Bridging Machine Learning and Experimental Design for Enhanced Data Analysis and Optimization

Qing Guo

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Statistics

Xinwei Deng, Chair
Hongxiao Zhu
Yili Hong
Xin Xing

June 7, 2024
Blacksburg, Virginia

Bridging Machine Learning and Experimental Design for Enhanced Data Efficiency and Optimization

Qing Guo

ABSTRACT

Experimental design is a powerful tool for gathering highly informative observations using a small number of experiments. The demand for smart data collection strategies is increasing due to the need to save time and budget, especially in online experiments and machine learning. However, the traditional experimental design method falls short in systematically assessing changing variables' effects. Specifically within Artificial Intelligence (AI), the challenge lies in assessing the impacts of model structures and training strategies on task performances with a limited number of trials. This shortfall underscores the necessity for the development of novel approaches. On the other side, the optimal design criterion has typically been model-based in classic design literature, which leads to restricting the flexibility of experimental design strategies. However, machine learning's inherent flexibility can empower the estimation of metrics efficiently using nonparametric and optimization techniques, thereby broadening the horizons of experimental design possibilities.

In this dissertation, the aim is to develop a set of novel methods to bridge the merits between these two domains: 1) applying ideas from statistical experimental design to enhance data efficiency in machine learning, and 2) leveraging powerful deep neural networks to optimize experimental design strategies.

This dissertation consists of 5 chapters. Chapter 1 provides a general introduction to mutual information, fractional factorial design, hyper-parameter tuning, multi-modality, *etc.* In Chapter 2, I propose a new mutual information estimator `FLO` by integrating techniques from variational inference (VAE), contrastive learning, and convex optimization. I apply `FLO` to broad data science applications, such as efficient data collection, transfer learning, fair learning*, etc.* Chapter 3 introduces a new design strategy called multi-layer sliced design (MLSD) with the application of AI assurance. It focuses on exploring the effects of hyper-parameters under different models and optimization strategies. Chapter 4 investigates classic vision challenges via multimodal large language models by implicitly optimizing mutual information and thoroughly exploring training strategies. Chapter 5 concludes this proposal and discusses several future research topics.

Key Words: Mutual Information, Sliced Design, Bayesian Optimal Design, Induced Lasso, Few-shot Learning, Variational Inference, Contrastive Learning

Bridging Machine Learning and Experimental Design for Enhanced Data Analysis and Optimization

Qing Guo

(GENERAL AUDIENCE ABSTRACT)

In the digital age, artificial intelligence (AI) is reshaping our interactions with technology through advanced machine learning models. These models are complex, often opaque mechanisms that present challenges in understanding their inner workings. This complexity necessitates numerous experiments with different settings to optimize performance, which can be costly. Consequently, it is crucial to strategically evaluate the effects of various strategies on task performance using a limited number of trials. The Design of Experiments (DoE) offers invaluable techniques for investigating and understanding these complex systems efficiently. Moreover, integrating machine learning models can further enhance the DoE. Traditionally, experimental designs pre-specify a model and focus on finding the best strategies for experimentation. This assumption can restrict the adaptability and applicability of experimental designs. However, the inherent flexibility of machine learning models can enhance the capabilities of DoE, unlocking new possibilities for efficiently optimizing experimental strategies through an information-centric approach. Moreover, the information-based method can also be beneficial in other AI applications, including self-supervised learning, fair learning, transfer learning, *etc.* The research presented in this dissertation aims to bridge machine learning and experimental design, offering new insights and methodologies that benefit both AI techniques and DoE.

To my family.

Acknowledgements

Firstly, I would like to express my heartfelt thanks to my PhD advisor Dr. Deng for his invaluable help, understanding, and support throughout my academic journey. He can always provide me with insightful solutions whenever I face challenges in my research or life. His unwavering passion and enthusiasm for both research and life have profoundly influenced me and shaped my career path. His encouragement and constant support have been fundamental in building my self-assurance and driving me to pursue my goals with determination. I truly could not have reached this point without his mentorship.

I would like to express my deepest gratitude to my committee members, Dr. Zhu, Dr. Hong, and Dr. Xing for their constructive comments, fruitful discussions, and thought-provoking questions that greatly contributed to my research. In addition, I would like to thank Dr. Zhu for providing me with valuable research opportunities for studying the changing trend of cortical thickness of monkeys, a project that enhanced my biological knowledge and computational skills. Also, I am grateful to Dr. Hong for his efforts in coordinating activities at the Virginia Tech Statistics and Artificial Intelligence (VTSAL) lab, which made me feel warm and created a sense of belonging.

I would also like to thank Dr. Leman for his constant help, encouragement, and support during my PhD program. I am particularly appreciative of the weekly statistics quiz activities he organizes, which have greatly enhanced my understanding of the history and background of the field. I am grateful to Dr. Leman for mentoring me in teaching. He is consistently available and willing to guide me and offer advice. Moreover, I would like to express my deepest gratitude to Dr. Leman and Dr. Higdon for their strong support during my job search. I would also extend my sincere gratitude to my friend and collaborator Dr. Tao for his unwavering help and support during my PhD study. I am grateful to Dr. Tao for his insightful suggestions and fruitful discussions about my research, which opened up a new door for me. He is always willing to help, for which I am immensely thankful.

I want to extend my heartfelt thanks to all my dear friends in Blacksburg for the wonderful times we have shared. I enjoyed all activities, like walking on the trail, hiking, eating hot-pot, and others, I cherished every moment. I am also grateful to the faculty and staff at the Department of Statistics at Virginia Tech for fostering such a diverse, warm, and welcoming environment. That makes me feel at home. Additionally, I would like to thank my managers and mentors during my internship at Amazon for their invaluable and professional guidance, which significantly enhanced my ability to engage with advanced research and allowed me to better apply my expertise to real-world applications.

Lastly, I am profoundly grateful to my family, whose unconditional love and support have always been the cornerstone of my journey. To my dear parents, thank you for your endless love, support, and understanding, which have empowered me from childhood to pursue my passions and explore new possibilities. To my grandparents, thank you for your ceaseless care and presence in my life. I would also like to extend my deepest gratitude to my aunt, Dr. Liu, who has provided invaluable help and advice at many critical moments in my life. Without all of you, none of these would have been possible.

Contents

List of Figures

List of Tables

Chapter 1 General Introduction

1.1 Experimental Design

Statistical experimental design plays a pivotal role in optimizing data collection and analysis processes, enabling researchers to obtain meaningful insights and make good inferences from experiments (Box et al., 2005; Wu and Hamada, 2011). Classic experimental design and Bayesian experimental design are two distinct approaches to designing and conducting experiments. In classic experimental design, researchers typically start with a hypothesis and design their experiments to test that hypothesis using frequentist statistics. Various methodologies exist within this framework, including split-plot design (Jones and Nachtsheim, 2009), fractional factorial design (Box and Hunter, 1961), and block design (Addelman, 1969). On the other hand, Bayesian experimental design is rooted in Bayesian statistics and offers a more flexible and iterative approach. It begins with prior beliefs or information about the parameters of interest and updates these beliefs as new data is collected. Bayesian experimental design allows for adaptive sampling, where sample sizes can be adjusted during the experiment based on the accumulating data. This makes it particularly useful in scenarios with limited resources or evolving research questions. Researchers incorporate various criteria to implement optimization, including the maximization of entropy and the minimization of predictive uncertainty (Sebastiani and Wynn, 2000; Leube et al., 2012; Gramacy, 2020). Recent studies have demonstrated the effectiveness of maximizing the expected information gain, commonly referred to as mutual information (Michaud, 2019; Kleinegesse and Gutmann, 2020, 2021).

This proposal highlights the potential of experimental design with application to AI assurance and develops a novel mutual information estimator that enhances experimental design but also finds practical applications in self-supervised learning, meta-learning, and fairness learning. In the following section, I will provide a brief introduction to mutual information

and fractional factorial design, and describe the potential of connections with machine learning.

1.1.1 Mutual Information

Efficient data collection plays a vital role in the fields of data science and machine learning. In the design of experiments (Wu and Hamada, 2011), a good criterion is essential to search for the optimal designs. Traditionally, various criteria like A-optimal (Alexanderian et al., 2016), D-optimal (de Aguiar et al., 1995), and I-optimal (Goos et al., 2016) have been employed, but they heavily depend on specific model assumptions. This limitation makes it challenging to perform analysis and draw inferences in situations where model assumptions are implicit. As a result, there is a growing need for the development of a flexible solution that can be applied when the likelihood of a model is unavailable. The recent works (Foster et al., 2019, 2020, 2021) focused on exploring the Bayesian optimal design via optimizing expected information gain (EIG). However, accurately estimating the EIG remains a challenging task. It is worth noting that the EIG is essentially to quantify the general dependence between pairs of variables, also known as mutual information (MI). Various MI estimation methods have been developed in the literature, such as kernel estimators (Gretton et al., 2003, 2005), naïve density-based estimator(s) and k-nearest neighbor estimators (Kraskov et al., 2004; Pérez-Cruz, 2008; Gao et al., 2015). The differentiability, scalability, and strong performance made variational objectives have been widely utilized recently (Oord et al., 2018). Instead of directly estimating data likelihoods, density ratios, or the corresponding gradients (Wen et al., 2020), variational approaches used mathematical inequalities to construct tractable lower or upper bounds of the mutual information (Poole et al., 2019), facilitated by the use of auxiliary critic functions[1].

[1]When estimates are sharp, these critic functions usually recover some transformation of the likelihood ratio.

1.1.2 Fractional Factorial Design

Considering an experimental design consisting of multiple factors, where each factor has several levels, the full factorial design is the collection of all possible level combinations. When the number of combinations becomes impractical, a fractional factorial design can be employed which narrows down the focus only to a subset of the possible combinations. In the literature, many studies developed methods for selecting a limited set of combinations in a thoughtful manner. For example, (Box and Hunter, 1961) proposed two-level fractional factorial designs to deal with the case that all factors have two levels. Taguchi Orthogonal Array is used to build non-regular fractional factorial design (Taguchi and Konishi, 1987). The work of (Zhou and Xu, 2014) explored space-filling fractional factorial designs under two popular measures discrepancy and maximin distance. Generalized fractional factorial design provided a more general minimum aberration criterion to work for symmetrical and asymmetrical designs, regular and nonregular designs (Wu and Xu, 2001). However, these works often consider all factors equally. In contrast, a recent work of (Sadeghi et al., 2020) investigated an issue in online experiments involving a platform factor with unique characteristics. Furthermore, for a more comprehensive examination of AI assurance, it is important to differentiate the roles of factors in model configuration and hyperparameters, delving into their respective influences on the AI system. To gain a better understanding of this situation with limited resources, fractional factorial designs offer an efficient solution to investigate the impacts of model configuration and hyperparameters.

1.2 Machine Learning

Machine learning algorithms, particularly deep learning, have played a key role in advancing Artificial Intelligence (AI) by enabling systems to recognize patterns, make predictions, and adapt to new information. A central challenge in machine learning revolves around optimizing model performance, an area that can be explored through the investigation of hyperparameters and model configurations (Probst et al., 2019; Lee et al., 2018; Mantovani et al., 2016).

Furthermore, transfer learning is an appealing facet of AI systems, involving leveraging knowledge gained from one task or domain to improve performance on a related but different task or domain (Pan and Yang, 2009). One specific application of transfer learning is few-shot learning, focusing on scenarios where very limited labeled data is available for the target task (Lifchitz et al., 2019). Few-shot learning algorithms aim to generalize from just a few examples and adapt a model's parameters accordingly. Different from the classic methods working on the few-shot learning of vision tasks, this proposal leverages the advanced multimodal large language models to tackle this challenge, which can process and generate content from multiple modalities, such as text and images.

1.2.1 Hyper-parameter Tuning

Hyperparameter tuning is a crucial process in machine learning, involving optimizing model parameters that are not learned directly from data. Unlike trainable model parameters that adjust automatically during training, hyperparameters are predetermined before training and critically influence model performance. They control model structure, and function, directly affecting outcomes. Effective hyperparameter tuning enables data scientists to enhance model performance to achieve optimal results. It is important for AI algorithms (Probst et al., 2019; Lee et al., 2018; Mantovani et al., 2016) but often costly in practice (Hutter et al., 2019). One reason is that searching for the optimal hyperparameter requires training a model multiple times with different combinations of hyperparameters to determine which configuration performs best. Each training iteration can be computationally intensive, especially for complex models like deep neural networks. Also, the search space for hyperparameters is often vast and multidimensional, with each hyperparameter having a range of possible values. The complexity of the search space increases exponentially with the number of hyperparameters, making it more challenging and resource-intensive to explore thoroughly. In the literature, researchers have explored various methods to improve the efficiency in finding the optimal hyperparameter settings, including genetic algorithm (GA) (Lessmann et al., 2005) and particle swarm optimization (PSO) (Lorenzo et al., 2017). Recently, Bayesian optimization has emerged as a

highly effective method because of its effectiveness, especially in complex models such as deep neural networks (Eggensperger et al., 2013; Feurer et al., 2015; Klein et al., 2017).

1.2.2 Multimodal Large Language Model (LLM)

The unification of architectures and the growing availability of pre-trained foundational models have catalyzed surging interests in general-purpose large multimodal foundation models. Substantial progress has been made in the past few years. Pioneering works such as CLIP (Radford et al., 2021) leverage rich multimodal data for self-supervised representation learning, which lays the foundation for various multimodal applications (Fang et al., 2023). While pioneering investigations on augmenting LLM with visual perception are very compute intensive (Tsimpoukelli et al., 2021; Alayrac et al., 2022; Huang et al., 2023; Li et al., 2023c), they demonstrated the potential of leveraging visually augmented LLM as a general purpose tooling for diverse visual language tasks (*e.g.*, VQA, captioning, chat, creative writing, *etc.*). The GPT4 further showed how visual understanding can multiply powerful LLM to massively boost productivity (OpenAI; Yang et al., 2023). Following that, the adoption of adapter architectures, the availability of powerful open LLMs and visual instruction tuning recipes (Li et al., 2023a; Xu et al., 2023; Zhao et al., 2023) have made multimodal LLM more accessible for academic studies and practical deployments (Zhu et al., 2023; Liu et al., 2023a; Dai et al., 2023; Ye et al., 2023; Girdhar et al., 2023; Zhang et al., 2023b,a; Chen et al., 2022; Li et al., 2023b). Relevant to chapter 4 are the works of (Lu et al., 2022; Zhang et al., 2023c), where chain-of-thought reasoning is applied for question answering in multimodal settings. Several researchers (Tsimpoukelli et al., 2021; Alayrac et al., 2022; Huang et al., 2023) also presented few-shot classification examples with multimodal LLM, but those are simple proof-of-concept demonstrations.

1.2.3 Few-shot Learning

Quick and robust learning from limited examples has been a long-standing challenge for Computer Vision (CV) (Fei-Fei et al., 2004, 2006). It can be framed under the context of

meta-learning which broadly refers to techniques that facilitate the learning of a new task in sample-scarce scenarios using prior knowledge from previously seen tasks, such as *learning to learn* (Thrun and Pratt, 1998), *domain adaptation* (Daumé III, 2009), *transfer learning* (Pan and Yang, 2009), *causal machine learning* (Schölkopf et al., 2021), *zero/few-shot learning* (Brown et al., 2020), *weakly supervised learning* (Robinson et al., 2020), etc. Here we focus on methods that are developed in the area of deep learning. Works such as `MAML` (Finn et al., 2017; Raghu et al., 2019; Rajeswaran et al., 2019; Nichol et al., 2018), `MetaNet` (Munkhdalai and Yu, 2017), and `FewTURE` (Hiller et al., 2022) builds a meta-learned a base model during episodic training then adapt to a task-specific model at inference time using gradient updates. Methods like `MatchingNet` (Vinyals et al., 2016), `ProtoNet` (Snell et al., 2017), and `FEAT` (Ye et al., 2020) applied different notations of affinity metrics to align query samples with support sets in the neural embedding spaces, while `MetaOptNet` (Lee et al., 2019) advocates the use of hyperplanes to separate classes. Motivated by the success of transformers, attention-based few-shot learners have also been proposed (Hou et al., 2019; Doersch et al., 2020). Other interesting directions include using Info-Max regularization (Boudiaf et al., 2020) or causal representation learning (Teshima et al., 2020; Xiu et al., 2021) to address the sample scarcity issue.

1.2.4 Task Adaptation

Inference time adaptation is important for improving low-shot learning performance (Hu et al., 2022). In the context of few-shot image classification, task adaptation can be considered as a procedure to reduce cross-class similarities and highlight discriminative features for the given task. This is often achieved by computing the class-prototypes (Snell et al., 2017), or applying (several) gradient updates to edit the input weights (Hiller et al., 2022) or model parameters (Finn et al., 2017). Applying closed-form or iterative solvers can further boost the efficiency and efficacy of the gradient-based adaptations (Bertinetto et al., 2019; Lee et al., 2019). A major drawback with the gradient-based adaptation is that error back-propagation is costly, especially for large networks.

For LLMs, task adaptation can be achieved more flexibly via prompting (Radford et al., 2019; Brown et al., 2020; Liu et al., 2023b). By consuming proper prompts (such as task examples, instructions, or a combination of both), language models are better conditioned for individual tasks, thus yielding substantial performance gains for zero/few-shot applications (Gao et al., 2021). This is more appealing operational-wise as a single generalist model can simultaneously serve many different tasks. Prior studies have shown vision-adapted LLMs also exhibit emergent few-shot learning capability on diverse vision-language tasks (Tsimpoukelli et al., 2021; Alayrac et al., 2022; Huang et al., 2023).

1.3 Overview

The rest of the dissertation is organized as follows. Chapter 2 presents a novel nonparametric mutual information estimator named `FLO`, applicable across a wide range of modern data science applications. Chapter 3 introduces a new design method multi-layer sliced design (MLSD) with application in AI assurance. This proposed method enables us to explore the influence of model configurations and hyper-parameters. Chapter 4 will investigate the classic vision classification tasks through advanced multi-modal models. Chapter 5 concludes the main results of the proposal and discusses several future research topics.

Bibliography

S. Addelman. The generalized randomized block design. *The American Statistician*, 23(4): 35–36, 1969.

J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

A. Alexanderian, N. Petra, G. Stadler, and O. Ghattas. A fast and scalable method for a-optimal design of experiments for infinite-dimensional bayesian nonlinear inverse problems. *SIAM Journal on Scientific Computing*, 38(1):A243–A272, 2016.

L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. In *ICLR*, 2019.

M. Boudiaf, Z. I. Masud, J. Rony, J. Dolz, P. Piantanida, and I. B. Ayed. Transductive information maximization for few-shot learning. In *NeurIPS*, 2020.

G. E. Box and J. S. Hunter. The 2 k—p fractional factorial designs. *Technometrics*, 3(3): 311–351, 1961.

G. E. Box, J. S. Hunter, W. G. Hunter, et al. Statistics for experimenters. In *Wiley series in probability and statistics*. Wiley Hoboken, NJ, 2005.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

J. Chen, H. Guo, K. Yi, B. Li, and M. Elhoseiny. Visualgpt: Data-efficient adaptation of pretrained language models for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18030–18040, 2022.

W. Dai, J. Li, D. Li, A. Meng Huat Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*, 2023.

H. Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.

P. F. de Aguiar, B. Bourguignon, M. Khots, D. Massart, and R. Phan-Than-Luu. D-optimal designs. *Chemometrics and intelligent laboratory systems*, 30(2):199–210, 1995.

C. Doersch, A. Gupta, and A. Zisserman. Crosstransformers: spatially-aware few-shot transfer. *Advances in Neural Information Processing Systems*, 33:21981–21993, 2020.

K. Eggensperger, M. Feurer, F. Hutter, J. Bergstra, J. Snoek, H. Hoos, K. Leyton-Brown, et al. Towards an empirical foundation for assessing bayesian optimization of hyperparameters. In *NIPS workshop on Bayesian Optimization in Theory and Practice*, volume 10, 2013.

Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023.

L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

M. Feurer, J. Springenberg, and F. Hutter. Initializing bayesian hyperparameter optimization via meta-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135. PMLR, 2017.

A. Foster, M. Jankowiak, E. Bingham, P. Horsfall, Y. W. Teh, T. Rainforth, and N. Goodman. Variational bayesian optimal experimental design. *Advances in Neural Information Processing Systems*, 32, 2019.

A. Foster, M. Jankowiak, M. O'Meara, Y. W. Teh, and T. Rainforth. A unified stochastic gradient approach to designing bayesian-optimal experiments. In *International Conference on Artificial Intelligence and Statistics*, pages 2959–2969. PMLR, 2020.

A. Foster, D. R. Ivanova, I. Malik, and T. Rainforth. Deep adaptive design: Amortizing sequential bayesian experimental design. In *International Conference on Machine Learning*, pages 3384–3395. PMLR, 2021.

S. Gao, G. Ver Steeg, and A. Galstyan. Efficient estimation of mutual information for strongly dependent variables. In *AISTATS*, 2015.

T. Gao, A. Fisch, and D. Chen. Making pre-trained language models better few-shot learners. In *ACL*, 2021.

R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023.

P. Goos, B. Jones, and U. Syafitri. I-optimal design of mixture experiments. *Journal of the American Statistical Association*, 111(514):899–911, 2016.

R. B. Gramacy. *Surrogates: Gaussian process modeling, design, and optimization for the applied sciences*. CRC press, 2020.

A. Gretton, R. Herbrich, and A. J. Smola. The kernel mutual information. In *ICASSP*, 2003.

A. Gretton, R. Herbrich, A. Smola, O. Bousquet, B. Schölkopf, et al. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 2005.

M. Hiller, R. Ma, M. Harandi, and T. Drummond. Rethinking generalization in few-shot classification. *Advances in Neural Information Processing Systems*, 35:3582–3595, 2022.

R. Hou, H. Chang, B. Ma, S. Shan, and X. Chen. Cross attention network for few-shot classification. *Advances in neural information processing systems*, 32, 2019.

S. X. Hu, D. Li, J. Stühmer, M. Kim, and T. M. Hospedales. Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference. In *CVPR*, pages 9068–9077, 2022.

S. Huang, L. Dong, W. Wang, Y. Hao, S. Singhal, S. Ma, T. Lv, L. Cui, O. K. Mohammed, Q. Liu, et al. Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*, 2023.

F. Hutter, L. Kotthoff, and J. Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

B. Jones and C. J. Nachtsheim. Split-plot designs: What, why, and how. *Journal of quality technology*, 41(4):340–361, 2009.

A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter. Fast bayesian optimization of machine learning hyperparameters on large datasets. In *Artificial intelligence and statistics*, pages 528–536. PMLR, 2017.

S. Kleinegesse and M. U. Gutmann. Bayesian experimental design for implicit models by mutual information neural estimation. In *International Conference on Machine Learning*, pages 5316–5326. PMLR, 2020.

S. Kleinegesse and M. U. Gutmann. Gradient-based bayesian experimental design for implicit models using mutual information lower bounds. *arXiv preprint arXiv:2105.04379*, 2021.

A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

K. Lee, S. Maji, A. Ravichandran, and S. Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, pages 10657–10665, 2019.

W.-Y. Lee, S.-M. Park, and K.-B. Sim. Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm. *Optik*, 172: 359–367, 2018.

S. Lessmann, R. Stahlbock, and S. F. Crone. Optimizing hyperparameters of support vector machines by genetic algorithms. In *IC-AI*, volume 74, page 82, 2005.

P. Leube, A. Geiges, and W. Nowak. Bayesian assessment of the expected data impact on prediction confidence in optimal sampling design. *Water Resources Research*, 48(2), 2012.

B. Li, Y. Zhang, L. Chen, J. Wang, F. Pu, J. Yang, C. Li, and Z. Liu. Mimic-it: Multi-modal in-context instruction tuning. *arXiv preprint arXiv:2306.05425*, 2023a.

B. Li, Y. Zhang, L. Chen, J. Wang, J. Yang, and Z. Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023b.

J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023c.

Y. Lifchitz, Y. Avrithis, S. Picard, and A. Bursuc. Dense classification and implanting for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9258–9267, 2019.

H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023a.

P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. Pre-train, prompt, and predict:

A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023b.

P. R. Lorenzo, J. Nalepa, M. Kawulok, L. S. Ramos, and J. R. Pastor. Particle swarm optimization for hyper-parameter selection in deep neural networks. In *Proceedings of the genetic and evolutionary computation conference*, pages 481–488, 2017.

P. Lu, S. Mishra, T. Xia, L. Qiu, K.-W. Chang, S.-C. Zhu, O. Tafjord, P. Clark, and A. Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

R. G. Mantovani, T. Horváth, R. Cerri, J. Vanschoren, and A. C. de Carvalho. Hyper-parameter tuning of a decision tree induction algorithm. In *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 37–42. IEEE, 2016.

I. J. Michaud. *Simulation-based bayesian experimental design using mutual information*. North Carolina State University, 2019.

T. Munkhdalai and H. Yu. Meta networks. In *ICML*, pages 2554–2563. PMLR, 2017.

A. Nichol, J. Achiam, and J. Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

OpenAI. Gpt-4v(ision) system card.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

F. Pérez-Cruz. Estimation of information theoretic measures for continuous random variables. In *NIPS*, 2008.

B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker. On variational bounds of mutual information. In *ICML*. PMLR, 2019.

P. Probst, M. N. Wright, and A.-L. Boulesteix. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, 9 (3):e1301, 2019.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

A. Raghu, M. Raghu, S. Bengio, and O. Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

A. Rajeswaran, C. Finn, S. Kakade, and S. Levine. Meta-learning with implicit gradients. 2019.

J. Robinson, S. Jegelka, and S. Sra. Strength from weakness: Fast learning using weak supervision. In *International Conference on Machine Learning*, pages 8127–8136. PMLR, 2020.

S. Sadeghi, P. Chien, and N. Arora. Sliced designs for multi-platform online experiments. *Technometrics*, 62(3):387–402, 2020.

B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

P. Sebastiani and H. P. Wynn. Maximum entropy sampling and optimal bayesian experimental design. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(1): 145–157, 2000.

J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

G. Taguchi and S. Konishi. *Orthogonal arrays and linear graphs: tools for quality engineering*. American Supplier Institute, 1987.

T. Teshima, I. Sato, and M. Sugiyama. Few-shot domain adaptation by causal mechanism transfer. In *ICML*, pages 9458–9469. PMLR, 2020.

S. Thrun and L. Pratt. Learning to learn: Introduction and overview. In *Learning to learn*, pages 3–17. Springer, 1998.

M. Tsimpoukelli, J. L. Menick, S. Cabi, S. Eslami, O. Vinyals, and F. Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.

O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

L. Wen, Y. Zhou, L. He, M. Zhou, and Z. Xu. Mutual information gradient estimation for representation learning. In *ICLR*, 2020.

C. Wu and H. Xu. Generalized minimum aberration for asymmetrical fractional factorial designs. *The Annals of Statistics*, 29(2):549–560, 2001.

C. J. Wu and M. S. Hamada. *Experiments: planning, analysis, and optimization*. John Wiley & Sons, 2011.

Z. Xiu, J. Chen, R. Henao, B. Goldstein, L. Carin, and C. Tao. Supercharging imbalanced data learning with energy-based contrastive representation transfer. In *NeurIPS*, 2021.

Z. Xu, Y. Shen, and L. Huang. Multiinstruct: Improving multi-modal zero-shot learning via instruction tuning. In *ACL*, 2023.

Z. Yang, L. Li, K. Lin, J. Wang, C.-C. Lin, Z. Liu, and L. Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 2023.

H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8808–8817, 2020.

Q. Ye, H. Xu, G. Xu, J. Ye, M. Yan, Y. Zhou, J. Wang, A. Hu, P. Shi, Y. Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023.

A. Zhang, H. Fei, Y. Yao, W. Ji, L. Li, Z. Liu, and T.-S. Chua. Transfer visual prompt generator across llms. *arXiv preprint arXiv:2305.01278*, 2023a.

R. Zhang, J. Han, A. Zhou, X. Hu, S. Yan, P. Lu, H. Li, P. Gao, and Y. Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023b.

Z. Zhang, A. Zhang, M. Li, H. Zhao, G. Karypis, and A. Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023c.

B. Zhao, B. Wu, and T. Huang. Svit: Scaling up visual instruction tuning. *arXiv preprint arXiv:2307.04087*, 2023.

Y.-D. Zhou and H. Xu. Space-filling fractional factorial designs. *Journal of the American Statistical Association*, 109(507):1134–1144, 2014.

D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

Chapter 2 Tight Mutual Information Estimation With Contrastive Fenchel-Legendre Optimization

Abstract

Successful applications of InfoNCE (Information Noise-Contrastive Estimation) and its variants have popularized the use of contrastive variational mutual information (MI) estimators in machine learning. While featuring superior stability, these estimators crucially depend on costly large-batch training, and they sacrifice bound tightness for variance reduction. To overcome these limitations, we revisit the mathematics of popular variational MI bounds from the lens of unnormalized statistical modeling and convex optimization. Our investigation yields a new unified theoretical framework encompassing popular variational MI bounds and leads to a new simple and powerful contrastive MI estimator we name `FLO`. Theoretically, we show that the `FLO` estimator is tight, and it converges under stochastic gradient descent. Empirically, the `FLO` estimator overcomes the limitations of its predecessors and learns more efficiently. The utility of `FLO` is verified using extensive benchmarks, and we further inspire the community with novel applications in meta-learning. Our presentation underscores the foundational importance of variational MI estimation in data-efficient learning.

Key Words: Mutual Information, Bayesian model, Experimental design, Variational inference, Contrastive learning

2.1 Introduction

Assessing the dependence between pairs of variables is integral to many scientific and engineering endeavors (Reshef et al., 2011; Shannon, 1948). *Mutual information* (MI) is a popular metric to quantify generic associations (MacKay, 2003), and its empirical estimators have been widely used in applications such as independent component analysis (Bach and Jordan, 2002), fair learning (Gupta et al., 2021), neuroscience (Palmer et al., 2015), Bayesian optimization (Kleinegesse and Gutmann, 2020), among others. Notably, the recent advances in deep *self-supervised learning* (SSL) heavily rely on nonparametric MI optimization (Tishby and Zaslavsky, 2015; Oord et al., 2018; He et al., 2020; Chen et al., 2020; Grill et al., 2020). In this study, we investigate the likelihood-free variational approximation of MI using only paired samples and improve the data-efficiency of current machine learning practices.

MI estimation has been extensively studied (Battiti, 1994; Maes et al., 1997; MacKay, 2003; Paninski, 2003; Pluim et al., 2003; Torkkola, 2003). While most classical estimators work reasonably well for low-dimensional cases, they scale poorly to big datasets: naïve density-based estimator(s) and k-nearest neighbor estimators (Kraskov et al., 2004; Pérez-Cruz, 2008; Gao et al., 2015) struggle with high-dimensional inputs, while kernel estimators are slow, memory demanding and sensitive to hyperparameters (Gretton et al., 2003, 2005). Moreover, these estimators are usually either non-differentiable or need to hold all data in memory. Consequently, they are not well suited for emerging applications where the data representation needs to be differentiably optimized based on small-batch estimation of MI (Hjelm et al., 2019). Alternatively, one can approach MI estimation through an estimated likelihood ratio (Suzuki et al., 2008; Hjelm et al., 2019), but the associated numerical instability has raised concerns (Arjovsky and Bottou, 2017).

To scale MI estimation to the growing size and complexity of modern datasets, and to accommodate the need for representation optimization (Bengio et al., 2013), variational objectives have been widely utilized recently (Oord et al., 2018). Instead of directly estimating data likelihoods, density ratios, or the corresponding gradients (Wen et al., 2020), variational

approaches appeal to mathematical inequalities to construct tractable lower or upper bounds of the mutual information (Poole et al., 2019), facilitated by the use of auxiliary critic functions[1]. This practice turns MI estimation into an optimization problem. Prominent examples include the *Barber-Agakov* (BA) estimator (Barber and Agakov, 2004), the *Donsker-Varadhan* (DV) estimator (Donsker and Varadhan, 1983), and the *Nguyen-Wainwright-Jordan* (NWJ) estimator (Nguyen et al., 2010). These variational estimators are closely connected to the variational objectives for likelihood inference (Alemi et al., 2018).

Despite reported successes, these variational estimators have a major limitation: their estimation variance grows exponentially to the ground-truth MI (McAllester and Stratos, 2018). This is especially harmful to applications involving deep neural nets, as it largely destabilizes training (Song and Ermon, 2020). An effective fix is to leverage multi-sample contrastive estimators, pioneered by the work of `InfoNCE` (Oord et al., 2018). However, the massive reduction in the variance comes at a price: the performance of the `InfoNCE` estimator is upper bounded by $\log K$, where K is the number of *negative* samples used (Poole et al., 2019). For a large MI, K needs to be sufficiently large to allow for an adequate estimate, consequently placing a significant burden on computation and memory. While variants of `InfoNCE` have been motivated to achieve more controllable bias and variance tradeoffs (Poole et al., 2019; Song and Ermon, 2020), little research has been conducted on the cost-benefit aspect of contrastive learning.

[1]When estimates are sharp, these critic functions usually recover some transformation of the likelihood ratio.

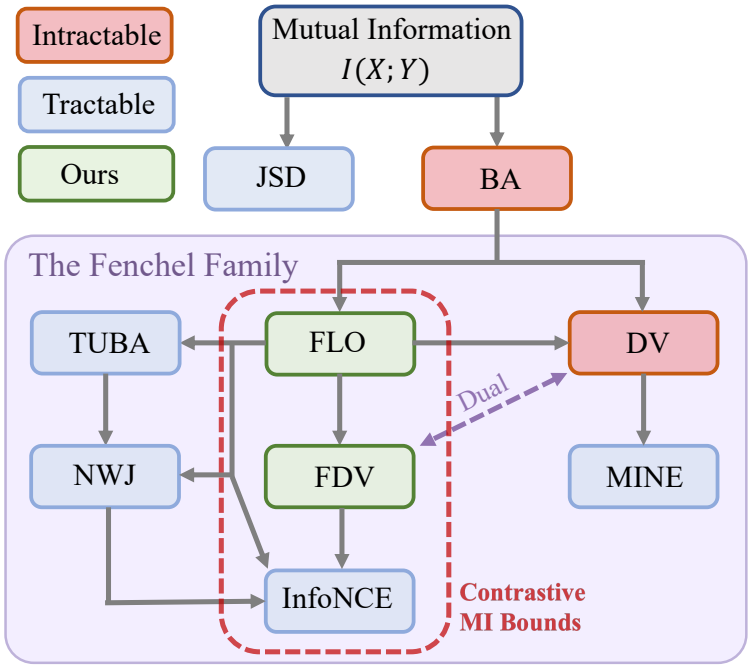

Figure 2.1: Schematic of variational lower bounds of mutual information. `FLO` provides a novel unified framework to analyze contrastive MI bounds.

A critical insight enabled by `InfoNCE` is that mutual information closely connects to contrastive learning (Gutmann and Hyvärinen, 2010; Oord et al., 2018). Paralleled by the empirical successes of instance discrimination-based self-supervision (Mnih and Kavukcuoglu, 2013; Wu et al., 2018; Chen et al., 2020; He et al., 2020) and multi-view supervision (Tian et al., 2019; Radford et al., 2021), `InfoNCE` offers an *InfoMax* explanation to why the ability to discriminate naturally paired *positive* instances from the randomly paired *negative* instances leads to universal performance gains in these applications (Linsker, 1988; Shwartz-Ziv and Tishby, 2017; Poole et al., 2019). Despite these encouraging developments, the big picture of MI optimization and contrastive learning is not yet complete: (*i*) There is an ongoing debate about to what extent MI optimization helps to learn (Tschannen et al., 2020); (*ii*) how does the contrastive view reconcile with those non-contrastive MI estimators; crucial for practical applications, (*iii*) are the empirical tradeoffs made by estimators such as `InfoNCE` absolutely necessary? And theoretically, (*iv*) formal guarantees on the statistical convergence of popular variational non-parametric MI estimation are missing currently.

In this work, we seek to bridge the above gaps by approaching the MI estimation from the

novel perspective of energy modeling. While this subject has recently been studied extensively using information-theoretic and variational inequalities, we embrace a new view from the lens of unnormalized statistical modeling. Our main contributions include:

- Unifying popular variational MI bounds under unnormalized statistical modeling;
- Deriving a simple but powerful novel contrastive variational bound called `FLO`;
- Providing theoretical justification of the `FLO` bound (tightness and convergence);
- Demonstrating strong empirical evidence of the superiority of `FLO` over its predecessors.
- Highlighting the importance of MI in data-efficient learning with novel applications

We contribute in-depth discussion to bridge the gaps between contrastive learning and MI estimation, along with principled practical guidelines informed by theoretical insights.

2.2 Fenchel-Legendre Optimization for Mutual Information Estimation

2.2.1 Preliminaries

Unnormalized statistical modeling defines a rich class of models of general interest. Specifically, we are interested in problems for which the system is characterized by an energy function $\tilde{p}_\theta(x) = \exp(-\psi_\theta(x))$, where θ is the system parameters and $\psi_\theta(x)$ is known as the *potential function*. The goal is to find a solution that is defined by a normalized version of $\tilde{p}_\theta(x)$, *i.e.*, $\min_\theta \left\{ \mathcal{L}\left(\frac{\tilde{p}_\theta}{\int \tilde{p}_\theta(x') \, \mathrm{d}\mu(x')} \right) \right\}$, where $\mathcal{L}(\cdot)$ is the loss function, μ is the base measure on $\mathcal{X}$ and $Z(\theta) \triangleq \int \tilde{p}_\theta(x') \, \mathrm{d}\mu(x')$ is called the *partition function* for $\tilde{p}_\theta(x)$. Problems in the above form arise naturally in statistical physics (Reichl, 2016), Bayesian analysis (Berger, 2013), and maximal likelihood estimation (Tao et al., 2019). A major difficulty with unnormalized statistical modeling is that the partition function $Z(\theta)$ is generally intractable for complex energy functions [2], and in many applications $Z(\theta)$ is further composed by $\log Z(\theta)$, whose

[2] In the sense that they do not render closed-from expressions.

concavity implies any finite sample estimate Monte-Carlo of $Z(\theta)$ will render the loss function biased (Rainforth et al., 2018; Zheng et al., 2018). Bypassing the difficulties caused by the intractable partition function is central to unnormalized statistical modeling (Geyer, 1994; Neal, 2001; Hinton, 2002; Hyvärinen, 2005; Gutmann and Hyvärinen, 2010).

Mutual information and unnormalized statistical models. As a generic score assessing the dependency between two random variables (X, Y), *mutual information* is formally defined as the *Kullback-Leibler divergence* (KL) between the joint distribution $p(x, y)$ and product of the respective marginals $p(x)p(y)$ (Shannon, 1948), *i.e.*, $I(X;Y) \triangleq \mathbb{E}_{p(x,y)}\left[\log \frac{p(x,y)}{p(x)p(y)}\right]$. The integrand $\log \frac{p(x,y)}{p(x)p(y)}$ is often known as the *point-wise mutual information* (PMI) in the literature. Mutual information has a few appealing properties: (i) it is invariant wrt invertible transformations of x and y, and (ii) it has the intuitive interpretation of reduced uncertainty of one variable given another variable[3]

To connect MI to unnormalized statistical modeling, we consider the classical *Barber-Agakov* (`BA`) estimator of MI (Barber and Agakov, 2003). To lower bound MI, `BA` introduces a variational approximation $q(y|x)$ for the posterior $p(y|x)$, and by rearranging the terms we obtain an inequality

$$\begin{aligned} I(X;Y) &= \mathbb{E}_{p(x,y)}\left[\log \frac{p(y|x)}{p(y)}\right] = \mathbb{E}_{p(x,y)}\left[\log \frac{q(y|x)}{p(y)}\right] + \mathbb{E}_{p(x)}[\mathrm{KL}(p(y|x) \parallel q(y|x))] \\ &\geq \mathbb{E}_{p(x,y)}\left[\log \frac{q(y|x)}{p(y)}\right] \triangleq I_{\texttt{BA}}(X;Y|q). \end{aligned} \tag{2.1}$$

Here we have used notation $I_{\texttt{BA}}(X;Y|q)$ to highlight the dependence on $q(y|x)$, and when $q(y|x) = p(y|x)$ this bound is sharp. Unfortunately, this naïve `BA` bound is not useful for sample-based MI estimation, as we do not know the ground-truth $p(y)$. But we can bypass this difficulty by setting $q_\theta(y|x) = \frac{p(y)}{Z_\theta(x)} e^{g_\theta(x,y)}$, where we call $e^{g_\theta(x,y)}$ the *tilting function* and recognize $Z_\theta(x) = \mathbb{E}_{p(y)}[e^{g_\theta(x,y)}]$ as the associated partition function. Substituting this $q_\theta(x|y)$ into (2.18) gives the following *unnormalized* ***BA*** bound (`UBA`) that pertains to unnormalized

[3]Formally, $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$, where $H(X)$ (resp. $H(X|Y)$) denotes the Shannon entropy (resp. conditional Shannon entropy) of a random variable.

statistical modeling (Poole et al., 2019)

$$I_{\texttt{UBA}}(X;Y|g_\theta) \triangleq \mathbb{E}_{p(x,y)}[g_\theta(x,y) - \log Z_\theta(x)] = \mathbb{E}_{p(x)}\left[\mathbb{E}_{p(y|x)}\left[\log \frac{e^{g_\theta(x,y)}}{Z_\theta(x)}\right]\right]. \tag{2.2}$$

While this UBA bound remains intractable, now with $Z_\theta(x)$ instead of $p(y)$ we can apply different techniques for empirical estimates of $Z_\theta(x)$ to render a tractable surrogate target. This has led to various popular MI bounds listed in Table 2.1 (see Appendix 2.5.1 for derivations).

InfoNCE and noise contrastive estimation. InfoNCE is a multi-sample mutual information estimator proposed in (Oord et al., 2018), built on the idea of *noise contrastive estimation* (NCE) (Gutmann and Hyvärinen, 2010). NCE learns statistical properties of a target distribution by comparing the *positive* samples from the target distribution to the "*negative*" samples from a carefully crafted noise distribution, and this technique is also known as *negative sampling* in some contexts (Mnih and Kavukcuoglu, 2013; Grover and Leskovec, 2016). The InfoNCE estimator implements this contrastive estimation idea via using the naïve empirical estimate of $Z_\theta(x)$ in UBA[4], *i.e.*

$$I^K_{\texttt{InfoNCE}}(X;Y|g_\theta) \triangleq \mathbb{E}_{p^K(x,y)}\left[\log \frac{e^{g_\theta(x_1,y_1)}}{\frac{1}{K}\sum_j e^{g_\theta(x_1,y_j)}}\right], I^K_{\texttt{InfoNCE}}(X;Y) \triangleq \max_{g_\theta \in \mathcal{F}} \{I^K_{\texttt{InfoNCE}}(X;Y|g_\theta)\}, \tag{2.3}$$

where g_θ is known as the *critic* in the nomenclature of contrastive learning, and we have used $p^K(x,y)$ to denote K independent draws from the joint density $p(x,y)$, and $\{(x_k,y_k)\}_{k=1}^K$ for each pair of samples. Here the positive and negative samples are respectively drawn from the joint $p(x,y)$ and product of marginals $p(x)p(y)$. Intuitively, InfoNCE tries to accurately classify the positive samples when they are mixed with negative samples, and the Proposition below formally characterizes InfoNCE's statistical properties as an MI estimator.

Proposition 2.2.1 ((Poole et al., 2019))**.** InfoNCE is an asymptotically tight mutual information lower bound, *i.e.* $I^K_{\texttt{InfoNCE}}(X;Y|g_\theta) \leq I(X;Y)$, $\lim_{K\to\infty} I^K_{\texttt{InfoNCE}}(X;Y) \to I(X;Y)$.

[4]This estimator is technically equivalent to the original definition due to the symmetry of K samples.

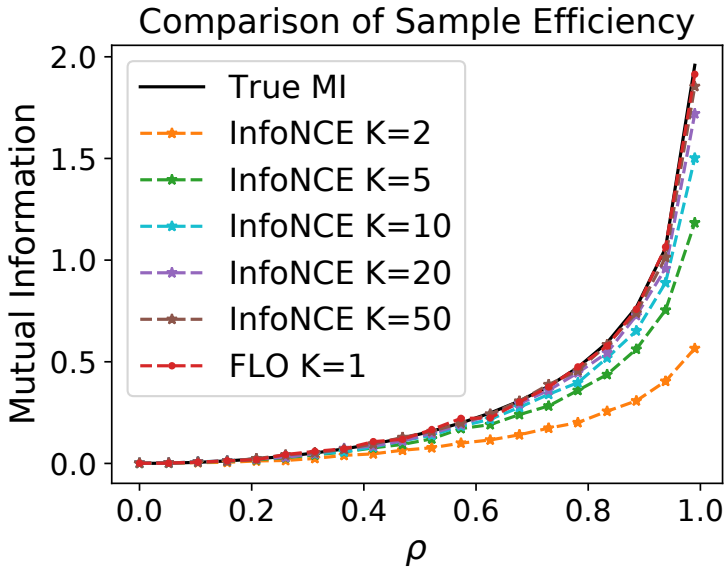

Figure 2.2: K-sample `InfoNCE` and single-sample `FLO`. Note `FLO` is tight regardless of sample size.

Fenchel-Legendre duality. Our key idea is to exploit the convex duality for MI estimation. Let $f(t)$ be a proper convex, lower-semicontinuous function; then its convex conjugate function is defined as $f^*(v) \triangleq \sup_{t\in\mathcal{D}(f)}\{tv - f(t)\}$, where $\mathcal{D}(f)$ is the domain of function f (Hiriart-Urruty and Lemaréchal, 2012). We call $f^*(v)$ the *Fenchel conjugate* of $f(t)$, which is also known as the *Legendre transform* in physics. The Fenchel conjugate pair (f, f^*) are dual to each other, in the sense that $f^{**} = f$, *i.e.*, $f(t) = \sup_{v\in\mathcal{D}(f^*)}\{vt - f^*(v)\}$. For $f(t) = -\log(t)$ and its Fenchel conjugate $f^*(v) = -1 - \log(-v)$, we have inequality

$$-\log(t) \geq -u - e^{-u}t + 1, \quad \text{for } u \in \mathbb{R} \tag{2.4}$$

with the equality holds when $u = \log(t)$.

2.2.2 Fenchel-Legendre Optimization for tight mutual information estimation

With the above mathematical tools, we are ready to present the main result of this paper: a tight, data-efficient variational MI lower bound that can be efficiently implemented.

Lower bounding MI with Fenchel-Legendre Optimization. Our key insight is that MI estimation is essentially an unnormalized statistical model, which can be efficiently handled

by the Fenchel-Legendre transform technique. Take the integrand from UBA in (2.21) and we can rewrite it as

$$\log \frac{\exp(g_\theta(x,y))}{Z_\theta(x)} = -\log\left\{\mathbb{E}_{p(y')}[\exp(g(x,y') - g(x,y))]\right\}, \tag{2.5}$$

where $p(y')$ is the same probability density as $p(y)$ (*i.e.*, Y' is an independent copy of Y). Now let us use the Fenchel inequality of $-\log(t)$ from (2.4), plugging it into the above equation and then we have

$$\log \frac{\exp(g_\theta(x,y))}{Z_\theta(x)} \geq \left\{-u - e^{-u}\mathbb{E}_{p(y')}[\exp(g(x,y') - g(x,y))]\right\} + 1. \tag{2.6}$$

for all $u \in \mathbb{R}$. This implies for any function $u_\phi(x,y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, the following inequality holds

$$\log \frac{\exp(g_\theta(x,y))}{Z_\theta(x)} \geq -\{u_\phi(x,y) + e^{-u_\phi(x,y)}\mathbb{E}_{p(y')}[\exp(g(x,y') - g(x,y))]\} + 1. \tag{2.7}$$

By putting (2.7) back to (2.21), we obtain our new Fenchel-Legendre Optimization (FLO) MI lower bound

$$I_{\texttt{FLO}}(X;Y|g_\theta,u_\phi) \triangleq \mathbb{E}_{p(x,y)}\left[-\{u_\phi(x,y) + e^{-u_\phi(x,y)}\mathbb{E}_{p(y')}[e^{g_\theta(x,y')-g_\theta(x,y)}]\}\right] + 1, \tag{2.8}$$

and concludes the proof for the following Proposition.

Proposition 2.2.2. $I_{\texttt{FLO}}(X;Y|g_\theta,u_\phi) \leq I_{\texttt{UBA}}(X;Y|g_\theta) \leq I(X;Y)$.

In practice, FLO can be estimated with the following naïve empirical K-sample estimator

$$\hat{I}^K_{\texttt{FLO}}(X;Y|g_\theta,u_\phi) \triangleq -\left\{u_\phi(x_i,y_i) + e^{-u_\phi(x_i,y_i)}\frac{1}{K-1}\sum_{j\neq i} e^{g_\theta(x_i,y_j)-g_\theta(x_i,y_i)}\right\} + 1. \tag{2.9}$$

Since the summation in $\hat{I}^K_{\texttt{FLO}}$ is not encapsulated by a convex log transformation, $I^K_{\texttt{FLO}} \triangleq \mathbb{E}_{p^K}[\hat{I}^K_{\texttt{FLO}}]$ is an unbiased estimator for $I_{\texttt{FLO}}(X;Y|g_\theta,u_\phi)$ independent of the batch size K (see

Figure 2.2).

Why is the FLO bound more appealing? At first sight, it may appear counter-intuitive that $I_{\texttt{FLO}}$ is a better MI bound compared to prior arts such as NWJ or InfoNCE: it seems to be more complicated as an extra variational function $u_\phi(x, y)$ has been introduced. To answer this question, we next explain the statistical meaning of the newly introduced $u_\phi(x, y)$, and establish some important statistical properties of FLO that make it more favorable: that $I_{\texttt{FLO}}$ is tight, meaning the ground-truth MI can be recovered for some specific choice of $g_\theta(x, y)$ and $u_\phi(x, y)$; and that $I^K_{\texttt{FLO}}$ for any batch size K is effectively optimizing InfoNCE with an infinite batch size. In Sec 2.2.4, we further justify FLO's advantages from optimization perspectives.

Given the close connection between FLO and UBA, we first recall UBA's optimal critic that gives the tight MI estimate is $g^*(x, y) = \log p(x|y) + c(x)$, where this $c(x)$ can be any function of x (Ma and Collins, 2018). This $g^*(x, y)$ is not directly meaningful in a statistical sense, however, by integrating out y', we have

$$\mathbb{E}_{p(y')}\left[e^{g^*(x,y')-g^*(x,y)}\right] = \mathbb{E}_{p(y')}\left[\frac{p(x|y')}{p(x|y)}\right] = \frac{p(x)}{p(x|y)} = \frac{p(x)p(y)}{p(x,y)}, \tag{2.10}$$

which is the likelihood ratio between the marginals and joints. On the other hand, based on the Fenchel-Legendre inequality (2.4), we know for fixed $g(x, y)$ our FLO bound in (2.8) can be maximized with $u_g(x, y) = \log \mathbb{E}_{p(y')}\left[e^{g(x,y')-g(x,y)}\right]$. Putting these all together we have $u_{g^*}(x, y) = -\log \frac{p(x,y)}{p(x)p(y)}$.

This shows the $u_\phi(x, y)$ introduced in FLO actually tries to recover the negative PMI. Comparing to the competing MI bounds that only optimize for g_θ, eliminating the drift term $c(x)$ reveals FLO enjoys the appealing *self-normalizing* property (Gutmann and Hyvärinen, 2010) that helps stabilize training. Plugging (g^*, u_{g^*}) into (2.8), we readily see $I_{\texttt{FLO}}(X; Y|u_{g^*}, g^*) = I(X; Y)$, proving FLO is a tight MI bound.

Proposition 2.2.3. The FLO estimator is tight, the eqaulity holds when $g(x, y) = \log p(x|y) + c(x)$ for arbitrary function $c(x)$ and $u(x, y) = -\log \frac{p(x,y)}{p(x)p(y)}$.

Corollary 2.2.4. Let (g^*, u_{g^*}) be the maximizers for (2.8), then $I(X; Y) = \mathbb{E}_{p(x,y)}[-u_{g^*}(x, y)]$.

Finally, we give a simple asymptotic argument showing FLO essentially optimizes InfoNCE with an infinite batch size. In virtue of the law of large numbers, we have the denominator in InfoNCE converging to $\lim_{K\to\infty} \frac{1}{K}\sum_{j=1}^{K} e^{g_\theta(x_i,y_j)} \to \mathbb{E}_{p(y')}[e^{g_\theta(x_i,y')}] = Z_\theta(x_i)$, and consequently it recovers the UBA bound. Since FLO is derived from UBA, we can view FLO as using the optimization of $u_\phi(x,y)$ to amortize the difficulty of evaluating infinite number of $e^{g_\theta(x_i,y_j)}$ with InfoNCE.

Efficient implementations of FLO. A lingering concern is that the newly introduced $u_\phi(x,y)$ can incur extra computation overhead. This is not true, as we can maximally encourage parameter sharing by jointly model $u_\phi(x,y)$ and $g_\theta(x,y)$ with a single neural network $f_\Psi(x,y) : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}^2$ with two output heads, *i.e.*, $[u_i, g_i] = f_\Psi(x_i, y_i)$. Consequently, while FLO adopts a dual critics design, it does not actually invoke extra modeling cost compared to its single-critic counterparts (*e.g.*, InfoNCE). Experiments show this shared parameterization in fact promotes synergies and speeds up learning (see our ablation studies in Appendix).

To further enhance the computation efficiency, we consider a massively parallelized *bilinear* critic design that uses all in-batch samples as negatives. Let $g_\theta(x,y) = \tau\cdot\langle h_\theta(x), \tilde{h}(y)\rangle$, where $h : \mathcal{X}\to\mathbb{S}^p$ and $\tilde{h} : \mathcal{Y}\to\mathbb{S}^p$ are respectively encoders that map data to unit sphere $\mathbb{S}^p$ embedded in $\mathbb{R}^{p+1}$, $\langle a,b\rangle = a^T b$ is the inner product operation, and $\tau > 0$ is the inverse temperature parameter. Thus the evaluation of the *Gram* matrix $G = \tau\cdot h(\mathbb{X})^T\tilde{h}(\mathbb{Y})$, where $[\mathbb{X},\mathbb{Y}] \in \mathbb{R}^{K\times(d_x+d_y)}$ is a mini-batch of K-paired samples and $g_\theta(x_i,y_j) = G_{ij}$, can be parallelized via matrix multiplication. In this setup, the diagonal terms of G are the positive scores while the off-diagonal terms are negative scores. A similar strategy has been widely employed in the contrastive representation learning literature (*e.g.*, (Chen et al., 2020))[5]. We can simply model the PMI critic as $u(x,y) = \texttt{MLP}(h(x), \tilde{h}(y))$, whose computation cost is almost neglectable in practice, where feature encoders $h, \tilde{h}$ dominate computing.

[5]As an important note to the community, most open source implementations for the bilinear contrastive loss have mechanically implemented $\frac{1}{T}\langle\cdot,\cdot\rangle$ following the practice from pioneering contrastive learning studies, which is numerically unstable compared to our parameterization $\tau\langle\cdot,\cdot\rangle$ proposed here.

Table 2.1: Comparison of popular variational MI estimators. Here $g(x,y), u(x,y)$ and $u(x)$ are variational functions to be optimized, $\sigma(u) = \frac{1}{1+\exp(-u)}$ is the Sigmoid function, $\mathcal{E}[f(u), \eta]$ denotes exponential average of function $f(u)$ with decay parameter $\eta \in (0,1)$, and $\alpha \in [0,1]$ is the balancing parameter used by α-`InfoNCE` trading off bias and variance between `InfoNCE` and `TUBA`. we use (x_i, y_i) to denote positive samples from the joint density $p(x,y)$, and (x_i, y_j) or (x'_k, y'_k) to denote negative samples drawn from the product of marginal $p(x)p(y)$. In context, $y_\oplus$ and $y_\ominus$ have the intuitive interpretation of positive and negative samples. We exclude variational upper bounds here because their computations typically involve the explicit knowledge of conditional likelihoods.

Name	Objective	Bias	Var.	Converge
$(x_i, y_i) \overset{iid}{\sim} p(x,y),\ (x'_k, y'_k) \overset{iid}{\sim} p(x)p(y),\ m_{\alpha,u}(x, y_{1:K}) \triangleq \alpha \frac{1}{K} \left\{ \sum_{k=1}^K \exp(g(x, y_k)) \right\} + (1-\alpha) \exp(u(x))$				
`DV` (Donsker and Varadhan, 1983)	$g(x_i, y_i) - \log(\sum_{k=1}^K \exp(g(x'_k, y'_k))/K)$	high	high	no
`MINE` (Belghazi et al., 2018)	$g(x_i, y_i) - \log(\mathcal{E}[\exp(g(x_i, y_j)), \eta])$	low	high	no
`NWJ` (Nguyen et al., 2010)	$g(x_i, y_i) - \exp(g(x_i, y_j) - 1)$	low	high	no
`JSD` (Hjelm et al., 2019)	$g^*(x_i, y_i) - \exp(g^*(x_i, y_j) - 1)$	low	high	no
	$g^* \xleftarrow{\arg\max} \{\log \sigma(g(x_i, y_i)) + \log \sigma(-g(x_i, y_j))\}$			
`TUBA` (Poole et al., 2019)	$g(x_i, y_i) + u(x_i) + 1 - \exp(g(x_i, y_j) - u(x_i))$	low	high	no
`InfoNCE` (Oord et al., 2018)	$g(x_i, y_i) - \log(\sum_j \exp(g(x_i, y_j))/K)$	high	low	no
α-`InfoNCE` (Poole et al., 2019)	$g(x_i, y_i) - g(x_i, y_j) - \log(m_{\alpha,u}(x, y_{1:K})) + \log(m_{\alpha,u}(x'_k, y'_k))$			no
α-`InfoNCE` interpolates between low-bias high-var ($\alpha \to 1$, `NWJ`) to high-bias low-var ($\alpha \to 0$, `InfoNCE`)				
`FLO` (ours)	$-u(x_i, y_i) - \exp(-u(x_i, y_i) + g(x_i, y_j) - g(x_i, y_i))$	**low**	**moderate**	**yes**

2.2.3 Connections to the existing MI bounds

Due to space limitations, we elaborate the connections to the existing MI bounds here and have relegated an extended related work discussion in a broader context to the Appendix.

From log-partition approximation to MI bounds. To embrace a more holistic understanding, we list popular variational MI bounds together with our `FLO` in Table 2.1, and visualize their connections in Figure 2.1. With the exception of JSD, these bounds can be viewed from the perspective of unnormalized statistical modeling, as they differ in how the log partition function $\log Z(x)$ is estimated. We broadly categorize these estimators into two families: the log-family (`DV`, `MINE`, `InfoNCE`) and the exponential-family (`NWJ`, `TUBA`, `FLO`). In the log-family, `DV` and `InfoNCE` are multi-sample estimators that leverage direct Monte-Carlo estimates $\hat{Z}$ for $\log Z(x)$, and these two differ in whether to include the positive sample in the denominator or not. To avoid the excessive in-batch computation of the normalizer and the associated memory drain, `MINE` further employed an *exponential moving average* (EMA) to

aggregate the normalizer across batches. Note for the log-family estimators, their variational gaps are partly caused by the log-transformation on finite-sample average due to Jensen's inequality (*i.e.*, $\log Z = \log \mathbb{E}[\hat{Z}] \geq \mathbb{E}[\log \hat{Z}]$). In contrast, the objective of exponential-family estimators do not involve such log-transformation, since they can all be derived from the Fenchel-Legendre inequality: `NWJ` directly applies the Fenchel dual of f-divergence for MI (Nowozin et al., 2016), while `TUBA` exploits this inequality to compute the log partition $\log Z(x) = \log \mathbb{E}_{p(y')}[\exp(g(x, y'))]$. Motivated from a contrastive view, our `FLO` applies the Fenchel-Legendre inequality to the log-partition of contrast scores.

A contrastive view for MI estimation. The MI estimators can also be categorized based on how they contrast the samples. For instance, `NWJ` and `TUBA` are generally considered to be non-contrastive estimators, as their objectives do not compare positive samples against negative samples on the same scale (*i.e.*, log versus exp), and this might explain their lack of effectiveness in representation learning applications. For JSD, it depends on a two-stage estimation procedure similar to that in adversarial training to assess the MI, by explicitly contrasting positive and negative samples to estimate the likelihood ratio. This strategy has been reported to be unstable in many empirical settings. The log-family estimators can be considered as a multi-sample, single-stage generalization of JSD. However, the `DV` objective can go unbounded thus resulting in a large variance, and the contrastive signal is decoupled by the EMA operation in `MINE`. Designed from contrastive perspectives, `InfoNCE` trades bound tightness for a lower estimation variance, which is found to be crucial in representation learning applications. Our `FLO` formalizes the contrastive view for exponential-family MI estimation, and bridges existing bounds: the PMI normalizer $\exp(-u(x, y))$ is a more principled treatment than the EMA in `MINE`, and compared to `DV` the positive and negative samples are explicitly contrasted and adaptively normalized.

Important FLO variants. We now demonstrate that `FLO` is a flexible framework that not only recovers existing bounds but also derives novel bounds such as

$$I_{\texttt{FDV}} \triangleq \texttt{StopGrad}[I_{\texttt{DV}}(\{(x_i, y_i)\})] + \frac{\sum_j \exp(c_\theta(x_i, y_i, y_j))}{\texttt{StopGrad}[\sum_j \exp(c_\theta(x_i, y_i, y_j))]} - 1. \tag{2.11}$$

Recall the optimal $g^*(x,y) = \log p(x|y) + c(x)$ and $u^*(x,y) = -\log \frac{p(x,y)}{p(x)p(y)}$, which motivates us to parameterize $u(x,y)$ in the form of $-g_\theta(x,y) + s_\psi(x)$, where $s_\psi(x)$ models the arbitrary drift $c(x)$, and this recovers the `TUBA` bound. Additionally, we note that (i) fixing either of u and g, and optimizing the other also gives a valid lower bound to MI; and (ii) a carefully chosen multi-input $u(\{(x_i,y_i)\})$ can be computationally appealing. As a concrete example, if we set u_ϕ to $\mathfrak{u}_\theta(\{(x_i,y_i)\}) \leftarrow \log\left(\frac{1}{K}\sum_j e^{c(x_i,y_i,y_j;g_\theta)}\right)$ and update $u_\theta(x,y)$ while artificially keeping the critic $g_\theta(x,y)$ fixed [6], then `FLO` falls back to `DV`. Alternatively, we can consider the Fenchel dual version of it: using the same multi-input $\mathfrak{u}_\theta(\{(x_i,y_i)\})$ above, treat u_ϕ as fixed and only update g_θ, and this gives us the novel MI objective in (2.11), we call it *Fenchel-Donsker-Varadhan* (FDV) estimator.

2.2.4 Gradient and convergence analysis of FLO

In this section, we will establish that FLO better optimizes the MI because its gradient is more accurate than competing variational bounds such as `NWJ` and `TUBA`; also, we provide the first convergence analysis for variational MI estimation by showing `FLO` converges under SGD.

First, recall most tractable variational MI bounds are derived from and upper bounded by the intractable `UBA` bound (Poole et al., 2019). For instance, with the same critic g_θ we have $I_{\texttt{NWJ}} \le I_{\texttt{TUBA}} \le I_{\texttt{UBA}}$. So if we can show $\nabla_\theta I_{\texttt{FLO}} \approx \nabla_\theta I_{\texttt{UBA}}$ then `FLO` is better optimized. To simplify notations, we denote $c_\theta(x,y,y') \triangleq g_\theta(x,y') - g_\theta(x,y)$ and $\mathcal{E}_\theta(x,y) \triangleq 1/\mathbb{E}_{p(y')}[e^{c_\theta(x,y,y')}]$, and we can easily verify

$$\mathbb{E}_{p(y')}\left[\nabla_\theta\left\{e^{c_\theta(x,y,y')}\right\}\right] = \nabla_\theta\left\{\frac{1}{\mathcal{E}_\theta(x,y)}\right\} = -\frac{\nabla\mathcal{E}_\theta(x,y)}{(\mathcal{E}_\theta(x,y))^2} = -\frac{\nabla_\theta \log \mathcal{E}_\theta(x,y)}{\mathcal{E}_\theta(x,y)}. \tag{2.12}$$

Since for fixed $g_\theta(x,y)$ the corresponding optimal $u^*_\theta(x,y)$ maximizing $I_{\texttt{FLO}}(u_\phi, g_\theta) \triangleq 1 - \left\{u_\phi(x,y) + \mathbb{E}_{p(y')}[e^{-u_\phi(x,y)+c(x,y,y';g_\theta)}]\right\}$ is given by $u^*_\theta(x,y) = \log \mathbb{E}_{p(y')}[e^{c_\theta(x,y,y')}] = -\log \mathcal{E}_\theta(x,y)$ (using (2.4)), we see that the term $e^{-u_\phi(x,y)}$ is essentially optimized to approximate $\mathcal{E}_\theta(x,y)$. To emphasize this point, we now write $\hat{\mathcal{E}}_\theta(x,y) \triangleq e^{-u_\phi(x,y)}$. When this approximation is

[6] That is to say g_θ in u_ϕ is an independent copy of g_θ.

sufficiently accurate (*i.e.*, $\mathcal{E}_\theta \approx \hat{\mathcal{E}}_\theta$), we can see that $\nabla I_{\texttt{FLO}}$ approximates $\nabla I_{\texttt{UBA}}$ as follows

$$\begin{aligned}\nabla_\theta\{I_{\texttt{FLO}}(u_\phi, g_\theta)\} &= -\mathbb{E}_{xy}\left[e^{-u_\phi(x,y)}\mathbb{E}_{y'}[\nabla_\theta e^{c_\theta(x,y,y')}]\right] = \mathbb{E}_{xy}\left[\tfrac{\hat{\mathcal{E}}_\theta(x,y)}{\mathcal{E}_\theta(x,y)}\nabla_\theta \log \mathcal{E}_\theta(x,y)\right] \\ &\approx \mathbb{E}_{xy}\left[\nabla_\theta \log \mathcal{E}_\theta(x,y)\right] = \nabla_\theta\left\{\mathbb{E}_{p(x,y)}[\log \mathcal{E}_\theta(x,y)]\right\} = \nabla_\theta\{I_{\texttt{UBA}}(g_\theta)\}.\end{aligned} \tag{2.13}$$

We can prove FLO will converge under much weaker conditions, even when this approximation $\hat{u}(x,y)$ is rough. The intuition is as follows: in (2.13), the term $\frac{\hat{\mathcal{E}}_{\theta_t}}{\mathcal{E}_{\theta_t}}$ only rescales the gradient, so the optimizer is still proceeding in the same direction as UBA in SGD. The informal version of our result is summarized in the Proposition below (see the Appendix for the formal version and proof).

Proposition 2.2.5 (Convergence of FLO, informal version)**.** Let $\{\eta_t\}_{t=1}^\infty$ be the stochastic *Robbins-Monro* sequence of learning rates: $\sum_t \mathbb{E}[\tilde{\eta}_t] = \infty$ and $\sum_t \mathbb{E}[\tilde{\eta}_t^2] < \infty$. If $\frac{\hat{\mathcal{E}}_{\theta_t}}{\mathcal{E}_{\theta_t}}$ is bounded between $[a,b]$ $(0 < a < b < \infty)$, then under the stochastic gradient descent scheme described in Algorithm 1, θ_t converges to a stationary point of $I_{\texttt{UBA}}(g_\theta)$ with probability 1, *i.e.*, $\lim_{t\to\infty}\|\nabla I_{\texttt{UBA}}(g_{\theta_t})\| = 0$. Additionally assume $I_{\texttt{UBA}}$ is convex with respect to θ, then FLO converges with probability 1 to the global optimum θ^* of $I_{\texttt{UBA}}$ from any initial point θ_0.

Algorithm 1 FLO

Empirical data $\hat{p}_d = \{(x_i, y_i)\}_{i=1}^n$

Model parameters $\Psi = (\theta, \phi)$

for $t = 1, 2, \cdots$ **do**

  Sample $i, j \overset{iid}{\sim} [n]$

  $u_{ii} = u_\phi(x_i, y_i), g_{ii} = g_\theta(x_i, y_i),$

  $g_{ij} = g_\theta(x_i, y_j)$

  $\mathcal{F} = u_{ii} + \exp(-u_{ii} + g_{ij} - g_{ii})$

$\Psi_t = \Psi_t - \eta_t \nabla_\Psi \mathcal{F}$

end for

Importantly, this marks the first convergence result for variational MI estimators. The convergence analyses for MI estimation are non-trivial and scarce even for those standard statistical estimators (Paninski, 2003; Gao et al., 2015; Rainforth et al., 2018). The lack of convergence guarantees has led to a proliferation of unstable MI-estimators used in practice (in particular, DV, JSD, and MINE) that critically rely on various empirical hacks to work well (see discussions in (Song and Ermon, 2020)). Our work establishes a family of variational MI estimators that provably converges, a contribution we consider significant as it fills an important gap in current literature on both theoretical and practical notes.

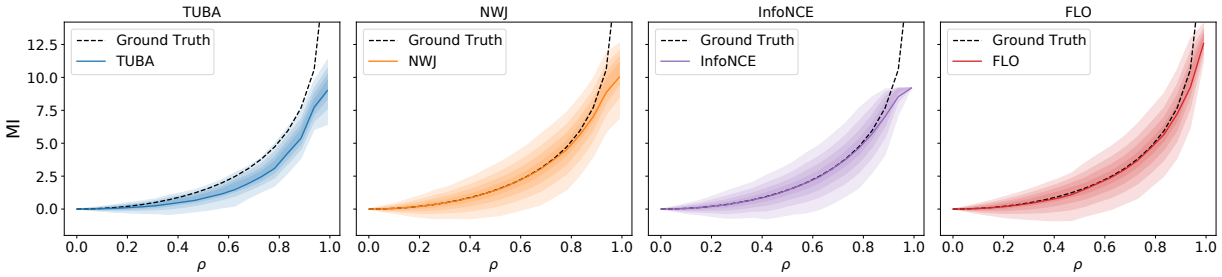

Figure 2.3: Bias-variance plot for popular variational MI bounds with the 10-D Gaussians. Estimators that are more concentrated around the dashed line is considered better (low-bias, low-variance). In the more challenging high-MI regime, `FLO` shows a clear advantage over competing alternatives, where FLO pays less price in variance to achieve even better accuracy when tight estimation is impossible.

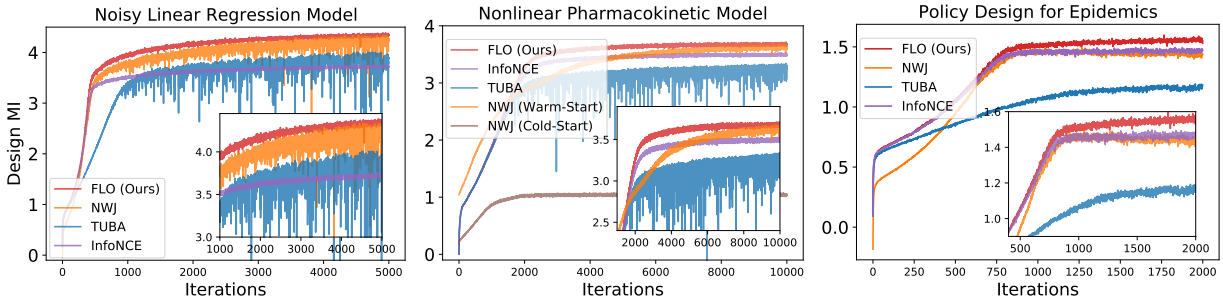

Figure 2.4: Bayesian Optimal Experiment Design results. `FLO` consistently performs best, demonstrating superior strength in learning efficiency and robustness. `NWJ` takes the runner-up, but it has a larger variance and is sensitive to network initializations. `InfoNCE` is less competitive due to low sample inefficiency, but its smaller variance helps in the more challenging dynamic case.

2.3 Experiments

We consider an extensive range of tasks to validate `FLO` and benchmark it against state-of-the-art solutions. To underscore the practical significance of MI in machine learning, we demonstrate example applications from data collection (in statistical parlance, experimental design), self-supervised pre-training, to meta/transfer learning. Limited by space, we present only the key results in the main text and defer ablation studies and details of our experimental setups to the Appendix. Our code is available from `https://github.com/qingguo666/FLO`. All experiments are implemented with `PyTorch`.

Comparison to baseline MI bounds. We start by comparing `FLO` to the following popular competing variational estimators: `NWJ`, `TUBA`, and `InfoNCE`. We use the bilinear critic

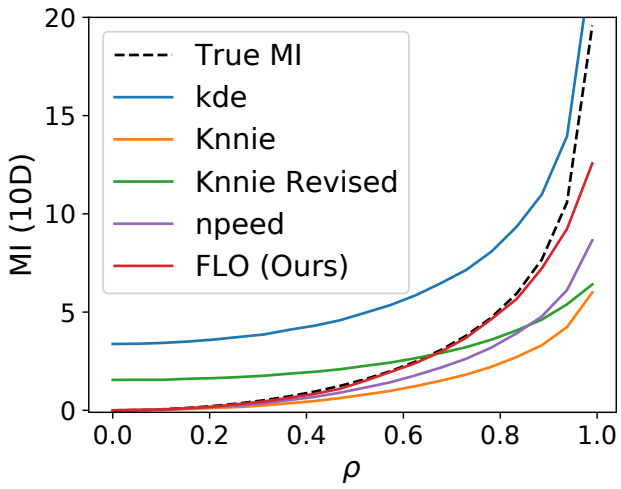

Figure 2.5: `FLO` compares favorably to classical MI estimators.

implementation for all models which maximally encourages both sample efficiency and code simplicity, and this strategy does perform best based on our observations. We consider the synthetic benchmark from (Poole et al., 2019), where $(X \in \mathbb{R}^d, Y \in \mathbb{R}^d)$ is jointly standard Gaussian with diagonal cross-correlation parameterized by $\rho \in [0, 1)$. We report $d = 10$ and $\rho \in [0, 0.99]$ here (other studies only report ρ up to 0.9, which is less challenging.), providing a reasonable coverage of the range of MI one may encounter in empirical settings.

To focus on the bias-variance trade-off, we plot the decimal quantiles in addition to the estimated MI in Figure 2.3, where `FLO` significantly outperformed its variational counterparts in the more challenging high-MI regime. In Figure 2.5, we show `FLO` also beats classical MI estimators (Kraskov et al., 2004; Ver Steeg and Galstyan, 2013; Gao et al., 2018). In the Appendix 2.5.8, we further discuss recent works on parametric estimators (Cheng et al., 2020; Brekelmans et al., 2021) and alternative information metrics (Xu et al., 2020).

Bayesian optimal experiment design (BOED). We next direct our attention to BOED, a topic of significant interest shared by the statistical and machine learning communities (Chaloner and Verdinelli, 1995; Wu and Hamada, 2011; Hernández-Lobato et al., 2014; Foster et al., 2020). The performance of machine learning models crucially relies on the quality of data supplied for training, and BOED is a principled framework that optimizes the data collection procedure (in statistical parlance, conducting *experiments*) (Foster

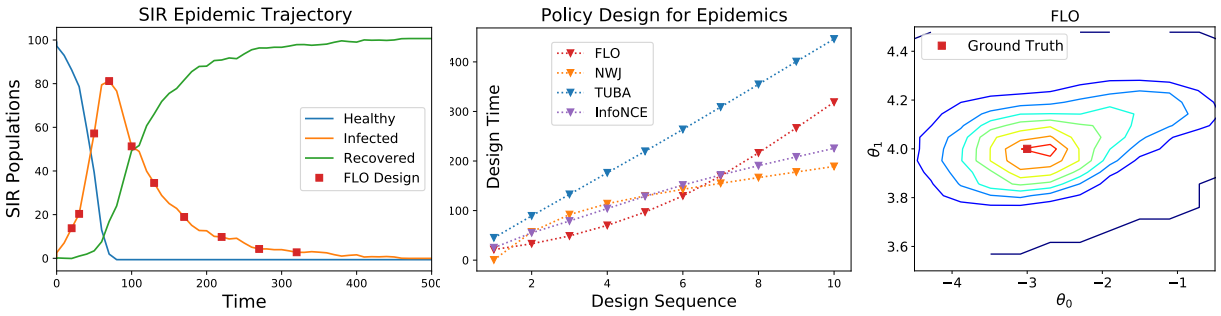

Figure 2.6: Diagnosis of learned sequential designs. The disease surveillance windows designed by `FLO` makes more sense: measures more frequently as infection spikes, and more sparsely when the pandemic slowly fades. The estimated parameter posterior (right) is consistent with the ground truth.

et al., 2019). Mathematically, let x be the data to be collected, θ be the parameters to be inferred, and d be the experiment parameters the investigator can manipulate (*a.k.a*, the *design parameters*), BOED tries to find the optimal data collection procedure that is expected to generate data that is most informative about the underlying model parameters, *i.e.*, solves for $\arg\max_d I(x;\theta;d)$. In this study, we focus on the more generic scenario where explicit likelihoods are not available, but we can still sample from the data generating procedure (Kleinegesse and Gutmann, 2020, 2021).

We consider three carefully selected models from recent literature for their progressive practical significance and the challenges involved (Foster et al., 2021; Ivanova et al., 2021; Kleinegesse et al., 2021): static designs of (*i*) a simple linear regression model and (*ii*) a complex nonlinear pharmacokinetic model for drug development; and the dynamic policy design for (*iii*) epidemic disease surveillance and intervention (*e.g.*, for Covid-19 modeling). Designs with higher MI are more favorable because it implies the data carries more information. In Figure 2.4 we compare design optimization curves using different MI optimization strategies, where `FLO` consistently leads. Popular `NWJ` and `InfoNCE` report different tradeoffs that are less susceptible to `FLO`. We also examine the `FLO` predicted posteriors and confirm they are consistent with the ground-truth parameters (Figure 2.6 right). For the dynamic policy optimization, we also manually inspect the design strategies reported by different models (Figure 2.6 left, middle). Consistent with human judgment, `FLO` policy better assigns budgeted surveillance resources at different stages of pandemic progression.

Table 2.2: Multi-view representation learning on `Cifar`

Model	`InfoNCE`	`SpecNCE` (HaoChen et al., 2021)[7]	`FLO`	`FDV`
MI	$5.73 \pm .07$	$4.76 \pm .08$	$5.83 \pm .08$	$\mathbf{5.93 \pm .08}$

A novel meta-learning framework. A second application of our work is to meta-learning, an area attracting substantial recent interest. In meta-learning, we are concerned with scenarios that at training time, there are abundant different labelled tasks, while upon deployment, only a handful of labeled instances are available to adapt the learner to a new task. Briefly, for an arbitrary loss $\ell_t(\hat{y}, y)$, where t is the task identifier and $\hat{y} = f(x)$ is the prediction made by the model, we denote the risk by $R_t(f) = \mathbb{E}_{p_t(x,y)}[\ell_t(f(x), y)]$. Denote $R(f) \triangleq \mathbb{E}_{t\sim p(t)}[R_t(f)]$ as the expected risk for all tasks and $\hat{R}(f)$ for the mean of empirical risks computed from all training tasks. Inspired by recent information-theoretic generalization theories (Xu and Raginsky, 2017), we derived a novel, principled objective

$$\mathcal{L}_{\texttt{Meta-FLO}}(f) = \hat{R}(f) + \lambda\sqrt{I_{\texttt{FLO}}(\hat{\mathcal{D}}_t; \hat{E}_t)}, \tag{2.14}$$

where λ is known given the data size and loss function, $(\hat{\mathcal{D}}_t, \hat{E}_t)$ are respectively data and task embeddings for training data, which for the first time lifts contrastive learning to the task and data distribution level. Our reasoning is that $\mathcal{L}_{\texttt{Meta-FLO}}(f)$ theoretically bounds $R(f)$ from above, and it is relatively sharp for being data-dependent. We give more information on this in the Appendix and defer a full exposition to a dedicated paper due to independent interest and space limits here. Note other MI bounds are not suitable for this task due to resource and variance concerns. In Figure 2.7 we show `Meta-FLO` wins big over the state-of-the-art *model agnostic meta-learning* (MAML) model on the regression benchmark from (Finn et al., 2017).

2.4 Conclusion

We have described a new framework for the contrastive estimation of mutual information from energy modeling perspectives. Our work not only encapsulates popular variational MI

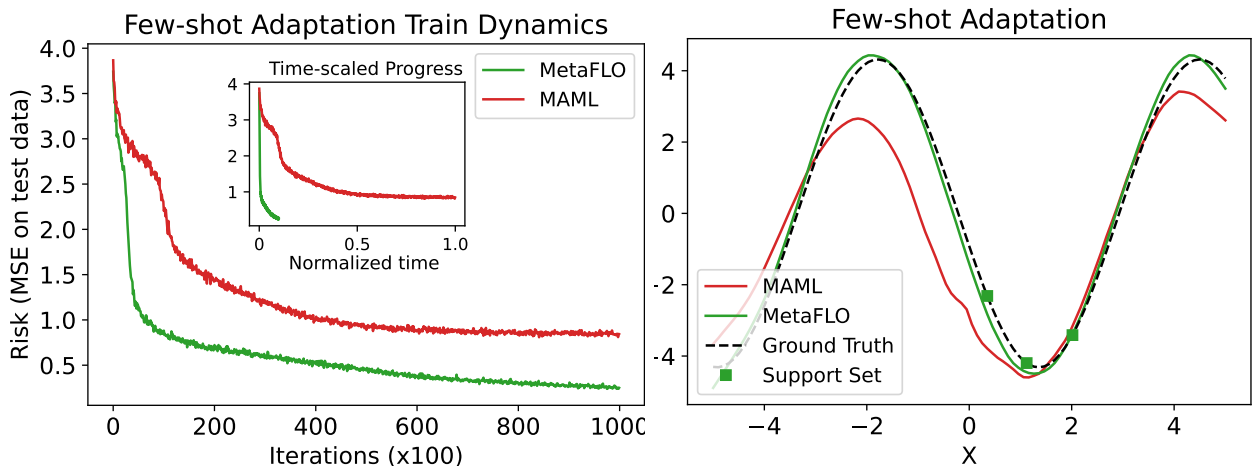

Figure 2.7: Few-shot adaptation with `Meta-FLO`.

bounds but also inspires novel objectives such as `FLO` and `FDV`, which comes with strong theoretical guarantees. In future work, we will leverage our theoretical insights to improve practical applications involving MI estimation, such as representation learning, fairness, and in particular, data efficient learning.

2.5 Appendix

2.5.1 Proof of Proposition 2.2.1 (InfoNCE Properties and derivation for some popular variational MI bounds)

Proof. Now let us prove `InfoNCE` is a lower bound to MI and under proper conditions this estimate is tight. Our proof is based on establishing that `InfoNCE` is a multi-sample extension of the `NWJ` bound. For completeness, we first repeat the proof for `BA` and `UBA` below, and then show `UBA` leads to `NWJ` and its multi-sample variant `InfoNCE`.

We can bound MI from below using an variational distribution $q(y|x)$ as follows:

$$
\begin{aligned}
I(X,Y) &= \mathbb{E}_{p(x,y)}\left[\log \frac{p(x,y)}{p(x)p(y)}\right] && (2.15)\\
&= \mathbb{E}_{p(x,y)}\left[\log \frac{p(y|x)p(x)q(y|x)}{p(x)p(y)q(y|x)}\right] \quad \#\ q(y—x)\ \textit{is the variational distributio} && (2.16)\\
&= \mathbb{E}_{p(x,y)}\left[\log \frac{q(y|x)}{p(y)}\right] + \mathbb{E}_{p(x)}[\mathrm{KL}(p(y|x)||q(y|x))] && (2.17)\\
&\geq \mathbb{E}_{p(x,y)}\left[\log q(y|x) - \log p(y)\right] \triangleq I_{\texttt{BA}}(X,Y;q) && (2.18)
\end{aligned}
$$

In sample-based estimation of MI, we do not know the ground-truth marginal density $p(y)$, which makes the above BA bound impractical. However, we can carefully choose an energy-based variational density that "cancels out" $p(y)$:

$$q_f(y|x) = \frac{p(y)}{Z_f(x)} e^{f(x,y)}, \quad Z_f(x) \triangleq \mathbb{E}_{p(y)}[e^{f(x,y)}]. \tag{2.19}$$

This auxiliary function $f(x, y)$ is known as the tilting function in importance weighting literature. Hereafter, we will refer to it the *critic function* in accordance with the nomenclature used in contrastive learning literature. The partition function $Z_f(x)$ normalizes this $q(y|x)$. Plugging this $q_f(y|x)$ into $I_{\texttt{BA}}$ yields:

$$\begin{aligned} I_{\texttt{BA}}(X, Y; q_f) &= \mathbb{E}_{p(x,y)}[f(x, y) + \log(p(y)) - \log(Z(x)) - \log p(y)] && (2.20) \\ &= \mathbb{E}_{p(x,y)}[f(x, y)] - \mathbb{E}_{p(x)}[\log(Z_f(x))] \triangleq I_{\texttt{UBA}}(X, Y; f) && (2.21) \end{aligned}$$

For $x, a > 0$, we have inequality $\log(x) \leq \frac{x}{a} + \log(a) - 1$. By setting $x \leftarrow Z(y)$ and $a \leftarrow e$, we have

$$\log(Z(y)) \leq e^{-1} Z(y). \tag{2.22}$$

Plugging this result into (2.21) we recover the celebrated NWJ bound, which lower bounds $I_{\texttt{UBA}}$:

$$I_{\texttt{UBA}}(X, Y) \geq \mathbb{E}_{p(x,y)}[f(x, y)] - e^{-1}\mathbb{E}_{p(x)}[Z_f(x)] \triangleq I_{\texttt{NWJ}}(X, Y; f). \tag{2.23}$$

When $f(x, y)$ takes the value of

$$f^*(x, y) = 1 + \log \frac{p(x|y)}{p(x)}, \tag{2.24}$$

this bound is sharp.

We next extend these bounds to the multi-sample setting. In this setup, we are given one paired sample (x_1, y_1) from $p(x, y)$ (*i.e.*, the positive sample) and $K-1$ samples independently drawn from $p(y)$ (*i.e.*, the negative samples). Note that when we average over x wrt $p(x)$ to

compute the MI, this equivalent to comparing positive pairs from $p(x, y)$ and negative pairs artificially constructed by $p(x)p(y)$. By the independence between X_1 and $Y_{k>1}$, we have

$$I(X; Y_{1:K}) = \mathbb{E}_{p(x_1,y_1)\prod_{k>1} p(y_k)}\left[\frac{p(x_1, y_1)\prod_{k>1} p(y_k)}{p(x_1)\prod_k p(y_k)}\right] = \mathbb{E}_{p(x_1,y_1)}\left[\frac{p(x_1, y_1)}{p(x_1)p(y_1)}\right] = I(X; Y) \tag{2.25}$$

So for arbitrary multi-sample critic $f(x; y_{1:K})$, we know

$$I(X; Y) = I(X_1; Y_{1:K}) \geq I_{\texttt{NWJ}}(X_1, Y_{1:K}; f) = \mathbb{E}_{p(x_1,y_1)\prod_{k>1} p(y_k)}[f(x_1, y_{1:K})] - e^{-1}\mathbb{E}_{p(x)}[Z_f(x)] \tag{2.26}$$

Now let us set

$$\tilde{f}(x_1; y_{1:K}) = 1 + \log\frac{e^{g(x_1,y_1)}}{m(x_1; y_{1:K})}, \quad m(x_1; y_{1:K}) = \frac{1}{K}\sum_k e^{g(x_1,y_k)}. \tag{2.27}$$

$$\begin{aligned} I_{\texttt{NWJ}}(X_1, Y_{1:K}; \tilde{f}) =& \mathbb{E}_{p(x_1,y_1)p^{K-1}(y_k)}\left[1 + \log\frac{e^{g(x_1,y_1)}}{m(x_1; y_{1:K})}\right] - \mathbb{E}_{p(x')p^K(y')}\left[e^{-1+1+\log\frac{e^{g(x'_1,y'_1)}}{m(x'_1;y'_{1:K})}}\right] \\ =& \mathbb{E}_{p(x_1,y_1)p^{K-1}(y_k)}\left[1 + \log\frac{e^{g(x_1,y_1)}}{m(x_1; y_{1:K})}\right] - \mathbb{E}_{p(x')p^K(y')}\left[\frac{e^{g(x'_1,y'_1)}}{m(x'_1; y'_{1:K})}\right] \end{aligned}$$

Due to the symmetry of $\{y_k\}_{k=1}^K$, we have

$$\mathbb{E}_{p(x')p^K(y')}\left[\frac{e^{g(x'_1,y'_1)}}{m(x'_1; y'_{1:K})}\right] = \mathbb{E}_{p(x')p^K(y')}\left[\frac{e^{g(x'_1,y'_k)}}{m(x'_1; y'_{1:K})}\right]. \tag{2.28}$$

So this gives

$$\mathbb{E}_{p(x')p^K(y')}\left[\frac{e^{g(x'_1,y'_1)}}{m(x'_1; y'_{1:K})}\right] = \mathbb{E}_{p(x')p^K(y')}\left[\frac{\frac{1}{K}e^{g(x'_1,y'_k)}}{m(x'_1; y'_{1:K})}\right] = 1, \tag{2.29}$$

and one can easily see this recovers the K-sample `InfoNCE` defined in (3)

$$I_{\texttt{NWJ}}(X_1, Y_{1:K}; \tilde{f}) = \mathbb{E}_{p(x_1,y_1)p^{K-1}(y_k)}\left[\log\frac{e^{g(x_1,y_1)}}{m(x_1; y_{1:K})}\right] = I^K_{\texttt{InfoNCE}}(X; Y|g) \tag{2.30}$$

Now we need to show this bound is sharp when $K \to \infty$. We only need to show that for some choice of $g(x, y)$, the inequality holds asymptotically. Recall the NWJ's optimal critic takes value of $f^*(x, y) = 1 + \frac{p(x|y)}{p(x)}$, so with reference to (2.27) let us plug in $g^*(x, y) = \frac{p(y|x)}{p(y)}$ into InfoNCE

$$\begin{aligned}
\mathcal{L}_K^* &= \mathbb{E}_{p^K}\left[\log\left(\frac{f^*(x_k, y_k)}{f^*(x_k, y_k) + \sum_{k' \neq k} f^*(x_k, y_{k'})}\right)\right] + \log K && (2.31)\\
&= -\mathbb{E}\left[\log\left(1 + \frac{p(y)}{p(y|x)}\sum_{k'}\frac{p(y_{k'}|x_k)}{p(y_{k'})}\right)\right] + \log K && (2.32)\\
&\approx -\mathbb{E}\left[\log\left(1 + \frac{p(y)}{p(y|x)}(K-1)\mathbb{E}_{y_{k'}}\frac{p(y_{k'}|x_k)}{p(y_{k'})}\right)\right] + \log K && (2.33)\\
&= -\mathbb{E}\left[\log\left(1 + \frac{p(y_k)}{p(y_k|x_k)}(K-1)\right)\right] + \log K && (2.34)\\
&\approx -\mathbb{E}\left[\log\frac{p(y)}{p(y|x)}\right] - \log(K-1) + \log K && (2.35)\\
(K \to \infty) &\to I(X;Y) && (2.36)
\end{aligned}$$

This concludes our proof. □

2.5.2 Proof of Proposition 2.2.2 (FLO lower bounds MI)

Proof. The proof is given in line in the main text. Basically we have applied the Fenchel duality trick to the log term in the UBA bound. Note that unlike UBA, our FLO bound can be unbiased estimated with finite samples (as UBA requires an infinite sum inside its log term, which makes finite-sample empirical estimate biased per Jensen's inequality). □

2.5.3 Gradient Analysis of FLO (More Detailed)

To further understand the workings of FLO, let us inspect the gradient of model parameters. Recall the intractable UBA MI estimator can be re-expressed in the following form:

$$I_{\text{UBA}'}(g_\theta) = \mathbb{E}_{p(x,y)}[-\log \mathbb{E}_{p(y')}[\exp(g_\theta(x, y') - g_\theta(x, y))]] \quad (2.37)$$

In this part, we want to establish the intuition that $\nabla_\theta\{I_{\texttt{FLO}}(u_\phi, g_\theta)\} \approx \nabla_\theta\{I_{\texttt{UBA}'}(g_\theta)\}$, where

$$I_{\texttt{FLO}}(u_\phi, g_\theta) \triangleq -\left\{u_\phi(x,y) + \mathbb{E}_{p(y')}[\exp(-u_\phi(x,y) + g_\theta(x,y') - g_\theta(x,y))]\right\} \tag{2.38}$$

is our FLO estimator.

By defining

$$\mathcal{E}_\theta(x,y) \triangleq \frac{1}{\mathbb{E}_{p(y')}[\exp(g_\theta(x,y') - g_\theta(x,y))]}, \tag{2.39}$$

we have

$$\nabla_\theta\left\{\frac{1}{\mathcal{E}_\theta(x,y)}\right\} = -\frac{\nabla\mathcal{E}_\theta(x,y)}{(\mathcal{E}_\theta(x,y))^2} = -\frac{\nabla_\theta \log \mathcal{E}_\theta(x,y)}{\mathcal{E}_\theta(x,y)}, \tag{2.40}$$

and

$$\begin{aligned}
\nabla_\theta\left\{\frac{1}{\mathcal{E}_\theta(x,y)}\right\} &= \nabla_\theta\mathbb{E}_{p(y')}[\{\exp(g_\theta(x,y') - g_\theta(x,y))\}] &(2.41)\\
&= \mathbb{E}_{p(y')}[\nabla_\theta\{\exp(g_\theta(x,y') - g_\theta(x,y))\}]. &(2.42)
\end{aligned}$$

We know fixing $g_\theta(x,y)$, the corresponding optimal $u^*_\theta(x,y)$ maximizing FLO is given by

$$u^*_\theta(x,y) = \log \mathbb{E}_{p(y')}[\exp(g_\theta(x,y') - g_\theta(x,y))] = -\log \mathcal{E}_\theta(x,y). \tag{2.43}$$

This relation implies the view that $\exp^{-u_\phi(x,y)}$ is optimized to approximate $\mathcal{E}_\theta(x,y)$. And to emphasize this point, we now write $\hat{\mathcal{E}}_\theta(x,y) \triangleq e^{-u_\phi(x,y)}$. Assuming this approximation is sufficiently accurate (*i.e.*, $\mathcal{E}_\theta \approx \hat{\mathcal{E}}_\theta$), we have

$$\begin{aligned}
\nabla_\theta\{I_{\texttt{FLO}}(u_\phi, g_\theta)\} &= -\mathbb{E}_{p(x,y)}\left[e^{-u_\phi(x,y)}\mathbb{E}_{p(y')}[\nabla_\theta \exp(g_\theta(x,y') - g_\theta(x,y))]\right] &(2.44)\\
&= \mathbb{E}_{p(x,y)}\left[\frac{e^{-u_\phi(x,y)}}{\mathcal{E}_\theta(x,y)}\nabla_\theta \log \mathcal{E}_\theta(x,y)\right] &(2.45)\\
&= \mathbb{E}_{p(x,y)}\left[\frac{\hat{\mathcal{E}}_\theta(x,y)}{\mathcal{E}_\theta(x,y)}\nabla_\theta \log \mathcal{E}_\theta(x,y)\right] &(2.46)\\
&\approx \mathbb{E}_{p(x,y)}\left[\nabla_\theta \log \mathcal{E}_\theta(x,y)\right] &(2.47)\\
&= \nabla_\theta\left\{\mathbb{E}_{p(x,y)}[\log \mathcal{E}_\theta(x,y)]\right\} = \nabla_\theta\{I_{\texttt{UBA}'}(g_\theta)\}. &(2.48)
\end{aligned}$$

While the above relation shows we can use FLO to amortize the learning of UBA, one major caveat with the above formulation is that $\hat{u}(x, y)$ has to be very accurate for it to be valid. As such, one needs to solve a cumbersome nested optimization problem: update g_θ, then optimize u_ϕ until it converges before the next update of g_θ. Fortunately, we can show that is unnecessary: the convergence can be established under much weaker conditions, which justifies the use of simple simultaneous stochastic gradient descent for both (θ, ϕ) in the optimization of FLO.

2.5.4 Proof of Proposition 2.2.5 (FLO Convergence under SGD)

Our proof is based on the convergence analyses of generalized stochastic gradient descent from (Tao et al., 2019). We cite the main assumptions and results below for completeness.

Definition 2.5.1 (Generalized SGD, Problem 2.1 in (Tao et al., 2019))**.** Let $h(\theta; \omega), \omega \sim p(\omega)$ be an unbiased stochastic gradient estimator for objective $f(\theta)$, $\{\eta_t > 0\}$ is the fixed learning rate schedule, $\{\xi_t > 0\}$ is the random perturbations to the learning rate. We want to solve for $\nabla f(\theta) = 0$ with the iterative scheme $\theta_{t+1} = \theta_t + \tilde{\eta}_t\, h(\theta_t; \omega_t)$, where $\{\omega_t\}$ are iid draws and $\tilde{\eta}_t = \eta_t \xi_t$ is the randomized learning rate.

Assumption 2.5.2. (Standard regularity conditions for Robbins-Monro stochastic approximation, Assumption D.1 (Tao et al., 2019)).

*A*1. $h(\theta) \triangleq \mathbb{E}_\omega[h(\theta; \omega)]$ is Lipschitz continuous;

*A*2. The ODE $\dot{\theta} = h(\theta)$ has a unique equilibrium point θ^*, which is globally asymptotically stable;

*A*3. The sequence $\{\theta_t\}$ is bounded with probability 1;

*A*4. The noise sequence $\{\omega_t\}$ is a martingale difference sequence;

*A*5. For some finite constants A and B and some norm $\|\cdot\|$ on $\mathbb{R}^d$, $\mathbb{E}[\|\omega_t\|^2] \leq A + B\|\theta_t\|^2$ a.s. $\forall t \geq 1$.

Proposition 2.5.3 (Generalized stochastic approximation, Proposition 2.2 in (Tao et al., 2019))**.** Under the standard regularity conditions listed in Assumption 2.5.2, we further assume $\sum_t \mathbb{E}[\tilde{\eta}_t] = \infty$ and $\sum_t \mathbb{E}[\tilde{\eta}_t^2] < \infty$. Then $\theta_n \to \theta^*$ with probability 1 from any initial point θ_0.

Assumption 2.5.4. (Weaker regularity conditions for generalized Robbins-Monro stochastic approximation, Assumption G.1 in (Tao et al., 2019)).

*B*1. The objective function $f(\theta)$ is second-order differentiable.

*B*2. The objective function $f(\theta)$ has a Lipschitz-continuous gradient, i.e., there exists a constant L satisfying

$$-LI \preceq \nabla^2 f(\theta) \preceq LI,$$

*B*3. The noise has a bounded variance, i.e., there exists a constant $\sigma > 0$ satisfying $\mathbb{E}\left[\|h(\theta_t; \omega_t) - \nabla f(\theta_t)\|^2\right] \leq \sigma^2$.

Proposition 2.5.5 (Weaker convergence results, Proposition G.2 in (Tao et al., 2019))**.** Under the technical conditions listed in Assumption 2.5.4, the SGD solution $\{\theta_t\}_{t>0}$ updated with generalized Robbins-Monro sequence ($\tilde{\eta}_t$: $\sum_t \mathbb{E}[\tilde{\eta}_t] = \infty$ and $\sum_t \mathbb{E}[\tilde{\eta}_t^2] < \infty$) converges to a stationary point of $f(\theta)$ with probability 1 (equivalently, $\mathbb{E}\left[\|\nabla f(\theta_t)\|^2\right] \to 0$ as $t \to \infty$).

Proof. Since $\hat{\mathcal{E}}_{\theta_t}/\mathcal{E}_{\theta_t}$ is bounded between $[a, b]$ ($0 < a < b < \infty$), results follow by a direct application of Proposition 2.5.3 and Proposition 2.5.5. □

2.5.5 Gaussian Toy Model Experiments

First, we start validating the properties and utility of the proposed `FLO` estimator by comparing it to competing solutions with the Gaussian toy models. Specifically, for the $2d$-D Gaussian model with correlation ρ, we have $X \in \mathbb{R}^d$ and $Y \in \mathbb{R}^d$ with covariance structure

$$\text{cov}[[X]_i, [X]_j] = \delta_{ij}, \text{cov}[[Y]_i, [Y]_j] = \delta_{ij}, \text{cov}[[X]_i, [Y]_j] = \delta_{ij} \cdot \rho \tag{2.49}$$

This allows us to have the ground-truth MI $I(X;Y) = -\frac{d}{2}\log(1-\rho^2)$ for reference and easily tune the difficulty of the task via varying d and ρ.

2.5.5.1 Choice of baselines

We choose TUBA, NWJ, InfoNCE and α-InfoNCE as our baselines. Note α-InfoNCE results are not reported in the main paper because we do not see a clear advantage via tuning α NWJ and InfoNCE are the two most popular estimators in practice that are employed without additional hacks. TUBA is included for its close relevance to FLO (*i.e.*, optimizing $u(x)$ instead of $u(x, y)$, and being non-contrastive). We do not include DV here because we find DV needs excessively a large negative sample size K to work. Variants like MINE are excluded for involving additional tuning parameters or hacks which complicate our analyses. The proposed FDV estimator is also excluded from our analyses for bound comparison since it includes $\hat{I}_{\texttt{DV}}$ in the estimator. Note that although not suitable for MI estimation, we find FDV works quite well in representation learning settings where the optimization of MI is targeted. This is because in FDV, the primal term $\hat{I}_{\texttt{DV}}$ term does not participate gradient computation, so it does not yield degenerated performance as that of DV. In the results reported below, we fixed $\alpha = 0.8$ for better visualization.

2.5.5.2 Experimental setups

We use the following baseline setup for all models unless otherwise specified. For the critic functions $g(x, y)$, $u(x, y)$ and $u(x)$, we use multi-layer perceptron (MLP) network construction with hidden-layers 512×512 and ReLU activation. For optimizer, we use Adam and set the learning rate to 10^{-4} unless otherwise specified. A default batch-size of 128 is used for training. To report the estimated MI, we use $10k$ samples and take the average. To visualize variance, we plot the decimal quantiles at $\{10\%, 20\%, \cdots, 80\%, 90\%\}$ and color code with different shades. We sample fresh data points in each iteration to avoid overfitting the data. All models are trained for $\sim 5,000$ iterations (each epoch samples $10k$ new data points, that is 78 iterations per epoch for a total of 50 epochs).

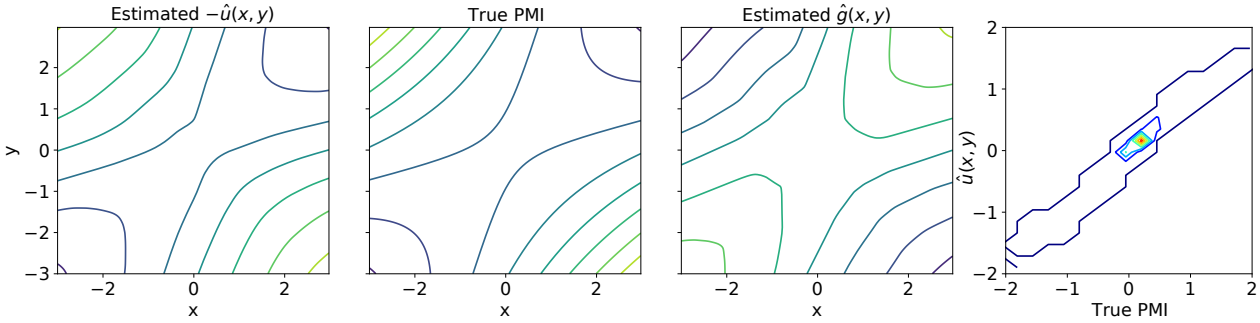

Figure 2.8: Comparison of estimated $u(x,y), g(x,y)$ and the ground-truth PMI $-\log \frac{p(x,y)}{p(x)p(y)}$ using the 2D Gaussian experiment. This confirms our analyses that the optimized $u(x,y)$ approximates the true PMI.

2.5.5.3 PMI approximation with $u(x,y)$

For Figure 2.8, we use the 2-D Gaussian with $\rho = 0.5$ to compare the estimated $u(x,y), g(x,y)$ with the ground-truth PMI, and the contour plot is obtained with a grid resolution of 2.5×10^{-2}. This confirms our analyses that the optimized $u(x,y)$ approximates the true PMI $-\log \frac{p(x,y)}{p(x)p(y)}$.

2.5.5.4 Ablation study: efficiency of parameter sharing for $g(x,y)$ and $u(x,y)$.

For the shared parameterization experiment for `FLO` (Figure 2.9), we used the more challenging 20-D Gaussian with $\rho = 0.5$, and trained the network with learning rate 10^{-3} and 10^{-4} respectively. We repeat the experiments for 10 times and plot the distribution of the MI estimation trajectories. Note that we intentionally used a setup such that the MLP network architecture we used is inadequate to get a sharp estimate (both for `FLO` and other MI estimators), which simulates the realistic scenario that the ground-truth MI is infeasible due to architecture constraints (refer to our ablation study on the influence network capacity in Sec 2.5.5.5). We observe the `FLO` estimator with a shared network learns faster than its separate network counterpart under both learning rates, validating the superior efficiency of parameter sharing.

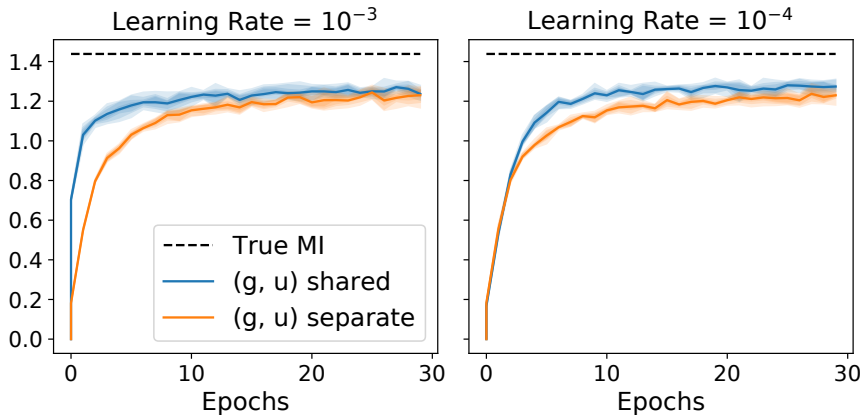

Figure 2.9: MI estimation with different critic parameter sharing strategies for FLO: shared network and separate networks under learning rates 10^{-3} and 10^{-4} for 2-D Gaussian. Note shared parameterization not only reduced half the network size, it also learns faster.

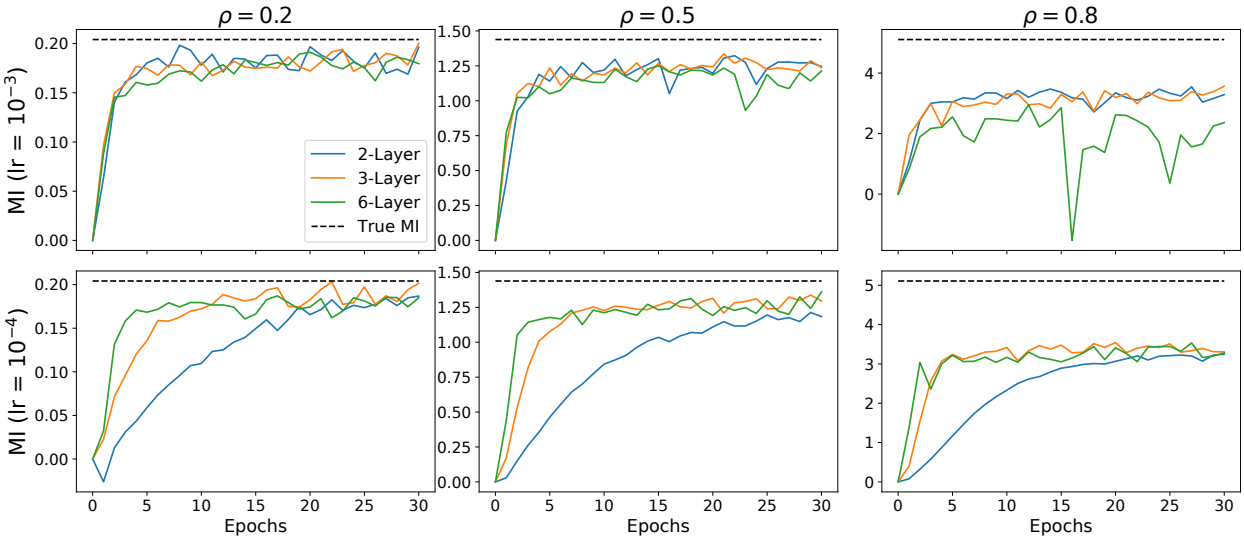

Figure 2.10: Abaltion study for network complexity with FLO. More complex networks lead to faster convergence and better MI estimates. However, the stability is more sensitive to the learning rate with a larger neural network.

2.5.5.5 Ablation study: network capacity and MI estimation accuracy

We further investigate how the neural network learning capacity affects MI estimation. In Figure 2.10 we compare the training dynamics of the FLO estimator with L-layer neural networks, where $L \in \{2, 3, 6\}$ and each hidden-layer has 512-units. A deeper network is generally considered to be more expressive. We see that using larger networks in general converge faster in terms of training iterations, and also obtain better MI estimates. However, more complex networks imply more computation per iteration, and it can be less stable when trained with

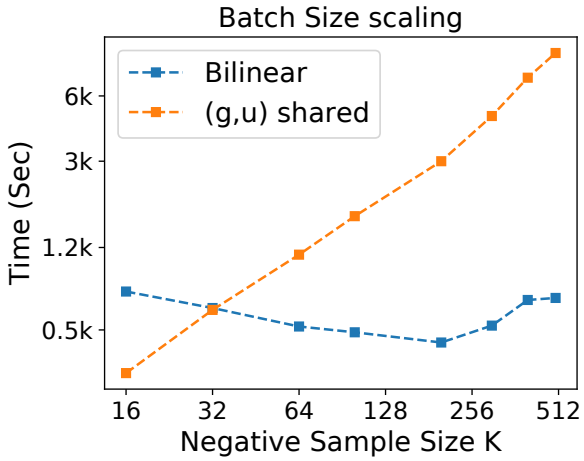

Figure 2.11: Comparison of computation time of the shared MLP critic and the bi-linear critic. Overall the bilinear implementation is more efficient than the shared MLP. `FLO`'s initial drop in computation time with a growing negative sample size is due to better exploitation of parallel computation.

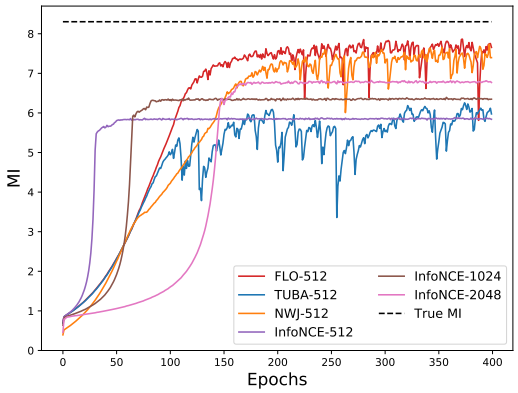

Figure 2.12: Comparison of learning dynamics with 20-D Gaussian at $\rho = 0.9$. We used bi-linear critics for all bounds. Note `InfoNCE` enjoys stable learning, and its convergence is fast in the small-sample regime but slow in the large-sample regime. In all cases, `InfoNCE` suffers from large biases. `NWJ` is more accurate but it learns slower. In contrast, our `FLO` learns fast and stably.

larger learning rates.

2.5.5.6 Ablation study: Bi-linear critics and scaling

We set up the *bi-linear* critic experiment as follows. For the naive baseline `FLO`, we use the shared-network architecture for $g(x, y)$ and $u(x, y)$, and use the in-batch shuffling to create the desired number of negative samples (`FLO-shuff`). For `FLO-BiL`, we adopt the following implementation: feature encoders $h(x), \tilde{h}(y)$ are respectively modeled with three layer MLP with 512-unit hidden layers and `ReLU` activations, and we set the output dimension to 512. Then we concatenate the feature representation to $z = [h(x), \tilde{h}(y)]$ and fed it to the $u(x, y)$ network, which is a two-layer 128-unit MLP. Note that is merely a convenient modeling choice and can be further optimized for efficiency. Each epoch contains $10k$ samples, and `FLO-shuff` is trained with fixed batch-size. For `FLO-BiL`, it is trained with batch-size set to the negative sample-size desired, because all in-batch data are served as negatives. We use the same learning rate 10^{-4} for both cases, and this puts large-batch training at disadvantage, as

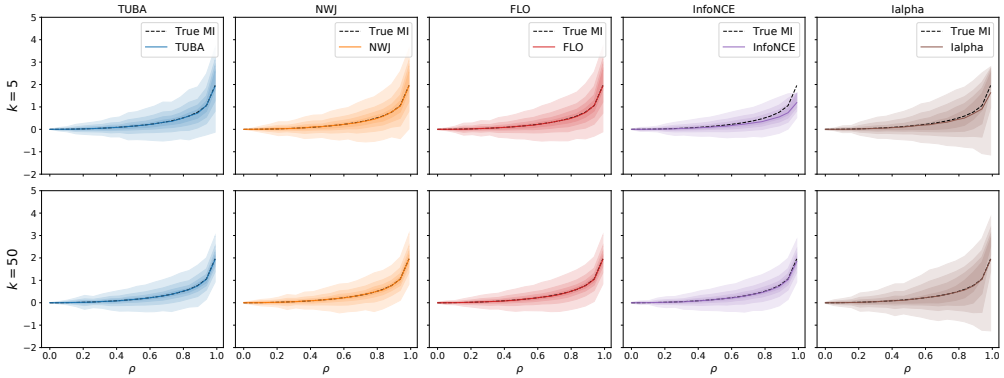

Figure 2.13: Bias variance plot for the popular MI bounds with the 2-D Gaussians. In this simpler case, `TUBA`, `NWJ` and `FLO` all give sharp estimate at $K = 5$. α-`InfoNCE` gives the worst variance profile. The reason is that because α-`InfoNCE` interpolates between the low-variance multi-sample `InfoNCE` and high-variance single-sample `NWJ` (see Figure 2.14), and in this case the variance from `NWJ` dominates.

fewer iterations are executed. To compensate for this, we use $T(K) = (\frac{K}{K_0})^{\frac{1}{2}} \cdot T_0$ to set the total number of iterations for `FLO-BiL`, where (T_0, K_0) are respectively the baseline training iteration and negative sample size used by `FLO-shuff`, and the number of negative sample K are $\{10, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$. We are mostly interested in computation efficiency here so we do not compare the bound. In Figure 2.11, we see the cost for training `FLO-shuff` grows linearly as expected. For `FLO-BiL`, a U-shape cost curve is observed. This is because bilinear implementation has three networks total, while the shared MLP only has one network. This implies more computations when the batch size is small, however, as the batch size grows, the computation overhead is amortized by better parallelism employed with the bilinear strategy, thus increasing overall efficiency until the device capacity has been reached. This explains the initial drop in cost, followed by the anticipated square-root growth.

2.5.5.7 Comparison of learning dynamics for different variational MI bounds

In Figure 2.12, we show the learning dynamics of competing estimators for the 20-D Gaussian when $\rho = 0.9$. We can find `FLO` achieves the best accuracy, it also learns fast and stably. `InfoNCE` learns very stably, yet its learning efficiency varies significantly in small-batch and large-batch setups.

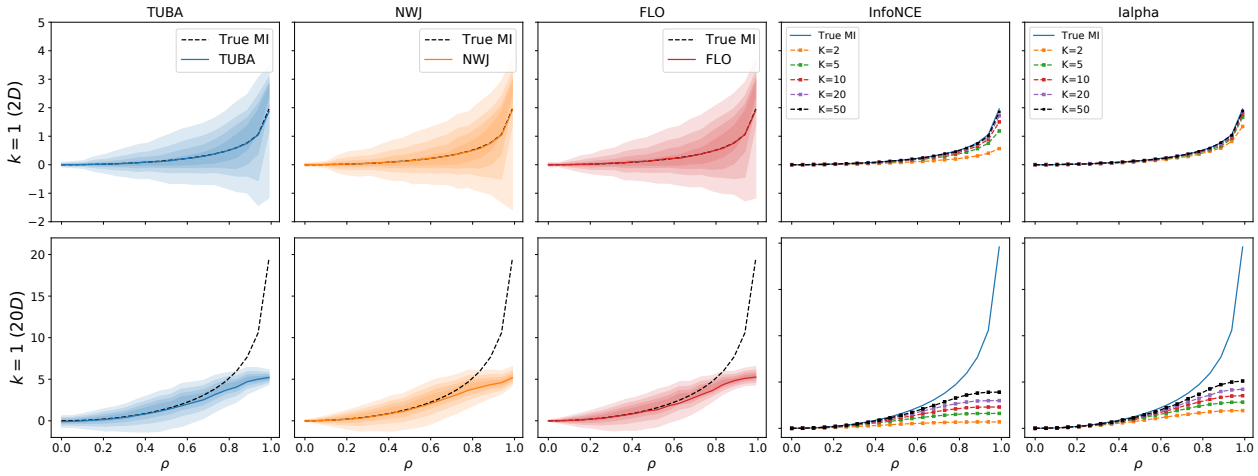

Figure 2.14: Bias variance plot for the popular MI bounds with the 2-D (upper panel) and 20-D (lower panel) Gaussians. Single-sample estimator of `TUBA`, `NWJ` and `FLO` (*i.e.*, $K = 1$) are compared to the multi-sample estimators of `InfoNCE` and α-`InfoNCE`.

2.5.5.8 Comprehensive analyses of bias-variance trade-offs

To supplement our results in the main paper, here we provide additional bias-variance plots for different MI estimators under various settings. In Figure 2.13 we show the bias-variance plot of MI estimates for 2-D Gaussians. In this case, the network used is sufficiently comprehensive so a sharp estimate is attainable. In all cases, the estimation variance grows with the MI value, which is consistent with the theoretical prediction that for tight estimators, the estimation variance grows exponentially with MI (McAllester and Stratos, 2018). In such cases, the argument for `InfoNCE`'s low-variance profile no longer holds: it is actually performing sub-optimally. For complex real applications, the negative sample size used might not provide an adequate estimate of ground-truth MI (*i.e.*, the $\log K$ cap), and that is when `InfoNCE`'s low-variance profile actually helps. We also notice that, when the MI estimate is not exactly tight, but very close to the true value, the variance dropped considerably. This might provide an alternative explanation (and opportunity) for the development of near-optimal MI estimation theories, which are not covered in existing literature.

We also tried the single-sample estimators for `NWJ`, `TUBA`, and `FLO` to their multi-sample `InfoNCE`-based counterparts (Figure 2.14), which is the comparison made by some of the prior studies (Note we do not apply Bilinear trick here, thus FLO seems similar to other methods).

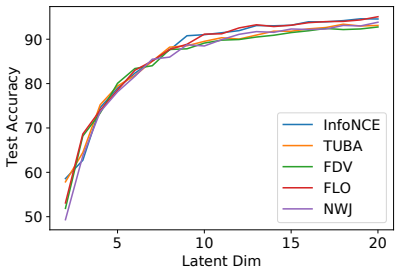

Figure 2.15: Extended results for the cross-view representation learning. `FDV` works best for smaller dimensions (≈ 5), and for higher dimensions (> 10) `FLO` and `InfoNCE` give the best results.

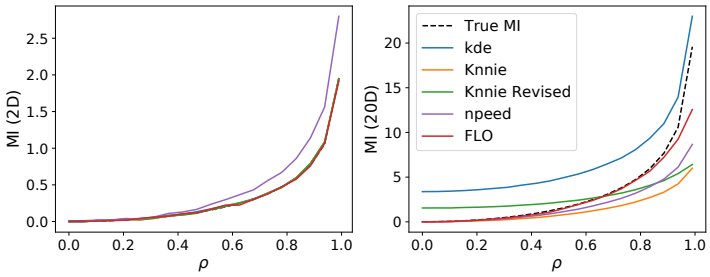

Figure 2.16: Comparison to classical MI estimators. (left) Easy 2D Gaussian, all models perform similarly. (right) Challenging 20D Gaussian, where `FLO` shows better overall accuracy. Note that the KDE accuracy in the high-dimensional setting is misjudged, as it is a well-known kernel-based density estimator scales poorly in high-dimensions.

In this setting, the variance single-sample estimators' variances are considerably larger, which explains their less favorable performance. Note that contradictory to theoretical predictions, a larger negative sample size does make `NWJ`, `TUBA`, and `FLO` tighter empirically, although the gains are much lesser compared to that of `InfoNCE` (partly because these three estimators are already fairly tight relative to `InfoNCE`). This might be explained by a better optimization landscape due to reduced estimation variance. We conjecture that for multi-sample `NWJ`, `TUBA`, and `FLO`, the performance in empirical applications such as self-supervised learning should be competitive to that of `InfoNCE`, which has never been reported in the literature.

2.5.6 Cross-view Representation Learning (Extended Analyses)

In addition to the results reported in the paper, we investigate how different latent dimension affect the results of the cross-view representation learning. We vary the latent dimension number from $d = 2$ to $d = 20$, and plot label prediction accuracy for the corresponding latent representations in Figure 2.15. The same setup for the bi-linear experiment is used for the MI estimation (for all MI estimators), where the images are flattened to be fed to the MLPs. The representations are trained for 50 epochs and the prediction model is trained for 50 epochs. We also trained the model for another 50 epochs and the conclusions are similar. We see that `FDV` works well for lower dimensions (*e.g.*, $d \approx 5$), and what works better for higher dimensions ($d > 10$) are `FLO` and `InfoNCE`.

2.5.7 Comparison with Classical MI Estimators

We also compare our `FLO` estimator to the classical MI estimators in Figure 2.16. The following implementations of baseline estimators for multi-dimensional data are considered: (*i*) *KDE*: we use kernel density estimators to approximate the joint and marginal likelihoods, then compute MI by definition; (*ii*) *NPEET* [8], a variant of Kraskov's K-nearest neighbor (KNN) estimator (Kraskov et al., 2004; Ver Steeg and Galstyan, 2013); (*iii*) *KNNIE* [9], the original KNN-estimator and its revised variant (Gao et al., 2018). These models are tested on 2-D and 20-D Gaussians with varying strength of correlation, with their hyper-parameters tuned for best performance. Note that the notation of "best fit" is a little bit subjective, as we will fix the hyper-parameter for all dependency strength, and what works better for weak dependency might necessarily not work well for strong dependency. We choose the parameter whose result is visually most compelling. In addition to the above, we have also considered other estimators such as maximal-likelihood density ratio [10] (Suzuki et al., 2008) and KNN with local non-uniformity correction [11]. However, these models either do not have a publicly available multi-dimensional implementation, or their codes do not produce reasonable results [12].

2.5.8 Comparison to Parametric Variational Estimators and Bounds Targeting Alternative Information Metrics

Parametric variational estimators are typically associated with the upper bound of MI (Cheng et al., 2020; Poole et al., 2019). Inspired by multi-sample variational bounds for likelihood estimation, (Brekelmans et al., 2021) derived a generic family of importance-weighted MI bounds that are provably tighter. These bounds usually require additional knowledge of likelihood, and consequently, they can not be directly used for data-driven MI estimations. On

[8] `https://github.com/gregversteeg/NPEET`
[9] `https://github.com/wgao9/knnie`
[10] `https://github.com/leomuckley/maximum-likelihood-mutual-information`
[11] `https://github.com/BiuBiuBiLL/NPEET_LNC`
[12] These are third-party Python implementations, so BUGs are highly likely.

the other hand, these models do not suffer from the exponential scaling of variance suffered by non-parametric MI estimators. Note that MI is not the only measure to assess association between two random variables, some alternatives can potentially do better for specific applications. Examples include $\mathcal{V}$ information (Xu et al., 2020), Rényi information (Lee and Shin, 2022), and the spectral information (HaoChen et al., 2021).

2.5.9 Regression with Sensitive Attributes (Fair Learning) Experiments

2.5.9.1 Introduction to fair machine learning

Nowadays consequential decisions impacting people's lives have been increasingly made by machine learning models. Such examples include loan approval, school admission, and advertising campaign, amongst others. While automated decision-making has greatly simplified our lives, concerns have been raised on (inadvertently) echoing, even amplifying societal biases. In particular, algorithms are vulnerable to inheriting discrimination from the training data and passing on such prejudices in their predictions.

To address the growing need for mitigating algorithmic biases, research has been devoted in this direction under the name fair machine learning. While discrimination can take many definitions that are not necessarily compatible, in this study we focus on the most widely recognized criteria *Demographic Parity* (DP), as defined below

Definition 2.5.6 (Demographic Parity, (Dwork et al., 2012))**.** The absolute difference between the selection rates of a decision rule $\hat{y}$ of two demographic groups defined by sensitive attribute s, *i.e.*,

$$\mathrm{DP}(\hat{Y}, S) = \left| \mathbb{P}(\hat{Y} = 1 | S = 1) - \mathbb{P}(\hat{Y} = 1 | S = 0) \right|. \tag{2.50}$$

With multiple demographic groups, it is the maximal disparities between any two groups:

$$\mathrm{DP}(\hat{Y}, S) = \max_{s \neq s'} \left| \mathbb{P}(\hat{Y} = 1 | S = s) - \mathbb{P}(\hat{Y} | S = s') \right|. \tag{2.51}$$

2.5.9.2 Experiment details and analyses

To scrub the sensitive information from data, we consider the *in-processing* setup

$$\mathcal{L} = \texttt{Loss}(\underbrace{\text{Predictor}(\text{Encoder}(x_i)), y_i)}_{\text{Primary loss}} + \lambda \underbrace{I(s_i, \text{Encoder}(x_i))}_{\text{Debiasing}}. \tag{2.52}$$

By regularizing model training with the violation of specified fairness metric $\Delta(\hat{y}, s)$, fairness is enforced during model training. In practice, people recognize that appealing to fairness sometimes costs the utility of an algorithm (*e.g.*, prediction accuracy) (Hardt et al., 2016). So most applications seek to find their own sweet points on the fairness-utility curve. In our example, it is the *DP-error* curve. A fair learning algorithm is considered good if it has a lower error at the same level of DP control.

In this experiment, we compare our MI-based fair learning solutions to the state-of-the-art methods. *Adversarial debiasing* tries to maximize the prediction accuracy for while minimize the prediction accuracy for sensitivity group ID (Zhang et al., 2018). We use the implementation from `AIF360`[13] package (Bellamy et al., 2018). FERMI is a density-based estimator for the *exponential Rényi mutual information* $\texttt{ERMI} \triangleq \mathbb{E}_{p(x,y)}[\frac{p(x,y)}{p(x)p(y)}]$, and we use the official codebase. For evaluation, we consider the *adult* data set from UCI data repository (Asuncion and Newman, 2007), which is the 1994 census data with 30k samples in the train set and 15k samples in the test set. The target task is to predict whether the income exceeds \$50k, where gender is used as a protected attribute. Note that we use this binary sensitive attribute data just to demonstrate our solution is competitive to existing solutions, where mostly developed for binary sensitive groups. Our solution can extend to more general settings where the sensitive attribute is continuous and high-dimensional.

We implement our fair regression model as follows. To embrace data uncertainty, we consider latent variable model $p_\theta(y, x, z) = p_\theta(y|z)p_\theta(x|z)p(z)$, where $v = \{x, y\}$ are the observed predictor and labels. Under the variational inference framework (Kingma and Welling, 2014),

[13] `https://github.com/Trusted-AI/AIF360`

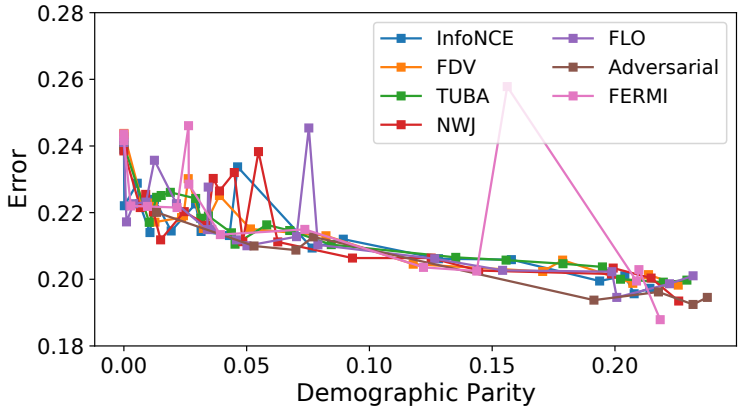

Figure 2.17: Fair Learning Result.

we write the ELBO$(v; p_\theta(v, z), q_\phi(z|v))$ as

$$\mathbb{E}_{Z\sim q_\phi(z|v)}[\log p_\theta(y|Z)] + \cancel{\mathbb{E}_{Z\sim q_\phi(z|v)}[\log p_\theta(x|Z)]} - \beta \mathrm{KL}(q_\phi(z|v) \parallel p(z)) \tag{2.53}$$

$p(z)$ is modeled with standard Gaussian, and the approximate posterior $q_\phi(z|v)$ is modeled by a neural network parameterizing the mean and variance of the latents (we use the standard mean-field approximation so cross-covariance is set to zero), and β is a hyperparameter controlling the relative contribution of the KL term to the objective. Note that unlike in the standard ELBO we have dropped the term $\mathbb{E}_{Z\sim q_\phi(z|v)}[\log p_\theta(x|Z)]$ because we are not interested in modeling the covariates. Note this coincides with the *variational information bottleneck* (VIB) formulation (Alemi et al., 2016). Additionally, the posterior $q_\phi(z|v)$ will not be conditioned on y, but only on x, because in practice, the labels y are not available at inference time. All networks used here are standard three-layer MLP with 512 hidden-units.

For Figure 2.17, we note that the adversarial de-biasing actually crashed in the DP range $[0.1, 0.18]$, so the results have to be removed. Since interpolation is used to connect different data points, it makes the adversarial scheme look good in this DP range, which is not the case. FERMI also gave unstable estimation in the DP range $[0.1, 0.18]$. Among the MI-based solutions, `NWJ` was the most unstable. Performance-wise, `InfoNCE`, `TUBA` and `FDV` are mostly tied, with the latter two slightly better in the "more fair" solutions (*i.e.*, at the low DP end).

Table 2.3: `MNIST` cross-view results.

Model	CCA	NWJ	TUBA	InfoNCE	FLO	FDV
Accuracy	67.78	76.71	79.49	79.27	79.47	**80.14**
$\hat{I}(x_l, x_r)$	NA	5.73	4.78	4.65	4.84	4.67

2.5.10 Self-supervised Learning

Our codebase is modified from a public `PyTorch` implementation[14]. Specifically, we train 256-dimensional feature representations by maximizing the self-MI between two random views of data, and report the test set classification accuracy using a linear classifier trained to convergence. We report performance based on `ResNet-50`, and some of the learning dynamics analyses are based on `ResNet-18` for reasons of memory constraints. Hyper-parameters are adapted from the original `SimCLR` paper. For the large-batch scaling experiment, we first grid-search the best learning rate for the base batch-size, then grow the learning rate linearly with batch-size.

2.5.11 Bayesian Experimental Design

2.5.11.1 Noisy Linear Model

Our setup is the same as the Noisy Linear Model in (Kleinegesse and Gutmann, 2020). We use 10 individual experimental designs. For encoder θ and encoder y, we use MLP with 2-layer, 128-dim hidden layer, and set the feature dim as 512. We train models in 5000 epochs, the batch size is 64, and the learning rate is $2 * 10^{-5}$. Four MI estimators (NWJ, TUBA, InfoNCE, and FLO) have been compared in this experiment and we got four optimized designs. Then, we use MCMC to estimate the posterior of the parameters.

[14] `https://github.com/sthalles/SimCLR`

2.5.11.2 Pharmacokinetic Model

The settings of this experiment refer to the Pharmacokinetic Model of (Kleinegesse and Gutmann, 2020). We use 10 individual experimental designs. The MLP is with 2-layer, 128-dim hidden layer, and the output feature dim as 512. We train 10000 epochs with a learning rate is 10^{-5} via four methods (NWJ, TUBA, InfoNCE, FLO).

2.5.11.3 SIR Model

We here consider the spread of a disease within a population of N individuals, modeled by stochastic versions of the well-known SIR (Allen et al., 2008). a susceptible state $S(t)$ and can then move to an infectious state $I(t)$ with an infection rate of β. These infectious individuals then move to a recovered state $R(t)$ with a recovery rate of γ, after which they can no longer be infected. The SIR model, governed by the state changes $S(t) \to I(t) \to R(t)$, thus has two model parameters $\boldsymbol{\theta}_1 = (\beta, \gamma)$.

The stochastic versions of these epidemiological processes are usually defined by a continuous-time Markov chain (CTMC), from which we can sample via the Gillespie algorithm (Allen, 2017). However, this generally yields discrete population states that have undefined gradients. In order to test our gradient-based algorithm, we thus resort to an alternative simulation algorithm that uses stochastic differential equations (SDEs), where gradients can be approximated.

We first define population vectors $X_1(t) = (S(t), I(t))$ for the SIR model and $X_2(t) = (S(t), E(t), I(t))$ for the SEIR model. We can effectively ignore the population of recovered because the total population is fixed. The system of Itô SDEs for the above epidemiological processes is

$$\mathrm{d}\boldsymbol{X}(t) = \boldsymbol{f}(\boldsymbol{X}(t))\,\mathrm{d}t + \boldsymbol{G}(\boldsymbol{X}(t))\,\mathrm{d}\boldsymbol{W}(t), \tag{2.54}$$

where $\boldsymbol{f}$ is the drift term, $\boldsymbol{G}$ is the diffusion term and $\boldsymbol{W}$ is the Wiener process. Euler-Maruyama algorithm is used to simulate the sample paths of the above SDEs.

$$\boldsymbol{f}_{\text{SIR}} = \begin{pmatrix} -\beta \frac{S(t)I(t)}{N} \\ \beta \frac{S(t)I(t)}{N} - \gamma I(t) \end{pmatrix}, \boldsymbol{G}_{\text{SIR}} = \begin{pmatrix} -\sqrt{\beta \frac{S(t)I(t)}{N}} & 0 \\ \sqrt{\beta \frac{S(t)I(t)}{N}} & -\sqrt{\gamma I(t)} \end{pmatrix} \tag{2.55}$$

We use the infection rate (I) as 0.1 and the recovery (R) rate as 0.01. The independent priors are N(0.1,0.02) and N(0.01, 0.002). The initial infection number is 10. We update MI one time after updating the sampler's three steps. We use an RNN network with 2 layer 64 dim hidden layer construction to decode the sequential design.

2.5.12 Meta Learning

Intuitions. Now let us describe the new `Meta-FLO` model for meta-learning. Given a model space $\mathcal{M}$ and a loss function $\ell : \mathcal{M} \times \mathcal{Z} \to \mathbb{R}$, the true risk and the empirical risk of $f \in \mathcal{M}$ are respectively defined as $R_t(f) \triangleq \mathbb{E}_{Z \sim \mu_t}[\ell(f, Z)]$ and $\hat{R}_t(f; \mathbb{S}_t) \triangleq \frac{1}{m} \sum_{i=1}^{m} \ell(f, Z_i)$.Let us denote R_τ is the generalization error for the task distribution τ where all tasks originate, and $\hat{R}_\tau$ is the empirical estimate. Our heuristic is simple, that is to optimize a tractable upper bound of the generalization risk given by

$$R_\tau \leq \underbrace{\hat{R}_\tau}_{\text{Utility}} + \underbrace{|R_\tau - \hat{R}_\tau|}_{\text{Generalization}} \triangleq \mathcal{L}_{\text{upper}}. \tag{2.56}$$

For meta-learning, we sample n-tasks for training and n'-tasks for testing, respectively denoted as $\mathbb{S}_{1:n}$ and $\mathbb{S}\text{test}_{1:n'}$. We further decouple the learning algorithm into two parts: the *meta-learner* $\mathcal{A}_{\text{meta}}(\mathbb{S}_{1:n})$ that consumes all train data to get the *meta-model* f_{meta}, and then *task-adaptation learner* $\mathcal{A}_{\text{adapt}}(f_{\text{meta}}, \mathbb{S}_t)$ which adapts the meta-model to the individual task data $\mathbb{S}_t$ to get task model f_t. For parameterized models such as deep nets, we denote Θ as our *meta parameters* and E_t as *task-parameters*, that is to say $\Theta \triangleq \mathcal{A}_{\text{meta}}(\mathbb{S}_{1:n})$, $E_t \triangleq \mathcal{A}_{\text{adapt}}(\Theta, \mathbb{S}_t)$, where Θ, E_t can be understood as weights of deep nets. In subsequent discussions, we will also call E_t the *task-embedding*. We can define the population *meta-risk* as $R_\tau(\Theta) \triangleq \mathbb{E}_{t, \Theta = \mathcal{A}_{\text{meta}}(\mathbb{S}_{1:n})}[\mathbb{E}_{E_t = \mathcal{A}_{\text{adapt}}(\Theta, \mathbb{S}_t)}[R_t(f_{E_t})]]$, and similarly for the empirical risk $\hat{R}_\tau$ evaluated on the query set $\mathbb{Q}_t$. Our model is based on the following inequality (Anonymous,

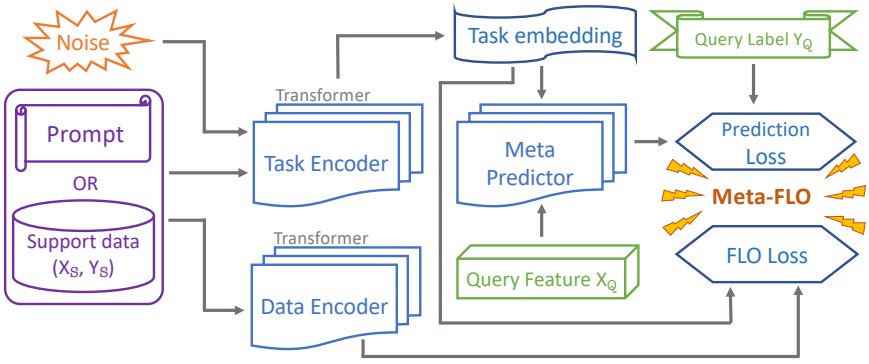

Figure 2.18: Model architecture of `Meta-FLO`.

2022):

$$\lim_{n\to\infty} |\mathbb{E}[R - \hat{R}]| \leq \sqrt{\frac{2\sigma^2}{m} I(E_t; \mathbb{S}_t|\Theta)} \tag{2.57}$$

which gives the main objective $\mathcal{L}_{\texttt{Meta-FLO}}(f) = \hat{R}(f) + \lambda\sqrt{I_{\texttt{FLO}}(\hat{\mathcal{D}}_t; \hat{E}_t)}$. We summarize our model architecture in Figure 2.18.

The sin-wave adaptation experiment involves regressing from the input ($x \sim \text{Uniform}([-5, 5])$) to the output of a sine wave $\kappa \sin(x - \gamma)$, where amplitude $\kappa \sim \text{Uniform}([0.1, 5])$ and phase ($\gamma \sim \text{Uniform}([0, \pi])$ of the sinusoid vary for each task. We use mean-squared error (MSE) as our loss and set the support-size $= 3$ and query-size $= 2$. We use simple three-layer MLPs for all the models: regressor, prompt encoder, and `FLO` critics, with hidden units all set to $[512, 512]$. During training, we use an episode-size of 64. For MAML, we use the first-order implementation (FOMAML) and set the inner learning rate to $\alpha = 10^{-4}$. For `Meta-FLO`, we set regularization strength to $\lambda = 10^{-2}$.

Bibliography

A. Alemi, B. Poole, I. Fischer, J. Dillon, R. A. Saurous, and K. Murphy. Fixing a broken ELBO. In *ICML*, pages 159–168, 2018.

A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. Deep variational information bottleneck. In *ICLR*, 2016.

L. J. Allen. A primer on stochastic epidemic models: Formulation, numerical simulation, and analysis. *Infectious Disease Modelling*, 2(2):128–142, 2017.

L. J. Allen, F. Brauer, P. Van den Driessche, and J. Wu. *Mathematical epidemiology*, volume 1945. Springer, 2008.

Anonymous. Meta-flo: Principled simple fast few-shot learning with stochastic prompt encoding networks. 2022.

M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017.

A. Asuncion and D. Newman. Uci machine learning repository, 2007.

F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3(Jul):1–48, 2002.

D. Barber and F. Agakov. The IM algorithm: a variational approach to information maximization. *NIPS*, 16:201, 2004.

D. Barber and F. V. Agakov. Information maximization in noisy channels: A variational approach. *NIPS*, 16, 2003.

R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE transactions on Neural Networks*, 5(4):537–550, 1994.

M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm. Mutual information neural estimation. In *ICML*, 2018.

R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. N. Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias, Oct. 2018. URL `https://arxiv.org/abs/1810.01943`.

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

J. O. Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

R. Brekelmans, S. Huang, M. Ghassemi, G. Ver Steeg, R. B. Grosse, and A. Makhzani. Improving mutual information estimation with annealed and energy-based bounds. In *ICLR*, 2021.

K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pages 273–304, 1995.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin. CLUB: A contrastive log-ratio upper bound of mutual information. In *ICML*, 2020.

M. D. Donsker and S. S. Varadhan. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on Pure and Applied Mathematics*, 36(2):183–212, 1983.

C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, 2012.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

A. Foster, M. Jankowiak, E. Bingham, P. Horsfall, Y. W. Teh, T. Rainforth, and N. Goodman. Variational bayesian optimal experimental design. In *NeurIPS*, 2019.

A. Foster, M. Jankowiak, M. O'Meara, Y. W. Teh, and T. Rainforth. A unified stochastic gradient approach to designing bayesian-optimal experiments. In *AISTATS*, 2020.

A. Foster, D. R. Ivanova, I. Malik, and T. Rainforth. Deep adaptive design: Amortizing sequential bayesian experimental design. In *ICML*, 2021.

S. Gao, G. Ver Steeg, and A. Galstyan. Efficient estimation of mutual information for strongly dependent variables. In *AISTATS*, 2015.

W. Gao, S. Oh, and P. Viswanath. Demystifying fixed k-nearest neighbor information estimators. *IEEE transactions on Information Theory*, 64(8):5629–5661, 2018.

C. J. Geyer. On the convergence of monte carlo maximum likelihood calculations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(1):261–274, 1994.

A. Gretton, R. Herbrich, and A. J. Smola. The kernel mutual information. In *ICASSP*, 2003.

A. Gretton, R. Herbrich, A. Smola, O. Bousquet, B. Schölkopf, et al. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 2005.

J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020.

A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *SIGKDD*, 2016.

U. Gupta, A. Ferber, B. Dilkina, and G. V. Steeg. Controllable guarantees for fair outcomes via contrastive information estimation. *arXiv preprint arXiv:2101.04108*, 2021.

M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010.

J. Z. HaoChen, C. Wei, A. Gaidon, and T. Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *NeurIPS*, 2021.

M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *NIPS*, 2016.

K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *NIPS*, 2014.

G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of convex analysis*. Springer Science & Business Media, 2012.

R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(Apr):695–709, 2005.

D. Ivanova, A. Foster, S. Kleinegesse, M. U. Gutmann, and T. Rainforth. Implicit deep adaptive design: Policy-based experimental design without likelihoods. *NeurIPS*, 2021.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.

S. Kleinegesse and M. U. Gutmann. Bayesian experimental design for implicit models by mutual information neural estimation. In *ICML*, 2020.

S. Kleinegesse and M. U. Gutmann. Gradient-based bayesian experimental design for implicit models using mutual information lower bounds. *arXiv preprint arXiv:2105.04379*, 2021.

S. Kleinegesse, C. Drovandi, and M. U. Gutmann. Sequential bayesian experimental design for implicit models via mutual information. *Bayesian Analysis*, 1(1):1–30, 2021.

A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

K. Lee and J. Shin. RényiCL: Contrastive representation learning with skew rényi divergence. In *NeurIPS*, 2022.

R. Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

Z. Ma and M. Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. *arXiv preprint arXiv:1809.01812*, 2018.

D. J. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE transactions on Medical Imaging*, 16(2):187–198, 1997.

D. McAllester and K. Stratos. Formal limitations on the measurement of mutual information. *arXiv preprint arXiv:1811.04251*, 2018.

A. Mnih and K. Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In *NIPS*, 2013.

R. M. Neal. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.

X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE transactions on Information Theory*, 56 (11):5847–5861, 2010.

S. Nowozin, B. Cseke, and R. Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *NIPS*, 2016.

A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

S. E. Palmer, O. Marre, M. J. Berry, and W. Bialek. Predictive information in a sensory population. *Proceedings of the National Academy of Sciences*, 112(22):6908–6913, 2015.

L. Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6): 1191–1253, 2003.

F. Pérez-Cruz. Estimation of information theoretic measures for continuous random variables. In *NIPS*, 2008.

J. P. Pluim, J. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE transactions on Medical Imaging*, 22(8):986–1004, 2003.

B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker. On variational bounds of mutual information. In *ICML*, 2019.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

T. Rainforth, R. Cornish, H. Yang, A. Warrington, and F. Wood. On nesting monte carlo estimators. In *International Conference on Machine Learning*, 2018.

L. E. Reichl. *A modern course in statistical physics*. John Wiley & Sons, 2016.

D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti. Detecting novel associations in large data sets. *science*, 334(6062):1518–1524, 2011.

C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

J. Song and S. Ermon. Understanding the limitations of variational mutual information estimators. In *ICLR*, 2020.

T. Suzuki, M. Sugiyama, J. Sese, and T. Kanamori. Approximating mutual information by maximum likelihood density ratio estimation. In *New challenges for feature selection in data mining and knowledge discovery*, 2008.

C. Tao, L. Chen, S. Dai, J. Chen, K. Bai, D. Wang, J. Feng, W. Lu, G. Bobashev, and L. Carin. On Fenchel mini-max learning. In *NeurIPS*, 2019.

Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.

K. Torkkola. Feature extraction by non-parametric mutual information maximization. *Journal of machine learning research*, 2003.

M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, and M. Lucic. On mutual information maximization for representation learning. *ICLR*, 2020.

G. Ver Steeg and A. Galstyan. Information-theoretic measures of influence based on content

dynamics. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 3–12, 2013.

L. Wen, Y. Zhou, L. He, M. Zhou, and Z. Xu. Mutual information gradient estimation for representation learning. In *ICLR*, 2020.

C. J. Wu and M. S. Hamada. *Experiments: planning, analysis, and optimization*, volume 552. John Wiley & Sons, 2011.

Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

A. Xu and M. Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In *NIPS*, 2017.

Y. Xu, S. Zhao, J. Song, R. Stewart, and S. Ermon. A theory of usable information under computational constraints. In *ICLR*, 2020.

B. H. Zhang, B. Lemoine, and M. Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340, 2018.

S. Zheng, J. Pacheco, and J. Fisher. A robust approach to sequential information theoretic planning. In *ICML*, 2018.

Chapter 3 Multi-layer Sliced Design and Analysis with Application to AI Assurance

Abstract

Enhancing AI assurance in tuning configurations and hyper-parameters of AI algorithms is an important problem in many applications. This work provides an experimental design method to address this challenging problem. The key idea of the method is to conduct an efficient experimental design to detect and quantify the effects of hyper-parameters on the performance of AI algorithms. Specifically, the method proposes a multi-layer sliced design to enable quantifying the effects of slice factors and design factors to account for hyper-parameters having different effects under different configurations of the AI algorithm. Moreover, this method develops an effective analysis procedure to estimate the effects of these factors and test their significance. The performance of the proposed design and analysis methods is successfully illustrated by simulation studies and real-world AI applications.

Key Words: Hyper-parameters, Sliced design, Induced lasso, Deep learning

3.1 Introduction

In the modern design of experiment applications, including online experiments and hyper-parameter tuning for AI algorithms, there often exist several factors of particular interest in comparison with other design factors. For example, (Sadeghi et al., 2020) considered an experiment on how to construct online designs of website layouts across multiple platforms such as laptops, cellphones, and iPads. In their work, the factor of "platform" is identified as a *slice factor* and other factors related to the website layout are considered as *design factors*. We note that the slice factors differ from design factors in the sense that the experimenters are interested in estimating the effects of design factors under different levels of the slice factor. The work of Sadeghi et al. (2020) mainly considers a sliced design with a single slice factor. However, many applications have multiple slice factors of interest. For example, a retail company would like to conduct online experiments of web advertisements across different user devices (e.g., desktops, cellphones) and apps (e.g., Facebook and Instagram). Such experiments will involve two slice factors: user devices and social-media apps. It is important to evaluate the impact of advertisements under every combination of the slice factors. Another example is that AI assurance faces the challenge of exploring the effects of hyper-parameters on model performance. Oftentimes, the effect of hyper-parameters on the model performance can be different across different models and optimization methods in the AI algorithm. From this viewpoint, we consider the modeling choices and optimization methods used in the AI algorithm can be considered as two slice factors, and other hyper-parameters in the neural network of the AI algorithm as design factors.

In the area of AI assurance (Batarseh et al., 2021; Batarseh and Freeman, 2022; Batarseh et al., 2023), it is important to investigate the effects of hyper-parameters on the model performance of AI algorithms under different configurations (e.g., different level combinations of model choice and optimization method). Investigating the effects of hyper-parameters on the AI algorithm has attracted considerable interest (Snoek et al., 2012; Bergstra and Bengio, 2012; Bardenet et al., 2013; Li et al., 2020a,b). To design an appropriate experiment to

evaluate the performance of the AI algorithm, a suitable design is needed to make the effects of hyper-parameters estimable under each configuration, which is closely related to the concept of the sliced design strategy (Sadeghi et al., 2020).

In this work, we propose a *multi-layer sliced design* (MLSD) to deal with multiple slice factors, with application to the investigation of the effect of hyper-parameters on the AI algorithm under different configurations. In this application, the factors involving configurations are considered slice factors. Multiple slice factors can have different importance or a hierarchical structure. Specifically, we consider the MLSD with each factor at two levels and propose the ordered word-length pattern for finding the ordered minimum aberration design for MLSD. Our proposed criterion is flexible in dealing with both equal importance and ordered importance of the slice factors. Moreover, we also develop a novel analysis method to obtain a parsimonious model by leveraging the sparsity principle (Box et al., 1978; Wu and Hamada, 2011) in the design of experiments (Yuan et al., 2007; Seeger et al., 2007; Dougherty et al., 2015). We further enable the hypothesis testing of significant effects among a variety of main effects, two-factor interaction effects. The proposed MLSD has a stronger estimation capability for slice factors, and it can estimate the corresponding factorial effects accurately. Recent work on sliced experimental design (Sadeghi et al., 2020) considered multiple submodel estimations for each platform. In contrast, our proposed analysis method can estimate these effects simultaneously by adopting the induced Lasso technique (Cilluffo et al., 2020). The proposed MLSD framework has practical applications beyond the investigation of hyper-parameters in AI algorithms. It can also be used in other aspects of AI assurance, such as investigating the robustness of AI algorithms.

The remainder of the article is organized as follows: In Section 3.2, we briefly review the factorial design and its use in AI-related applications. Section 3.3 details the proposed multi-layer sliced design. In Section 3.4, we focus on the analysis method for estimation of the effects of interest based on MLSD. Section 3.5 conducts simulations to examine the performance of the proposed design and analysis method. Section 3.6 presents a practical application of our method in AI assurance. Finally, we conclude the article with some discussions in Section 3.7.

3.2 Literature Review

Hyper-parameter tuning for AI algorithms is important(Mantovani et al., 2016; Lee et al., 2018; Probst et al., 2019) but often costly in practice (Hutter et al., 2019). Traditionally, this is mainly a manual process heavily relying on the experience of investigators. For simple settings with one or two hyper-parameters involved (*e.g.*, bandwidth parameter selection for kernel learning) in the AI algorithm, straightforward exhaustive approaches such as grid search usually work well (Bergstra et al., 2011). However, this simple approach quickly becomes impractical when there are a large number of hyper-parameters, where Monte Carlo approaches including random search (Bergstra and Bengio, 2012) and the one-factor-at-time procedure are often used. Such methods are unfortunately ineffective for high-dimensional hyper-parameters or can miss the interaction between different hyper-parameters. Methods such as genetic algorithm (GA) (Lessmann et al., 2005) and particle swarm optimization (PSO) (Lorenzo et al., 2017) are also used as heuristics to prioritize the settings of hyper-parameters. Recently, Bayesian optimization has gained great attention by its effectiveness, especially in complex models such as deep neural networks (Eggensperger et al., 2013; Feurer et al., 2015; Klein et al., 2017) and knowledge transfer (Yogatama and Mann, 2014; Joy et al., 2016). Many existing Bayesian optimization techniques face a common challenge. The acquisition criterion is often non-convex and potentially non-differentiable, making it difficult for standard local numerical optimization methods to find the optimal solution reliably. Recent work has explored Delaunay triangulation to address this challenge (Gramacy et al., 2021).

When emphasizing the main effects and two-factor interaction effects, one can exploit fractional factorial design (Box and Hunter, 1961; Gunst and Mason, 2009; Wu and Hamada, 2011) to use a small number of experimental trials (i.e., a level combination of factors) to adequately estimate the effects up to the second order. We use ideas from the design of experiments literature, the hyper-parameter tuning can be investigated from a new and different angle. By considering the hyper-parameters with possible discrete values the factorial design can be applied to estimate the effects of different hyper-parameters (Cheng, 2016). Due to

limitations on resources, the fractional factorial design aims at economically investigating the cause-and-effect relationships (Box and Hunter, 1961; Gunst and Mason, 2009). It allows for more efficient use of resources by reducing the number of experiments. To find optimal fractional factorial designs, a widely used criterion is the maximum resolution criterion (Box and Hunter, 1961) and the minimum aberration criterion (Box and Hunter, 1961; Fries and Hunter, 1980; Tang and Wu, 1996), both of which are based on using the word-length pattern (Fries and Hunter, 1980; Cheng et al., 1999; Wu and Hamada, 2011).

In the direction of using fractional factorial designs for novel applications, Sadeghi et al. (2020) proposed the sliced design for the multi-platform online experiments. Their research focused on identifying the optimal sliced design through the sliced minimum aberration criterion, and they developed linear models to estimate all effects. Chang (2022) provides theoretical support for the sliced minimum aberration design from the view of Bayesian analysis. However, the sliced design is not readily applicable to scenarios with multiple slicing factors. Additionally, their methodology requires separate modeling and estimation processes for different platforms.

The sliced design approach proposed by Sadeghi et al. (2020), which treats factors differently, has connections with other existing methodologies in the literature. For instance, in a split-plot design (Jones and Nachtsheim, 2009; Wu and Hamada, 2011), the whole plot factors are assigned to main plots, and subplot factors are applied within these subplots (Fisher, 1970). Similarly, robust parameter designs explore interactions between control factors and noise factors (i.e., uncontrollable variables) (Taguchi, 1987). The branching and nested design (Phadke, 1995; Hung et al., 2009) includes branching and nested factors and the nested factors differ for the levels of branching factors.

3.3 Multi-Layer Sliced Design

This section details the proposed multi-layer sliced design (MLSD). In the MLSD, we consider two classes of factors, slice (platform) factors, and design factors, as shown in Table 3.1. We

denote the slice factors are $S_i, i = 1, ..., m$, where each S_i has l_i levels. The design factors are $X_j, j = 1, ..., k$, where each X_j has h_j levels. Different level combinations of the design factors are to be conducted to understand the effects of design factors on the response (i.e., experiment outputs). The slice factors are often of great importance to be considered as platform effects, i.e., the experimenter expects that the effect of design factors can vary according to the different settings of slice factors. It is important to distinguish the roles of the slice factors and design factors in both design criterion and data analysis. Additionally, considering multiple slice factors allows for the incorporation of more complex statistical models that can address both the main effects of each individual factor and their interaction effects, thereby enhancing the flexibility and depth of the analysis.

Table 3.1: Factors in the Multi-Layer Sliced Design

	Slice Factors			Design Factors		
	S_1	...	S_m	X_1	...	X_k
Number of Levels	l_1	...	l_m	h_1	...	h_k

For the example of a two-layer sliced Design in a webpage layout application, online platforms can be regarded as slice factors. The S_1 can be the electronic devices such as cell phones, and laptops for web browsing. And S_2 is the social apps such as Instagram or Facebook used by customers. Different advertisement designs under the ith level of S_1 and jth level of S_2 can be constructed as $\mathbf{D}_{ij}$ for the design factors $X_1, \ldots, X_k$. The $\mathbf{D}_{ij}$ can be a full factorial design or a fractional factorial design, while all level combinations of the slice factors will be considered. Using such a multi-layer sliced design, the experimenter can investigate how the slice factors and design factors affect customer shopping behaviors.

For ease of presentation, we will start with our proposed method under the two-layer sliced design with $m = 2$. The presented definition, properties, and analysis methods can be extended to multi-layer sliced design with $m \geq 3$.

Definition 3.3.1 (Two-layer sliced Design)**.** Consider two slice factors S_1 with l_1 levels and S_2 with l_2 levels and k design factors $X_1, X_2, ..., X_k$. The whole design for slice factors and

design factors, denoted as $\mathbf{D}$, consists of subdesigns, $\mathbf{D}_{11}, ..., \mathbf{D}_{1,l_2}, ..., \mathbf{D}_{l_1,1}, ..., \mathbf{D}_{l_1,l_2}$ associated with each level combination of slice factors.

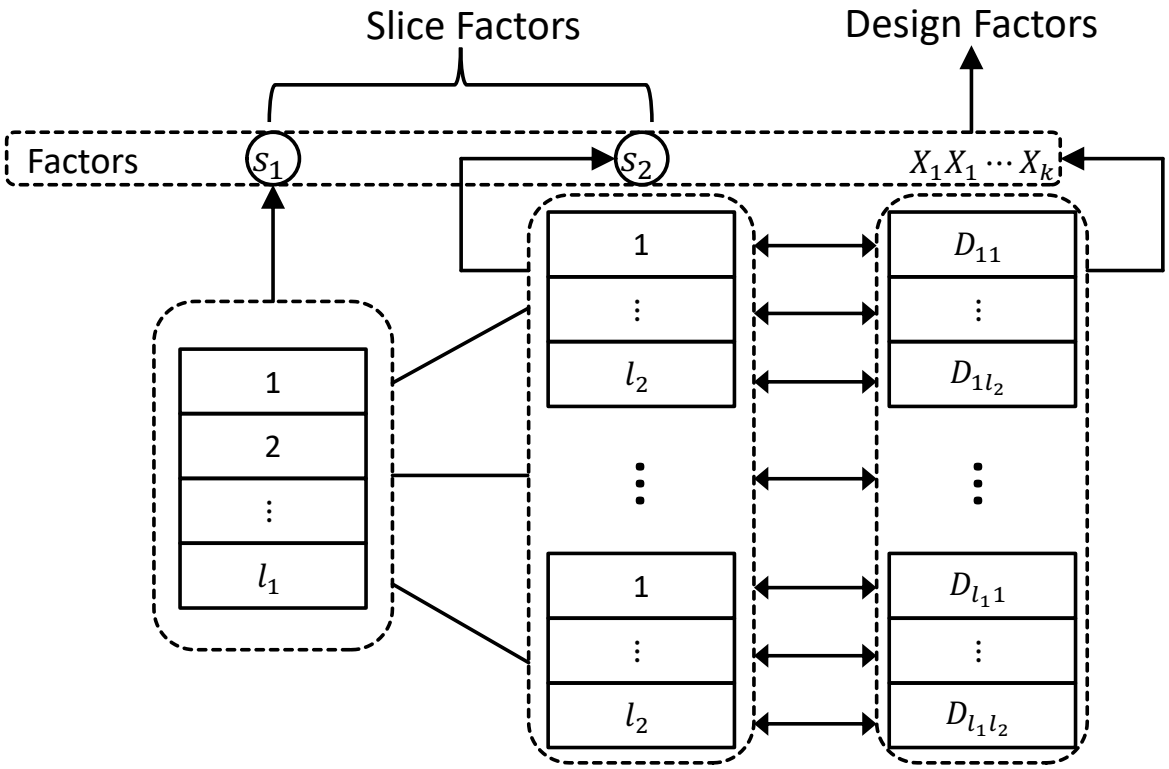

Figure 3.1: An Illustration of a Two-Layer Sliced Design in Definition 3.3.1

Figure 3.1 presents the design set $\mathbf{D}$ of the experiment in Definition 3.3.1. When one considers both the slice factor and design factor with two levels, a full factorial two-layer sliced design $\mathbf{D}$ can be denoted as $2^2 2^k$. To reduce the run size, especially in a situation with a large number of design factors, we would consider the $\mathbf{D}_{ij}$ to be the fractional factorial design. Consequently, the whole MLSD design $\mathbf{D}$ will also be a fractional factorial design. To enable the investigation of how the design factors affect the response under different level combinations of the slice factors, a suitable two-layer sliced design should have the following two characteristics:

(i) The subdesigns $\mathbf{D}_{ij}, i = 1, ..., l_1; j = 1, ..., l_2$ *should attain a preferable estimation for the effects of design factors.*

(ii) The whole design $\mathbf{D}$ *for slice factors and design factors can estimate the effects of slice factors and the two-way interaction effects between slice factors and design factors.*

To construct the MLSD design with the above properties, we need to differentiate the importance of different effects. Without loss of generality, a factorial effect can be expressed as a word consisting of slice factors (e.g., S_1, S_2) and design factors (e.g., $A, B, C, \ldots$). Let E_I be the set of all factorial effects with words that exclude slice factor S_1 and S_2 (*e.g.* A, AB, ABC), and E_S be the set of all factorial effects with words that include the slice factor S_1 or S_2 (*e.g.* AS_1, ABS_2, $ABCS_1S_2$). Furthermore, we consider the importance of slice factors in two situations: (i) one of the slice factors is more important than the other slice factor (*e.g.*, $S_1 \succ S_2$ or $S_2 \succ S_1$); and (*ii*) two slice factors have equal importance ($S_1 \stackrel{\Delta}{=} S_2$). When $S_1 \succ S_2$, we denote S_1 to be the primary slice factor and S_2 to be the secondary slice factor. Specifically, we propose the following hierarchy principle for the MLSD design.

Principle 1 (Effect hierarchy for MLSD design)**.** The ordering of importance for effects is determined by the following rules:

(i) For the union of E_I and E_S, lower-order effects are more important than higher-order effects.

(ii) For E_I, effects of the same order are equally important.

(iii) For E_S, effects with the primary slice factor are more important than the ones with a secondary slice factor of the same order. If slice factors are the same important, effects of the same order are equally important.

(iv) Any effect in the set E_S is more important than an effect in E_I with the same order.

Next, we will establish some criteria to compare different MLSD designs given the number of slice factors and design factors. To facilitate our discussion, we will consider all slice factors and design factors at two levels. Following Definition 3.3.1 with both S_1 and S_2 having two levels, we consider the fractional factorial design for the whole MLSD design $\mathbf{D}$ as $2^2 \cdot 2^{k-p}$.

Here p represents that the run size of $\mathbf{D}$ is a 2^{-p}th fractional of the full factorial design of slice factors and design factors. For simplicity, we will demonstrate the MLSD design at two levels for both slice factors and design factors as a $2^{2+(k-p)}$ design. In fractional factorial design, the defining relation is essential for constructing word-length patterns (Cheng, 2016; Wu and Hamada, 2011). These patterns allow statisticians to assess the capability of a design to distinguish between the effects of various factors and their interactions. In addition, word-length patterns serve as a measure of a design's resolution, effectively indicating the extent of confounding among factors. Similarly, it is essential to establish criteria that guide the selection of designs for MLSD. To compare different $2^{2+(k-p)}$ MLSD designs, we introduce the concept of *sliced defining relations* for the two slice factors. In the MLSD, slice factors are of important interest, thus their defining relation takes priority over other defining relations. For the design with one slice factor, the sliced defining relation of design is composed of the slice factor S and the group of its aliasing effects. It is formed by multiplying the defining relation of design by the slice factor S (*i.e.* $S = ABCS$ is obtained from $I = ABC$) (Sadeghi et al., 2020). We note that S is aliased with the effects $ABCS$. The word ABC is called the generator of the design and it has length 3. In the two-layer situation, we will have the sliced defining relation for each slice factor. For example, we will have two sliced defining relations $S_1 = ABCS_1$ and $S_2 = ABCS_2$ for a two-layer sliced design. Correspondingly, there are separate word-length patterns for different slice factors. Next, we will define the sliced word-length pattern.

Definition 3.3.2 (Sliced word-length pattern)**.** The sliced defining relations of $\mathbf{D}$ are the aliasing relations involving slice factors $S_i, i = 1, 2$. The sliced word-length pattern is

$$\mathrm{SW} = \{\mathrm{SW}_1, \mathrm{SW}_2\},$$

where SW_i represents the set of word-length counts for the sliced relation of slice factor S_i.

Specifically,

$$\mathrm{SW}_i = \left\{(3^{B_{S_i,3}}), \ldots, ((k+2)^{B_{S_i,k+2}})\right\},$$

where $B_{S_i,j}$ denotes the number of effects with length j for the sliced relation of S_i, and k is the number of design factors.

The number of sliced defining relations equals the number of slice factors. The sliced defining relations can be derived from multiplying the defining contrast subgroup of **D** by the slice factors. For illustration, consider a 2^{2+3-1} MLSD. Assume that the whole design **D** is a 2^{5-1} fractional factorial design that consists of four subdesigns, each of which is a 2^{3-1} fractional factorial design for design factors. We now consider three strategies to construct an MLSD design. $d^{(1)}$: $I = ABC$; $d^{(2)}$: $I = ABCS_1S_2$; $d^{(3)}$: $I = ABCS_2$. For $d^{(1)}$, the sliced defining relations are $S_1 = ABCS_1$ and $S_2 = ABCS_2$. The sliced word-length pattern is $SW = \{(3^0, 4^1, 5^0), (3^0, 4^1, 5^0)\}$. For $d^{(2)}$, the sliced defining relations are $S_1 = ABCS_2$ and $S_2 = ABCS_1$. The sliced word-length pattern is $SW = \{(3^0, 4^1, 5^0), (3^0, 4^1, 5^0)\}$. For $d^{(3)}$, the sliced defining relations are $S_1 = ABCS_1S_2$ and $S_2 = ABC$. The sliced word-length pattern is $SW = \{(3^0, 4^0, 5^1), (3^1, 4^0, 5^0)\}$.

Since the slice factors S_1 and S_2 can be of different importance, we can order the two sliced relations and define the ordered sliced word-length pattern as follows.

Definition 3.3.3. With the ordered slice factors, the ordered sliced word-length pattern can be defined as follows.

a) When slice factor S_1 and S_2 are equally important (i.e., $S_1 \stackrel{\Delta}{=} S_2$), then the ordered sliced word-length pattern grouping is $SW = \{\underbrace{(3^{B_{S_1,3}+B_{S_2,3}}, ..., (k+2)^{B_{S_1,k+2}+B_{S_2,k+2}})}_{\text{Part 1}}\}$.

b) When slice factor S_i is more important than $S_{i'}$ (i.e., $S_1 \succ S_2$ or $S_2 \succ S_1$) then the ordered sliced word-length pattern grouping is

$$SW = \{\underbrace{\mathrm{SW}_i}_{\text{Part 1}}, \underbrace{\mathrm{SW}_{i'}}_{\text{Part 2}}\}.$$

When the slice factors are equally important, the sliced word-length pattern only has one part. For other situations, it will contain several parts, and the number of parts is determined by the number of slice factors. With the defined ordered slice word-length pattern, we can continue to compare two design strategies $d^{(2)}$ with $I = ABCS_1S_2$ and $d^{(3)}$ with $I = ABCS_2$ in the 2^{2+3-1} MLSD. If $S_1 \succ S_2$, the ordered sliced word-length pattern for $d^{(2)}$ is $\{(3^0, 4^1, 5^0), (3^0, 4^1, 5^0)\}$ and that for $d^{(3)}$ is $\{(3^0, 4^0, 5^1), (3^1, 4^0, 5^0)\}$. When $S_1 \stackrel{\Delta}{=} S_2$, the word-length pattern of $d^{(2)}$ is $SW = \{(3^0, 4^2, 5^0)\}$ and the word-length pattern of $d^{(3)}$ is $SW = \{(3^1, 4^0, 5^1)\}$.

Next, we develop proper criteria to compare the MLSD. The maximum resolution (Box and Hunter, 1961) and minimum aberration (Fries and Hunter, 1980) are two popular criteria for selecting the optimal design. We now extend resolution and aberration to accommodate ordered slice factors and propose the ordered sliced resolution as follows:

Criterion 1 (Ordered Sliced Resolution)**.** The ordered sliced resolution of a $2^{2+(k-p)}$ complete design **D** is defined to be the smallest j such that $B_{S_i,j} \geq 1$ or $B_{S_1,j} + B_{S_2,j} \geq 1$ in Part 1 based on Definition 3.3.3.

According to the sliced hierarchy principle, a suitable design is to maximize the ordered resolution. The design with a large resolution can ensure the capability to estimate important slice factors and their interaction with design factors. Here, we use sliced defining relation and sliced word pattern to find a sliced minimum aberration design. The objective is to minimize the aliasing of slice factors with higher-order effects, thereby preserving their estimability.

Criterion 2 (Ordered Sliced Minimum Aberration)**.** Suppose that two $2^{2+(k-p)}$ MLSD $\ddot{d}$ and $\tilde{d}$ are to be compared. Let r be the smallest integer such that $\sum_{S_i} B_{s_i,r}(\ddot{d}) \neq \sum_{S_i} B_{s_i,r}(\tilde{d})$, S_i are all the primary slice factors. Design $\ddot{d}$ is said to have less sliced aberration if $\sum_{S_i} B_{s_i,r}(\ddot{d}) < \sum_{S_i} B_{s_i,r}(\tilde{d})$. If there is no design with less sliced aberration than $\ddot{d}$, then $\ddot{d}$ is called a sliced minimum aberration design.

To construct a sliced minimum aberration design, the length of effects in sliced defining relation plays a key role. Next, we will establish a property to determine the number of effects

containing the secondary slice factor in the defining relation (i.e., defining contract subgroup) for the case of $S_1 \succ S_2$. The secondary slice factor S_2 can help extend the length of the words in the primary sliced defining relation.

Theorem 3.3.4. *If $S_1 \succ S_2$, then for a $2^{2+(k-p)}$ MLSD* $\mathbf{D}$*, the largest number of effects in defining relation that can contain the secondary slice factor (S_2) is 2^{p-1}.*

Based on the above theorem, the number of effects containing the secondary slice factor in the defining relation can be determined. It will help identify the minimum aberration design. For a concrete example, consider an MLSD 2^{2+6-2} with slice factors S_1, S_2 and design factors A, B, C, D, E, F. The minimum aberration scheme of fractional factorial design 2^{6-2} for design factors A, B, C, D, E, F is $I = ABCD = CDEF = ABEF$. If $S_1 \succ S_2$, the best defining relation is $I = ABCDS_2 = CDEFS_2 = ABEF$. We can find the defining relation that contains a secondary slice factor is $2^{2-1} = 2$. By Theorem 3.3.4, there is no other defining relation that can increase the number of effects containing the secondary slice factor. This means the primary sliced defining relation can get the minimum aberration design.

Although we present the above results under the two-layer sliced design, the definition, criterion, and properties for the multi-layer sliced design can be extended and generalized. Similarly, multiple slice factors $S_1, \ldots, S_m$ can be ordered, such as $S_1 \prec S_2 \prec \ldots \prec S_m$, $S_1 \triangleq S_2 \triangleq \ldots \triangleq S_m$, $S_1 \succ S_2 \succ \ldots \succ S_m$, and $S_1 \succ \ldots \succ S_i \triangleq \ldots \triangleq S_j \succ \ldots \succ S_m$. Next, we will discuss several properties of the multi-layer sliced design.

Proposition 3.3.5. *In the multi-layer sliced design $2^m 2^{(k-p)}$ with one or two primary slice factors, the corresponding sliced minimum aberration design can be obtained by not including the primary slice factors in the defining relation.*

The statement in Proposition 3.3.5 aligns with Theorem 3.3.4. In a two-layer sliced design, as described in Theorem 3.3.4, the objective of achieving sliced minimum aberration is to incorporate a larger number of secondary slice factors in the defining relation. This approach aids in obtaining a suitable primary sliced defining relation. Moreover, Proposition 3.3.5 offers

valuable guidance for the exploration of sliced minimum aberration designs in general. The following remarks serve as useful guidance for finding a sliced minimum aberration design.

Remark 3.3.6. *In the multi-layer sliced fractional factorial design* $2^m 2^{k-p}$*, there exists a sliced word-length pattern by* $(3^{B_{S,3}}, ..., (k+2)^{B_{S,k+2}})$ *for each slice factor. The design and its properties are determined by the grouping of the ordered sliced word-length pattern set* $\{(3^{B_{S_1,3}}, ..., (k+2)^{B_{S_1,k+2}}), (3^{B_{S_m,3}}, ..., (k+2)^{B_{S_m,k+2}})\}$.

The estimation capability of the design is determined by the grouping of the ordered sliced word-length pattern. In the MLSD with $m > 1$, the order of the importance of the slice factors affects how to determine the sliced word-length pattern and search for the sliced minimum aberration design.

Remark 3.3.7. *In the multi-layer sliced fractional factorial design* $2^m 2^{k-p} (m > 2, m > p)$ *with slice factors of the same importance, the sliced minimum aberration design can be obtained from the defining relation of design* 2^{m-p} *and design* 2^{k-p} *with minimum aberration.*

As an illustration, consider a $2^5 2^{5-2}$ MLSD with slice factors $S_1, ..., S_5$ and design factors A, B, C, D, E. A minimum aberration design for slice factors is $I = S_1S_2S_4S_5 = S_1S_2S_3 = S_3S_4S_5$. In this case, the aliasing effects in the defining relation can be arranged in a sequence from long to short. Similarly, a minimum aberration design for design factors is $I = ABC = CDE = ABDE$. However, in this defining relation, the sequence of aliasing effects should be ordered from short to long. Combining these two defining relations, we can obtain the optimal defining relation for the MLSD as follows: $I = S_1S_2S_4S_5ABC = S_1S_2S_3CDE = S_3S_4S_5ABDE$.

Remark 3.3.8. *In the multi-layer sliced fractional factorial design* $2^m 2^{k-p}$*, if slice factors are of the same importance, the sliced minimum aberration design corresponds to the design with sliced defining relations where all words contain slice factors.*

Consider a $2^2 2^{6-2}$ MLSD. We will examine two design strategies, denoted as $d^{(1)}$ and $d^{(2)}$. For $d^{(1)}$, the defining relation is given by $I = ABCD = CDEF = ABEF$, and the

sliced defining relations are $S_1 = ABCDS_1 = CDEFS_1 = ABEFS_1$ and $S_2 = ABCDS_2 = CDEFS_2 = ABEFS_2$. The corresponding sliced word-length pattern is $(4^0, 5^6, 6^0)$. On the other hand, for $d^{(2)}$, the defining relation is $I = ABCDS_2 = CDEFS_2 = ABEF$, and the sliced defining relations are $S_1 = ABCDS_1S_2 = CDEFS_1S_2 = ABEF$ and $S_2 = ABCD = CDEF = ABEFS_2$. The sliced word-length pattern associated with $d^{(2)}$ is $(4^3, 5^1, 6^2)$. Considering the scenario where $S_1 \succ S_2$, we have demonstrated that $d^{(2)}$ represents the sliced minimum aberration design. However, as $S_1 \overset{\Delta}{=} S_2$, it becomes evident that $d^{(2)}$ is not the sliced minimum aberration design since there exists a design $d^{(1)}$ with a smaller aberration. In $d^{(2)}$, we observe that there are several words in the sliced defining relation that lack slice factors. Additionally, we can establish that $d^{(1)}$ is a minimum sliced aberration design where all words in its sliced defining relation contain slice factors.

3.4 The Estimation Method

In this section, we introduce an estimation procedure aimed at identifying significant effects. To streamline the model estimations and avoid the complexity of multiple sub-models, we propose the use of conditional effects. This approach is supported by several studies that have explored conditional effects with minimum aberration (Mukerjee et al., 2017; Chang, 2023). In the multi-layer sliced design, we assume that the slice factors $S_1, \ldots, S_m$, and design factors $X_1, \ldots, X_m$ all have two levels "0" and "1", but we use -1 and 1 in data analysis.

Let x_i, for $i = 1, \ldots, k$, and s_j, for $j = 1, \ldots, m$, be the corresponding values of X_i and S_j, respectively. The conditional value of X_i given $S_1 = s_1^*, \ldots, S_m = s_m^*$ can be defined as follows:

$$X_i|_{S_1=s_1^*,\ldots,S_m=s_m^*} = \begin{cases} x_i & \text{if } s_i = s_i^* \text{ for } i = 1, \ldots, m, \\ 0 & \text{otherwise.} \end{cases}$$

Due to the limited experimental run size, we will focus on the estimable main effects and

two-factor interaction effects by considering a linear model as follows:

$$
\begin{aligned}
y = \beta_0 &+ \sum_{i=1}^{k} \sum_{s_1=0}^{1} ... \sum_{s_m=0}^{1} \beta_{X_i s_1 ... s_m} (X_i | S_1 = s_1, ..., S_m = s_m) \\
&+ \sum_{i=1}^{m} \beta_{S_i} S_i + \sum_{\substack{i=1 \\ i!=j}}^{m} \sum_{j=1}^{m} \beta_{S_i S_j} S_i S_j \\
&+ \sum_{i=1}^{k} \sum_{j=1}^{m} \beta_{X_i S_j} X_i S_j \\
&+ \epsilon,
\end{aligned}
$$

where $\boldsymbol{\beta} = (\beta_0, ..., \beta_{X_i s_1 .., s_l}, ..., \beta_{S_1}, ..., \beta_{S_i S_j}, ..., \beta_{X_i S_j}, ...)$ are the coefficients of the linear model and ϵ is the error term with $\epsilon \sim N(0, \sigma^2)$. The set of coefficients can be simply written as $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_q)$, where q is total number of coefficients. The total number of conducted experiments is n. The vector of response can be denoted as $\boldsymbol{y}$ and the corresponding regression matrix can be written as $\boldsymbol{X}$. In this situation where $n < q$, one possible method for analysis and inference is Lenth's method (Lenth, 1989). However, it may not yield accurate parameter estimates for our setting. Some assumptions underlying Lenth's method—such as equal variance among factorial effects and a regular design structure—may not be valid for our experimental designs. According to the sparsity principle in experimental design (Wu and Hamada, 2011), it is assumed that only a small of factorial effects will significantly influence the outcome. Therefore, we consider estimating the parameters using the Lasso method (Tibshirani, 1996) by minimizing the penalized least squares as

$$
L(\boldsymbol{\beta}) = \frac{1}{2} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_1, \tag{3.1}
$$

where $||\boldsymbol{\beta}||_1$ is the l_1 norm of $\boldsymbol{\beta}$ and $\lambda \geq 0$ is a tuning parameter. Here we adopt the AIC for the choice of tuning parameter λ (Shao, 1997). Due to the non-smoothness of the l_1 norm, the objective function in (3.1) is not differentiable at zero with respect to $\boldsymbol{\beta}$, which makes it difficult to obtain the derivative of the parameter $\boldsymbol{\beta}$ at all points. Thus one cannot directly

apply the sandwich formula to obtain the variance of $\hat{\boldsymbol{\beta}}$ for inference. The work of Cilluffo et al. (2020) incorporates the idea from the induced smoothing (Brown and Wang, 2005) to allow estimation and inference on the model coefficients in the Lasso regression. Specifically, the estimating equation from the first "pseudo" derivative of $L(\boldsymbol{\beta})$ is

$$l(\boldsymbol{\beta}) = -\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda\{2\mathbb{1}(\boldsymbol{\beta} > 0) - \mathbf{1}_q\},$$

where $\mathbb{1}$ is the indicator function, i.e., $\mathbb{1}(a > b) = 1$ if $a > b$, and 0 otherwise. Then we can use $l(\boldsymbol{\beta})$ to get the estimated $\hat{\boldsymbol{\beta}}$ and the probability distribution $\boldsymbol{\Sigma}^{-1/2}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \sim f(\boldsymbol{z})$, where $\boldsymbol{\Sigma} = (\sigma_{ij})_{q\times q} = Cov(\hat{\boldsymbol{\beta}})$ is the covariance matrix of $\hat{\boldsymbol{\beta}}$. Here $f(\boldsymbol{z}) = \Pi_{j=1}^{q} f(z_j)$. The Lasso method can result in an estimated coefficient being exactly equal to 0. One can assign two-components mixture distribution to z_j, i.e., $f(z_j) \approx c_j\phi(z_j) + (1 - c_j)\phi_{\tilde{\epsilon}}(z_j)$ with ϕ is the probability density function (pdf) of standard normal distribution and $\phi_{\tilde{\epsilon}}$ is the pdf of a zero-mean normal with very small variance (e.g., $\tilde{\epsilon} = 1e - 6$). The key of the induced smoothing is adding a scaled perturbation of parameters to form the new estimating equation as

$$\begin{aligned}
\tilde{l}(\boldsymbol{\beta}) &= \mathbb{E}_{\boldsymbol{z}}[l(\boldsymbol{\beta} + \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{z})] \\
&= \int l(\boldsymbol{\beta} + \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{z}) f(\boldsymbol{z}) d\boldsymbol{z} \\
&= -\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\eta}(\boldsymbol{\beta}, \boldsymbol{z}; \boldsymbol{c}),
\end{aligned}$$

where the q-dimensional penalty vector $\boldsymbol{\eta}(\boldsymbol{\beta}, \boldsymbol{z}; \boldsymbol{c})$ has the elements $\eta_j = c_j\{2\Phi(\beta_j/\sqrt{\sigma_{jj}}) - 1\} + (1 - c_j)\{2\Phi_{\tilde{\epsilon}}(\beta_j/\sqrt{\sigma_{jj}}) - 1\}$. Here Φ and $\Phi_{\tilde{\epsilon}}$ are the corresponding cumulative density function of standard normal distribution. Since the $\tilde{l}(\boldsymbol{\beta})$ is a smooth function, we can use the sandwich formula to compute the estimation covariance as

$$\hat{\boldsymbol{\Sigma}} = \tilde{l}'(\hat{\boldsymbol{\beta}})^{-1}\boldsymbol{V}\tilde{l}'(\hat{\boldsymbol{\beta}})^{-1},$$

Here $\boldsymbol{V} = Cov(\tilde{l}(\boldsymbol{\beta})) \propto \boldsymbol{X}^T\boldsymbol{X}$. Consequently, we can obtain the approximate distribution of

$\hat{\boldsymbol{\beta}}$ and perform the statistical hypothesis testing on $\boldsymbol{\beta}$. For example of the hypothesis testing $H_0 : \beta_j = 0$, the Wald statistic under H_0 is

$$W_j = \frac{\hat{\beta}_j}{\sqrt{var(\hat{\beta}_j)}} \xrightarrow{d} N(0,1).$$

In the following simulation and application studies, we will use the Wald statistic to test the significance of the coefficients for models. For implementation, we adopt the *islasso* R package for the induced smoothing approach to estimate the effect size and determine statistical significance (Cilluffo et al., 2020). Insignificant effects will be shrunk to zero.

3.5 Simulation

This section presents the results of several simulation studies that demonstrate the accuracy and robustness of the MLSD model. We extensively tested our model under two-layer and three-layer sliced designs under different noise levels.

3.5.1 Two-layer sliced design

We first consider a two-layer sliced design 2^{2+3-1}. Assume that the slice factor S_1 and S_2 are of the same importance. Based on the sliced minimum aberration, we can obtain two best design schemes $d^{(1)}$: $I = ABC$ and $d^{(2)}$: $I = ABCS_1S_2$. The ordered sliced word-length pattern grouping for $d^{(1)}$ and $d^{(2)}$ are $SW_{d^{(1)}} = \{(3^0, 4^2, 5^0)\}$ and $SW_{d^{(2)}} = \{(3^0, 4^2, 5^0)\}$. It means that these two designs both have the ordered sliced resolution of IV. Based on the factorial strategies, the following two designs are obtained with 16 observations. (Hung et al., 2009) introduces the concepts of branching and nested factors to describe situations where certain factors only exist within the levels of other factors. These are referred to as nested factors, with the factor containing other factors being called a branching factor. In this context, we consider slice factors as branching factors and design factors as nested factors. The unique aspect here is that all levels of the nested factor can exist under the branching factor. An optimal branching Latin hypercube design (BLHD) is generated by maximizing the

minimum inter-site distance. This optimal BLHD is used as the benchmark for comparison. Here, 0 and 1 represent two levels of factors.

Table 3.2: The Design Points for Different Experiments

$d^{(1)}$					$d^{(2)}$					BLHD				
S_1	S_2	A	B	C	S_1	S_2	A	B	C	S_1	S_2	A	B	C
0	0	0	0	0	0	0	0	0	0	1	1	0	0	1
0	0	1	1	0	0	0	0	1	1	0	0	0	1	1
0	0	1	0	1	0	0	1	0	1	0	0	0	0	1
0	0	0	1	1	0	0	1	1	0	1	1	1	1	0
1	0	0	0	0	1	0	0	0	1	0	1	1	0	0
1	0	1	1	0	1	0	0	1	0	0	1	1	1	1
1	0	1	0	1	1	0	1	0	0	1	0	1	0	0
1	0	0	1	1	1	0	1	1	1	1	0	1	1	1
0	1	0	0	0	0	1	0	0	1	0	1	0	0	0
0	1	1	1	0	0	1	0	1	0	1	0	0	1	0
0	1	1	0	1	0	1	1	0	0	1	1	0	1	1
0	1	0	1	1	0	1	1	1	1	1	0	0	0	0
1	1	0	0	0	1	1	0	0	0	0	1	0	1	0
1	1	1	1	0	1	1	0	1	1	0	0	1	1	0
1	1	1	0	1	1	1	1	0	1	0	0	1	0	1
1	1	0	1	1	1	1	1	1	0	1	1	1	0	1

Given a design matrix, we consider the underlying model for the response as follows.

$$
\begin{aligned}
y = \beta_0 &+ \sum_{i=0}^{1}\sum_{j=0}^{1} \beta_{A_{ij}}(A|S_1 = i, S_2 = j) \\
&+ \sum_{i=0}^{1}\sum_{j=0}^{1} \beta_{B_{ij}}(B|S_1 = i, S_2 = j) \\
&+ \sum_{i=0}^{1}\sum_{j=0}^{1} \beta_{C_{ij}}(C|S_1 = i, S_2 = j) \\
&+ \sum_{i=1}^{2} \beta_{S_i} S_i + \beta_{S_1S_2} S_1 S_2 \\
&+ \sum_{i=1}^{2} \beta_{AS_i} A S_i + \sum_{i=1}^{2} \beta_{BS_i} B S_i + \sum_{i=1}^{2} \beta_{CS_i} C S_i \\
&+ \epsilon,
\end{aligned}
\tag{3.2}
$$

where ϵ is the error term following $N(0, \sigma^2)$. Here, take $\beta_0 = 5, \beta_{A_{00}} = 3, \beta_{B_{01}} = 4, \beta_{C_{10}} = 10, \beta_{S_1} = 10, \beta_{S_2} = -3, \beta_{CS_1} = 5, \beta_{S_1S_2} = 5$, the other coefficients as 0. Two scenarios of the error terms are considered: $\epsilon \sim N(0, 1)$ and $\epsilon \sim N(0, 0.5)$. When the responses are generated, the proposed analysis method in Section 3.4 is used to analyze the data.

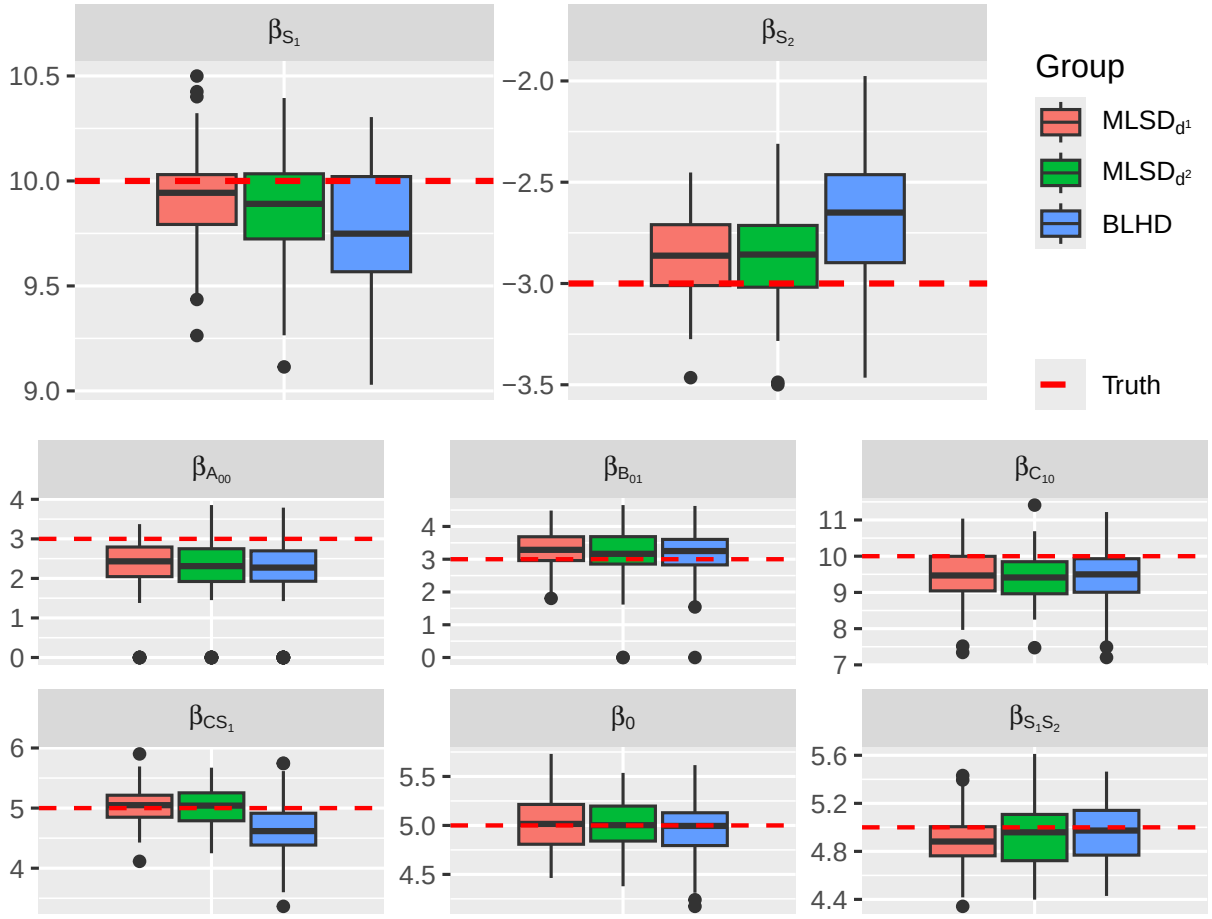

Figure 3.2: The comparison of simulation results for three different designs

Figure 3.2 presents the results of analyses conducted across different experiments with 100 replications. The performance of $d^{(1)}$ and $d^{(2)}$ surpasses that of BLHD in terms of accuracy and precision for the coefficients of S_1 and S_2. The means of the estimated coefficients for the proposed designs are closer to the true values, and the corresponding boxplots are noticeably narrower. For other coefficients, the three designs exhibit comparable performance, with the exception of β_{CS_1}. It is observed that the competitor's estimates for β_{CS_1} deviate more from the truth compared to those from the proposed MLSD method. Overall, the significant coefficients

Table 3.3: The Comparison of Estimated Coefficients in Two-Layers Platform Design

		$\epsilon \sim N(0,1)$			$\epsilon \sim N(0,0.5)$		
	Truth	$d^{(1)}$	$d^{(2)}$	BLHD	$d^{(1)}$	$d^{(2)}$	BLHD
β_0	5	5.01 (0.28)	4.99 (0.27)	4.96 (0.27)	4.98 (0.12)	5.00 (0.11)	4.95 (0.13)
$\beta_{A_{00}}$	3	2.29 (0.77)	2.20 (0.86)	2.08(0.99)	2.27 (0.54)	2.42 (0.30)	2.38 (0.30)
$\beta_{A_{01}}$	0	0 (0)	0(0)	0 (0)	0 (0)	0 (0)	0 (0)
$\beta_{A_{10}}$	0	0 (0)	0(0)	0.01 (0.12)	0 (0)	0 (0)	0 (0)
$\beta_{A_{11}}$	0	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
$\beta_{B_{00}}$	0	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
$\beta_{B_{01}}$	3	3.30 (0.55)	3.25 (0.64)	3.19 (0.67)	3.32 (0.30)	3.31 (0.34)	3.31 (0.31)
$\beta_{B_{10}}$	0	0 (0)	0(0)	0.04 (0.24)	0 (0)	0 (0)	0 (0)
$\beta_{B_{11}}$	0	0 (0)	0 (0)	-0.01 (0.13)	0 (0)	0 (0)	0 (0)
$\beta_{C_{00}}$	0	0 (0)	0 (0)	0(0)	0 (0)	0 (0)	0(0)
$\beta_{C_{01}}$	0	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
$\beta_{C_{10}}$	10	9.46 (0.71)	9.46 (0.72)	9.45(0.08)	9.51 (0.38)	9.50(0.34)	9.56(0.49)
$\beta_{C_{11}}$	0	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
β_{S_1}	10	9.92 (0.23)	9.92 (0.23)	9.76 (0.29)	9.89 (0.11)	9.88 (0.12)	9.78 (0.17)
β_{S_2}	-3	-2.87 (0.20)	-2.89 (0.20)	-2.67 (0.32)	-2.90 (0.12)	-2.90 (0.13)	-2.65 (0.15)
β_{AS_1}	0	-0.01 (0.11)	-0.01 (0.10)	0 (0)	0 (0)	0 (0)	0 (0)
β_{AS_2}	0	0 (0)	-0.01 (0.06)	-0.06 (0.22)	0(0)	0 (0)	0 (0)
β_{BS_1}	0	0 (0.06)	-0.02 (0.12)	0 (0)	0 (0)	0 (0)	0 (0)
β_{BS_2}	0	0.01 (0.07)	0.01 (0.07)	0 (0)	0 (0)	0 (0)	0 (0)
β_{CS_1}	5	5.03 (0.29)	5.03 (0.29)	4.65 (0.47)	4.99 (0.15)	5.02 (0.16)	4.62 (0.28)
β_{CS_2}	0	0 (0)	-0.01 (0.09)	0 (0)	0 (0)	0 (0)	0 (0)
$\beta_{S_1S_2}$	5	4.89 (0.23)	4.89 (0.23)	4.94 (0.25)	4.87 (0.12)	4.89 (0.12)	4.97 (0.13)

can be estimated accurately, as demonstrated by the coverage of the true values within the boxplots. It is interesting to note that the estimation capabilities of significant slice factor effects are more accurate than design factors, as indicated by their small variance in Table 3.3. In addition, the proposed method also works better when the noise level is small. One can also obtain more accurate estimations for slice factors under low noise levels. In Table 3.3, the insignificant coefficients can be accurately evaluated under small noise conditions. However, under large noise conditions, there is a risk of incorrectly identifying insignificant factors. For example, the true value of $\beta_{A_{01}}$ is 0, but the mean estimated by the BLHD method is small, albeit non-zero. This issue is also present in the other two methods. Nonetheless, although the estimated values are not exactly zero, they are very small. Moreover, we calculate the signal-to-noise ratios (SNR) using formula $Var(\boldsymbol{X}\hat{\boldsymbol{\beta}})/Var(\boldsymbol{Y})$ to access the meaningful information captured in the data. In both levels of noise, the SNR is approximately 95%, indicating 95%

of the variability in the data can be attributed to the true signal, while the remaining 5% is due to noise. It suggests that the results of the analysis are likely to be reliable with findings supported by the data.

3.5.2 Three-layer sliced design

This section examines the performance of the proposed methods under the three-layer situations. Consider a three-layer sliced design 2^{3+3-1} with two design schemes $d^{(1)}$: $I = ABC$ and $d^{(2)}$: $I = ABCS_1S_2S_3$. The ordered sliced word-length pattern grouping for $d^{(1)}$ and $d^{(2)}$ are $SW_{d^{(1)}} = \{(3^0, 4^3, 5^0)\}$ and $SW_{d^{(2)}} = \{(3^0, 4^0, 5^3)\}$, respectively. Note that $d^{(1)}$ is ordered sliced resolution IV and $d^{(2)}$ is ordered sliced resolution V. Thus, according to the sliced minimum aberration criterion, the design $d^{(1)}$ is better than the design $d^{(2)}$. For comparison, we also take an optimal BLHD generated by maximizing the minimum inter-site distance as a benchmark.

Based on the constructed design, we consider the following model to generate response y as

$$\begin{aligned} y = \beta_0 &+ \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1}\beta_{A_{ijk}}(A|S_1 = i, S_2 = j, S_3 = k) \\ &+ \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1}\beta_{B_{ijk}}(B|S_1 = i, S_2 = j, S_3 = k) \\ &+ \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1}\beta_{C_{ijk}}(C|S_1 = i, S_2 = j, S_3 = k) \\ &+ \sum_{i=1}^{3}\beta_{S_i}S_i + \sum_{\substack{i=1 \\ i\neq j}}^{3}\sum_{j=1}^{3}\beta_{S_iS_j}S_iS_j \\ &+ \sum_{i=1}^{3}\beta_{AS_i}AS_i + \sum_{i=1}^{3}\beta_{BS_i}BS_i + \sum_{i=1}^{3}\beta_{CS_i}CS_i \\ &+ \epsilon, \end{aligned}$$

where ϵ is the error term following $N(0, \sigma^2)$. Here, we take $\beta_0 = 5$, $\beta_{A_{000}} = 3$, $\beta_{A_{011}} = 2$,

$\beta_{B_{010}} = 4$, $\beta_{B_{001}} = 2$, $\beta_{C_{100}} = 10$, $\beta_{C_{101}} = 3$, $\beta_{S_1} = 10$, $\beta_{S_2} = -3$, $\beta_{S_3} = 5$, $\beta_{CS_1} = 5$, $\beta_{S_1S_2} = 5$ and the other coefficients are 0's. We also consider two scenarios of the noise levels as $\epsilon \sim N(0, 1)$ and $\epsilon \sim N(0, 0.5)$.

Figure 3.3 displays the simulation outcomes based on 100 replications. This figure demonstrates that the two MLSDs accurately estimate the true parameters, as evidenced by the coverage of the true values by all boxplots. Regarding the BLHD, some simulations suggest that $\beta_{A_{011}}$ and $\beta_{B_{001}}$ are statistically insignificant. Although BLHD effectively estimates the slice factors with reasonable accuracy and precision, it exhibits lesser precision and accuracy compared to MLSDs. This is indicated by the means of deviating more from the red line and broader boxplots. This discrepancy is attributed to the BLHD's design, which is more suited for factors with continuous values. When applied to a discrete design, the optimization algorithms face challenges in achieving a global optimum due to the presence of ties, leading to inefficiencies in the algorithm thus, there is no guarantee that the BLHD identified is the optimal one. Table 3.4 indicates the use of $d^{(2)}$ provides slightly better estimates for slice factors than the use of $d^{(1)}$. That is the estimates of β_{S1}, β_{S2}, β_{S3} of $d^{(2)}$ move the mean slightly closer to true values of parameters than that of $d^{(1)}$. Overall, we can find the performance of BLHD is less competitive to MLSD when the layers of design are growing. For $d^{(2)}$, we calculate the SNR is around 98% which is much better than that of two two-layer slice designs. Here, we conduct more designs to explore the variability of responses.

3.6 Case Study

In this section, we use the proposed MLSD for enhancing the AI assurance (Batarseh et al., 2021; Batarseh and Freeman, 2022; Batarseh et al., 2023). In AI assurance, it is important to understand the hyperparameter effects under different models and optimization strategies when training a deep learning model. The combinatorial complexity of hyper-parameters is a common challenge for machine learning practitioners and researchers since it often requires significant computational resources to find a good combination of hyper-parameters for a

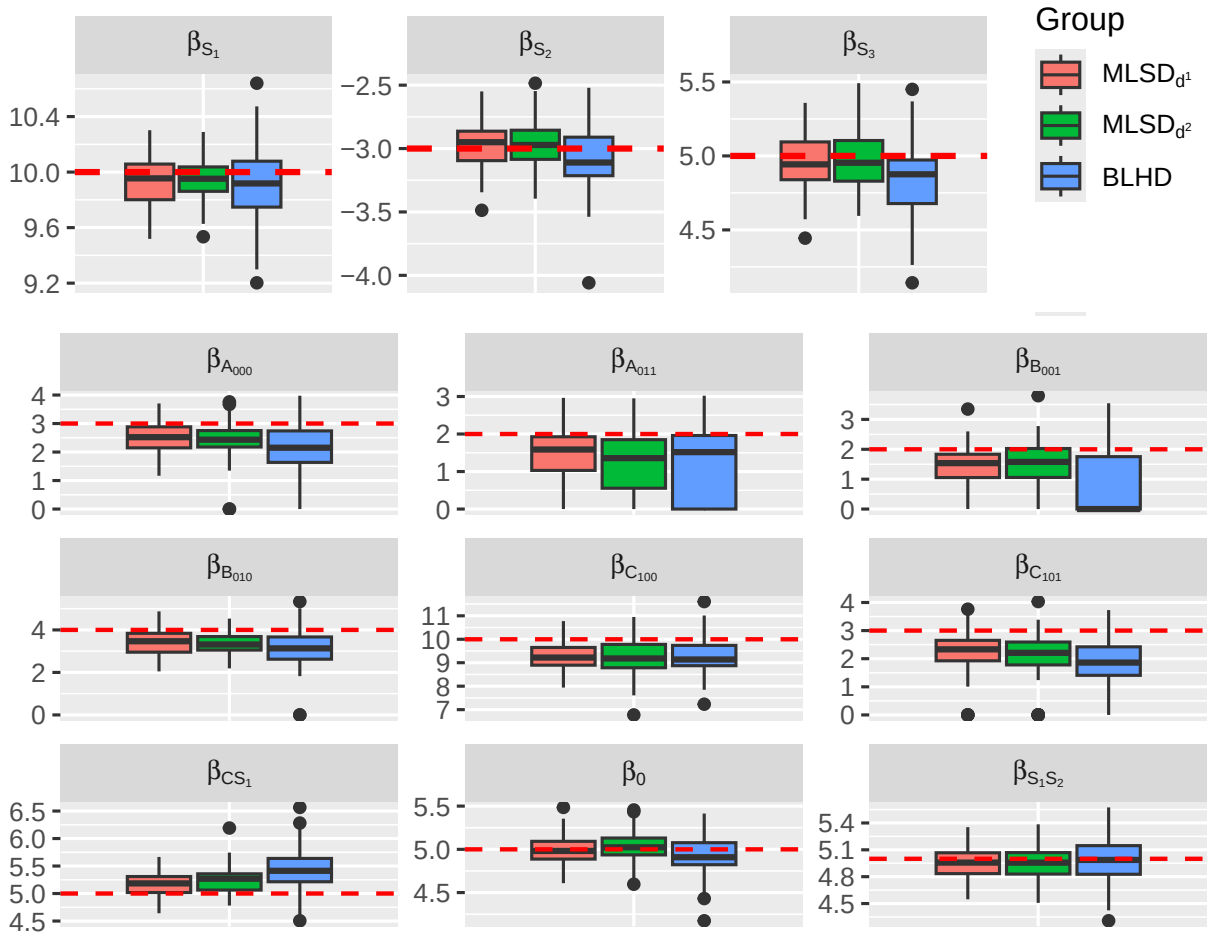

Figure 3.3: The comparison of simulation results for three different designs under three-layer platform design

Table 3.4: The Comparison of Estimated Coefficients in Three-Layers Platform Design

		$\epsilon \sim N(0,1)$			$\epsilon \sim N(0,0.5)$		
	Truth	$d^{(1)}$	$d^{(2)}$	BLHD	$d^{(1)}$	$d^{(2)}$	BLHD
β_0	5	5 (0.18)	5.03 (0.18)	4.93 (0.21)	5.03 (0.09)	5.03 (0.09)	4.96 (0.09)
$\beta_{A_{000}}$	3	2.50 (0.57)	2.41 (0.60)	2.03 (1.02)	2.55 (0.30)	2.58 (0.32)	2.27 (0.40)
$\beta_{A_{011}}$	2	1.39 (0.75)	1.24 (0.85)	1.33 (0.94)	1.54 (0.33)	1.55 (0.27)	1.56 (0.29)
$\beta_{B_{010}}$	4	3.43 (0.62)	3.33 (0.47)	3.12 (0.84)	3.44 (0.61)	3.57 (0.53)	3.14 (0.41)
$\beta_{B_{001}}$	2	1.34 (0.80)	1.46 (0.82)	0.85 (0.98)	1.54 (0.30)	1.59 (0.30)	1.37 (0.57)
$\beta_{C_{100}}$	10	9.25 (0.58)	9.26 (0.70)	9.25 (0.72)	9.41 (0.34)	9.37 (0.36)	9.24 (0.40)
$\beta_{C_{101}}$	3	2.27 (0.73)	2.15 (0.76)	1.71 (0.99)	2.45 (0.35)	2.38 (0.31)	2.08 (0.41)
β_{S_1}	10	9.93 (0.18)	9.95 (0.15)	9.90 (0.27)	9.96 (0.09)	9.96 (0.10)	9.88 (0.12)
β_{S_2}	-3	-2.97 (0.19)	-2.98 (0.18)	-3.07 (0.25)	-2.95 (0.08)	-2.97 (0.11)	-3.05 (0.14)
β_{S_3}	5	4.95 (0.18)	4.97 (0.18)	4.83 (0.25)	4.94 (0.10)	4.96 (0.09)	4.84 (0.12)
β_{CS_1}	5	5.16 (0.21)	5.23 (0.24)	5.42 (0.37)	5.09 (0.12)	5.11 (0.12)	5.38 (0.17)
$\beta_{S_1S_2}$	5	4.96 (0.16)	4.96 (0.18)	4.98 (0.24)	4.95 (0.08)	4.94 (0.08)	4.98 (0.10)

specific task. Moreover, a good combination of hyper-parameters can be different for deep learning methods under different models and optimization strategies.

For investigating the effects of hyper-parameters in the deep learning model, practitioners typically rely on their own experience to determine the values of each parameter. They might change one element at a time while keeping the others constant, similar to a one-factor-at-a-time analysis (Wu and Hamada, 2011). Based on their results and intuition, they then decide the optimal combination of settings. Such a procedure cannot identify the best combinations of hyper-parameters. Alternatively, practitioners could implement all possible combinations to find the best one, but such an exhaustive search requires costly computational resources. Using the proposed MLSD and analysis method, one can provide a statistical tool to facilitate a more efficient exploration of the hyper-parameter, model, and optimization space for AI assurance.

Specifically, we consider the deep learning algorithm of classification for a popular dataset named MNIST (Deng, 2012), which is composed of handwritten digits formatted as 28×28 pixel monochrome images. Figure 3.4 illustrates some image examples. The objective of the deep learning algorithm is to accurately classify the images and achieve good prediction accuracy. There are several key factors, including model architecture, optimization strategies, batch sizes, and epoch and learning rates, for consideration in the deep learning algorithm. For model architecture, two popular models used in image classification problems are the Convolutional Neural Network (CNN) and Multi-layer Perceptron (MLP). In general, CNN is the preferred choice for image classification tasks over MLP due to its ability to capture spatial hierarchies and local patterns in images, leading to better performance. However, MLP typically requires fewer computational resources compared to CNN. If computational resources are limited, an MLP might be a more feasible choice. In addition, MLP is generally simpler and faster to train than CNN, especially for small datasets or simple image classification tasks where the spatial hierarchies are not as critical. Therefore, when the performance of MLP is comparable to that of CNN without significant sacrifices in accuracy, MLP becomes an attractive option. Optimizers are the tools we use to estimate the parameters by minimizing a

loss function. Some well-known optimizers are adaptive gradients, such as AdamW (Loshchilov and Hutter, 2017) and Adagrad (Lydia and Francis, 2019). The learning rate can control how much model weights should be updated. As deep learning algorithms often require significant computational resources, it may not be feasible to optimize a model using all available data at once. As a result, the data is split into smaller subsets or batches, and the model estimation is iterated over each batch. Thus the number of data points in each batch is known as the batch size, and the number of times the algorithm runs on the entire training dataset is known as the number of epochs.

Figure 3.4: Examples of MNIST Dataset

To investigate the effects of the above key factors in the deep learning algorithm, we cast this parameter selection problem as a two-layer sliced design. The model and optimizer can be considered as the slice factors, which are crucial components in deep learning. The remaining factors, including the number of epochs, batch size, and learning rate, can be considered design factors. While some design factors are continuous, users of deep learning algorithms typically choose discrete levels, for example, 32, 64, or 128 for the batch size. By using the proposed 2^{2+3-1} two-layer sliced design, we can effectively investigate the effects of these factors on the performance of the deep learning algorithm. Table 3.5 lists the factors and their respective levels in this study.

Table 3.5: Five Factors and Their Levels

	Slice Factors		Design Factors		
	Model (S_1)	Optimizer (S_2)	Epoch (A)	Batch Size (B)	Learning Rate (C)
Level 0	MLP	AdamW	20	32	10^{-3}
Level 1	CNN	Adagrad	50	64	10^{-4}

In this two-layer sliced design, we consider two optimal designs with 16 runs as described

in Section 3.5.1. For each design, we take the prediction accuracy as our response. The corresponding results for $d^{(1)}$, $d^{(2)}$, and the BLHD are reported in Table 3.6. After collecting all the responses (y), we conduct parameter estimations and summarize the significant parameters for all strategies in Table 3.7. All other parameters not reported in the table are zero across all three designs.

Table 3.6: The Classification Accuracy (%) for Each Design Strategy in Comparison

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
D_1	95.36	94.81	93.15	91.23	98.54	98.34	98.44	96.72	88.05	89.42	78.28	69.56	94.25	95.92	77.97	62.45
D_2	94.96	90.42	92.99	94.76	74.6	88.51	89.52	74.24	97.23	98.24	98.79	97.98	94.25	66.78	78.58	95.31
BLHD	66.83	90.73	91.68	95.46	89.42	75.45	98.69	97.78	88.86	98.14	63.36	98.44	88.51	94.96	92.94	78.53

Table 3.7 compares results for different designs. The coefficients estimation of I, β_{S_1}, β_{S_2}, $\beta_{S_1S_2}$, $\beta_{C_{00}}$, and $\beta_{C_{01}}$ in $d^{(1)}$ and $d^{(2)}$ are very close. A plausible explanation is that the estimates of $d^{(1)}$ and $d^{(2)}$ are somehow different due to a large random noise. Thus, there exists an estimation gap between $d^{(1)}$ and $d^{(2)}$. Among all the coefficients, the signs of each detected effect in the two optimal strategies are consistent. However, for the BLHD, the magnitude of S_1 and S_1S_2 are different from the coefficients of $d^{(1)}$ and $d^{(2)}$. Moreover, the BLHD design fails to identify the effects of $\beta_{C_{00}}$, indicating a potential limitation in its detection capability.

Based on the result for $d^{(1)}$, the average accuracy of the deep learning algorithm is 88.90%. The estimate of β_{S_1} is 1.41, which means that changing the model from MLP to CNN can improve the accuracy by 1.41%. This is consistent with the general observation that CNN works better in computer vision tasks than MLP. In terms of β_{S_2}, we can infer that the optimizer AdamW performs better than the Adagrad and it can enhance the accuracy by 6.9%. The estimate of $\beta_{S_1S_2}$ shows that the model and optimizer interact with each other and the interaction harms the prediction accuracy. The estimate of $\beta_{A_{00}}$ presents improving epochs from 20 to 50 can increase 2.48% accuracy when we use the MLP model and AdamW optimizer.

The estimate of $\beta_{A_{11}}$ implies that improving epochs from 20 to 50 increases 4.26% accuracy when we use the CNN model and Adagrad optimizer. Overall, the main effect of epoch (i.e.,

design factor A) in every slice factor plays a positive impact on accuracy. This is because the model does not converge at 20 epochs and it needs more epochs to improve the model's performance. The effects of batch size (i.e., design factor B) always hurt the prediction. This can be explained by using a larger batch size means fewer model updates in each model epoch. In addition, the significant effect of the learning rate (i.e., design factor C) shows that prediction accuracy can benefit from the bigger learning rate since it can accelerate the model learning process. The results of $d^{(2)}$ can be interpreted similarly.

Table 3.7: Parameters Estimates in Two-Layer Sliced Design

	β_0	$\beta_{A_{01}}$	$\beta_{A_{11}}$	$\beta_{B_{01}}$	$\beta_{B_{11}}$	$\beta_{C_{00}}$	$\beta_{C_{01}}$	$\beta_{C_{11}}$	β_{S_1}	β_{S_2}	$\beta_{S_1S_2}$
D_1	88.90	2.48	4.26	-1.80	-3.43	-1.41	-7.37	-12.40	1.41	-6.90	-0.75
D_2	89.19	0	2.87	0	-2.27	-1.24	-6.95	-10.7	1.61	-6.39	-0.61
BLHD	88.59	0	6.02	0	-1.09	0	-6.19	-9.38	0.80	-6.94	-1.66

From the analysis, one can obtain several insightful observations, guiding the selection of factors for model performance. Notably, the first slice factor indicates a positive sign, suggesting that CNN models outperform MLP models in the classification of handwritten digital images, enhancing accuracy by approximately 1.5%. If computational resources are limited and some decrease in accuracy is acceptable, MLP models may be considered a viable alternative. Moreover, the choice of optimization strategy plays a crucial role in performance, with the AdamW optimizer boosting accuracy by about 6.5%, as demonstrated by the second slice factor. Based on the insights from the slice factors, coupled with considerations such as budget and computation time, researchers can make informed decisions regarding the most suitable model and optimization method. Furthermore, under specific combinations of slice factors, experimenters can identify optimal hyperparameter settings. In a short summary, the proposed approach is useful to facilitate simultaneous model selection, optimization strategies choosing, and hyperparameter tuning within a single fractional experimental design, enabling efficient and effective exploration of model configurations.

3.7 Discussion

We proposed a multi-layer sliced design to quantify the effects of slice factors and design factors to account for design factors with different effects under different level combinations of slice factors. We also developed a criterion for finding the minimum aberration design in this new situation. Moreover, we developed an effective analysis method to estimate the effects of these factors and test their significance. It enhances the reduction of estimation bias through the combination of sub-model estimations. The application of the proposed design to AI assurance is particularly important in practice as it can effectively detect the effects of hyper-parameters affecting the performance of AI algorithms.

Our proposed method can also be adapted for online experiments and other AI applications. In online experiments, slice factors can be different mediums where an experiment is conducted, and they can significantly influence the results of the study. For example, these factors can include device types such as laptops and cellphones, web browsers such as Google Chrome and Safari, and e-commerce platforms like Amazon and eBay. The importance of a slice factor can vary depending on the context and objectives of the study. Here is a scenario where one slice factor is more important than another. Consider conducting a study to understand online shopping behavior, with a focus on comparing user interactions and purchase decisions on laptops versus cellphones. In this experiment, it involves two slice factors: device types (cellphone and laptop) and web browsers (Google Chrome and Firefox). The device type can be more important than the web browser for several reasons. First, users might use cellphones for quick purchases or while on the go whereas desktop shopping might be associated with more extensive research and comparison. Understanding these differences is crucial for the study's objectives. Second, cellphones might offer features like push notifications and personalized recommendations that can increase user engagement and conversion rates. Analyzing the differences in shopping behavior between laptops and cellphones can provide valuable insights for e-commerce businesses. In the future, it will be interesting to apply the proposed MLSD to online shopping experiments, taking into account slice factors

of varying importance.

The proposed estimation method can estimate the effects of interest simultaneously. An interesting finding is that in the case of only one slice factor, the conditional value X_1 given S_1 can be viewed from the angle of conditional main effects (Mak and Wu, 2019). Note that there is a close connection between interaction effects and conditional main effects. It is important to remark that we only consider the conditional main effects on the slice factors, which differs from the conventional conditional main effect model. In addition, it is important to clarify the differences between the slice factor and the blocking factor in experimental design. While the design strategy for constructing these two types of designs may have some similarities, their underlying mechanisms are distinct. In the case of the blocking factor, it is essential to control for the variation it introduces. While for the slice factor, it is crucial to accurately detect its effect, which is often the primary focus. Additionally, when constructing the design, priority should be given for accurate estimation of the slice factor's effect.

We would like to remark that exploring the potential of experimental design in modern applications is a promising direction and is gaining increasing attention. Some recent works have shown that experimental design can help improve the performance of AI. Lim et al. (2020) employed experimental design to enhance the efficiency of AI-driven optimization for complex disease treatments using a minimal number of experiments. Lian et al. (2021, 2022) explored the design of experiments to improve the robustness and assurance of AI algorithms. On the other hand, other researchers study using AI to improve the experimental design. Kleinegesse and Gutmann (2020, 2021) and Guo et al. (2021) used contrastive variational mutual information estimators to better find the optimal design. Ren et al. (2021) considered the smart device to collect the most informative data by optimizing the knowledge graph of the customer. Our study presents useful results demonstrating that the application of statistical experimental design can enhance AI research. In the future, one can consider experimenting with the Amazon Mechanical Turk platform to gather data for a multi-layer sliced design and analysis. Additionally, we identify several directions for future exploration. Firstly, it would be valuable to examine sliced design under non-normal response scenarios, extending

to both design construction and analysis methodologies. Secondly, researchers could consider linking the slice aberration criterion to estimation capability with statistical foundation and utilizing alternative design criteria that incorporate prior information about design factors to construct experimental designs, such as the I-WLP criterion Li et al. (2015, 2019). Thirdly, it is interesting to explore the minimum aberration design under the constructed estimation model which can be inspired by the work of Mukerjee et al. (2017); Chang (2023).

3.8 Appendix

Proof for Theorem 3.3.4

Proof. In order to construct a $2^{2+(k-p)}$ MLSD, we need to use p independent generators, $G_1, ..., G_p$, then the total effects in the defining contrast subgroup is $2^p - 1$. The defining relation can be written as

$$I = \underbrace{G_1 = \ldots = G_p}_{\text{Independent generators}} = G_1G_2 = \ldots = G_1G_p = \ldots = G_1G_2 \ldots G_p.$$

Let n ($n \leq p$) denote an arbitrary number of generators containing the secondary slice factors. If $n = 0$, the number of effects containing the secondary slice factor is 0. If $n = 1$, suppose that G_1 is the only one generator with a secondary slice factor among the first p generators, then the number of effects containing a secondary slice factor can be computed as

$$1 + \binom{p-1}{1} + \binom{p-1}{2} + \ldots + \binom{p-1}{p-1} = 2^{p-1}.$$

If $n = 2$, we can regard $G_1, G_1G_2, G_3, \ldots, G_p$ as the p independent generators. The number of effects containing the secondary slice factor is 2^{p-1}. Similarly, for $n = 3, ..., p$, the number of effects containing secondary slice factor is 2^{p-1}. □

Proof for Proposition 3.3.5

Proof. Given we have one primary slice factor, putting the primary slice factor into the defining relation will lead to a suboptimal design, because the primary slice factor will be offset in the primary sliced defining relation. This will shorten the length of the aliasing effects in the primary sliced defining relation. Thus, we should exclude the primary slice factor from the defining relation.

We discuss another situation with two primary slice factors S_1 and S_2. For example, in a $2^m 2^{n-2}$ MLSD, an arbitrary defining relation is $I = G_1 = G_2 = G_1G_2$, where G_1 and G_2 are two independent generators without primary slice factors. We consider the achievable methods that can help generate a better sliced defining relation for S_1 by putting primary slice factors to it. Obviously, directly putting S_1 to the defining relation does not work. A possible way is to put S_2 to extend the length of the words. But this method will make the sliced defining relation of S_2 worse. The design would be less effective due to its weakest component limiting overall performance. Another solution is to put S_1S_2 to the defining relation. Without loss of generality, the defining relation can be $I = G_1S_1S_2 = G_2 = G_1G_2S_1S_2$. In this situation, the sliced word-length pattern for S_1 would be the same as $I = G_1 = G_2 = G_1G_2$. Similarly, we know no way can help obtain a better sliced defining relation for S_2. Based on Criterion 2, the primary factors can be excluded from the defining relation to obtaining a sliced minimum aberration design. □

Bibliography

R. Bardenet, M. Brendel, B. Kégl, and M. Sebag. Collaborative hyperparameter tuning. In *International conference on machine learning*, pages 199–207. PMLR, 2013.

F. A. Batarseh and L. Freeman. *AI Assurance: Towards Trustworthy, Explainable, Safe, and Ethical AI*. Elsevier, 2022.

F. A. Batarseh, L. Freeman, and C.-H. Huang. A survey on artificial intelligence assurance. *Journal of Big Data*, 8(1):60, 2021.

F. A. Batarseh, J. Chandrasekaran, and L. J. Freeman. An introduction to ai assurance. In *AI Assurance*, pages 3–12. Elsevier, 2023.

J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

G. E. Box and J. S. Hunter. The 2 k—p fractional factorial designs. *Technometrics*, 3(3): 311–351, 1961.

G. E. Box, W. H. Hunter, S. Hunter, et al. *Statistics for experimenters*, volume 664. John Wiley and sons New York, 1978.

B. M. Brown and Y.-G. Wang. Standard errors and covariance matrices for smoothed rank estimators. *Biometrika*, 92(1):149–158, 2005.

M.-C. Chang. A unified framework for minimum aberration. *Statistica Sinica*, 32(1):251–69, 2022.

M.-C. Chang. Bayesian-inspired minimum contamination designs under a double-pair conditional effect model. *Statistical Theory and Related Fields*, pages 1–14, 2023.

C.-S. Cheng. *Theory of Factorial Design*. Chapman and Hall/CRC, 2016.

C.-S. Cheng, D. M. Steinberg, and D. X. Sun. Minimum aberration and model robustness for two-level fractional factorial designs. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(1):85–93, 1999.

G. Cilluffo, G. Sottile, S. La Grutta, and V. M. Muggeo. The induced smoothed lasso: A practical framework for hypothesis testing in high dimensional regression. *Statistical methods in medical research*, 29(3):765–777, 2020.

L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

S. Dougherty, J. R. Simpson, R. R. Hill, J. J. Pignatiello, and E. D. White. Effect of heredity and sparsity on second-order screening design performance. *Quality and Reliability Engineering International*, 31(3):355–368, 2015.

K. Eggensperger, M. Feurer, F. Hutter, J. Bergstra, J. Snoek, H. Hoos, K. Leyton-Brown, et al. Towards an empirical foundation for assessing bayesian optimization of hyperparameters. In *NIPS workshop on Bayesian Optimization in Theory and Practice*, volume 10, 2013.

M. Feurer, J. Springenberg, and F. Hutter. Initializing bayesian hyperparameter optimization via meta-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

R. A. Fisher. Statistical methods for research workers. In *Breakthroughs in statistics: Methodology and distribution*, pages 66–70. Springer, 1970.

A. Fries and W. G. Hunter. Minimum aberration 2 k–p designs. *Technometrics*, 22(4):601–608, 1980.

R. B. Gramacy, A. Sauer, and N. Wycoff. Triangulation candidates for bayesian optimization. *arXiv preprint arXiv:2112.07457*, 2021.

R. F. Gunst and R. L. Mason. Fractional factorial design. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(2):234–244, 2009.

Q. Guo, J. Chen, D. Wang, Y. Yang, X. Deng, L. Carin, F. Li, and C. Tao. Tight mutual information estimation with contrastive fenchel-legendre optimization. *arXiv preprint arXiv:2107.01131*, 2021.

Y. Hung, V. R. Joseph, and S. N. Melkote. Design and analysis of computer experiments with branching and nested factors. *Technometrics*, 51(4):354–365, 2009.

F. Hutter, L. Kotthoff, and J. Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

B. Jones and C. J. Nachtsheim. Split-plot designs: What, why, and how. *Journal of quality technology*, 41(4):340–361, 2009.

T. T. Joy, S. Rana, S. Gupta, and S. Venkatesh. Hyperparameter tuning for big data using bayesian optimisation. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2574–2579. IEEE, 2016.

A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter. Fast bayesian optimization of machine learning hyperparameters on large datasets. In *Artificial intelligence and statistics*, pages 528–536. PMLR, 2017.

S. Kleinegesse and M. U. Gutmann. Bayesian experimental design for implicit models by mutual information neural estimation. In *International Conference on Machine Learning*, pages 5316–5326. PMLR, 2020.

S. Kleinegesse and M. U. Gutmann. Gradient-based bayesian experimental design for implicit models using mutual information lower bounds. *arXiv preprint arXiv:2105.04379*, 2021.

W.-Y. Lee, S.-M. Park, and K.-B. Sim. Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm. *Optik*, 172: 359–367, 2018.

R. V. Lenth. Quick and easy analysis of unreplicated factorials. *Technometrics*, 31(4):469–473, 1989.

S. Lessmann, R. Stahlbock, and S. F. Crone. Optimizing hyperparameters of support vector machines by genetic algorithms. In *IC-AI*, volume 74, page 82, 2005.

H. Li, P. Chaudhari, H. Yang, M. Lam, A. Ravichandran, R. Bhotika, and S. Soatto. Rethinking the hyperparameters for fine-tuning. *arXiv preprint arXiv:2002.11770*, 2020a.

L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, J. Ben-Tzur, M. Hardt, B. Recht, and A. Talwalkar. A system for massively parallel hyperparameter tuning. *Proceedings of Machine Learning and Systems*, 2:230–246, 2020b.

W. Li, Q. Zhou, and R. Zhang. Effective designs based on individual word length patterns. *Journal of Statistical Planning and Inference*, 163:43–47, 2015.

W. Li, R. W. Mee, and Q. Zhou. Using individual factor information in fractional factorial designs. *Technometrics*, 61(1):38–49, 2019.

J. Lian, L. Freeman, Y. Hong, and X. Deng. Robustness with respect to class imbalance in artificial intelligence classification algorithms. *Journal of Quality Technology*, 53(5):505–525, 2021.

J. Lian, K. Choi, B. Veeramani, A. Hu, L. Freeman, E. Bowen, and X. Deng. Do-aiq: A design-of-experiment approach to quality evaluation of ai mislabel detection algorithm. *arXiv preprint arXiv:2208.09953*, 2022.

J. J. Lim, J. Goh, M. B. M. A. Rashid, and E. K.-H. Chow. Maximizing efficiency of artificial intelligence-driven drug combination optimization through minimal resolution experimental design. *Advanced Therapeutics*, 3(4):1900122, 2020.

P. R. Lorenzo, J. Nalepa, M. Kawulok, L. S. Ramos, and J. R. Pastor. Particle swarm optimization for hyper-parameter selection in deep neural networks. In *Proceedings of the genetic and evolutionary computation conference*, pages 481–488, 2017.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

A. Lydia and S. Francis. Adagrad—an optimizer for stochastic gradient descent. *Int. J. Inf. Comput. Sci*, 6(5):566–568, 2019.

S. Mak and C. J. Wu. cmenet: A new method for bi-level variable selection of conditional main effects. *Journal of the American Statistical Association*, 114(526):844–856, 2019.

R. G. Mantovani, T. Horváth, R. Cerri, J. Vanschoren, and A. C. de Carvalho. Hyper-parameter tuning of a decision tree induction algorithm. In *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 37–42. IEEE, 2016.

R. Mukerjee, C. J. Wu, and M.-C. Chang. Two-level minimum aberration designs under a conditional model with a pair of conditional and conditioning factors. *Statistica Sinica*, pages 997–1016, 2017.

M. S. Phadke. *Quality engineering using robust design*. Prentice Hall PTR, 1995.

P. Probst, M. N. Wright, and A.-L. Boulesteix. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, 9 (3):e1301, 2019.

X. Ren, H. Yin, T. Chen, H. Wang, Z. Huang, and K. Zheng. Learning to ask appropriate questions in conversational recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 808–817, 2021.

S. Sadeghi, P. Chien, and N. Arora. Sliced designs for multi-platform online experiments. *Technometrics*, 62(3):387–402, 2020.

M. Seeger, F. Steinke, and K. Tsuda. Bayesian inference and optimal design in the sparse linear model. In *Artificial Intelligence and Statistics*, pages 444–451. PMLR, 2007.

J. Shao. An asymptotic theory for linear model selection. *Statistica sinica*, pages 221–242, 1997.

J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

G. Taguchi. *System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Costs*. American Suppliers Institute, 1987.

B. Tang and C. Wu. Characterization of minimum aberration 2n-k designs in terms of their complementary designs. *The Annals of Statistics*, pages 2549–2559, 1996.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

C. J. Wu and M. S. Hamada. *Experiments: planning, analysis, and optimization*. John Wiley & Sons, 2011.

D. Yogatama and G. Mann. Efficient transfer learning method for automatic hyperparameter tuning. In *Artificial intelligence and statistics*, pages 1077–1085. PMLR, 2014.

M. Yuan, V. R. Joseph, and Y. Lin. An efficient variable selection approach for analyzing designed experiments. *Technometrics*, 49(4):430–439, 2007.

Chapter 4 How Do Large Multimodal Models Really Fare in Classical Vision Few-Shot Challenges? A Deep Dive

Abstract

Recent advances in multimodal foundational models have demonstrated marvelous in-context learning capabilities for diverse vision-language tasks. However, existing literature has mainly focused on few-shot learning tasks similar to their NLP counterparts. It is unclear whether these foundation models can also address classical vision challenges such as few-shot classification, which in some settings (*e.g.*, 5-way 5-shot) necessitates sophisticated reasoning over several dozens of images – a challenging task for learning systems. In this work, we take a deep dive to probe the potential and limitations of existing multimodal models on this problem. Our investigation reveals that while these models under careful calibration can outperform dedicated visual models in complex narratable scenes, they can falter with more abstract visual inputs. Moreover, we also investigate curriculum learning and find out how it can mitigate the performance gap via smoothly bridging verbal and nonverbal reasoning for vision language tasks.

Key Words: Large language model, Computer vision, Multi-modal alignment, Few-shot learning, Curriculum learning

4.1 Introduction

How humans rapidly adapt across different tasks with little supervision has long fascinated the science community (Thorpe et al., 1996; Graves et al., 2014; Biederman, 1987). In many real-world situations, acquiring large datasets for training is either impractical, infeasible, or cost prohibitive, thus making reliable predictions about unseen cases from sparse exemplars a vital research direction for machine learning (Thrun, 1998; Fei-Fei et al., 2004; Vinyals et al., 2016; Finn et al., 2017).

As humans, our ability to generalize well without extensive supervision is widely believed to come from prior experiences, knowledge, and the ability to integrate information and *think*. For example, when adults learned about the animal *llama* for the first time, it rarely takes more than a couple of images to register the concept of this species for its resemblance to more common animals such as sheep, camels, or horses[1]. This fast learning happens by (i) activating prior **knowledge** from previously seen tasks (*e.g.*, similar animals); and (ii) extracting useful **information** from the limited exemplars for the new task. To enable robust generalizations, especially in situations of incomplete information, (iii) **reasoning** kicks in: creating higher levels of abstraction to relate disparate pieces of information, formulating/testing hypotheses, and making logical extrapolations.

These intuitions have all been mathematically formalized under various learning theories, such as *Bayesian methods* (Ding et al., 2021), *weakly supervised learning* (Robinson et al., 2020), *causal machine learning* (Schölkopf et al., 2021), *information theoretic generalization bounds* (Chen et al., 2021), *etc.* While different technical notations of knowledge, information, and reasoning can provably improve few-shot generalization, for real-world problems there has been a lack of generic learning frameworks to accommodate knowledge manipulation and complex reasoning at scale. Consequently, domain knowledge-driven or regularization-based methods have ruled few-shot learning leaderboards in practice (Hu et al., 2022).

In recent years, the rise of foundational models has initiated a paradigm shift in building

[1]In fact, in some languages *llama* literally translates into *sheep-camel*, *camel-horse*, *etc.*

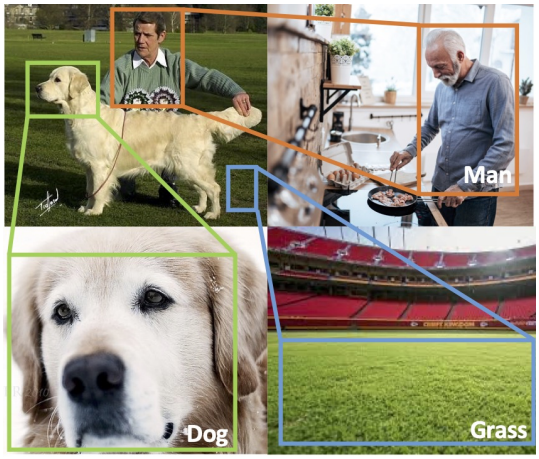

Figure 4.1: Multiple concepts often co-exist in complex visual scenes, making 1-shot classification an ill-posed problem. Thus, the system must reason from multi-shot examples for accurate concept binding.

general-purpose tools for machine learning (Eloundou et al., 2023). Large language models (LLMs) such as `ChaptGPT` have demonstrated human-like problem-solving skills when strong reasoning and encyclopedic knowledge are seamlessly integrated, and they are receptive to human instructions to collaboratively complete more complex tasks. This has also ignited significant interest in transcending the domain boundaries to synergize visual and language foundational models (Lu et al., 2021; OpenAI) to build *Large Multimodal Models* (LMM). Recent studies have shown the effectiveness & efficiency of interfacing different modalities using light-weight adapters (Tsimpoukelli et al., 2021; Huang et al., 2023; Li et al., 2023; Zhu et al., 2023), and the impressive emerging generalization capabilities of these fused models on both traditional and novel vision-language tasks, either with or without few-shot task adaptation.

While existing LMMs perform strongly on certain dimensions (*e.g.*, few-shot task following, captioning, VQA, *etc.*), they are highly reliant on what the base LLMs have been heavily tuned on. This raises an interesting question: to what extent do broad knowledge and strong logic benefit classical vision challenges? For example, the multi-way few-shot classification also necessitates advanced cognitive function, and solving this challenge entails several core competence dimensions not adequately covered by popular multimodal benchmarks (Liu et al., 2023; Xu et al., 2023; Fu et al., 2023): models need to compose reasoning over a large number

of images, possibly with complex visual scenes and subject to heavy confounding (see Figure 4.1 for example, the same scene can be associated with multiple categories).

In this work, we make the following contributions: (*i*) ablating different few-shot classification strategies for LMM; (*ii*) benchmarking on an extensive set of datasets with varying difficulty; (*iii*) proposing auxiliary tasks that boost performance and interpretability. Our findings show that with proper tuning, LMMs can do exceptionally well on complex narratable visual inputs, and even beat state-of-art dedicated vision models; however, their effectiveness degrades on more abstract visual inputs and struggle with subtle differences that require very sophisticated, verbose descriptions. These observations reveal current gaps for multimodal models: they have mostly learned shallow object concept mappings during training and heavily rely on verbal reasoning to perform inference. To facilitate future research, we also release synthetic data annotations used in this study.

4.2 Background and Problem Setup

Large Multimodal Models (LMM). In this work, we focus on the popular multimodal adapter architecture, where models are comprised of three components: a base LLM to process instructions and generate responses, a visual encoder to embed visual inputs, and an adapter to align visual embeddings to LLM inputs. This modularized design is highly flexible and parameter efficient: one can easily plug in different pre-trained foundational models and only train the lightweight adapter to achieve impressive performance on a variety of tasks. Specifically, we follow MiniGPT-4's recipe as our starting point (Zhu et al., 2023): LLaMA-based `Vicuna` model is used as our base LLM, and the vision encoder is the pre-trained Q-former from `BLIP-2` (Li et al., 2023) along with its `ViT` backbone (Fang et al., 2023), and a simple linear adaptation layer is used to connect the two. See Figure 4.2 for our model architecture. We expect our conclusions to generalize well to other LMMs as MiniGPT-4's architecture and training recipe are simple and representative.

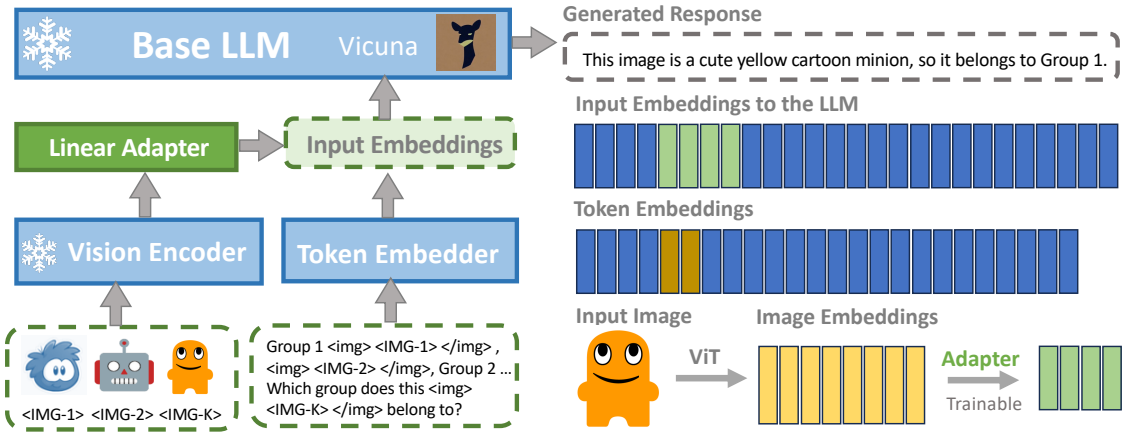

Figure 4.2: Model architecture of multimodal adapter.

Vision-language alignment and instruction tuning. Similar to their language-only counterparts, LMMs typically need to go through a two-stage training process to become useful. In the first alignment stage, the models are supervised with diverse text-image pairs or interleaved multimodal texts to establish the mapping between corresponding visual and language tokens. Massive amounts of data are used in this phase to teach various visual concepts, and typically, they are noisy and not directly tied to specific tasks. To teach LMM to understand human intents and accomplish regular vision-language tasks, instruction fine-tuning coach models with high-quality task data in the second stage. Perhaps surprisingly, if the base LLM is already well-tuned on diverse instructions, then the second-stage learning can be highly efficient: models tuned with a few hundred examples on simple generic tasks such as image captioning are already capable of carrying out a wide range of common visual-language tasks, and it only takes a few minutes on a single GPU. More extensive tuning on a richer set of tasks typically yields further gains on quantifiable dimensions, as evaluated by comprehensive benchmarks (Xu et al., 2023; Liu et al., 2023).

Few-shot classification. In this work, we adopt the classical *N-way K-shot* setup for the few-shot classification problem. Specifically, the models are trained and evaluated in an episodic fashion: each episode $\mathfrak{e}$ is composed of a support set $X^{\mathfrak{e}}_{\text{supp}} = \{(x_{nk}, y_{nk})|n \in [1, N], k \in [1, K]\}$ and a query set $X^{\mathfrak{e}}_{\text{query}} = \{(x_l, y_l)|l \in [1, L]\}$, where n indexes different classes. The goal is to make predictions on the query labels using the K exemplars for each

class within the episode. Compared to traditional "static" classification where classes are fixed beforehand at training time, few-shot classification poses challenges on capturing generalizable intra/inter-class patterns dynamically with only a handful of examples. As such, the classes used in our evaluation are disjoint from the classes used in training, *i.e.*, $\mathcal{C}_{\texttt{train}} \bigcap \mathcal{C}_{\texttt{test}} = \emptyset$. If powerful learners fail to come up with a good hypothesis, it easily overfits.

Importance of in-context few-shot classification for large visual systems. In open-world interactive settings, new visual concepts or categories not present during training can emerge (*e.g.*, personal items, pets, *etc.*), and the visual system needs to act on such unseen visual entities. Often in such situations, the system may receive some limited directions, blending both language instruction and a few visual examples. For complex visual scenes, the system also has to deal with additional difficulties such as distracting backgrounds, visual occlusion, confounding elements, *etc.* Also, for large visual systems, it is typically impractical to perform gradient-based task adaptations. Hence, in-context few-shot classification becomes a critical capability, enabling the system to comprehend and reason about these unfamiliar concepts to successfully complete the tasks. To the best of the author(s)' knowledge, this critical dimension has not been rigorously investigated in LMM literature.

4.3 Few-Shot Classification With LMM

Compared to existing few-shot methods, LMMs have several key advantages: (*i*) it can leverage the enormous parametric knowledge; (*ii*) powerful reasoning of various kinds (*e.g.*, logical, commonsense, causal, counterfactual, inductive, abductive, *etc.*); (*iii*) flexibility to receive additional language guidance such as instruction or corrective feedback. It is natural to hypothesize that these models should also work reasonably well on the classic N-way K-shot classification:

- For simpler cases where all input images feature a salient object that has a common name, models can directly invoke their zero-shot classification ability to classify;

- For harder cases where more complex visual scenes are involved, models should activate higher levels of cognitive features, first decompose the scene into more elementary components, then apply reasoning to summarize the intra-class commonalities & inter-class differences to derive decision rules.

Despite the above reasons for optimism, we also foresee a few risks that may prevent LMMs from being performant in few-shot image classification:

- **Hallucination:** LMMs also suffer from hallucinations, and such errors can propagate through the decision chain and lead to wrong conclusions;
- **Unstructured visual tokens:** It is unclear how LLMs perceive and process visual inputs that are presented as cluttered semantic embeddings, the standard reasoning process learned from structured texts may break down in such scenarios;
- **Mismatched correspondence:** LMMs are generally tuned to optimize text-image correspondence during training, while few-shot vision classification mainly assesses the understanding of image-image correspondence;
- **Visual information overflow:** At training time, LMMs typically see no more than a couple of images in the input; while in the standard 5-way 5-shot setup for example, the models need to reason over $5 \times 5 + 1 = 26$ images, which is cognitively overwhelming.

With these doubts in mind, we ran a quick sanity test on the 5-way 5-shot `miniImageNet` classification task using the `OpenFlamingo` model. Not surprisingly, the emerging few-shot task following capability failed this classical challenge: we only observed a meager accuracy of 27.5%, slightly better than random guessing.

Having established that emergence alone is inadequate for few-shot classification, we are interested in exploring how additional supervised training can help. To formalize this, let us denote the input sequence of images as X, and the target label as Y, then we can finetune the model with respect to the maximal likelihood loss $-\mathbb{E}_{p(x,y)}[\log q(y|x)]$, where $p(x, y)$ is the underlying ground-truth distribution and $q(y|x)$ is the predictive distribution parameterized

by the neural network (in this case, LMM). However, as we will see later in the experiments (Section 4.4), while directly training on the label loss significantly improves over the vanilla baseline, this is not competitive with other state-of-the-art (dedicated) few-shot classification solutions while our base LLM model is very strong at reasoning. And contrary to our expectations, training converges slowly.

After carefully analyzing the model outputs and learning dynamics, we attributed the sub-optimal results to the catastrophic forgetting of prior knowledge. More specifically, we observed that the model quickly overfitted to the task of predicting image labels, while significantly regressed on general visual language understanding capabilities. We conjectured that fitting on labels only caused the model to forget subtle visual details important for reasoning the right answer. Such motivated, we propose to combat catastrophic forgetting by encouraging the model to maximally retain detailed knowledge about input image sequence X, which can empower the LMMs to reason and make predictions better.

To simplify our discussion, let us extend the notation Y to cover both the label and detailed description (*i.e.*, caption) of the input images. The idea is that the caption contains rich details of the image, and if we use the LMM to model both the caption and labels, then the LMM's intermediate representation will retrain more useful information to reason about the label. Therefore, we can train the model to maximize the mutual information between X and Y

$$I(X, Y) = \iint \log \frac{p(x, y)}{p(x)p(y)} \, dx \, dy. \tag{4.1}$$

From earlier discussions in Chapter 2, we have learned that explicitly optimizing mutual information is non-trivial, especially for practical applications where large neural networks are involved. In particular, contrastive variational algorithms like `InfoNCE` and `FLO` introduced in the previous chapter are not directly applicable due to the special network architectures adopted by the transformer-based large language models: i) the feature space is the sequence of embedding vectors rather than one single embedding necessitating special network module to model the projection head; ii) contrastive variational algorithms are best suited for train-

ing encoders but not generators, which entails using sophisticated algorithms such as REINFORCE to backpropagate the gradient. To bypass these difficulties, the following theory offers clever approaches to tackle the challenges associated with optimizing mutual information for auto-regressive large language models.

Theorem 4.3.1 (Rate-distortion Inequality (Alemi et al., 2018)). *The following inequality holds*

$$H \ - \ D \ \leq \ I(X, Y), \tag{4.2}$$

where

$$H = - \int p(y) \ \log p(y) \, dy \tag{4.3}$$

$$D = - \int p(y) \, dy \ \int p(x|y) \ \log q(y|x) \, dx \tag{4.4}$$

Proof. The proof is based on the fact that Kullback-Leibler (KL) divergences are positive semidefinite

$$\mathrm{KL} \left[p(y|x) \ || \ q(y|x) \right] = \int p(y|x) \ \log \frac{p(y|x)}{q(y|x)} \, dy \ \geq 0, \tag{4.5}$$

which means for any distribution $q(y|x)$:

$$\int p(y|x) \log p(y|x) \, dy \geq \int p(y|x) \ \log q(y|x) \, dy, \tag{4.6}$$

Based on the above statement,

$$
\begin{aligned}
I(X,Y) &= \iint \log \frac{p(x,y)}{p(x)p(y)}\, dx\, dy && (4.7)\\
&= \int p(x)\, dx \int p(y|x)\ \log \frac{p(y|x)}{p(y)}\, dy && (4.8)\\
&= \int p(x)\, dx\ \left[\int p(y|x)\ \log p(y|x)\, dy - \int p(y|x)\ \log p(y)\, dy\right] && (4.9)\\
&\geq \int p(x)\, dx\ \left[\int p(y|x)\ \log q(y|x)\, dy - \int p(y|x)\ \log p(y)\, dy\right] && (4.10)\\
&= \int p(y)\, dy \int p(x|y)\ \log \frac{q(y|x)}{p(y)}\, dx && (4.11)\\
&= \left(-\int p(y)\ \log p(y)\, dy\right) - \left(-\int p(y)\, dy\ \int p(x|y)\ \log q(y|x)\, dx\right) && (4.12)\\
&= H\ -\ D, && (4.13)
\end{aligned}
$$

H is the *data entropy* which measures the complexity of data and can be treated as a constant, and D is known as the *distortion* between $p(x|y)$ and $q(y|x)$.

□

Remark. The complete version of rate-distortion inequality also dictates the mutual information is upper-bounded by the rate term. $R = \int p(y)\, dy\ \int p(x|y)\ \log \frac{p(x|y)}{m(x)}\, dx$, where $m(x)$ is a variational approximation to $p(x)$.

According to the rate-distortion theory (Alemi et al., 2018), optimizing mutual information can be translated into minimizing its lower bound. The entropy H, corresponding to the ground truth distribution $p(y)$, is fixed since it is inherent to the distribution. The term D represents the maximum likelihood loss concerning the model distribution $q(y|x)$. For simplicity, we only consider D as the training loss. This formulation not only addresses the challenges of mutual information optimization but also underscores the significance of detailed captioning as a primary training task. This is due to its ability to provide extensive information about images. In the following sections, we outline a series of meticulously designed experiments aimed at exploring the model's capabilities in this domain. Details regarding the training and the prompt templates used are provided in the Appendix 4.6.1.

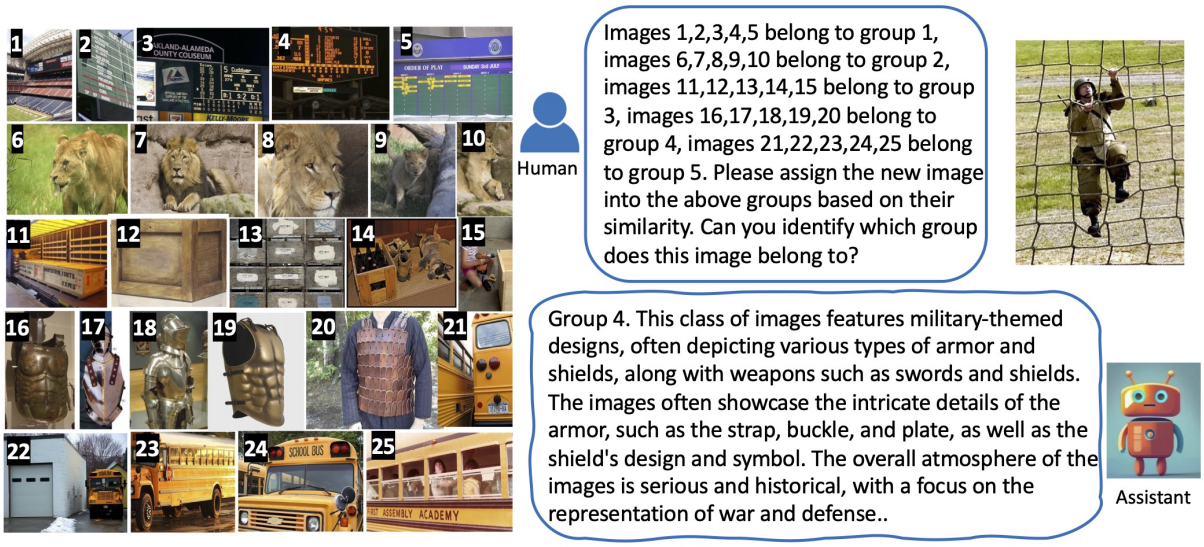

Figure 4.3: An example of reasoning augmented few-shot classification using LMM.

Caption-guidance (two-step). Given most LMMs are pretrained on image captioning, we propose a two-step captioning guided classification strategy as our baseline: in the first step, we generate a descriptive caption for the query image; in the second step, we prompt the model with the support image embeddings and the generated query caption to predict the query image label.

Reasoning-guidance (one-step). Our second strategy is motivated by chain-of-thought (CoT) reasoning and the consideration to make the model's decision more interpretable: in addition to the label prediction, models also need to generate the reasoning for the decision (Figure 4.3). We adopt an inductive setup to synthesize reasoning examples for training: sample 5 images from the query class, caption them using LMM, and then use a rewriter LLM to summarize captions' commonalities as the supporting argument. To improve reasoning quality, we both inject manually authored seed examples to guide the rewriter and apply *ad-hoc* rule-filtering. After experimenting with multiple candidates, we pick `Vicuna-7B` v1.5 as our rewriter.

Selective focusing (auxiliary task). One key observation we made is that vision-adapted LLMs suffer cognitive overloading when processing dense image inputs, and they easily con-

fuse the contents from different images. This inspired us to augment the training with an auxiliary task we call selective focusing, where the models are instructed to generate detailed descriptions for one randomly selected image from its inputs. Given the absence of such granular level annotations, we use synthetic dense captions generated by the LMM as training data.

Curriculum training. Our initial results reveal substantial performance gaps on datasets with unnarratable visual inputs or visually similar categories. We conjecture this is due to the sharp transition from verbal to non-verbal reasoning that traps the model in bad local optima. To remedy this, we adopt curriculum training for these more challenging settings, where we warm-start the model on narratable classifying diverse narratable scenes (*i.e.*, `ImageNet`).

4.4 Experiments

Table 4.1: MiniImageNet 5-way 5-shot test classification accuracy (%).

Baselines	Acc	Baselines	Acc	Baselines	Acc	MM-LLM	Acc
ProtoNet (Snell et al., 2017)	79.4	FRN (Wertheimer et al., 2021)	82.8	PAL (Ma et al., 2021)	84.4	`CAP`	79.5
FEAT (Ye et al., 2020)	82.0	BML (Zhou et al., 2021)	83.6	COSOC (Luo et al., 2021)	85.2	`NVR`	85.3
DeepEMD (Zhang et al., 2020)	82.4	Meta-NVG (Zhang et al., 2021a)	83.8	CNL (Zhao et al., 2021)	83.4	`REASON+SF`	89.3
MELR (Fei et al., 2020)	83.4	MetaQDA (Zhang et al., 2021b)	84.3	FewTURE (Hiller et al., 2022)	86.4	`NVR+SF`	**92.8**

`CAP`: caption-guided; `NVR`: non-verbal reasoning; `REASON`: reasoning-guided; `SF`: selective focusing.

We used the following popular few-shot classification benchmarks in our experiments: *miniImageNet* (Vinyals et al., 2016), *tieredImageNet* (Ren et al., 2018), *CIFAR-FS* (Bertinetto et al., 2019), and *Meta-Dataset* (Triantafillou et al., 2020). Limited by space, we present only the key results in the main text and defer details of training & evaluation setups to the Appendix 4.6.2. Our code will be available from `https://github.com/qingguo666/MMFSL`.

We start by ablating different learning strategies on the `miniImageNet` with 5-way 5-shot classification. Table 4.1 summarizes the main results along with baseline numbers from prior arts [2]. Despite being a general-purpose tool, LMM performs strongly on the few-shot clas-

[2]We exclude works using visual encoder trained from full ImageNet data due to information leakage con-

Table 4.2: 5-way 5-shot accuracies (%)

	tiredIN	miniIN	CIFAR-FS
ProtoNet (Snell et al., 2017)	84.01	79.46	-
FEAT (Ye et al., 2020)	84.79	82.05	-
MetaQDA (Zhang et al., 2021b)	89.56	84.28	88.79
FewTURE (Hiller et al., 2022)	89.96	86.38	88.90
MM-LLM	**91.83**	**92.8**	**92.5**
cross-domain	[Source]	97.17†	89.17

† miniIN might overlap with tiredIN.

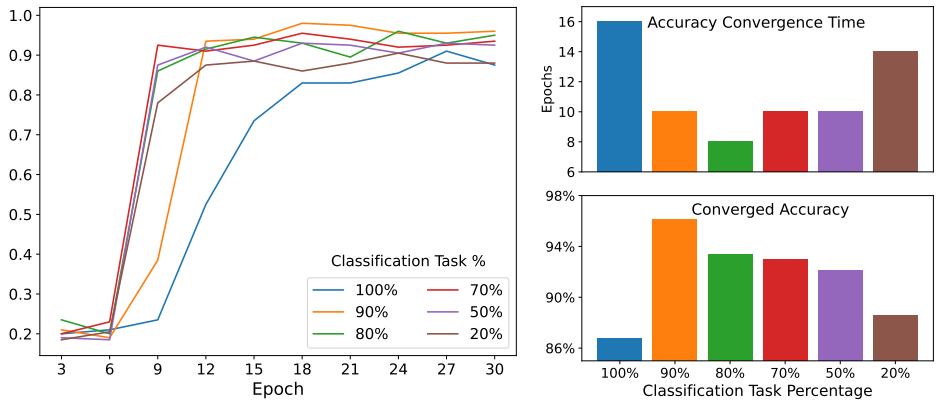

Figure 4.4: Ablation on the effect of adding different portions of selective focusing tasks. A small fraction of description teaches the model to better understand subtle differences between the images

sification task: without exploiting any special architecture, feature engineering, or inductive bias, they are able to match the SOTA performance of dedicated vision FSL models. Our close analyses of the less performant caption-guided strategy revealed errors are mostly due to hallucinated objects in the caption or the target object not visually salient.

While we saw the generated reasoning does generalize reasonably well for unseen inputs, contrary to our original expectation, reasoning-guided learning is less accurate compared to their no reasoning counterpart. The fact that text generation-free strategies worked better indicates: (1) harmful hallucination happens in the text decoding step; (2) the visual embeddings received by the model do capture richer signals that can be conveyed in text. We note visual information overflow seems to be the main blocker, as adding selective-focusing enables

cerns.

LMM to beat SOTA results by a large margin.

Figure 4.4 further ablates how different balancing ratios between classification and selective-focusing impact final accuracy and convergence speed, a small fraction ($10 \sim 20\%$) of focusing can be most beneficial. Based on the learnings from `miniImageNet`, we apply the best setting to `tieredImageNet` and `CIFAR-FS`. Table 4.2 shows that the results are consistent with our observations from the `miniImageNet` experiments, and the knowledge appears to transfer well across domains.

Next, we move to more challenging settings where the models must discriminate between: (*i*) categories that are visually similar (*i.e.*, `Flower`, `Airplane`, `Fungi`, `Birds`); and (*ii*) categories that are hard to describe (*e.g.*, `Omniglot`, `Texture`). In Table 4.3, direct cross-domain transfer from `tiredImageNet` trained model does not do well, which is consistent with our observation that descriptions for images from these datasets are often too broad to make a discriminative call with high confidence (see examples in the Appendix 4.6.3). However, if we warm start from the *tiredImageNet* checkpoint, they become highly competitive to the SOTA results only after 5 epochs' training. We hypothesize this is because after being proficient in leveraging verbal reasoning to solve problems, it becomes easier to grasp non-verbal reasoning skills on top of that.

As a final remark, recent studies have warned of potential catastrophic forgetting after the fine-tuning of LMMs (Zhai et al., 2023). We therefore interacted with the few-shot optimized models and noticed models can still perform other vision-language tasks, but are less compelling than before.

4.5 Conclusion

We have demonstrated that LMM provides a simple, effective, interpretable approach to address the challenge of few-shot image classification. Our results showed model experienced hardships of visual overloading and non-verbal reasoning, which can be mitigated via smoothing the learning curves through introducing auxiliary tasks and adopting curriculum learning.

Table 4.3: 5-way 5-shot accuracies (%) and 95% confidence interval on `Meta Dataset`

Method	Flower	Airplane	Fungi	Birds	Texture	Omniglot
Train on ImageNet only						
k-NN (Triantafillou et al., 2020)	83.10±0.68	46.81±0.89	36.16±1.02	50.13±1.00	66.36±0.75	37.07±1.15
MatchingNet (Triantafillou et al., 2020)	80.13±0.71	48.97±0.93	33.97±1.00	62.21±0.95	64.15±0.85	52.27±1.28
ProtoNet (Doersch et al., 2020)	86.96±0.73	58.04±0.96	40.73±1.15	74.07±0.92	68.76±0.77	68.50±1.27
fo-MAML (Triantafillou et al., 2020)	81.74±0.83	56.24±1.11	32.10±1.10	63.61±1.06	68.04±0.81	55.55±1.54
RelationNet (Triantafillou et al., 2020)	68.76±0.83	40.73±0.83	30.55±1.04	49.51±1.05	52.97±0.69	45.35±1.36
BOHB (Saikia et al., 2020)	87.34±0.59	54.12±0.90	41.38±1.12	70.69±0.90	68.34±0.76	67.57±1.21
TSA (Li et al., 2022)	94.05±0.45	80.13±1.01	51.38±1.17	83.39±0.80	**79.61±0.68**	82.58±1.11
Train on MetaDataset						
MatchingNet (Triantafillou et al., 2020)	81.90±0.72	69.17±0.96	33.70±1.04	56.40±1.00	61.80±0.74	78.25±1.01
ProtoNet (Triantafillou et al., 2020)	86.85±0.71	71.14±0.86	40.26±1.13	67.01±1.02	65.18±0.84	79.56±1.12
RelationNet (Triantafillou et al., 2020)	76.08±0.76	69.71±0.83	32.56±1.08	54.14±0.99	56.56±0.73	86.57±0.79
fo-Proto-MAML (Triantafillou et al., 2020)	88.72±0.67	75.23±0.76	41.99±1.17	69.88±1.02	68.25±0.81	82.69±0.97
CNAPs (Requeima et al., 2019)	88.90±0.50	83.70±0.60	50.20±1.10	73.60±0.90	59.50±0.70	91.70±0.50
SUR-pnf (Dvornik et al., 2020)	90.00±0.60	79.70±0.80	49.80±1.10	75.90±0.90	72.50±0.70	90.00±0.60
URT (Liu et al., 2021)	88.20±0.60	85.80±0.60	63.50±1.00	76.30±0.80	71.80±0.70	94.40±0.40
FLUTE (Triantafillou et al., 2021)	91.60±0.60	87.20±0.50	58.10±1.10	79.20±0.80	68.80±0.80	93.20±0.50
URL (Li et al., 2021)	92.11±0.48	88.59±0.46	68.75±0.95	80.54±0.69	76.17±0.67	94.51±0.41
TSA (Li et al., 2022)	92.18±0.52	89.33±0.44	67.40±0.99	81.42±0.74	76.74±0.72	**94.96±0.38**
LMM (starting from `tiredImageNet` trained checkpoint)						
Direct transfer	69.6±1.11	27.6±2.31	30.5±1.71	64.8±1.34	59.3±2.16	34.2±1.42
Curriculum on each dataset	96.7±0.62	89.9±0.85	81.2±0.61	96.7 ±1.09	77.0±1.58	94.3±0.77
Curriculum on all datasets	**97.3±0.30**	**91.6±0.50**	**83.2±1.80**	**96.9±0.58**	78.2±0.09	91.9±1.40

4.6 Appendix

4.6.1 Prompt Templates

We present the prompt templates used in our experiments below. Note placeholders such as `<ImageHere>`, `<ImageDescriptionHere>` , `<Answer>`, `<ReasonHere>` will be replaced with actual contents, and highlighted parts (*e.g.*, `Group <Answer>.`) will be used for the loss computation.

Caption generation:

```
###Human:  <Img><ImageHere></Img> Describe this image in detail.
###Assistant:
```

Caption-guided classification.

```
###Human: You will be given a few different images. Image 1:
<Img><ImageHere></Img>; Image 2: <Img><ImageHere></Img>; Image 3:
<Img><ImageHere></Img>; ... Image 24: <Img><ImageHere></Img>; Image
25: <Img><ImageHere></Img>. Images 1,2,3,4,5 belong to group 1, images
6,7,8,9,10 belong to group 2, images 11,12,13,14,15 belong to group 3,
images 16,17,18,19,20 belong to group 4, images 21,22,23,24,25 belong
to group 5. Please assign the following images into the above groups
based on their similarity. Here is a description of a new image:
<ImageDescriptionHere> Can you identify which group does this image
belong to? ###Assistant: Group <Answer>.
```

Non-verbal classification.

```
###Human: You will be given a few different images. Image 1:
<Img><ImageHere></Img>; Image 2: <Img><ImageHere></Img>; Image 3:
<Img><ImageHere></Img>; ... Image 24: <Img><ImageHere></Img>; Image
25: <Img><ImageHere></Img>. Images 1,2,3,4,5 belong to group 1, images
6,7,8,9,10 belong to group 2, images 11,12,13,14,15 belong to group 3,
images 16,17,18,19,20 belong to group 4, images 21,22,23,24,25 belong to
group 5. Please assign the following images into the above groups based
on their similarity. Here is a new image: <Img><ImageHere></Img>.
Can you identify which group does this image belong to? ###Assistant:
Group <Answer>.
```

Reasoning-guided classification.

```
 ###Human:  You will be given a few different images.  Image 1:
<Img><ImageHere></Img>; Image 2:  <Img><ImageHere></Img>; Image
3:  <Img><ImageHere></Img>; ...  Image 24:  <Img><ImageHere></Img>;
Image 25:  <Img><ImageHere></Img>.  Images 1,2,3,4,5 belong to
group 1, images 6,7,8,9,10 belong to group 2, images 11,12,13,14,15
belong to group 3, images 16,17,18,19,20 belong to group 4, images
21,22,23,24,25 belong to group 5.  Please assign the following images
into the above groups based on their similarity.  Here is a new image:
<Img><ImageHere></Img>.  Can you identify which group does this image
belong to?  Group <Answer>.  <ReasonHere>
```

Reasoning generation.

Remark. We have used `Vicuna` V1.5 as our rewriter model, which is using different prompt template compared to the `Vicuna` V0 base model used by MiniGPT4. Specifically, speaker tags are

```
A chat between a curious user and an artificial intelligence assistant.
The assistant gives helpful, detailed, and polite answers to the user's
questions. USER: We have a lot of images coming from 50 different
classes, where images from the same class share some distinctive
common features (each image belongs to only one class). Give a list
of detailed descriptions of images from one of the class, you need
to summarize the common features for these descriptions in one or
two sentence. This summary will be used to instruct data annotators
to classify new images to this class. A good example of the summary
will be something like "this is the domes/buildings class, the images
consistently mention architectural structures, particularly domes, and
often provide details of intricate designs, decorations, and historical
context."

Here are the image descriptions for you to summarize:
1 <ImageDescriptionHere>
2 <ImageDescriptionHere>
...
5 <ImageDescriptionHere> ASSISTANT:
```

Selective focusing.

```
###Human: You will be given a few different images. Image 1:
<Img><ImageHere></Img>; Image 2: <Img><ImageHere></Img>; Image
3: <Img><ImageHere></Img>; ... Image 24: <Img><ImageHere></Img>;
Image 25: <Img><ImageHere></Img>. Describe image <ID> in detail.
###Assistant: <ImageDescriptionHere>
```

4.6.2 Experiment Details

Platform. We use `PyTorch` in our experiments. All training jobs are done on one Nvidia A100-80G GPU, and some of the evaluations are offloaded to Nvidia V100-32G GPU.

Base model. We have used MiniGPT4 for all our fine-tuning experiments. Specifically, we use the original official release built on `Vicuna V0`, and pick the 7B model for quick experimentation. Since MiniGPT4 only saw single pair of text-image data during training, we need to find a model trained on interleaved image-text to baseline few-shot classification accuracy. Since canidate models like `FROZEN`, `Flamingo`, `KOSMOS` are not publicly available, we tested the 9B version of `OpenFlamingo` model (Awadalla et al., 2023) built on top of `MPT`.

Training. In our experiments, we freeze all parameters except for the linear adapter. Models are trained for 30 epochs unless otherwise specified, with 3k iterations in each epoch. To avoid OOM error, we restrict the batch size to 1 and use gradient accumulation to update weights every 16 iterations. A cosine learning rate scheduler is used, with 200 warmup steps from 10^{-6} to 3×10^{-5}, then gradually anneal to 10^{-5} towards the end. A weight decay factor of 0.05 is also applied. Inputs are formatted using the templates from Appendix 4.6.1, and only the `Assistant` response tokens are used in the loss computation.

Evaluation. Given the limited computation resource we can use, different evaluation strategies have been applied to each experiment. For the main `ImagetNet` and `Cifar` results (Table 4.1 & 4.2), we used 600 episodes to compute the accuracy. For the ablation on the task mixing ratio and learning dynamics, we evaluated every three epochs and used 200 episodes for each checkpoint. Note evaluating with 200 episodes has higher variance, so the numbers in Figure 4.4 is slightly different from the numbers reported in previous tables (which used 600 episodes). But our observations are reliable as we saw consistent trends across different experiments. Finally, for the `Meta Dataset` experiments, we run a total of 1,800 episodes for each checkpoint. The additional episodes evaluated helped us to derive a reliable estimate of

the 95% confidence interval. For the "curriculum on each dataset", we trained 5 epochs; for the "curriculum with all `Meta Dataset`", we trained 50 epochs.

4.6.3 Examples Where Verbal Reasoning Struggle

Figure S1 gave a few examples of image descriptions generated for `Omniglot` characters. These abstract hieroglyph characters are difficult to describe verbally, which explains the bad transfer performance for the models trained on the more narratable `ImageNet`.

Original Minigpt4: The image shows a small black and white picture of a person in profile with their eyes closed. The person is standing in front of a white background with their head tilted to the left and their eyes closed. The image appears to be a simple line drawing of the person.

TieredImagenet trained: This is a simple black and white drawing of a cross. It has a simple design and is easy to recognize. The cross is made up of four intersecting lines, with two vertical lines on the top and two horizontal lines on the bottom. The cross is not very detailed, but it is still easy to recognize.

Ominiglot trained: The image shows an illustration of a horizontal line with a square in the middle. The line has a horizontal extension at one end and a vertical extension at the other end.

Original Minigpt4: This image appears tend. The triangle has one long side and two short sides, with the point facing downwards. It could be interpreted as a symbol for stability or balance, as it has a solid, structured form. The triangle could also be seen as a representation of the human body, with the long side representing the spine and the short sides representing the limbs. However, without more context, it is difficult to determine the exact meaning or significance of this image.o be a black and white depiction of a triangular shape with a small, pointed

TieredImagenet trained: The image shows a letter 'a' written in the Arabic script.

Ominiglot trained: The image shows a triangle made out of two lines. The lines are connected at their tips and have one point on top. The point is not visible in the image. The triangle could be used to represent a tent, a pyramid, or a point in space.

Figure S1: Image descriptions for `Omniglot` characters generated by different models.

Bibliography

A. Alemi, B. Poole, I. Fischer, J. Dillon, R. A. Saurous, and K. Murphy. Fixing a broken elbo. In *International conference on machine learning*, pages 159–168. PMLR, 2018.

A. Awadalla, I. Gao, J. Gardner, J. Hessel, Y. Hanafy, W. Zhu, K. Marathe, Y. Bitton, S. Gadre, S. Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023.

L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. In *ICLR*, 2019.

I. Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.

Q. Chen, C. Shui, and M. Marchand. Generalization bounds for meta-learning: An information-theoretic analysis. In *NeurIPS*, volume 34, 2021.

N. Ding, X. Chen, T. Levinboim, S. Goodman, and R. Soricut. Bridging the gap between practice and pac-bayes theory in few-shot meta-learning. *Advances in Neural Information Processing Systems*, 34:29506–29516, 2021.

C. Doersch, A. Gupta, and A. Zisserman. Crosstransformers: spatially-aware few-shot transfer. *Advances in Neural Information Processing Systems*, 33:21981–21993, 2020.

N. Dvornik, C. Schmid, and J. Mairal. Selecting relevant features from a multi-domain representation for few-shot classification. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 769–786. Springer, 2020.

T. Eloundou, S. Manning, P. Mishkin, and D. Rock. Gpts are gpts: An early look at the labor market impact potential of large language models. *arXiv preprint arXiv:2303.10130*, 2023.

Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023.

N. Fei, Z. Lu, T. Xiang, and S. Huang. Melr: Meta-learning via modeling episode-level relationships for few-shot learning. In *ICLR*, 2020.

L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135. PMLR, 2017.

C. Fu, P. Chen, Y. Shen, Y. Qin, M. Zhang, X. Lin, Z. Qiu, W. Lin, J. Yang, X. Zheng, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

M. Hiller, R. Ma, M. Harandi, and T. Drummond. Rethinking generalization in few-shot classification. *Advances in Neural Information Processing Systems*, 35:3582–3595, 2022.

S. X. Hu, D. Li, J. Stühmer, M. Kim, and T. M. Hospedales. Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference. In *CVPR*, pages 9068–9077, 2022.

S. Huang, L. Dong, W. Wang, Y. Hao, S. Singhal, S. Ma, T. Lv, L. Cui, O. K. Mohammed, Q. Liu, et al. Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*, 2023.

J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

W.-H. Li, X. Liu, and H. Bilen. Universal representation learning from multiple domains for few-shot classification. In *CVPR*, pages 9526–9535, 2021.

W.-H. Li, X. Liu, and H. Bilen. Cross-domain few-shot learning with task-specific adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7161–7170, 2022.

L. Liu, W. Hamilton, G. Long, J. Jiang, and H. Larochelle. A universal representation transformer layer for few-shot image classification. In *ICLR*, 2021.

Y. Liu, H. Duan, Y. Zhang, B. Li, S. Zhang, W. Zhao, Y. Yuan, J. Wang, C. He, Z. Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023.

K. Lu, A. Grover, P. Abbeel, and I. Mordatch. Pretrained transformers as universal computation engines. *arXiv preprint arXiv:2103.05247*, 1, 2021.

X. Luo, L. Wei, L. Wen, J. Yang, L. Xie, Z. Xu, and Q. Tian. Rectifying the shortcut learning of background for few-shot learning. *NeurIPS*, 34:13073–13085, 2021.

J. Ma, H. Xie, G. Han, S.-F. Chang, A. Galstyan, and W. Abd-Almageed. Partner-assisted learning for few-shot image classification. In *CVPR*, pages 10573–10582, 2021.

OpenAI. Gpt-4v(ision) system card.

M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel. Meta-learning for semi-supervised few-shot classification. In *ICLR*, 2018.

J. Requeima, J. Gordon, J. Bronskill, S. Nowozin, and R. E. Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. *Advances in Neural Information Processing Systems*, 32, 2019.

J. Robinson, S. Jegelka, and S. Sra. Strength from weakness: Fast learning using weak supervision. In *International Conference on Machine Learning*, pages 8127–8136. PMLR, 2020.

T. Saikia, T. Brox, and C. Schmid. Optimized generic feature learning for few-shot classification across domains. *arXiv preprint arXiv:2001.07926*, 2020.

B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

S. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. *nature*, 381(6582):520–522, 1996.

S. Thrun. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. Springer, 1998.

E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, U. Evci, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *ICLR*, 2020.

E. Triantafillou, H. Larochelle, R. Zemel, and V. Dumoulin. Learning a universal template for few-shot dataset generalization. In *ICML*, pages 10424–10433. PMLR, 2021.

M. Tsimpoukelli, J. L. Menick, S. Cabi, S. Eslami, O. Vinyals, and F. Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.

O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

D. Wertheimer, L. Tang, and B. Hariharan. Few-shot classification with feature map reconstruction networks. In *CVPR*, pages 8012–8021, 2021.

P. Xu, W. Shao, K. Zhang, P. Gao, S. Liu, M. Lei, F. Meng, S. Huang, Y. Qiao, and P. Luo. Lvlm-ehub: A comprehensive evaluation benchmark for large vision-language models. *arXiv preprint arXiv:2306.09265*, 2023.

H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8808–8817, 2020.

Y. Zhai, S. Tong, X. Li, M. Cai, Q. Qu, Y. J. Lee, and Y. Ma. Investigating the catastrophic forgetting in multimodal large language models. *arXiv preprint arXiv:2309.10313*, 2023.

C. Zhang, Y. Cai, G. Lin, and C. Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *CVPR*, pages 12203–12213, 2020.

C. Zhang, H. Ding, G. Lin, R. Li, C. Wang, and C. Shen. Meta navigator: Search for a good adaptation policy for few-shot learning. In *CVPR*, pages 9435–9444, 2021a.

X. Zhang, D. Meng, H. Gouk, and T. M. Hospedales. Shallow bayesian meta learning for real-world few-shot recognition. In *CVPR*, pages 651–660, 2021b.

J. Zhao, Y. Yang, X. Lin, J. Yang, and L. He. Looking wider for better adaptive representation in few-shot learning. In *AAAI*, volume 35, pages 10981–10989, 2021.

Z. Zhou, X. Qiu, J. Xie, J. Wu, and C. Zhang. Binocular mutual learning for improving few-shot classification. In *CVPR*, pages 8402–8411, 2021.

D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

Chapter 5 Summary and Discussion

This dissertation explores the intersection between machine learning and experimental design, aiming to harness the strengths of each field to enhance the other. By integrating flexible machine learning algorithms with cost-effective experimental design methodologies, this project seeks to develop innovative approaches that can lead to significant improvements in both domains.

Chapter 2 proposed a novel, provable, and efficient estimator for mutual information (MI) called `FLO` from the insight of information theory. This method can be applied to a broad range of data science applications. By integrating techniques from variational inference, contrastive learning, and convex optimization, the proposed FLO no longer requires a close-form likelihood of the data-generating process, thus overcoming a major limitation of classical experiment design strategies. This enables us to use a powerful deep neural net for nonparametric experiment design optimization. The proposed solution is Bayesian optimal and highly flexible: it uses FLO to optimize MI without using likelihood functions and can be easily generalized to sequential decision-making scenarios in complex environments. The proposed method has also been successfully applied in a wide range of complex tasks including epidemiological surveillance, pharmaceutical drug metabolism monitoring, and large-scale conversational recommendation systems. In addition, mutual information assesses the dependency between pairs of variables which is key to many scientific and engineering problems. Our proposed estimator `FLO` can be utilized to enhance feature extraction in these domains. For example, recent developments in self-supervised learning, particularly in training large foundational models, have extensively utilized the concept of mutual information optimization. This technique enhances the model's ability to extract representative features by maximizing the shared information between different views or transformations of the same data. By doing so, it ensures that the features captured are robust and meaningful, thereby improving the model's performance on a variety

of downstream tasks. Moreover, `FLO` provides a valuable technique for promoting fairness in machine learning models. By minimizing the mutual information between sensitive attributes (such as age and gender) and the model's responses, we can effectively reduce the dependence of the outputs on these attributes. This approach ensures that the model does not use information about these sensitive characteristics to make decisions, thereby helping to prevent biases that could lead to discriminatory practices.

Chapter 3 introduces the multi-layer sliced design (MLSD), which applies experimental design ideas to explore the impact of hyperparameters across various models and optimization strategies in AI algorithms. Given the high costs associated with evaluating every possible combination of factors, we employ a fractional factorial design. This approach allows us to identify significant factors using fewer experiments. Specifically, I developed the MLSD framework to conduct the slice minimum aberration design, ensuring it possesses sufficient power to estimate the effects of sliced factors. Furthermore, I introduced a novel analytical method designed to derive a parsimonious model that effectively captures the influence of hyperparameters under diverse settings. This methodology offers multiple benefits, including reduced manual effort and more efficient use of computational resources in AI applications. I estimate the effects of factors under the assumption that significant factors are sparse, applying a variable selection technique to discover effective factors. The method employed, named induced lasso, differs from traditional variable selection methods like the classic lasso. It not only selects variables but also conducts tests to assess the coefficients' significance. The versatility of the proposed MLSD framework allows for its application in online experiments to examine the impact of different advertising strategies across various platforms, including electronic devices (such as laptops and smartphones) and social media channels (like Instagram and Facebook).

Chapter 4 explores the few-shot learning capability of multi-modal models and leverages it to address classic vision classification tasks. Few-shot learning, where models learn from as few as one or two images per class, is pivotal especially when data collection is expensive and slow. It offers a solution that considerably reduces training resources and time. To

address the challenge of few-shot image classification, I have implemented a novel solution based on a large vision-language model, incorporating innovative training strategies such as caption guidance, reasoning guidance, selective focusing, and curriculum training. These methods help bridge the gap between verbal and non-verbal reasoning in vision language tasks. Additionally, I demonstrated that minimizing cross-entropy loss indirectly optimizes mutual information, underscoring the significance of image captioning tasks. These tasks enable the model to more effectively learn and capture complex information from images. Also, I showed that this multi-modal model not only enables the large vision language model to excel at vision classification tasks but also makes the machine decisions interpretable. Specifically, I applied a chain-of-thought reasoning technique to the model to generate a step-by-step rationale for the model predictions in natural language. The efficacy of this approach is demonstrated through its superior performance over ten state-of-the-art baselines across nine benchmarks, particularly excelling in complex scenarios marked by visual confounders and non-verbal reasoning challenges. These include distinguishing between visually similar categories that are intricately describable, like different classes of flowers. This method not only allows the model to effectively leverage prior knowledge and extract pertinent information from limited examples but also to deliver precise and reliable classification outcomes.

In the future, I will delve deeper into the intersection of statistics, machine learning, and artificial intelligence. By embracing AI and leveraging analytical tools from statistics and mathematics, I am empowered to tackle the challenges in data science and AI systems from novel perspectives. Specifically, I intend to innovate semi-parametric formulations of self-supervised learning to overcome the challenges posed by low-resource learning. Additionally, I am interested in improving the multi-turn collaborative problem-solving efficiency between AI agents and humans. It involves teaching AI agents to optimize their action planning and collaborative decision-making in scenarios with limited interactive turns and applying ideas from experimental design.