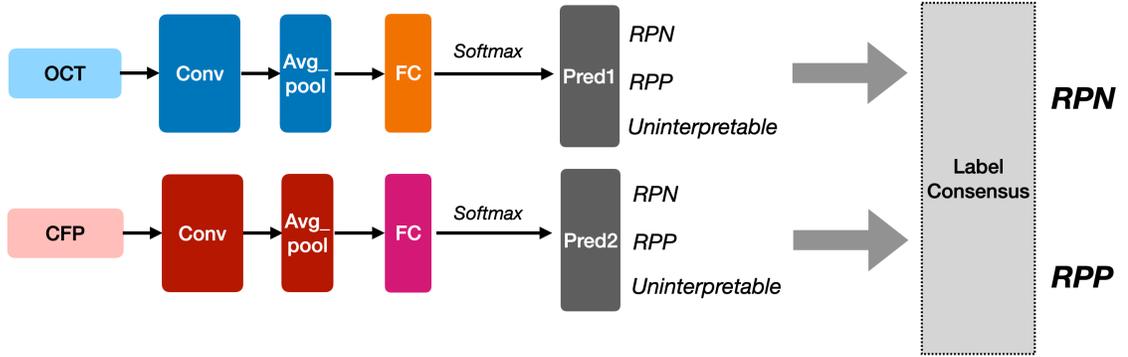


FIGURE C.1: Overview of Baseline A. Each modality was processed with convolutional blocks, followed by a fully connected layer to generate the corresponding predictions. Then the label consensus mechanism was used to determine the final outputs.

we configured the gradient descent solver (e.g., stochastic gradient descent, Adam, etc.) to minimize $J_1(\theta_1)$, $J_2(\theta_2)$, and $J_3(\theta_3)$ alternately to update θ_1 , θ_2 , and θ_3 respectively.

C.3.2 Transfer Learning

For transfer learning, given that two transfer learning datasets did not contain any uninterpretable images, we pre-trained the network by minimizing the binary cross-entropy loss – i.e.,

$$J_{\text{pretrain}}(\theta_1, \theta_2, \theta_3) = \frac{1}{N} \sum_{k=1}^N [y_k \cdot \log(p_k) + (1 - y_k) \cdot \log(1 - p_k)]. \quad (\text{C.4})$$

C.4 Intuition of Designing Baseline A and B and Their Implementation Details

Unlike our model that could handle the uninterpretable inputs internally and only classify the pairs of OCT and CFP inputs as RPN and RPP, the existing methods could not consider the uninterpretable inputs during the training process. However, they could still be used to detect uninterpretable images by directly classifying them into an additional category (i.e., uninterpretable) besides RPN and RPP. Therefore, Baseline A and B were designed to evaluate the classification performance where the uninterpretable images were directly classified into a third-class besides RPN and RPP.

Baseline A was designed to evaluate the classification performance when two single-

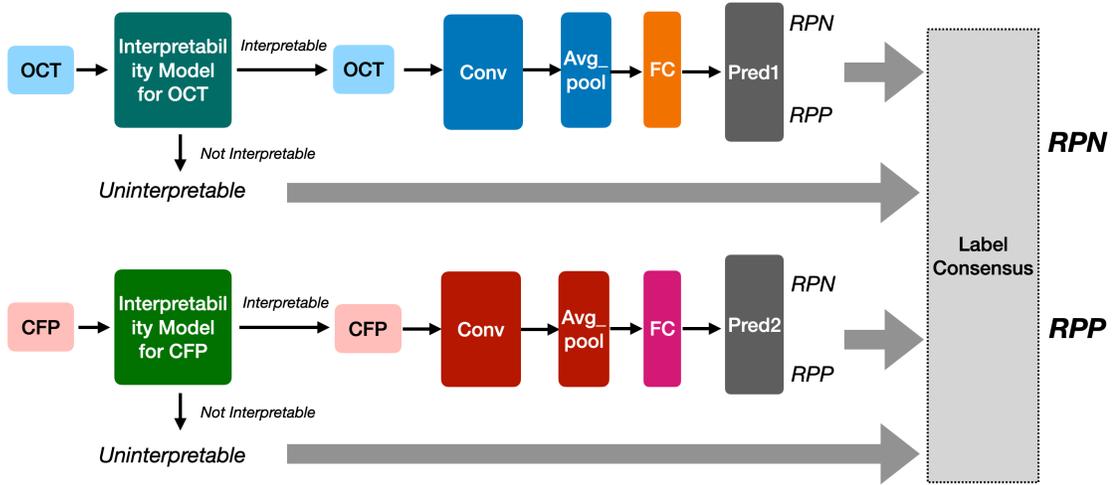


FIGURE C.2: Overview of Baseline B. Each modality was fed into a model that classifies the interpretability. Then, if it is determined as interpretable, subsequent convolutional layers and fully connected layers were used to classify the input as normal/abnormal. Otherwise, the modality was classified as uninterpretable.

modal CNNs were employed to classify the corresponding input modality as RPN, RPP, and uninterpretable. The pipeline is shown in Figure C.1. Specifically, each modality was processed with convolutional blocks, followed by a fully connected layer to generate the corresponding predictions. Then the LCM was used to determine the final outputs. For fair comparison, the architecture of the convolutional layers was configured to use the same layout as in our approach – inception-v3 (Szegedy et al., 2016a).

Baseline B was designed to show the prediction performance when each single-modal CNN model only classified the inputs as RPN and RPP, while the interpretability was determined separately. The overview of this baseline is shown in Figure C.2. Specifically, each imaging modality was fed into the model. Then if an image was determined as interpretable, subsequent convolutional layers and fully connected layers were used to classify the input as RPN/RPP. Otherwise, the image was classified as uninterpretable. Finally, the LCM was used to couple the individual results from each image modality, in order to produce a final prediction.

Bibliography

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., ... Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265–283. <https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf>
- Abeyruwan, S. W., Graesser, L., D'Ambrosio, D. B., Singh, A., Shankar, A., Bewley, A., Jain, D., Choromanski, K. M., & Sanketi, P. R. (2023). I-sim2real: Reinforcement learning of robotic policies in tight human-robot interaction loops. *Conference on Robot Learning*, 212–224.
- Anderson, J. R., Boyle, C. F., & Reiser, B. J. (1985). Intelligent tutoring systems. *Science*, 228(4698), 456–462.
- Arjona-Medina, J. A., Gillhofer, M., Widrich, M., Unterthiner, T., Brandstetter, J., & Hochreiter, S. (2019). Rudder: Return decomposition for delayed rewards. *Advances in Neural Information Processing Systems*, 32.
- Arlotti, M., Rosa, M., Marceglia, S., Barbieri, S., & Priori, A. (2016). The adaptive deep brain stimulation challenge. *Parkinsonism & related disorders*, 28, 12–17.
- Arlotti, M., Rossi, L., Rosa, M., Marceglia, S., & Priori, A. (2016). An external portable device for adaptive deep brain stimulation (adbs) clinical research in advanced parkinson's disease. *Medical engineering & physics*, 38(5), 498–505.
- Azur, M. J., Stuart, E. A., Frangakis, C., & Leaf, P. J. (2011). Multiple imputation by chained equations: What is it and how does it work? *International journal of methods in psychiatric research*, 20(1), 40–49.
- Ba, J., Mnih, V., & Kavukcuoglu, K. (2014). Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*.
- Bagnall, A., Lines, J., Bostrom, A., Large, J., & Keogh, E. (2017). The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31, 606–660.
- Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Bashir, F., & Wei, H.-L. (2018). Handling missing data in multivariate time series using a vector autoregressive model-imputation (var-im) algorithm. *Neurocomputing*, 276, 23–30.

- Benabid, A. L. (2003). Deep brain stimulation for parkinson's disease. *Current opinion in neurobiology*, 13(6), 696–706.
- Bennett, T. J. (2009). Maximizing quality in ophthalmic digital imaging. *J. Ophthalmic Photogr*, 31(1), 32–39.
- Beric, A., Kelly, P. J., Rezai, A., Sterio, D., Mogilner, A., Zonenshayn, M., & Kopell, B. (2001). Complications of deep brain stimulation surgery. *Stereotactic and functional neurosurgery*, 77(1-4), 73–78.
- Beudel, M., & Brown, P. (2016). Adaptive deep brain stimulation in parkinson's disease. *Parkinsonism & related disorders*, 22, S123–S126.
- Bishop, C. (2006). *Pattern recognition and machine learning*. Springer.
- Brown, P., Oliviero, A., Mazzone, P., Insola, A., Tonali, P., & Di Lazzaro, V. (2001). Dopamine dependency of oscillations between subthalamic nucleus and pallidum in parkinson's disease. *Journal of Neuroscience*, 21(3), 1033–1038.
- Butt, A. H., Rovini, E., Dolciotti, C., De Petris, G., Bongioanni, P., Carboncini, M., & Cavallo, F. (2018). Objective and automatic classification of parkinson disease with leap motion controller. *Biomedical engineering*, 17(1), 1–21.
- Calvert, J. S., Price, D. A., Chettipally, U. K., Barton, C. W., Feldman, M. D., Hoffman, J. L., Jay, M., & Das, R. (2016). A computational approach to early sepsis detection. *Computers in biology and medicine*, 74, 69–73.
- Cao, W., Wang, D., Li, J., Zhou, H., Li, L., & Li, Y. (2018). Brits: Bidirectional recurrent imputation for time series. *arXiv preprint arXiv:1805.10572*.
- Chen, M., Xu, C., Gatto, V., Jain, D., Kumar, A., & Chi, E. (2022). Off-policy actor-critic for recommender systems. *Proceedings of the 16th ACM Conference on Recommender Systems*, 338–349.
- Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. *International conference on machine learning*, 1597–1607.
- Chen, W., Kirkby, L., Kotzev, M., Song, P., Gilron, R., & Pepin, B. (2021). The role of large-scale data infrastructure in developing next-generation deep brain stimulation therapies. *Frontiers in Human Neuroscience*, 15, 717401.
- Cheng, J., Dong, L., & Lapata, M. (2016). Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.

- Chesnaye, N. C., Stel, V. S., Tripepi, G., Dekker, F. W., Fu, E. L., Zoccali, C., & Jager, K. J. (2022). An introduction to inverse probability of treatment weighting in observational research. *Clinical Kidney Journal*, 15(1), 14–20.
- Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.
- Christopher, M., Belghith, A., Bowd, C., & et al. (2018). Performance of deep learning architectures and transfer learning for detecting glaucomatous optic neuropathy in fundus photographs. *Scientific reports*, 8(1), 1–13.
- Cui, Y., Zhu, Y., Wang, J. C., Lu, Y., Zeng, R., Katz, R., Vingopoulos, F., Le, R., Láíns, I., Wu, D. M., et al. (2021). Comparison of widefield swept-source optical coherence tomography angiography with ultra-widefield colour fundus photography and fluorescein angiography for detection of lesions in diabetic retinopathy. *British Journal of Ophthalmology*, 105(4), 577–581.
- Dai, B., Nachum, O., Chow, Y., Li, L., Szepesvári, C., & Schuurmans, D. (2020). Coindice: Off-policy confidence interval estimation. *arXiv preprint arXiv:2010.11652*.
- Darwiche, A., & Mukherjee, S. (2018). Machine learning methods for septic shock prediction. *Proceedings of the 2018 International Conference on Artificial Intelligence and Virtual Reality*, 104–110.
- Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J. M., Parikh, D., & Batra, D. (2017). Visual Dialog. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Deng, J., Dong, W., Socher, R., & et al. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, 248–255.
- Deuschl, G., Schade-Brittinger, C., Krack, P., Volkmann, J., Schäfer, H., Bötzel, K., Daniels, C., Deutschländer, A., Dillmann, U., Eisner, W., et al. (2006). A randomized trial of deep-brain stimulation for parkinson's disease. *New England Journal of Medicine*, 355(9), 896–908.
- Diabetic retinopathy detection – identify signs of diabetic retinopathy in eye images [Updated February 17, 2015. Accessed July 12, 2020.]. (n.d.).
- Dong, C., Loy, C. C., He, K., & Tang, X. (2015). Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2), 295–307.

- Doughty, H., Damen, D., & Mayol-Cuevas, W. (2018). Who's better? who's best? pairwise deep ranking for skill determination. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6057–6066.
- Farajtabar, M., Chow, Y., & Ghavamzadeh, M. (2018). More robust doubly robust off-policy evaluation. *International Conference on Machine Learning*, 1447–1456.
- Fleuren, L. M., Klausch, T. L., Zwager, C. L., Schoonmade, L. J., Guo, T., Roggeveen, L. F., Swart, E. L., Girbes, A. R., Thorald, P., Ercole, A., et al. (2020). Machine learning for the prediction of sepsis: A systematic review and meta-analysis of diagnostic test accuracy. *Intensive care medicine*, 46(3), 383–400.
- Follett, K. A., Weaver, F. M., Stern, M., Hur, K., Harris, C. L., Luo, P., Marks Jr, W. J., Rothlind, J., Sagher, O., Moy, C., et al. (2010). Pallidal versus subthalamic deep-brain stimulation for parkinson's disease. *New England Journal of Medicine*, 362(22), 2077–2091.
- Fortuin, V., Baranchuk, D., Rätsch, G., & Mandt, S. (2020). Gp-vae: Deep probabilistic time series imputation. *International Conference on Artificial Intelligence and Statistics*, 1651–1661.
- Fortuin, V., Dresdner, G., Strathmann, H., & Rätsch, G. (2018). Scalable gaussian processes on discrete domains. *arXiv preprint arXiv:1810.10368*.
- Fortuin, V., & Rätsch, G. (2019). Deep mean functions for meta-learning in gaussian processes. *arXiv preprint arXiv:1901.08098*.
- Freedman, D., Pisani, R., & Purves, R. (2007). *Statistics (international student edition)*. Pisani, R. Purves, 4th edn. WW Norton & Company, New York.
- Fu, J., Kumar, A., Nachum, O., Tucker, G., & Levine, S. (2020). D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*.
- Fu, J., Norouzi, M., Nachum, O., Tucker, G., Novikov, A., Yang, M., Zhang, M. R., Chen, Y., Kumar, A., Paduraru, C., et al. (2020). Benchmarks for deep off-policy evaluation. *ICLR*.
- Gao, G., & Chi, M. (2023). Trace augmentation with missing ehcs for sepsis treatments. *2023 IEEE 11th International Conference on Healthcare Informatics (ICHI)*, 480–480.
- Gao, G., Gao, Q., Yang, X., Ju, S., Pajic, M., & Chi, M. (2023). On trajectory augmentations for off-policy evaluation. *The Twelfth International Conference on Learning Representations*.

- Gao, G., Gao, Q., Yang, X., Pajic, M., & Chi, M. (2022a). A reinforcement learning-informed pattern mining framework for multivariate time series classification. *IJCAI*.
- Gao, G., Gao, Q., Yang, X., Pajic, M., & Chi, M. (2022b). A reinforcement learning-informed pattern mining framework for multivariate time series classification. *International Joint Conference on Artificial Intelligence (IJCAI)*.
- Gao, G., Ju, S., Ausin, M. S., & Chi, M. (2023). Hope: Human-centric off-policy evaluation for e-learning and healthcare. *AAMAS*.
- Gao, G., Khoshnevisan, F., & Chi, M. (2022). Reconstructing missing ehRs using time-aware within-and cross-visit information for septic shock early prediction. *2022 IEEE 10th International Conference on Healthcare Informatics (ICHI)*, 151–162.
- Gao, G., Yang, X., & Chi, M. (2024). Get a head start: On-demand pedagogical policy selection in intelligent tutoring. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(11), 12136–12144.
- Gao, Q., Amason, J., Cousins, S., Pajic, M., & Hadziahmetovic, M. (2021). Automated identification of referable retinal pathology in teleophthalmology setting. *Translational vision science & technology*, 10(6), 30–30.
- Gao, Q., Gao, G., Chi, M., & Pajic, M. (2023a). Variational latent branching model for off-policy evaluation. *ICLR*.
- Gao, Q., Gao, G., Chi, M., & Pajic, M. (2023b). Variational latent branching model for off-policy evaluation. *The Eleventh International Conference on Learning Representations (ICLR)*.
- Gao, Q., Gao, G., Dong, J., Tarokh, V., Chi, M., & Pajic, M. (2024). Off-policy evaluation for human feedback. *Advances in Neural Information Processing Systems*, 36.
- Gao, Q., Hajinezhad, D., Zhang, Y., Kantaros, Y., & Zavlanos, M. (2019). Reduced variance deep reinforcement learning with temporal logic specifications. *ICCPs*.
- Gao, Q., Hajinezhad, D., Zhang, Y., Kantaros, Y., & Zavlanos, M. M. (2019). Reduced variance deep reinforcement learning with temporal logic specifications. *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, 237–248.
- Gao, Q., Naumann, M., Jovanov, I., Lesi, V., Kamaravelu, K., Grill, W. M., & Pajic, M. (2020). Model-based design of closed loop deep brain stimulation controller using reinforcement learning. *2020 ACM/IEEE 11th Int. Conf. on Cyber-Physical Systems (ICCPs)*, 108–118.

- Gao, Q., Pajic, M., & Zavlanos, M. M. (2020). Deep imitative reinforcement learning for temporal logic robot motion planning with noisy semantic observations. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 8490–8496.
- Gao, Q., Schmidt, S. L., Chowdhury, A., Feng, G., Peters, J. J., Genty, K., Grill, W. M., Turner, D. A., & Pajic, M. (2023). Offline learning of closed-loop deep brain stimulation controllers for parkinson disease treatment. *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, 44–55.
- Gao, Q., Schmidt, S. L., Kamaravelu, K., Turner, D. A., Grill, W. M., & Pajic, M. (2022). Offline policy evaluation for learning-based deep brain stimulation controllers. *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCCPS)*, 80–91.
- Gao, Q., Wang, D., Amason, J. D., Yuan, S., Tao, C., Henao, R., Hadziahmetovic, M., Carin, L., & Pajic, M. (2022a). Gradient importance learning for incomplete observations. *International Conference on Learning Representations*.
- Gao, Q., Wang, D., Amason, J. D., Yuan, S., Tao, C., Henao, R., Hadziahmetovic, M., Carin, L., & Pajic, M. (2022b). Gradient importance learning for incomplete observations. *International Conference on Learning Representations*.
- García-Laencina, P. J., Sancho-Gómez, J.-L., & Figueiras-Vidal, A. R. (2010). Pattern classification with missing data: A review. *Neural Computing and Applications*, 19(2), 263–282.
- Gargeya, R., & Leng, T. (2017). Automated identification of diabetic retinopathy using deep learning. *Ophthalmology*, 124(7), 962–969.
- Gonzalez, R., & Woods, R. (2007). *Digital image processing*. Addison-Wesley Publishing Company.
- Gould, S., Gao, T., & Koller, D. (2009). Region-based segmentation and object detection. *Advances in Neural Information Processing Systems*, 22.
- Gu, S., Holly, E., Lillicrap, T., & Levine, S. (2017). Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. *Int. Conf. on robotics and automation (ICRA)*, 3389–3396.
- Guez, A., Vincent, R. D., Avoli, M., & Pineau, J. (2008). Adaptive treatment of epilepsy via batch-mode reinforcement learning. *AAAI*, 1671–1678.

- Gulshan, V., Peng, L., Coram, M., & et al. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, *316*(22), 2402–2410.
- Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *ICML*, 1861–1870.
- Habets, J., Heijmans, M., Kuijf, M. L., Janssen, M. L., Temel, Y., & Kubben, P. L. (2018). An update on adaptive deep brain stimulation in parkinson’s disease. *Movement Disorders*, *33*(12), 1834–1843.
- Hadziahmetovic, M., Nicholas, P., Jindal, S., Mettu, P. S., & Cousins, S. W. (2019). Evaluation of a remote diagnosis imaging model vs dilated eye examination in referable macular degeneration. *JAMA Ophthalmol*, *137*(7), 802–808.
- Hafner, D., Lillicrap, T., Ba, J., & Norouzi, M. (2020). Dream to control: Learning behaviors by latent imagination. *International Conference on Learning Representations*.
- Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., & Davidson, J. (2019). Learning latent dynamics for planning from pixels. *International conference on machine learning*, 2555–2565.
- Hafner, D., Lillicrap, T. P., Norouzi, M., & Ba, J. (2020). Mastering atari with discrete world models. *International Conference on Learning Representations*.
- Han, B., Ren, Z., Wu, Z., Zhou, Y., & Peng, J. (2022). Off-policy reinforcement learning with delayed rewards. *International Conference on Machine Learning*, 8280–8303.
- Hanin, B., & Rolnick, D. (2018). How to start training: The effect of initialization and architecture. *Advances in Neural Information Processing Systems*, *31*.
- Hanna, J., Niekum, S., & Stone, P. (2019). Importance sampling policy evaluation with an estimated behavior policy. *International Conference on Machine Learning*, 2605–2613.
- Hao, B., Ji, X., Duan, Y., Lu, H., Szepesvari, C., & Wang, M. (2021). Bootstrapping fitted q-evaluation for off-policy inference. *International Conference on Machine Learning*, 4074–4084.
- Hinton, G., Vinyals, O., Dean, J., et al. (n.d.). Distilling the knowledge in a neural network.
- Hoang, K. B., Cassar, I. R., Grill, W. M., & Turner, D. A. (2017). Biomarkers and stimulation algorithms for adaptive brain stimulation. *Frontiers in neuroscience*, *11*, 564.

- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.
- Honaker, J., & King, G. (2010). What to do about missing values in time-series cross-section data. *American journal of political science*, 54(2), 561–581.
- Hsu, H.-L., Gao, Q., & Pajic, M. (2024). ϵ -neural thompson sampling of deep brain stimulation for parkinson disease treatment. *arXiv preprint arXiv:2403.06814*.
- Ipsen, N., Mattei, P.-A., & Frellsen, J. (2020). How to deal with missing data in supervised deep learning? *ICML Workshop on the Art of Learning with Missing Values (Artemiss)*.
- Ishii, S., Yoshida, W., & Yoshimoto, J. (2002). Control of exploitation–exploration meta-parameter in reinforcement learning. *Neural networks*, 15, 665–687.
- Jia, Y., Burden, J., Lawton, T., & Habli, I. (n.d.). Safe reinforcement learning for sepsis treatment. *2020 IEEE International conference on healthcare informatics (ICHI)*, 1–7.
- Jiang, N., & Li, L. (2016). Doubly robust off-policy value evaluation for reinforcement learning. *ICML*, 652–661.
- Johnson, A. E., Pollard, T. J., Shen, L., Li-Wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., & Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(1), 1–9.
- Jovanov, I., Naumann, M., Kumaravelu, K., Grill, W. M., & Pajic, M. (2018). Platform for model-based design and testing for deep brain stimulation. *ICCPs*.
- Julien, C., Hache, G., Dulac, M., Dubrou, C., Castelnovo, G., Giordana, C., Azulay, J.-P., & Fluchère, F. (2021). The clinical meaning of levodopa equivalent daily dose in parkinson’s disease. *Fundamental & Clinical Pharmacology*, 35(3), 620–630.
- Kaggle diabetic retinopathy detection competition [Accessed: 2021-05-08]. (n.d.).
- Kam, H. J., & Kim, H. Y. (2017). Learning representations for the early detection of sepsis with deep neural networks. *Computers in biology and medicine*, 89, 248–255.
- Keremany, D. S., Goldbaum, M., Cai, W., & et al. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5), 1122–1131.
- Keremany, D. S., Goldbaum, M., Cai, W., Valentim, C., & et al. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5), 1122–1131.

- Keremany, D. S., Goldbaum, M., Cai, W., Valentim, C. C., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., et al. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5), 1122–1131.
- Khoshnevisan, F., et al. (2020). A variational recurrent adversarial multi-source domain adaptation framework for septic shock early prediction across medical systems.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Körber, M., Lange, J., Rediske, S., Steinmann, S., & Glück, R. (2021). Comparing popular simulation environments in the scope of robotics and reinforcement learning. *arXiv preprint arXiv:2103.04616*.
- Kostrikov, I., & Nachum, O. (2020). Statistical bootstrapping for uncertainty estimation in off-policy evaluation. *arXiv preprint arXiv:2007.13609*.
- Kostrikov, I., Nair, A., & Levine, S. (2022). Offline reinforcement learning with implicit q-learning. *ICLR*.
- Kühn, A., Kupsch, A., Schneider, G., & Brown, P. (2006). Reduction in subthalamic 8–35 hz oscillatory activity correlates with clinical improvement in parkinson’s disease. *Euro. J. of Neuroscience*, 23(7), 1956–1960.
- Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1), 79–86.
- Kumar, A., Zhou, A., Tucker, G., & Levine, S. (2020). Conservative q-learning for offline reinforcement learning. *NeurIPS*.
- Kuncel, A. M., & Grill, W. M. (2004). Selection of stimulus parameters for deep brain stimulation. *Clinical neurophysiology*, 115(11), 2431–2441.
- Le, H., Voloshin, C., & Yue, Y. (2019). Batch policy learning under constraints. *International Conference on Machine Learning*, 3703–3712.
- LeCun, Y., & Cortes, C. (2010). MNIST handwritten digit database. <http://yann.lecun.com/exdb/mnist/>
- LeCun, Y., Haffner, P., Bottou, L., & Bengio, Y. (1999). Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision* (pp. 319–345). Springer.

- L'Ecuyer, P., & Tuffin, B. (2008). Approximate zero-variance simulation. *2008 Winter Simulation Conference*, 170–181.
- Lee, A. X., Nagabandi, A., Abbeel, P., & Levine, S. (2020). Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, 33, 741–752.
- Li, B., Powell, A., Hooper, P., & Sheidow, T. (2015). Prospective evaluation of teleophthalmology in screening and recurrence monitoring of neovascular age-related macular degeneration: A randomized clinical trial. *Jama Ophthalmology*, 133(3), 276–282.
- Li, L., Chu, W., Langford, J., & Wang, X. (2011). Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. *Proceedings of the fourth ACM International Conference on Web Search and Data Mining*, 297–306.
- Li, S. C.-X., Jiang, B., & Marlin, B. (2019). Misgan: Learning from incomplete data with generative adversarial networks. *arXiv preprint arXiv:1902.09599*.
- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2016). Continuous control with deep reinforcement learning. *ICLR*.
- Lipton, Z. C., Kale, D., & Wetzel, R. (2016). Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. *Machine learning for health-care conference*, 253–270.
- Lis, C. G., Patel, K., & Gupta, D. (2015). The relationship between patient satisfaction with service quality and survival in non-small cell lung cancer—is self-rated health a potential confounder? *PloS one*, 10(7), e0134617.
- Little, R. J., & Rubin, D. B. (2019). *Statistical analysis with missing data* (Vol. 793). John Wiley & Sons.
- Little, S., Pogosyan, A., Neal, S., Zavala, B., Zrinzo, L., Hariz, M., Foltynie, T., Limousin, P., Ashkan, K., FitzGerald, J., et al. (2013). Adaptive deep brain stimulation in advanced parkinson disease. *Annals of neurology*, 74(3), 449–457.
- Little, S., Tripoliti, E., Beudel, M., Pogosyan, A., Cagnan, H., Herz, D., Bestmann, S., Aziz, T., Cheeran, B., Zrinzo, L., et al. (2016). Adaptive deep brain stimulation for parkinson's disease demonstrates reduced speech side effects compared to conventional stimulation in the acute setting. *J Neurol Neurosurg Psychiatry*, 87(12), 1388–1389.
- Liu, E., Stephan, M., Nie, A., Piech, C., Brunskill, E., & Finn, C. (2022). Giving feedback on interactive student programs with meta-exploration. *Advances in Neural Information Processing Systems*, 35, 36282–36294.

- Liu, R., Greenstein, J. L., Granite, S. J., Fackler, J. C., Bembea, M. M., Sarma, S. V., & Winslow, R. L. (2019). Data-driven discovery of a novel sepsis pre-shock state predicts impending septic shock in the icu. *Scientific reports*, 9(1), 1–9.
- Liu, S., Paranjape, A. S., Elmaanaoui, B., Dewelle, J., & et al. (2009). Quality assessment for spectral domain optical coherence tomography (oct) images. *Multimodal Biomedical Imaging IV*, 7171, 71710X.
- Liu, Y., Bacon, P.-L., & Brunskill, E. (2020). Understanding the curse of horizon in off-policy evaluation via conditional importance sampling. *ICML*.
- Lu, C., Ball, P. J., Rudner, T. G., Parker-Holder, J., Osborne, M. A., & Teh, Y. W. (2022). Challenges and opportunities in offline reinforcement learning from visual observations. *arXiv preprint arXiv:2206.04779*.
- Lu, W., Tong, Y., Yu, Y., Xing, Y., Chen, C., & Shen, Y. (2018). Deep learning-based automated classification of multi-categorical abnormalities from optical coherence tomography images. *Translational Vision Science & Technology*, 7(6), 41.
- Ma, C., Tschitschek, S., Palla, K., Hernández-Lobato, J. M., Nowozin, S., & Zhang, C. (2018). Eddi: Efficient dynamic discovery of high-value information with partial VAE. *arXiv preprint arXiv:1809.11142*.
- MacGlashan, J., Ho, M. K., Loftin, R., Peng, B., Wang, G., Roberts, D. L., Taylor, M. E., & Littman, M. L. (2017). Interactive learning from policy-dependent human feedback. *International Conference on Machine Learning*, 2285–2294.
- Mamandipoor, B., Majd, M., Moz, M., & Osmani, V. (2019). Blood lactate concentration prediction in critical care patients: Handling missing values. *arXiv preprint arXiv:1910.01473*.
- Mandel, T., Liu, Y.-E., Brunskill, E., & Popović, Z. (2017). Where to add actions in human-in-the-loop reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).
- Mandel, T., Liu, Y.-E., Levine, S., Brunskill, E., & Popovic, Z. (2014). Offline policy evaluation across representations with applications to educational games. *AAMAS*, 1077.
- Mann, H. B., & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, 50–60.
- Mao, Q., Jay, M., Hoffman, J. L., Calvert, J., Barton, C., Shimabukuro, D., Shieh, L., Chetipally, U., Fletcher, G., Kerem, Y., et al. (2018). Multicentre validation of a sepsis prediction algorithm using only vital sign data in the emergency department, general ward and icu. *BMJ open*, 8(1).

- Marras, C., Beck, J., Bower, J., Roberts, E., Ritz, B., Ross, G., Abbott, R., Savica, R., Van Den Eeden, S., Willis, A., et al. (2018). Prevalence of parkinson's disease across north america. *NPJ Parkinson's disease*, 4(1), 21.
- Mattei, P.-A., & Frellsen, J. (2019). Miwae: Deep generative modelling and imputation of incomplete data sets. *International Conference on Machine Learning*, 4413–4423.
- Mazumder, R., Hastie, T., & Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11, 2287–2322.
- Medeiros, F. A., Jammal, A. A., & Thompson, A. C. (2019). From machine to machine: An oct-trained deep learning algorithm for objective quantification of glaucomatous damage in fundus photographs. *Ophthalmology*, 126(4), 513–521.
- Mehrotra, A., Chernew, M., Linetsky, D., Hatch, H., & Cutler, D. (2017). The impact of the covid-19 pandemic on outpatient visits: A rebound emerges. *To the Point (blog)*, Commonwealth Fund.
- Mehrotra, R., McInerney, J., Bouchard, H., Lalmas, M., & Diaz, F. (2018). Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2243–2251.
- Michalak, S., & Hadziahmetovic, M. (2019). Teleophthalmology in retinal disease. *Retinal Physician*, 17, 39–43.
- Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. *ICML*, 1928–1937.
- Mnih, V., Heess, N., Graves, A., et al. (2014). Recurrent models of visual attention. *Advances in neural information processing systems*, 27, 2204–2212.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529.
- Mookiah, M. R. K., Acharya, U. R., Chua, C. K., Lim, C. M., Ng, E., & Laude, A. (2013). Computer-aided diagnosis of diabetic retinopathy: A review. *Computers in biology and medicine*, 43(12), 2136–2155.
- Moreno-Barea, F. J., Strazzera, F., Jerez, J. M., Urda, D., & Franco, L. (2018). Forward noise adjustment scheme for data augmentation. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 728–734.

- Morvan, M. L., Josse, J., Moreau, T., Scornet, E., & Varoquaux, G. (2020). Neumiss networks: Differentiable programming for supervised learning with missing values. In *Neurips*.
- Muhammad, H., Fuchs, T. J., De Cuir, N., & et al. (2017). Hybrid deep learning on single wide-field optical coherence tomography scans accurately classifies glaucoma suspects. *Journal of glaucoma*, 26(12), 1086.
- Nachum, O., Chow, Y., Dai, B., & Li, L. (2019). Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *NeurIPS*, 32.
- Nagaraj, V., Lamperski, A., & Netoff, T. I. (2017). Seizure control in a computational model using a reinforcement learning stimulation paradigm. *International J. of Neural Sys.*, 27(07), 1750012.
- Namkoong, H., Keramati, R., Yadlowsky, S., & Brunskill, E. (2020). Off-policy policy evaluation for sequential decisions under unobserved confounding. *Advances in Neural Information Processing Systems*, 33, 18819–18831.
- Nie, A., Flet-Berliac, Y., Jordan, D., Steenbergen, W., & Brunskill, E. (2022). Data-efficient pipeline for offline reinforcement learning with limited data. *Advances in Neural Information Processing Systems*, 35, 14810–14823.
- Okun, M. S. (2012a). Deep-brain stimulation for parkinson’s disease. *New England Journal of Medicine*, 367(16), 1529–1538.
- Okun, M. S. (2012b). Deep-brain stimulation for parkinson’s disease. *New England Journal of Medicine*, 367(16), 1529–1538.
- Opri, E., Cernera, S., Molina, R., Eisinger, R. S., Cagle, J. N., Almeida, L., Denison, T., Okun, M. S., Foote, K. D., & Gunduz, A. (2020). Chronic embedded cortico-thalamic closed-loop deep brain stimulation for the treatment of essential tremor. *Science translational medicine*, 12(572), eaay7680.
- Parvinian, B., Scully, C., Wiyor, H., Kumar, A., & Weininger, S. (2018). Regulatory considerations for physiological closed-loop controlled medical devices used for automated critical care: Food and drug administration workshop discussion topics. *Anesthesia and analgesia*, 126(6), 1916.
- Patil, V. P., Hofmarcher, M., Dinu, M.-C., Dorfer, M., Blies, P. M., Brandstetter, J., Arjona-Medina, J. A., & Hochreiter, S. (2020). Align-rudder: Learning from few demonstrations by reward redistribution. *arXiv preprint arXiv:2009.14108*.

- Pineau, J., Guez, A., Vincent, R., Panuccio, G., & Avoli, M. (2009). Treating epilepsy via adaptive neurostimulation: A reinforcement learning approach. *Int. J. of Neural Sys.*, 19(04), 227–240.
- Powers, R., Etezadi-Amoli, M., Arnold, E. M., Kianian, S., Mance, I., Gibiansky, M., Trietsch, D., Alvarado, A. S., Kretlow, J. D., Herrington, T. M., et al. (2021). Smartwatch inertial sensors continuously monitor real-world motor fluctuations in parkinson’s disease. *Science translational medicine*, 13(579), eabd7865.
- Precup, D. (2000). Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, 80.
- Rafailov, R., Yu, T., Rajeswaran, A., & Finn, C. (2021). Offline reinforcement learning from images with latent space models. *Learning for Dynamics and Control*, 1154–1168.
- Raghu, M., Zhang, C., Kleinberg, J., & Bengio, S. (2019). Transfusion: Understanding transfer learning for medical imaging. *Advances in neural information processing systems*, 3347–3357.
- Ramaker, C., Marinus, J., Stiggelbout, A. M., & Van Hilten, B. J. (2002). Systematic evaluation of rating scales for impairment and disability in parkinson’s disease. *Movement disorders*, 17(5), 867–876.
- Raman, R., Srinivasan, S., Virmani, S., Sivaprasad, S., Rao, C., & Rajalakshmi, R. (2019). Fundus photograph-based deep learning algorithms in detecting diabetic retinopathy. *Eye*, 33(1), 97–109.
- Rathi, S., Tsui, E., Mehta, N., Zahid, S., & Schuman, J. (2017). The current state of teleophthalmology in the united states. *Ophthalmology*, 124(12), 1729–1734.
- Rocco, I., Arandjelović, R., & Sivic, J. (2018). End-to-end weakly-supervised semantic alignment. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6917–6925.
- Rossi, S., Michiardi, P., & Filippone, M. (2019). Good initializations of variational bayes for deep models. *International Conference on Machine Learning*, 5487–5497.
- Ruan, S., Nie, A., Steenbergen, W., He, J., Zhang, J., Guo, M., Liu, Y., Nguyen, K. D., Wang, C. Y., Ying, R., et al. (2023). Reinforcement learning tutor better supported lower performers in a math task. *arXiv preprint arXiv:2304.04933*.
- Rusu, A. A., Colmenarejo, S. G., Gülçehre, Ç., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K., & Hadsell, R. (2016). Policy distillation. *ICLR*.

- Rybkin, O., Zhu, C., Nagabandi, A., Daniilidis, K., Mordatch, I., & Levine, S. (2021). Model-based reinforcement learning via latent-space collocation. *International Conference on Machine Learning*, 9190–9201.
- Saad, D. (1998). Online algorithms and stochastic approximations. *Online Learning*, 5, 6–3.
- Saito, Y., Udagawa, T., Kiyohara, H., Mogi, K., Narita, Y., & Tateno, K. (2021). Evaluating the robustness of off-policy evaluation. *Fifteenth ACM Conference on Recommender Systems*, 114–123.
- Saquil, Y., Chen, D., He, Y., Li, C., & Yang, Y.-L. (2021). Multiple pairwise ranking networks for personalized video summarization. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1718–1727.
- Scherpf, M., Gräßer, F., Malberg, H., & Zaunseder, S. (2019). Predicting sepsis with a recurrent neural network using the mimic iii database. *Computers in biology and medicine*, 113, 103395.
- Schmidt, S. L., Chowdhury, A. H., Mitchell, K. T., Peters, J. J., Gao, Q., Lee, H.-J., Genty, K., Chow, S.-C., Grill, W. M., Pajic, M., et al. (2024). At home adaptive dual target deep brain stimulation in parkinson’s disease with proportional control. *Brain*, 147(3), 911–922.
- Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., & Joachims, T. (2016). Recommendations as treatments: Debiasing learning and evaluation. *international conference on machine learning*, 1670–1679.
- Segal, A., Gal, K., Kamar, E., Horvitz, E., & Miller, G. (2018). Optimizing interventions via offline policy evaluation: Studies in citizen science. *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Sheetrit, E., Nissim, N., Klimov, D., Fuchs, L., Elovici, Y., & Shahar, Y. (2017). Temporal pattern discovery for accurate sepsis diagnosis in icu patients. *arXiv preprint arXiv:1709.01720*.
- Shibata, N., Tanito, M., Mitsuhashi, K., Fujino, Y., & et al. (2018). Development of a deep residual learning algorithm to screen for glaucoma from fundus photography. *Scientific reports*, 8(1), 1–9.
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 60.
- Silva, I., Moody, G., Scott, D. J., Celi, L. A., & Mark, R. G. (2012). Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. *2012 Computing in Cardiology*, 245–248.

- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587), 484–489.
- Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014). Deterministic policy gradient algorithms.
- Singer, M., Deutschman, C. S., Seymour, C. W., Shankar-Hari, M., Annane, D., Bauer, M., Bellomo, R., Bernard, G. R., Chiche, J.-D., Coopersmith, C. M., et al. (2016). The third international consensus definitions for sepsis and septic shock (sepsis-3). *Jama*, 315(8), 801–810.
- Singh, R. K., & Gorantla, R. (2020). Dmenet: Diabetic macular edema diagnosis using hierarchical ensemble of cnns. *PloS one*, 15(2), e0220677.
- Snell, C., Kostrikov, I., Su, Y., Yang, M., & Levine, S. (2022). Offline rl for natural language generation with implicit language q learning. *arXiv preprint arXiv:2206.11871*.
- So, R. Q., Kent, A. R., & Grill, W. M. (2012). Relative contributions of local cell and passing fiber activation and silencing to changes in thalamic fidelity during deep brain stimulation and lesioning: A computational modeling study. *Journal of computational neuroscience*, 32(3), 499–519.
- Sportisse, A., Boyer, C., Dieuleveut, A., & Josses, J. (2020). Debiasing averaged stochastic gradient descent to handle missing values. *Advances in Neural Information Processing Systems*, 33.
- Sreelatha, O., & Ramesh, S. (2016). Teleophthalmology: Improving patient outcomes? *Clinical Ophthalmology (Auckland, NZ)*, 10, 285.
- Stanslaski, S., Herron, J., Chouinard, T., Bourget, D., Isaacson, B., Kremen, V., Opri, E., Drew, W., Brinkmann, B. H., Gunduz, A., et al. (2018). A chronically implantable neural coprocessor for investigating the treatment of neurological disorders. *IEEE transactions on biomedical circuits and systems*, 12(6), 1230–1245.
- Stekhoven, D. J., & Bühlmann, P. (2012). Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1), 112–118.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- Swann, N. C., de Hemptinne, C., Miocinovic, S., Qasim, S., Wang, S. S., Ziman, N., Ostrem, J. L., San Luciano, M., Galifianakis, N. B., & Starr, P. A. (2016). Gamma oscillations in the hyperkinetic state detected with chronic human brain recordings in parkinson's disease. *Journal of Neuroscience*, 36(24), 6445–6458.

- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016a). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016b). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.
- Tang, S., & Wiens, J. (2021). Model selection for offline reinforcement learning: Practical considerations for healthcare settings. *Machine Learning for Healthcare Conference*, 2–35.
- Tang, Z., Feng, Y., Li, L., Zhou, D., & Liu, Q. (2019). Doubly robust bias reduction in infinite horizon off-policy estimation. *ICLR*.
- Tejedor, M., Woldaregay, A. Z., & Godtliebsen, F. (2020). Reinforcement learning application in diabetes blood glucose control: A systematic review. *Artificial intelligence in medicine*, 104, 101836.
- Thomas, P., & Brunskill, E. (2016). Data-efficient off-policy policy evaluation for reinforcement learning. *ICML*, 2139–2148.
- Thomas, P. S. (2015). Safe reinforcement learning.
- Todorov, E., Erez, T., & Tassa, Y. (2012). Mujoco: A physics engine for model-based control. *2012 IEEE/RSJ international conference on intelligent robots and systems*, 5026–5033.
- Treder, M., Lauermann, J. L., & Eter, N. (2018a). Automated detection of exudative age-related macular degeneration in spectral domain optical coherence tomography using deep learning. *Graefe's Archive for Clinical and Experimental Ophthalmology*, 256(2), 259–265.
- Treder, M., Lauermann, J. L., & Eter, N. (2018b). Deep learning-based detection and classification of geographic atrophy using a deep convolutional neural network classifier. *Graefe's Archive for Clinical and Experimental Ophthalmology*, 256(11), 2053–2060.
- Usher, D., Dumskyj, M., Himaga, M., Williamson, T. H., Nussey, S., & Boyce, J. (2004). Automated detection of diabetic retinopathy in digital retinal images: A tool for diabetic retinopathy screening. *Diabetic Medicine*, 21(1), 84–90.
- Vaghefi, E., Hill, S., Kersten, H. M., & Squirrell, D. (2020). Multimodal retinal image analysis via deep learning for the diagnosis of intermediate dry age-related macular degeneration: A feasibility study. *Journal of Ophthalmology*.

- Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 5998–6008.
- Vecerik, M., Hester, T., Scholz, J., Wang, F., Pietquin, O., Piot, B., Heess, N., Rothörl, T., Lampe, T., & Riedmiller, M. (2017). Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*.
- Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782), 350–354.
- Wang, J., De Vries, A. P., & Reinders, M. J. (2006). Unifying user-based and item-based collaborative filtering approaches by similarity fusion. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 501–508.
- Wang, W., Xu, Z., Yu, W., & et al. (2019a). Two-stream cnn with loose pair training for multi-modal amd categorization. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 156–164.
- Wang, W., Xu, Z., Yu, W., & et al. (2019b). Two-stream cnn with loose pair training for multi-modal amd categorization. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 156–164.
- Wang, X., Zhang, R., Shen, C., Kong, T., & Li, L. (2021). Dense contrastive learning for self-supervised visual pre-training. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3024–3033.
- Wang, Y., Zhang, Y., Yao, Z., Zhao, R., & Zhou, F. (2016). Machine learning based detection of age-related macular degeneration (amd) and diabetic macular edema (dme) from optical coherence tomography (oct) images. *Biomedical optics express*, 7(12), 4928–4940.
- Wattenberg, M., Viégas, F., & Johnson, I. (2016). How to use t-sne effectively. *Distill*, 1(10), e2.
- Wen, J., Dai, B., Li, L., & Schuurmans, D. (2020). Batch stationary distribution estimation. *International Conference on Machine Learning*, 10203–10213.
- Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4), 229–256.

- Wilson, A. G., Hu, Z., Salakhutdinov, R., & Xing, E. P. (2016). Deep kernel learning. *Artificial intelligence and statistics*, 370–378.
- Winder, R. J., Morrow, P. J., McRitchie, I. N., Bailie, J., & Hart, P. M. (2009). Algorithms for digital image processing in diabetic retinopathy. *Computerized medical imaging and graphics*, 33(8), 608–622.
- Wong, J. K., Deuschl, G., Wolke, R., Bergman, H., Muthuraman, M., Groppa, S., Sheth, S. A., Bronte-Stewart, H. M., Wilkins, K. B., Petrucci, M. N., et al. (2022). Proc. the 9th annual deep brain stimulation think tank: Advances in cutting edge technologies, artificial intelligence, neuromodulation, neuroethics, pain, interventional psychiatry, epilepsy, and traumatic brain injury. *Frontiers in Human Neuroscience*, 25.
- Wu, Y., Rosca, M., & Lillicrap, T. (2019). Deep compressed sensing. *International Conference on Machine Learning*, 6850–6860.
- Wu, Y., Mansimov, E., Grosse, R. B., Liao, S., & Ba, J. (2017). Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. *NeurIPS*.
- Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. *International conference on machine learning*, 2048–2057.
- Xu, Z., Wang, W., Yang, J., & et al. (2020). Automated diagnoses of age-related macular degeneration and polypoidal choroidal vasculopathy using bi-modal deep convolutional neural networks. *British Journal of Ophthalmology*.
- Yang, M., Dai, B., Nachum, O., Tucker, G., & Schuurmans, D. (2021). Offline policy selection under uncertainty. *Deep RL Workshop NeurIPS 2021*.
- Yang, M., Nachum, O., Dai, B., Li, L., & Schuurmans, D. (2020). Off-policy evaluation via the regularized lagrangian. *NeurIPS*, 33.
- Yang, X., Gao, G., & Chi, M. (2023a). Hierarchical apprenticeship learning for disease progression modeling. *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 2388–2396.
- Yang, X., Gao, G., & Chi, M. (2023b). An offline time-aware apprenticeship learning framework for evolving reward functions. *arXiv preprint arXiv:2305.09070*.
- Yang, X., Zhang, Y., & Chi, M. (2018). Time-aware subgroup matrix decomposition: Imputing missing data using forecasting events. *2018 IEEE International Conference on Big Data (Big Data)*, 1524–1533.

- Yao, Y., Vehtari, A., Simpson, D., & Gelman, A. (2018). Yes, but did it work?: Evaluating variational inference. *International Conference on Machine Learning*, 5581–5590.
- Yee, C. R., Narain, N. R., Akmaev, V. R., & Vemulapalli, V. (2019). A data-driven approach to predicting septic shock in the intensive care unit. *Biomedical informatics insights*, 11, 1178222619885147.
- Yoo, T. K., Choi, J. Y., Seo, J. G., Ramasubramanian, B., Selvaperumal, S., & Kim, D. W. (2019). The possibility of the combination of oct and fundus images for improving the diagnostic accuracy of deep learning for age-related macular degeneration: A preliminary experiment. *Medical & biological engineering & computing*, 57(3), 677–687.
- Yoon, J., Jordon, J., & Schaar, M. (2018). Gain: Missing data imputation using generative adversarial nets. *International Conference on Machine Learning*, 5689–5698.
- Yu, H.-F., Rao, N., & Dhillon, I. S. (2016). Temporal regularized matrix factorization for high-dimensional time series prediction. *NIPS*, 847–855.
- Yu, T., Kumar, A., Rafailov, R., Rajeswaran, A., Levine, S., & Finn, C. (2021). Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34, 28954–28967.
- Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., & Ma, T. (2020). Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33, 14129–14142.
- Zhang, M., Vikram, S., Smith, L., Abbeel, P., Johnson, M., & Levine, S. (2019). Solar: Deep structured representations for model-based reinforcement learning. *International Conference on Machine Learning*, 7444–7453.
- Zhang, M. R., Paine, T., Nachum, O., Paduraru, C., Tucker, G., Norouzi, M., et al. (2020). Autoregressive dynamics models for offline policy evaluation and optimization. *International Conference on Learning Representations*.
- Zhang, R., Dai, B., Li, L., & Schuurmans, D. (2020). Gendice: Generalized offline estimation of stationary values. *International Conference on Learning Representations*.
- Zhang, S., Liu, B., & Whiteson, S. (2020). Gradientdice: Rethinking generalized offline estimation of stationary values. *ICML*, 11194–11203.
- Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., & Irving, G. (2019). Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

Biography

Qitong Gao earned his B.Eng. in Chemical Engineering from Tianjin University (Tianjin, China) in the summer of 2016, followed by an M.Sc. in Mechanical Engineering and Materials Science (MEMS) from Duke University (Durham, NC, USA) in the spring of 2018. Following the completion of his M.Sc., he continued to work with Prof. Michael Zavlanos for an additional year before transitioning to Prof. Miroslav Pajic's Cyber Physical Systems Lab (CPSL) in the Department of Electrical and Computer Engineering at Duke University in 2019 to pursue his Ph.D. He completed internships at Meta Platforms, Inc., during the summers of 2020 and 2021, where he focused on developing machine learning and deep learning pipelines to automate and enhance enterprise products (Workplace), and operational research processes. Subsequently, he joined Netflix, Inc., in the summer of 2022 as a research scientist intern, concentrating on the development of an off-policy deep reinforcement learning approach to enhance Netflix's recommendation system. His current research focuses on offline reinforcement learning and off-policy evaluation, along with multi-modal data analysis and reasoning. He applies these interests to various domains such as human-centric systems (e.g., healthcare), recommendation systems, robotics, and cyber-physical systems.