On Exploiting Structures for Deep Learning Algorithms with Matrix Estimation

by

Yuzhe Yang

B.S., Peking University (2018)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2020

© Massachusetts Institute of Technology 2020. All rights reserved.

Author .....
Department of Electrical Engineering and Computer Science
May 15, 2020

Certified by .....
Dina Katabi
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

On Exploiting Structures for Deep Learning Algorithms with Matrix Estimation

by

Yuzhe Yang

Submitted to the Department of Electrical Engineering and Computer Science
on May 15, 2020, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Despite recent breakthroughs of deep learning, the intrinsic structures within tasks have not yet been fully explored and exploited for better performance. This thesis proposes to harness the structured properties of deep learning tasks using matrix estimation (ME). Motivated by the theoretical guarantees and appealing results, we apply ME to study the following two important learning problems:

1. Adversarial robustness. Deep neural networks are vulnerable to adversarial attacks. This thesis proposes ME-Net, a defense method that leverages ME. In ME-Net, images are preprocessed using two steps: first pixels are randomly dropped from the image; then, the image is reconstructed using ME. We show that this process destroys the adversarial structure of the noise, while re-enforcing the global structure in the original image. Comparing ME-Net with state-of-the-art defense mechanisms shows that ME-Net consistently outperforms prior techniques, improving robustness against both black-box and white-box attacks.
2. Value-based planning and deep reinforcement learning (RL). This thesis proposes to exploit the underlying *low-rank* structures of the state-action value function, i.e., Q function. We verify empirically the existence of low-rank Q functions in the context of control and deep RL tasks. As our key contribution, by leveraging ME, we propose a generic framework to exploit the underlying low-rank structure in Q functions. This leads to a more efficient planning procedure for classical control, and additionally, a simple scheme that can be applied to any value-based RL techniques to consistently achieve better performance on "low-rank" tasks.

The results of this thesis demonstrate the value of using matrix estimation to capture the internal structures of deep learning tasks, and highlight the benefits of leveraging structure for analyzing and improving modern learning algorithms.

Thesis Supervisor: Dina Katabi
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Dina Katabi, for her unstinting support and invaluable guidance. I feel fortunate to work with Dina. She is an extremely insightful researcher, influencing me the way to do research. Dina constantly inspired me with her passion for finding answers to the unknowns and doing research with real impact. I thank her for her patience to spend a huge amount of time brainstorming with me, listening to my ideas, reading my drafts/presentations and teaching me how to structure and improve them. I could not ask for a better advisor. I am truly grateful.

This thesis would not have been possible without my collaborators' wisdom and efforts. Zhi brings me to the field of matrix estimation; Discussing and brainstorming with Zhi has always been fruitful (together we have come up with so many crazy ideas!). Without his help and guidance, this thesis would not have been possible. Hao Wang managed to be a mentor, a friend, and an older brother; He not only showed me the ropes of doing research but also taught me the way to become a successful graduate student. I am also very fortunate to have learnt from my other co-authors, Shichao and Guo, who were fantastic collaborators.

I am really grateful for being a part of NETMIT and CSAIL, and working with amazing colleagues/friends. I owe my thanks to Deepak, Zach, Chen-Yu, Mingmin, Hao He, Lijie, Tianhong, Yuan, Abbas, Aniruddh, Colin, and Yingcheng, for their friendship and insightful discussions on various research topics. I also want to thank Jie, Zeyuan, Wenbo, Lei, Hongzi, and all other friends for their warm companionship when I was lost and frustrated.

Last but not least, I would like to thank my parents, my girlfriend Luxin Zhang, and my family, for their unconditional love, guidance, encouragement, and unwavering support of all my decisions over the years. Thank you.

Previously Published Material

Parts of this thesis are based on previously published materials and were done in collaboration with other authors. My contributions to these works centered around proposing and refining the ideas, implementation and empirical evaluation of the proposed algorithms. We collaborated closely in the brainstorming, algorithmic design and writing of the papers.

Chapter 2 revises a previous publication [1]: Yuzhe Yang, Guo Zhang, Dina Katabi, and Zhi Xu. ME-Net: Towards Effective Adversarial Robustness with Matrix Estimation. ICML, 2019.

Chapter 3 revises a previous publication [2]: Yuzhe Yang, Guo Zhang, Zhi Xu, and Dina Katabi. Harnessing Structures for Value-Based Planning and Reinforcement Learning. ICLR, 2020.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent breakthroughs in deep learning, especially by using deep neural networks (DNNs), have achieved impressive accuracy and wide adoption in the field of computer vision [9–13], natural language processing [14], control and planning [15–18], reinforcement learning (RL) [19–22], and countless applications on real systems in the physical world [23–27]. Yet, many real-world problems can exhibit intrinsic structures. Stochastic control often needs to control complex systems. While the state space could be high-dimensional, the dynamics are likely to possess some structured forms, such as being governed by physical laws, or partial differential equations. In addition, for deep reinforcement learning (DRL), while the dimension of images is quite large, it is likely that only a few latent features are actually informative, and hence sufficient for learning useful representations. Furthermore, for intelligent agents such as playing go, while the input, the board configuration, is complex, there are often cases or patterns where people have developed particular tactics. In those scenarios, there are only few if not many good moves. Overall, because of the structured dynamics or latent low-dimensional features, it is fairly reasonable to expect that, certain underlying structures will be imposed on different deep learning tasks. The key of this thesis is to study meaningful structure that naturally arises in deep learning problems, and design corresponding algorithms to our benefit.

In what follows, two case studies are performed under such structural viewpoint. We take a close look at two prevailing yet important deep learning problems: (1) *adversarial robustness*, which aims to enhance the robustness of modern DNNs against

small perturbations to the input; and (2) *deep reinforcement learning*, which introduces deep learning (architectures) to RL principles to create efficient algorithms.

1.1 Adversarial Robustness

Deep neural networks (NNs) are shown to be vulnerable to adversarial attacks, where the natural data is perturbed with human-imperceptible, carefully crafted noises [10]. By adding such small, indistinguishable perturbation to the inputs, an adversary can fool neural networks to produce incorrect outputs with high probabilities. This phenomena raises increasing concerns for safety-critical scenarios such as the self-driving cars where NNs are widely deployed.

Images contain noise: even the "clean" images taken from a camera contain small white noise from the environment. Such small, unstructured noise seems to be tolerable for modern deep neural networks, which achieve human-level performance. However, the story is completely different for carefully constructed noise. Structured, adversarial noise (i.e., adversarial examples) can easily corrupt the results, leading to incorrect prediction from human's perspective. This posts a natural conjecture: if one could somehow "revert" the noisy images back to some common, underlying global structure, then training and inference should be more robust. After all, images are structured data, and it is such global structures that make human perception stable.

As adversarial perturbations are carefully generated structured noise, a natural conjecture for defending against them is to destroy, or denoise such structured, adversarial noise. The most naive approach is to randomly mask some pixels: with probability p, independently for each pixel, one keeps the original value; otherwise, drop it (e.g., set the value to 0). While such method can eliminate the adversarial structure within the noise through random information drop, it is almost certain to fail since it equally destroys the information of the original image, making NN inference even worse.

While sub-optimal, random masking does possess some nice features as mentioned before, it leads to an interesting suggestion: perhaps completely eliminating adversarial noise is too ambitious and impossible; instead, we could try to reconstruct the images from the masked ones in an attempt to reduce the overall adversarial noise. After

all, images contain some internal *structures*. An image classified as cat should have at least a cat as its main body. If both training and testing are performed under the same underlying structures (i.e., there is no distributional shift in training and testing), we should hope the network to be generalizable and robust. In addition, if the reconstruction in terms of maintaining the underlying structure is satisfactory, the randomness in this pipeline (masking and reconstruction) is likely to redistribute the carefully constructed, originally adversarial noise into some other non-adversarially designed noise which are less likely to invalidate the predictions.

1.2 Planning & Deep Reinforcement Learning

Value-based methods are widely used in control, planning and reinforcement learning tasks [16, 18, 22, 28]. To solve a Markov Decision Process (MDP), one common method is to use value iteration, which finds the optimal value function and the optimal policy. This process can be done by iteratively computing and updating the state-action value function, represented by $Q(s, a)$ (i.e., the Q-value function). In simple cases with small state and action spaces, value iteration can be ideal for efficient and accurate planning. However, for modern MDPs, the data that encodes the value function usually lies in thousands or millions of dimensions [16, 17], let alone continuous state space, such as images in deep reinforcement learning [22, 29]. These practical constraints significantly hamper the efficiency and applicability of the vanilla value iteration.

However, the Q-value function is intrinsically induced by the underlying system dynamics. These dynamics are likely to possess some structured forms in various settings, such as being governed by partial differential equations. It is also possible that states and actions may contain latent features (e.g., similar states could have similar optimal actions). Thus, in those scenarios, it is reasonable to expect the structured dynamic to impose certain global *structure* on the Q-value. Since the Q function can be treated as a giant matrix, with rows as states and columns as actions, a structured Q function naturally translates to a *structured* Q matrix.

To begin with, we take a linear algebraic view by treating the Q function as a giant matrix, where each row represents a state, each column represents an action, and its ij-th entry represents the Q value of the corresponding state-action pair.

One of the fundamental global structures in studying matrices, is the rank of the matrix. Subsequently, we propose to explore the global *low-rank* structure of the Q matrix. Low-rank structure has been widely observed and exploited in modern big data (matrix) analysis [30]. As can be demonstrated empirically, the majority of the benchmarking Atari games, as well as many stochastic control tasks all exhibit low-rank Q matrices. This leads us to a natural question: How do we leverage the low-rank structure in Q matrices to allow value-based techniques to achieve better performance on "low-rank" tasks? In short, when the underlying tasks contain certain desired structures, a framework that is able to exploit such structured information to improve both planning and deep RL methods would be much desired.

1.3 Design Overview: Structured Framework with Matrix Estimation

Having set up the stage, we are now ready to introduce our structured framework for deep learning tasks. Specifically, we employ Matrix Estimation (ME) as our oracle to help harnessing the underlying structures in algorithms. ME is a fairly mature topic with strong theoretical guarantees and appealing practical performance. In the sequel, we will briefly describe the basic problem formulation, and then importantly, connect it with our setup to see why this technique should be a natural and effective method.

Matrix estimation is concerned with recovering a data matrix from noisy and incomplete observations of its entries. Consider a true, unknown data matrix $M \in \mathbb{R}^{n\times m}$. Often, we have access to a subset Ω of entries from a noisy matrix $X \in \mathbb{R}^{n\times m}$ such that $\mathbb{E}[X] = M$. For example, in recommendation system, there are true, unknown ratings for each product from each user. One often observes a subset of noisy ratings if the user actually rates the product online. Technically, it is often assumed that each entry of X, X_{ij}, is a random variable independent of the others, which is observed with probability $p \in (0, 1]$ (i.e., missing with probability $1 - p$). The theoretical question is then formulated as finding an estimator $\hat{M}$, given noisy, incomplete observation matrix X, such that $\hat{M}$ is "close" to M. The closeness is typically measured by some matrix norm, $||\hat{M} - M||$, such as the Frobenius norm.

Over the years, extensive algorithms have been proposed. They range from simple spectral method such as universal singular value thresholding (USVT) [31], which performs SVD on the observation matrix X and discards small singular values (and corresponding singular vectors), to convex optimization based methods, which minimize the nuclear norm [32], i.e.:

$$\min_{\hat{M}\in\mathbb{R}^{n\times m}} ||\hat{M}||_* \quad \text{s.t. } \hat{M}_{ij} \approx X_{ij},\ \forall\,(i,j)\in\Omega, \tag{1.1}$$

where $||\hat{M}||_*$ is the nuclear norm of the matrix (i.e., sum of the singular values). To speed up the computation, the Soft-Impute algorithm [33] reformulates the optimization using a regularization parameter $\lambda \geq 0$:

$$\min_{\hat{M}\in\mathbb{R}^{n\times m}} \frac{1}{2}\sum_{(i,j)\in\Omega}\left(\hat{M}_{ij} - X_{ij}\right)^2 + \lambda||\hat{M}||_*. \tag{1.2}$$

In this thesis, we view ME as a principled oracle to effectively exploit the low-rank structures.

The key message in the field of ME is: if the true data matrix M has some *global structures*, exact or approximate recovery of M can be theoretically guaranteed [31, 32, 34]. In the literature, the most studied global structure is low rank. This strong theoretical guarantee serves as the foundation for employing ME to exploit structures across different deep learning tasks.

Adversarial Robustness. As we treat the input images as matrices, to destroy the structure of adversarial noises while enforcing the structure of the original image, we can employ a masking-and-reconstruction pipeline, where ME is applied for reconstruction of the masked version. This process, as performed before sending the inputs into the DNNs, can redistribute the carefully constructed adversarial noises to non-adversarial structures.

In this thesis, we propose to leverage matrix estimation (ME) as our reconstruction scheme. We view a masked adversarial image as a noisy and incomplete realization of the underlying clean image, and propose ME-Net, a preprocessing-based defense that reverts a noisy incomplete image into a denoised version that maintains the underlying global structures in the clean image. ME-Net realizes adversarial robustness by using

such denoised global-structure preserving representations.

We note that the ME-Net pipeline can be combined with different training procedures. In particular, we show that ME-Net can be combined with standard stochastic gradient descent (SGD) or adversarial training, and in both cases improves adversarial robustness. This is in contrast with many preprocessing techniques which cannot leverage the benefits of adversarial training [6–8], and end up failing under the recent strong white-box attack [5].

Planning and Deep Reinforcement Learning. Following the intuitions from enforcing the structures in Q function, we propose a generic framework that leverages matrix estimation to exploit the low-rank structure in both classical planning and modern deep RL tasks. In particular, for classical control tasks, we propose Structured Value-based Planning (SVP). For the Q matrix of dimension $|\mathcal{S}| \times |\mathcal{A}|$, at each value iteration, SVP randomly updates a small portion of the $Q(s, a)$ and employs ME to reconstruct the remaining elements. We show that planning problems can greatly benefit from such a scheme, where much fewer samples (only sample around 20% of (s, a) pairs at each iteration) can achieve almost the same performance as the optimal policy.

For more advanced deep RL tasks, we extend our intuition and propose Structured Value-based Deep RL (SV-RL), applicable for deep Q-value based methods such as DQN [22]. Here, instead of the full Q matrix, SV-RL naturally focuses on the "sub-matrix", corresponding to the sampled batch of states at the current iteration. For each sampled Q matrix, we again apply ME to represent the deep Q learning target in a structured way, which poses a low rank regularization on this "sub-matrix" throughout the training process, and hence eventually the Q-network's predictions. Intuitively, as learning a deep RL policy is often noisy with high variance, if the task possesses a low-rank property, this scheme will give a clear guidance on the learning space during training, after which a better policy can be anticipated.

1.4 Thesis Structure

The remainder of this thesis is organized as follows. Chapter 2 introduces the ME-Net model and studies how one can improve adversarial robustness of DNNs using the

intrinsic global structure. Chapter 3 introduces the structured viewpoint of both classical control and modern deep RL tasks, and proposes corresponding algorithms for exploiting the low-rank structures. In both chapters, we demonstrate the effectiveness of the proposed structured solutions through rich experiments. The missing details as well as supporting results in Chapters 2 and 3 can be found in Appendices A and B, respectively. Finally, we conclude this thesis in Chapter 4 and point out some interesting future directions.

Chapter 2

ME-Net: Towards Effective Adversarial Robustness with Matrix Estimation

2.1 Problem & Motivation

State-of-the-art deep neural networks (NNs) are vulnerable to adversarial examples [10]. However, by adding small human-indistinguishable perturbation to the inputs, an adversary can fool neural networks to produce incorrect outputs with high probabilities. This phenomena raises increasing concerns for safety-critical scenarios such as the self-driving cars where NNs are widely deployed.

An increasing body of research has been aiming to either generate effective perturbations, or construct NNs that are robust enough to defend against such attacks. Currently, many effective algorithms exist to craft these adversarial examples, but defense techniques seem to be lagging behind. For instance, the state-of-the-art defense can only achieve less than 50% adversarial accuracy for ℓ_∞ perturbations on datasets such as CIFAR-10 [4]. Under recent strong attacks, most defense methods have shown to break down to nearly 0% accuracy [5].

As adversarial perturbations are carefully generated structured noise, a natural conjecture for defending against them is to destroy their structure. A naive approach for doing so would randomly mask (i.e., zero out) pixels in the image. While such

method can eliminate the adversarial structure within the noise through random information drop, it is almost certain to fail since it equally destroys the information of the original image, making NN inference even worse.

However, this naive starting point raises an interesting suggestion: instead of simply applying a random mask to the images, a preferable method should also reconstruct the images from their masked versions. In this case, the random masking destroys the crafted structures, but the reconstruction recovers the global structures that characterize the objects in the images. Images contain some global structures. An image classified as cat should have at least a cat as its main body. Humans use such global structure to classify images. In contrast the structure in adversarial perturbation is more local and defies the human eye. If both training and testing are performed under the same underlying global structures (i.e., there is no distributional shift in training and testing), the network should be generalizable and robust. If the reconstruction can successfully maintain the underlying global structure, the masking-and-reconstruction pipeline can redistribute the carefully constructed adversarial noises to non-adversarial structures.

In this work, we leverage matrix estimation (ME) as our reconstruction scheme. ME is concerned with recovering a data matrix from noisy and incomplete observations of its entries, where exact or approximate recovery of a matrix is theoretically guaranteed if the true data matrix has some *global structures* (e.g., low rank). We view a masked adversarial image as a noisy and incomplete realization of the underlying clean image, and propose ME-Net, a preprocessing-based defense that reverts a noisy incomplete image into a denoised version that maintains the underlying global structures in the clean image. ME-Net realizes adversarial robustness by using such denoised global-structure preserving representations.

We note that the ME-Net pipeline can be combined with different training procedures. In particular, we show that ME-Net can be combined with standard stochastic gradient descent (SGD) or adversarial training, and in both cases improves adversarial robustness. This is in contrast with many preprocessing techniques which cannot leverage the benefits of adversarial training [6–8], and end up failing under the recent strong white-box attack [5].

We provide extensive experimental validation of ME-Net under the strongest

black-box and white-box attacks on established benchmarks such as MNIST, CIFAR-10, SVHN, and Tiny-ImageNet, where ME-Net outperforms state-of-the-art defense techniques. Our implementation is available at: `https://github.com/YyzHarry/ME-Net`.

2.1.1 Contributions

This thesis makes the following contributions in this chapter:

- We are the first to leverage matrix estimation as a general pipeline for image classification and defending against adversarial attacks.
- We show empirically that ME-Net improves the robustness of neural networks under various ℓ_∞ attacks:
    1. ME-Net alone significantly improves the state-of-the-art results on black-box attacks;
    2. Adversarially trained ME-Net consistently outperforms the state-of-the-art defense techniques on white-box attacks, including the strong attacks that counter gradient obfuscation [5].

    Such superior performance is maintained across various datasets: CIFAR-10, MNIST, SVHN, and Tiny-ImageNet.
- We show additional benefits of ME-Net such as improving generalization (i.e., performance on clean images).

2.2 ME-Net

We first describe the motivation and high level idea underlying our design. We then provide the formal algorithm.

2.2.1 Our Design

Images contain noise: even "clean" images taken from a camera contain white noise from the environment. Such small, unstructured noise seems to be tolerable for modern deep NNs, which achieve human-level performance. However, the story is

different for carefully constructed noise. Structured, adversarial noise (i.e., adversarial examples) can easily corrupt the NN results, leading to incorrect prediction from human's perspective. This means that to achieve robustness to adversarial noise, we need to eliminate/reduce the crafted adversarial structure. Of course, while doing so, we need to maintain the intrinsic structures in the image that allow a human to make correct classifications.

We can model the problem as follows: An image is a superposition of: 1) intrinsic true structures of the data in the scene, 2) adversarial carefully-structured noise, and 3) non-adversarial noise. Our approach is first to destroy much of the crafted structure of the adversarial noise by randomly masking (zeroing out) pixels in the image. Of course, this process also increases the overall noise in the image (i.e., the non-adversarial noise) and also negatively affects the underlying intrinsic structures of the scene. Luckily however there is a well-established theory for recovering the underlying intrinsic structure of data from noisy and incomplete (i.e., masked) observations. Specifically, if we think of an image as a matrix, then we can leverage a well-founded literature on matrix estimation (ME) which allows us to recover the true data in a matrix from noisy and incomplete observations [31, 32, 35]. Further, ME provides provable guarantees of exact or approximate recovery of the true matrix if the true data has some global structures (e.g., low rank) [34, 36]. Since images naturally have global structures (e.g., an image of a cat, has a cat as a main structure), ME is guaranteed to restore the intrinsic structures of the clean image.

Another motivation for our method comes from adversarial training, where an NN is trained with adversarial examples. Adversarial training is widely adopted to increase the robustness of neural networks. However, recent theoretical work formally argues that adversarial training requires substantially more data to achieve robustness [37]. The natural question is then how to automatically obtain more data, with the purpose of creating samples that can help robustness. Our masking-then-reconstruction pipeline provides exactly one such automatic solutions. By using different random masks, we can create variations on each image, where all such variations maintain the image's underlying true global structures. We will see later in our results that this indeed provides significant gain in robustness.

2.2.2 Matrix Estimation Pipeline

Having described the intuition underlying ME-Net, we next provide a formal description of matrix estimation (ME), which constitutes the reconstruction step in our pipeline.

Matrix Estimation. Matrix estimation is concerned with recovering a data matrix from noisy and incomplete observations of its entries. Consider a true, unknown data matrix $M \in \mathbb{R}^{n\times m}$. Often, we have access to a subset Ω of entries from a noisy matrix $X \in \mathbb{R}^{n\times m}$ such that $\mathbb{E}[X] = M$. For example, in recommendation system, there are true, unknown ratings for each product from each user. One often observes a subset of noisy ratings if the user actually rates the product online. Technically, it is often assumed that each entry of X, X_{ij}, is a random variable independent of the others, which is observed with probability $p \in (0, 1]$ (i.e., missing with probability $1 - p$). The theoretical question is then formulated as finding an estimator $\hat{M}$, given noisy, incomplete observation matrix X, such that $\hat{M}$ is "close" to M. The closeness is typically measured by some matrix norm, $||\hat{M} - M||$, such as the Frobenius norm.

Over the years, extensive algorithms have been proposed. They range from simple spectral method such as universal singular value thresholding (USVT) [31], which performs SVD on the observation matrix X and discards small singular values (and corresponding singular vectors), to convex optimization based methods, which minimize the nuclear norm [32], i.e.:

$$\min_{\hat{M}\in\mathbb{R}^{n\times m}} ||\hat{M}||_* \quad \text{s.t. } \hat{M}_{ij} \approx X_{ij},\ \forall\,(i,j) \in \Omega, \tag{2.1}$$

where $||\hat{M}||_*$ is the nuclear norm of the matrix (i.e., sum of the singular values). To speed up the computation, the Soft-Impute algorithm [33] reformulates the optimization using a regularization parameter $\lambda \geq 0$:

$$\min_{\hat{M}\in\mathbb{R}^{n\times m}} \frac{1}{2} \sum_{(i,j)\in\Omega} \left(\hat{M}_{ij} - X_{ij}\right)^2 + \lambda||\hat{M}||_*. \tag{2.2}$$

In this work, we view ME as a reconstruction oracle from masked images, rather than focusing on specific algorithms.

The key message in the field of ME is: if the true data matrix M has some *global structures*, exact or approximate recovery of M can be theoretically guaranteed [31,32,

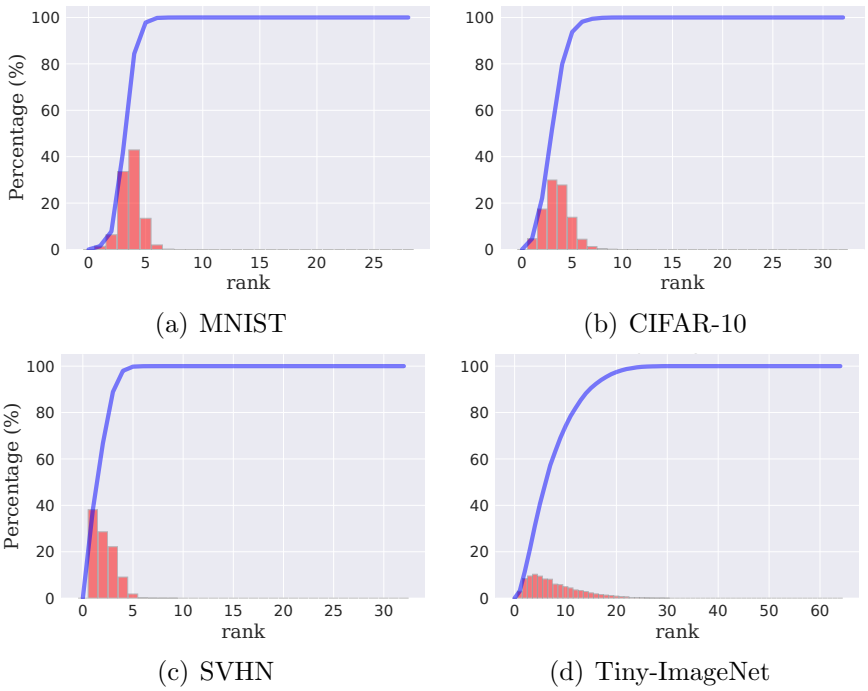


Figure 2-1: **The approximate rank of different datasets.** We plot the histogram (in red) and the empirical CDF (in blue) of the approximate rank for images in each dataset.

34]. This strong theoretical guarantee serves as the foundation for employing ME to reconstruct structures in images. In the literature, the most studied global structure is low rank. Latent variable models, where each row i and each column j are associated with some features $u_i \in \mathbb{R}^r$ and $v_j \in \mathbb{R}^r$ and $M_{ij} = f(u_i, v_j)$ for some function f, have also been investigated [31, 38]. To some extent, both could be good models for images.

Empirical Results. Before closing, we empirically show that images have strong global structures (i.e., low rank). We consider four datasets: MNIST, CIFAR-10, SVHN, and Tiny-ImageNet. We perform SVD on each image and compute its approximate rank, which is defined as the minimum number of singular values necessary to capture at least 90% of the energy in the image. Fig. 2-1 plots the histogram and the empirical CDF of the approximate ranks for each dataset. As expected, images in all datasets are relatively low rank. Specifically, the vast majority of images in MNIST, CIFAR-10, and SVHN have a rank less than 5. The rank of images in Tiny-ImageNet is larger but still significantly less than the image dimension ($\sim$10 vs. 64). This result shows that images tend to be low-rank, which implies the validity of using ME as our reconstruction oracle to find global structures.

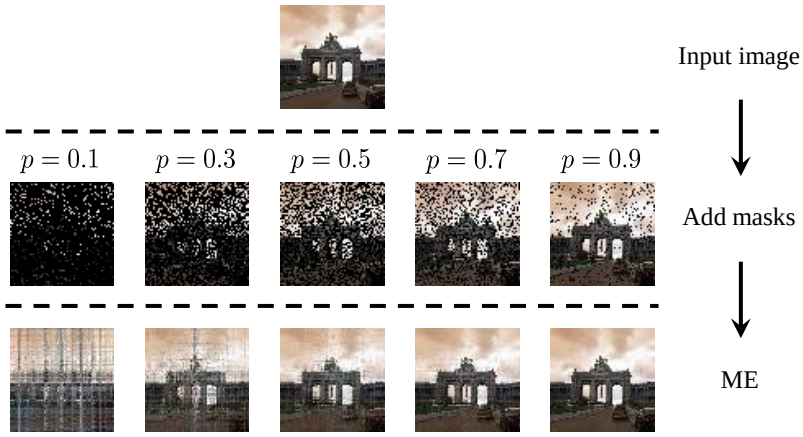


Figure 2-2: **An example of how ME affects the input images.** We apply different masks and show the reconstructed images by ME.

Next, we show in Fig. 2-2 the results of ME-based reconstruction for different masks. Evidently, the global structure (the gate in the image) has been maintained even when p, the probability of observing the true pixel, is as low as 0.3. This shows that despite random masking we should be able to reconstruct the intrinsic global image structure from the masked adversarial images. Our intuition is that humans use such underlying global structures for image classification, and if we can maintain such global structures while weakening other potentially adversarial structures, we can force both training and testing to focus on human recognizable structures and increase robustness to adversarial attacks.

2.2.3 Model

We are now ready to formally describe our technique, which we refer as ME-Net. The method is illustrated in Fig. 2-3 and summarized as follows:

- **ME-Net Training:** Define a mask as an image transform in which each pixel is preserved with probability p and set to zero with probability $1 - p$. For each training image X, we apply n masks with probabilities $\{p_1, p_2, \ldots, p_n\}$, and obtain n masked images $\{X^{(1)}, X^{(2)}, \ldots, X^{(n)}\}$. An ME algorithm is then applied to obtain reconstructed images $\{\hat{X}^{(1)}, \hat{X}^{(2)}, \ldots, \hat{X}^{(n)}\}$. We train the network on the reconstructed images $\{\hat{X}^{(1)}, \hat{X}^{(2)}, \ldots, \hat{X}^{(n)}\}$ as usual via SGD. Alternatively, adversarial training can also be readily applied in our framework.

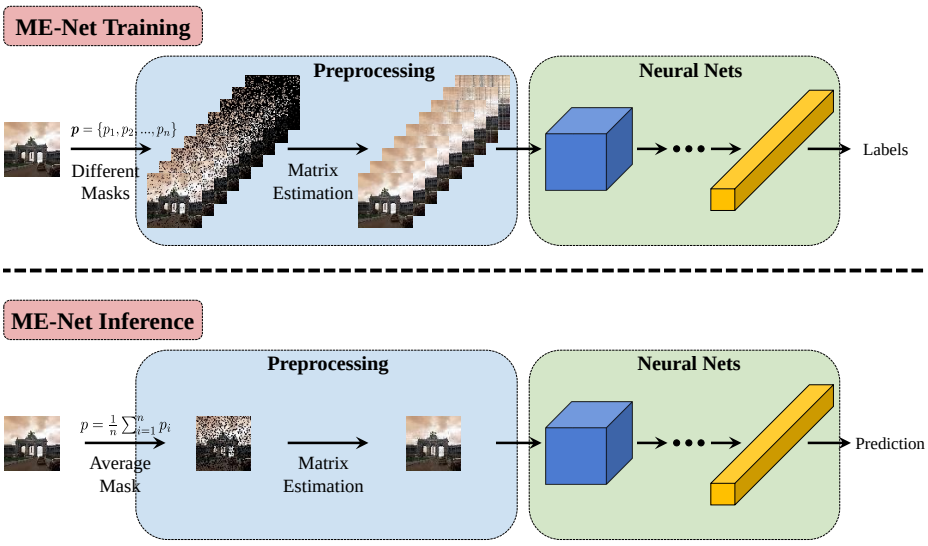


Figure 2-3: An illustration of ME-Net training and inference process.

- **ME-Net Inference:** For each test image X, we randomly sample a mask with probability $p = \frac{1}{n}\sum_{i=1}^{n} p_i$, i.e., the average of the masking probabilities during training. The masked image is then processed by the same ME algorithm used in training to obtain $\hat{X}$. Finally, $\hat{X}$ is fed to the network for prediction.

Note that we could either operate on the three RGB channels separately as independent matrices or jointly by concatenating them into one matrix. In this work, we take the latter approach as their structures are closely related. The pseudo code for ME-Net is provided in Algorithm 1. We provide additional details of ME-Net in Appendix A.1.

2.3 Evaluation

We evaluate ME-Net empirically under ℓ_∞-bounded attacks and compare it with state-of-the-art defense techniques.

Experimental Setup: We implement ME-Net as described in Section 2.2.3. During training, for each image we randomly sample 10 masks with different p values and apply matrix estimation for each masked image to construct the training set. During testing, we sample a single mask with p set to the average of the values used during training, apply the ME-Net pipeline, and test on the reconstructed image.

Algorithm 1: ME-Net training & inference

`/* ME-Net Training */`
Input: training set $S = \{(X_i, y_i)\}_{i=1}^{M}$, prescribed masking probability $\boldsymbol{p} = \{p_1, p_2, \ldots, p_n\}$, network N
for all $X_i \in S$ **do**
  Randomly sample n masks with probability $\{p_1, p_2, \ldots, p_n\}$
  Generate n masked images $\{X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(n)}\}$
  Apply ME to obtain reconstructed images $\{\hat{X}_i^{(1)}, \hat{X}_i^{(2)}, \ldots, \hat{X}_i^{(n)}\}$
  Add $\{\hat{X}_i^{(1)}, \hat{X}_i^{(2)}, \ldots, \hat{X}_i^{(n)}\}$ into new training set S'
end for
Randomly initialize network N
for number of training iterations **do**
  Sample a mini-batch $B = \{(\hat{X}_i, y_i)\}_{i=1}^{m}$ from S'
  Do one training step of network N using mini-batch B
end for

`/* ME-Net Inference */`
Input: test image X, masking probability $\boldsymbol{p} = \{p_1, p_2, \ldots, p_n\}$ used during training
Output: predicted label y
Randomly sample one mask with probability $p = \frac{1}{n}\sum_{i=1}^{n} p_i$
Generate masked image and apply ME to reconstruct $\hat{X}$
Input $\hat{X}$ to the trained network N to get the predicted label y

Unless otherwise specified, we use the Nuclear Norm minimization method [32] for matrix estimation.

We experiment with two versions of ME-Net: the first version uses standard stochastic gradient descent (SGD) to train the network, and the second version uses adversarial training, where the model is trained with adversarial examples.

For each attack type, we compare ME-Net with state-of-the-art defense techniques for the attack under consideration. For each technique, we report accuracy as the percentage of adversarial examples that are correctly classified.[1] As common in prior work [4, 6, 7], we focus on robustness against ℓ_∞-bounded attacks, and generate adversarial examples using standard methods such as the CW attack [39], Fast Gradient Sign Method (FGSM) [9], and Projected Gradient Descent (PGD) which is a more powerful adversary that performs a multi-step variant of FGSM [4].

[1] To be consistent with literature, we generate adversarial examples from the whole dataset and use all of them to report accuracy.

Organization: We first perform an extensive study on CIFAR-10 to validate the effectiveness of ME-Net against black-box and white-box attacks. We then extend the results to other datasets such as MNIST, SVHN, and Tiny-ImageNet. We also provide additional supporting results in Appendix A.2, A.3, A.4, A.5, and A.8. Additional hyper-parameter studies, such as random restarts and different number of masks, can be found in Appendix A.7, A.6 and A.9.

2.3.1 Black-box Attacks

In black-box attacks, the attacker has no access to the network model; it only observes the inputs and outputs. We evaluate ME-Net against three kinds of black-box attacks:

- **Transfer-based attack:** A copy of the victim network is trained with the same training settings. We apply CW, FGSM and PGD attacks on the copy network to generate black-box adversarial examples. We use the same attack parameters as in [4]: total perturbation ε of $8/255$ (0.031), step size of $2/255$ (0.01). For PGD attacks, we use 7, 20 and 40 steps. Note that we only consider the *strongest* transfer-based attacks, i.e., we use *white-box* attacks on the independently trained copy to generate black-box examples.

- **Decision-based attack:** We apply the newly proposed Boundary attack [40] which achieves better performance than transfer-based attacks. We apply 1000 attack steps to ensure convergence.

- **Score-based attack:** We also apply the state-of-the-art SPSA attack [41] which is strong enough to bring the accuracy of several defenses to near zero. We use a batch-size of 2048 to make the SPSA strong, and leave other hyper-parameters unchanged.

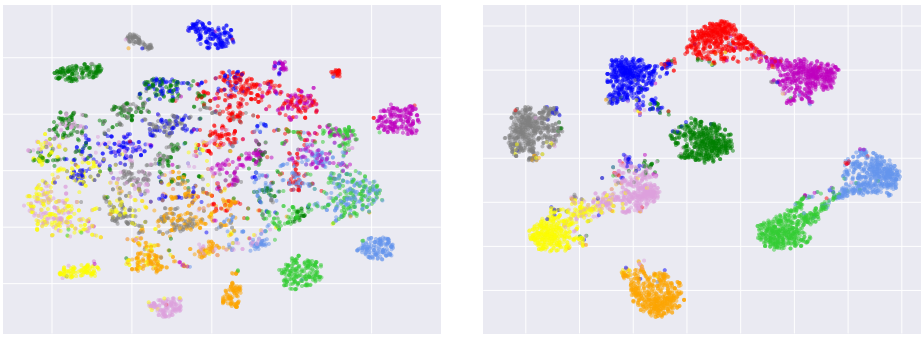
As in past work that evaluates robustness on CIFAR-10 [4, 6], we use the standard ResNet-18 model in [11]. In training ME-Net, we experiment with different settings for p. We report the results for $p \in [0.8, 1]$ below, and refer the reader to the Appendix for the results with other p values.

Since most defenses experimented only with transfer-based attacks, we first compare ME-Net to past defenses under transfer-based attacks. For comparison, we select a

Method	Training	CW	FGSM	PGD (7 steps)
Vanilla	SGD	8.9%	24.8%	7.6%
Madry	Adv. train	78.7%	67.0%	64.2%
Thermometer	SGD	–	–	53.5%
Thermometer	Adv. train	–	–	77.7%
ME-Net	SGD	**93.6%**	**92.2%**	**91.8%**

Table 2.1: **CIFAR-10 black-box results under transfer-based attacks.** We compare ME-Net with state-of-the-art defense methods under both SGD and adversarial training.

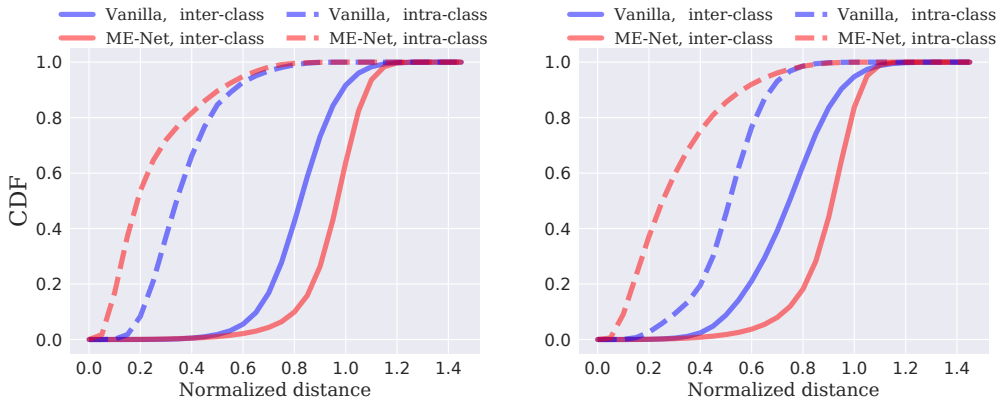
state-of-the-art adversarial training defense [4] and a preprocessing method [6]. We compare these schemes against ME-Net with standard SGD training. The results are shown in Table 2.1. They reveal that even without adversarial training, ME-Net is much more robust than prior work to black-box attacks, and can improve accuracy by 13% to 25%, depending on the attack.



(a) Vanilla under adv. attack. (b) ME-Net under adv. attack.

Figure 2-4: **Class separation under black-box adversarial attack.** The vectors right before the softmax layer are projected to a 2D plane using t-SNE [3].

To gain additional insight, we look at the separation between different classes under black-box transfer-based attack, for the vanilla network and ME-Net. Fig. 2-4(a) and 2-4(b) show the 2D projection of the vectors right before the output layer (i.e., softmax layer), for the test data in the vanilla model and ME-Net. The figures show that when the vanilla model is under attack, it loses its ability to separate different classes. In contrast, ME-Net can sustain clear separation between classes even in the presence of black-box attack.



(a) Clean data. (b) Black-box adv. attack.

Figure 2-5: **The empirical CDF of the distance within and among classes.** We quantitatively show the intra-class and inter-class distances between vanilla model and ME-Net on clean data and under black-box adversarial attacks.

Attacks	**CW**	**FGSM**	**PGD**			**Boundary**	**SPSA**
			7 steps	20 steps	40 steps		
Vanilla	8.9%	24.8%	7.6%	1.8%	0.0%	3.5%	1.4%
ME-Net	**93.6%**	**92.2%**	**91.8%**	**91.8%**	**91.3%**	**87.4%**	**93.0%**

Table 2.2: **CIFAR-10 extensive black-box results.** We show significant adversarial robustness of ME-Net under different strong black-box attacks.

To further understand this point, we compute the Euclidean distance between classes and within each class. Fig. 2-5 plots the empirical CDFs of the intra-class and inter-class distance between the vectors before the output layer, for both the vanilla classifier and ME-Net. The figure shows results for both clean data and adversarial examples. Comparing ME-Net (in red) with the vanilla classifier (in blue), we see that ME-Net both reduces the distance within each class, and improves the separation between classes; further this result applies to both clean and adversarial examples. Overall, these visualizations offer strong evidence supporting the improved robustness of ME-Net.

Finally, we also evaluate ME-Net under other strong black-box attacks. Table 2.2 summarizes these results demonstrating that ME-Net consistently achieves high robustness under different black-box attacks.

2.3.2 White-box Attacks

In white-box attacks, the attacker has full information about the neural network model (architecture and weights) and defense methods. To evaluate robustness against such white-box attacks, we use the BPDA attack proposed in [5], which has successfully circumvented a number of previously effective defenses, bringing them to near 0 accuracy. Specifically, most defense techniques rely on preprocessing methods which can cause *gradient masking* for gradient-based attacks, either because the preprocessing is not differentiable or the gradient is useless. BPDA addresses this issue by using a "differentiable approximation" for the backward pass. As such, until now no preprocessing method is effective under white-box attacks. In ME-Net, the backward pass is not differentiable, which makes BPDA the strongest white-box attack. We use PGD-based BPDA and experiment with different number of attack steps.

For white box attacks, we distinguish two cases: defenses that use only preprocessing (without adversarial training), and defenses that incorporate adversarial training. All defenses that incorporate adversarial training, including ME-Net, are trained with PGD with 7 steps.

Table 2.3 shows a comparison of the performance of various preprocessing methods against the BPDA white-box attack. We compare ME-Net with three preprocessing defenses, i.e., the PixelDefend method [7], the Thermometer method [6], and the total variation (TV) minimization method [8]. The results in the table for [6, 7] are directly taken from [5]. Since the TV minimization method is not tested on CIFAR-10, we implement this method using the same setting used with ME-Net. The table shows that preprocessing alone is vulnerable to the BPDA white-box attack, as all schemes perform poorly under such attack. Interestingly however, the table also shows that ME-Net's preprocessing is significantly more robust to BPDA than other preprocessing methods. We attribute this difference to that ME-Net's preprocessing step focuses on protecting the global structures in images.

Next we report the results of white-box attacks on schemes that use adversarial training. One key characteristic of ME-Net is its orthogonality with adversarial training. Note that many preprocessing methods propose combining adversarial training, but the combination actually performs worse than adversarial training alone [5]. Since ME-Net's preprocessing already has a decent accuracy under the strong white-box

Method	Type	Steps	Accuracy
Thermometer	Prep.	40	0.0%*
PixelDefend	Prep.	100	9.0%*
TV Minimization	Prep.	100	0.4%
ME-Net	Prep.	1000	**40.8%**

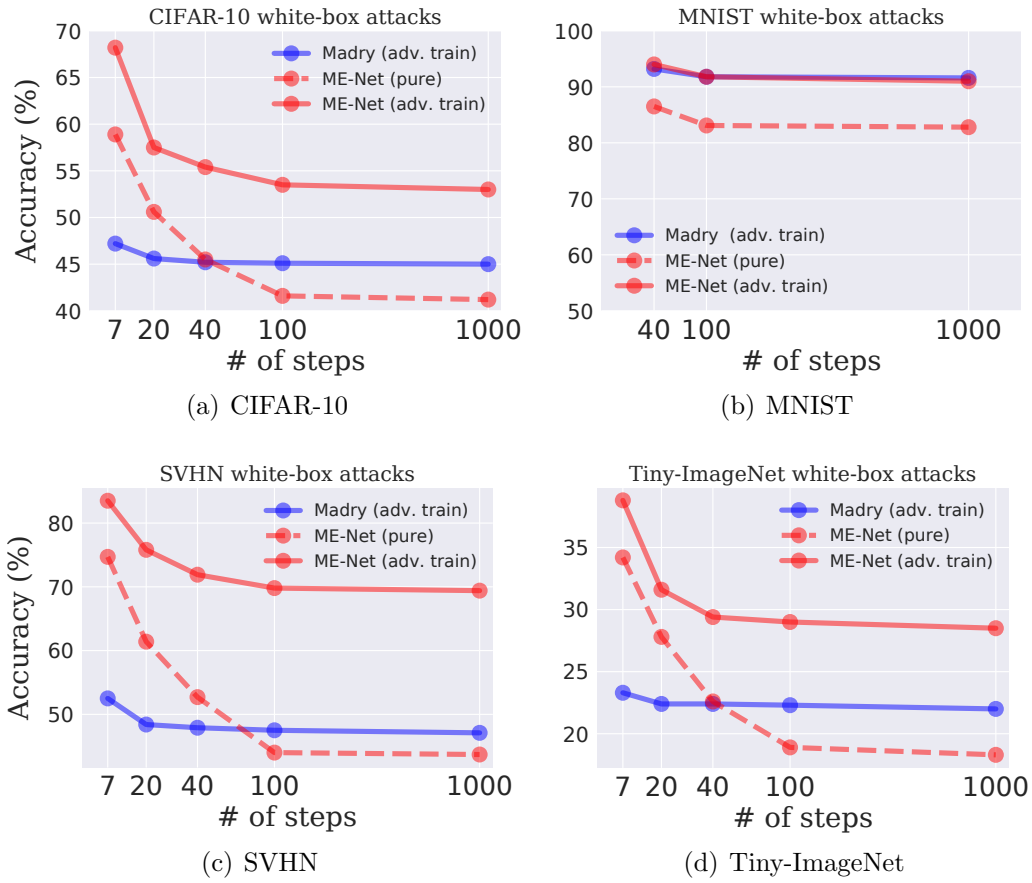
Table 2.3: **White-box attack against pure preprocessing schemes.** We use PGD or BPDA attacks in white-box setting. Compared to other pure preprocessing methods, ME-Net can increase robustness by a significant margin. *Data from [5].

Network	Method	Type	Steps	Accuracy
ResNet-18	Madry	Adv. train	1000	45.0%
	ME-Net	Prep. + Adv. train	1000	**52.8%**
WideResNet	Madry	Adv. train	1000	46.8%
	Thermometer	Prep. + Adv. train	1000	12.3%
	ME-Net	Prep. + Adv. train	1000	**55.1%**

Table 2.4: **White-box attack results for adversarial training**. We use 1000 steps PGD or BPDA attacks in white-box setting to ensure the results are convergent. ME-Net achieves state-of-the-art white-box robustness when combined with adversarial training.

attacks, we envision a further improvement when combining with adversarial training. We compare ME-Net against two baselines: we compare against [4], which is the state-of-the-art in defenses against white-box attacks. We also compare with the Thermometer technique in [6], which like ME-Net, combines a preprocessing step with adversarial training. For all compared defenses, adversarial training is done using PGD with 7 steps. We also use BPDA to approximate the gradients during the backward pass. For our comparison we use ResNet-18 and its wide version since they were used in past work on robustness with adversarial training. As for the attacker, we allow it to use the *strongest possible* attack, i.e., it uses BPDA with 1000 PGD attack steps to ensure the results are convergent. Note that previous defenses (including the state-of-the-art) only consider up to 40 steps.

Table 2.4 summarizes the results. As shown in the table, ME-Net combined with adversarial training outperforms the state-of-the-art results under white-box attacks, achieving a 52.8% accuracy with ResNet and a 55.1% accuracy with WideResNet.



(a) CIFAR-10 (b) MNIST

(c) SVHN (d) Tiny-ImageNet

Figure 2-6: **White-box attack results on different datasets**. We compare ME-Net with [4] under PGD or BPDA attack with different attack steps up to 1000. We show both the pure ME-Net without adversarial training, and ME-Net with adversarial training. For Tiny-ImageNet, we report the Top-1 adversarial robustness.

In contrast, the Thermometer method that also uses preprocessing plus adversarial training cannot survive the strong white-box adversary.

2.3.3 Evaluation with Different Datasets

We evaluate ME-Net on MNIST, SVHN, CIFAR-10, and Tiny-ImageNet and compare its performance across these datasets. For space limitations, we present only the results for the white-box attacks. We provide results for black-box attacks and additional attacks in Appendix A.2, A.3, A.4, and A.5.

For each dataset, we use the network architecture and parameters commonly used in past work on adversarial robustness to help in comparing our results to past work. For MNIST, we use the LeNet model with two convolutional layers as in [4]. We also

use the same attack parameters as total perturbation scale of 76.5/255 (0.3), and step size 2.55/255 (0.01). Besides using 40 and 100 total attack steps, we also increase to 1000 steps to further strengthen the adversary. For ME-Net with adversarial training, we follow their settings to use 40 steps PGD during training. We use standard ResNet-18 for SVHN and CIFAR-10, and DenseNet-121 for Tiny-ImageNet, and set attack parameters as follows: total perturbation of 8/255 (0.031), step size of 2/255 (0.01), and with up to 1000 total attack steps. Since in [4] the authors did not examine on SVHN and Tiny-ImageNet, we follow their methods to retrain their model on these datasets. We use 7 steps PGD for adversarial training. We keep all the training hyper-parameters the same for ME-Net and [4].

Fig. 2-6 shows the performance of ME-Net on the four datasets and compares it with [4], a state-of-the-art defense against white-box attacks. We plot both the result of a pure version of ME-Net, and ME-Net with adversarial training. The figure reveals the following results. First, it shows that ME-Net with adversarial training outperforms the state-of-the-art defense against white-box attacks. Interestingly however, the gains differ from one dataset to another. Specifically, ME-Net is comparable to [4] on MNIST, provides about 8% gain on CIFAR-10 and Tiny-ImageNet, and yields 23% gain on SVHN.

We attribute the differences in accuracy gains across datasets to differences in their properties. MNIST is too simple (single channel with small 28×28 pixels), and hence ME-Net and [4] both achieve over 90% accuracy. The other datasets are all more complex and have 3 RGB channels and bigger images. More importantly, Fig. 2-1 shows that the vast majority of images in SVHN have a very low rank, and hence very strong global structure, which is a property that ME-Net leverages to yield an accuracy gain of 23%. CIFAR-10 and Tiny-ImageNet both have relatively low rank images but not as low as SVHN. The CDF shows that 90% of the images in CIFAR have a rank lower than 5, whereas 90% of the images in Tiny-ImageNet have a rank below 10. When taking into account that the dimension of Tiny-ImageNet is twice as CIFAR (64×64 vs. 32×32), one would expect ME-Net's gain on these datasets to be comparable, which is compatible with the empirical results.

Method	Training	Steps	Approx. Input	Projected BPDA
ME-Net	Pure	1000	**41.5%**	**64.9%**
	Adversarial	1000	**62.5%**	**74.7%**

Table 2.5: **Results of ME-Net against adaptive white-box attacks on CIFAR-10.** We use 1000 steps PGD-based BPDA for the two newly proposed attacks, and report the accuracy of ME-Net.

2.3.4 Evaluation against Adaptive Attacks

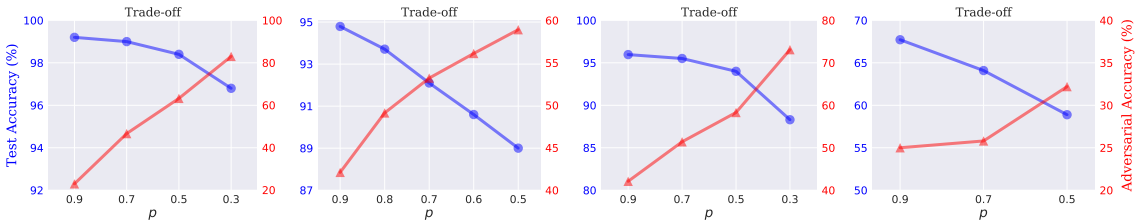
Since ME-Net provides a new preprocessing method, we examine customized attacks where the adversary takes advantage of knowing the details of ME-Net's pipeline. We propose two kinds of white-box attacks: 1) *Approximate input attack*: since ME-Net would preprocess the image, this adversary attacks not the original image, but uses the exact preprocess method to approximate/reconstruct an input, and attacks the newly constructed image using the BPDA procedure [5]. 2) *Projected BPDA attack*: since ME-Net focuses on the global structure of an image, this adversary aims to attack directly the main structural space of the image. Specifically, it uses BPDA to approximate the gradient, and then projects the gradient to the low-rank space of the image iteratively, i.e., it projects on the space constructed by the top few singular vectors of the original image, to construct the adversarial noise. Note that these two attacks are based on the BPDA white-box attack which has shown most effective against preprocessing. Table 2.5 shows the results of these attacks, which demonstrates that ME-Net is robust to these adaptive white-box attacks.

2.3.5 Adversarial Robustness vs. Generalization

In this section, we briefly discuss the trade-off between standard generalization and adversarial robustness, which can be affected by training ME-Net with different hyper-parameters. When the masks are generated with higher observing probability p, the recovered images will contain more details and are more similar to the original ones. In this case, the generalization ability will be similar to the vanilla network (or even be enhanced). However, the network will be sensible to the adversarial noises, as the adversarial structure in the noise is only destroyed a bit, and thus induces low

robustness. On the other hand, when given lower observing probability p, much of the adversarial structure in the noise will be eliminated, which can greatly increase the adversarial robustness. Nevertheless, the generalization on clean data can decrease as it becomes harder to reconstruct the images and the input images may not be similar to the original ones. In summary, there exists an inherent trade-off between standard generalization and adversarial robustness. The trade-off should be further studied to acquire a better understanding and performance of ME-Net.

We provide results of the inherent trade-off between adversarial robustness and standard generalization on different datasets. As shown in Fig. 2-7, we change the observing probability p of the masks to train different ME-Net models, and apply 7 steps white-box BPDA attack to each of them. As p decreases, the generalization ability becomes lower, while the adversarial robustness grows rapidly. We show the consistent trade-off phenomena on different datasets.



(a) MNIST (b) CIFAR-10 (c) SVHN (d) Tiny-ImageNet

Figure 2-7: **The trade-off between adversarial robustness and standard generalization on different datasets**. We use pure ME-Net during training, and apply 7 steps white-box BPDA attack for the adversarial accuracy. For Tiny-ImageNet we only report the Top-1 accuracy. The results verify the consistent trade-off across different datasets.

2.3.6 Comparison of Different ME Methods

Matrix estimation (ME) is a well studied topic with several established ME techniques. The results in the other sections are with the Nuclear Norm minimization algorithm [32]. Here we compare the performance of three ME methods: the Nuclear Norm minimization algorithm, the Soft-Impute algorithm [33], and the universal singular value thresholding (USVT) approach [31].

We train ME-Net models using different ME methods on CIFAR-10 with ResNet-

Method	Complexity	Clean	Black-box	White-box
Vanilla	–	93.4%	0.0%	0.0%
ME-Net - USVT	Low	**94.8%**	**89.4%**	**51.9%**
ME-Net - Soft-Imp.	Medium	**94.9%**	**91.3%**	**52.3%**
ME-Net - Nuc. Norm	High	**94.8%**	**91.0%**	**52.8%**

Table 2.6: **Comparisons between different ME methods.** We report the generalization and adversarial robustness of three ME-Net models using different ME methods on CIFAR-10. We apply transfer-based 40 steps PGD attack as black-box adversary, and 1000 steps PGD-based BPDA as white-box adversary.

18. We apply transfer-based PGD black-box attacks with 40 attack steps, as well as white-box BPDA attack with 1000 attack steps. We compare the complexity, generalization and adversarial robustness of these methods. More details can be found in Appendix A.6.

Table 2.6 shows the results of our comparison. The table shows that all the three ME methods are able to improve the original standard generalization, and achieve almost the same test accuracy. The nuclear norm minimization algorithm takes much longer time and more computation power. The Soft-Impute algorithm simplifies the process but still requires certain computation resources, while the USVT approach is much simpler and faster. The performance of different ME methods is slightly different, as more complex algorithms may gain better performances.

2.3.7 Improving Generalization

As a preprocessing method, ME-Net also serves as a data augmentation technique during training. We show that besides adversarial robustness, ME-Net can also improve generalization (i.e., the test accuracy) on clean data. We distinguish between two training procedures: 1) non-adversarial training, where the model is trained only with clean data, and 2) adversarial training where the model is trained with adversarial examples. For each case we compare ME-Net with the best performing model for that training type. We show results for different datasets, where each dataset is trained with the typical model in past work as stated in Section 2.3.3. Table 2.7 shows the results, which demonstrate the benefit of ME-Net as a method for improving

Method	Training	MNIST	CIFAR-10	SVHN	Tiny-ImageNet
Vanilla	Pure	98.8%	93.4%	95.0%	66.4%
ME-Net	Pure	**99.2%**	**94.9%**	**96.0%**	**67.7%**
Madry	Adversarial	98.5%	79.4%	87.4%	45.6%
ME-Net	Adversarial	**98.8%**	**85.5%**	**93.5%**	**57.0%**

Table 2.7: **Generalization performance on clean data.** For each dataset, we use the same network for all the schemes. ME-Net improves generalization for both adversarial and non-adversarial training. For Tiny-ImageNet, we report the Top-1 accuracy.

generalization under both adversarial and non-adversarial training.

2.4 Related Work

Due to the large body of work on adversarial robustness, we focus on methods that are most directly related to our work, and refer readers to the survey [42] for a more comprehensive and broad literature review.

Adversarial Training. Currently, the most effective way to defend against adversarial attacks is adversarial training, which trains the model on adversarial examples generated by different kinds of attacks [4, 9, 10, 43]. Authors of [4] showed that training on adversarial examples generated by PGD with a random start can achieve state-of-the-art performance on MNIST and CIFAR-10 under ℓ_∞ constraint. One major difficulty of adversarial training is that it tends to overfit to the adversarial examples. Authors in [37] thus demonstrated and proved that much more data is needed to achieve good generalization under adversarial training. ME-Net can leverage adversarial training for increased robustness. Further its data augmentation capability helps improving generalization.

Preprocessing. Many defenses preprocess the images with a transformation prior to classification. Typical preprocessing includes image re-scaling [44], discretization [45], thermometer encoding [6], feature squeezing [46], image quilting [8], and neural-based transformations [7,47]. These defenses can cause *gradient masking* when using gradient-based attacks. However, as shown in [5], by applying the Backward Pass Differentiable

Approximation (BPDA) attacks designed for obfuscated gradients, the accuracy of all of these methods can be brought to near zero. ME-Net is the first preprocessing method that remains effective under the strongest BPDA attack, which could be attributed to its ability to leverage adversarial training.

Matrix Estimation. Matrix estimation recovers a data matrix from noisy and incomplete samples of its entries. A classical application is recommendation systems, such as the Netflix problem [48], but it also has richer connections to other learning challenges such as graphon estimation [38, 49], community detection [50, 51], and recently even in deep reinforcement learning [2]. Many efficient algorithms exist such as the universal singular value thresholding approach [31], the convex nuclear norm minimization formulation [32] and even non-convex methods [52–54]. The key promise is that as long as there are some structures underlying the data matrix, such as being low-rank, then exact or approximate recovery can be guaranteed. As such, ME is an ideal reconstruction scheme for recovering global structures.

2.5 Summary & Discussion

In this chapter, we introduced ME-Net, which leverages matrix estimation to improve the robustness to adversarial attacks. Extensive experiments under strong black-box and white-box attacks demonstrated the significance of ME-Net, where it consistently improves the state-of-the-art robustness in different benchmark datasets. Furthermore, ME-Net can easily be embedded into existing networks, and can also bring additional benefits such as improving standard generalization.

Chapter 3

Harnessing Structures for Value-Based Planning and Reinforcement Learning

3.1 Problem & Motivation

Value-based methods are widely used in control, planning, and reinforcement learning [16, 18, 22]. To solve a Markov Decision Process (MDP), one common method is value iteration, which finds the optimal value function. This process can be done by iteratively computing and updating the state-action value function, represented by $Q(s, a)$ (i.e., the Q-value function). In simple cases with small state and action spaces, value iteration can be ideal for efficient and accurate planning. However, for modern MDPs, the data that encodes the value function usually lies in thousands or millions of dimensions [16, 17], including images in deep reinforcement learning [22, 29]. These practical constraints significantly hamper the efficiency and applicability of the vanilla value iteration.

Yet, the Q-value function is intrinsically induced by the underlying system dynamics. These dynamics are likely to possess some structured forms in various settings, such as being governed by partial differential equations. In addition, states and actions may also contain latent features (e.g., similar states could have similar optimal actions). Thus, it is reasonable to expect the structured dynamic to impose a *structure* on the

Q-value. Since the Q function can be treated as a giant matrix, with rows as states and columns as actions, a structured Q function naturally translates to a *structured* Q matrix.

In this work, we explore the *low-rank* structures. To check whether low-rank Q matrices are common, we examine the benchmark Atari games, as well as 4 classical stochastic control tasks. As we demonstrate in Sections 3.3 and 3.4, more than 40 out of 57 Atari games and all 4 control tasks exhibit low-rank Q matrices. This leads us to a natural question: How do we leverage the low-rank structure in Q matrices to allow value-based techniques to achieve better performance on "low-rank" tasks?

We propose a generic framework that allows for exploiting the low-rank structure in both classical planning and modern deep RL. Our scheme leverages Matrix Estimation (ME), a theoretically guaranteed framework for recovering low-rank matrices from noisy or incomplete measurements [34]. In particular, for classical control tasks, we propose Structured Value-based Planning (SVP). For the Q matrix of dimension $|\mathcal{S}| \times |\mathcal{A}|$, at each value iteration, SVP randomly updates a small portion of the $Q(s, a)$ and employs ME to reconstruct the remaining elements. We show that planning problems can greatly benefit from such a scheme, where fewer samples (only sample around 20% of (s, a) pairs at each iteration) can achieve almost the same performance as the optimal policy.

For more advanced deep RL tasks, we extend our intuition and propose Structured Value-based Deep RL (SV-RL), applicable for deep Q-value based methods such as DQN [22]. Here, instead of the full Q matrix, SV-RL naturally focuses on the "sub-matrix", corresponding to the sampled batch of states at the current iteration. For each sampled Q matrix, we again apply ME to represent the deep Q learning target in a structured way, which poses a low rank regularization on this "sub-matrix" throughout the training process, and hence eventually the Q-network's predictions. Intuitively, as learning a deep RL policy is often noisy with high variance, if the task possesses a low-rank property, this scheme will give a clear guidance on the learning space during training, after which a better policy can be anticipated. We confirm that SV-RL indeed can improve the performance of various value-based methods on "low-rank" Atari games: SV-RL consistently achieves higher scores on those games. Interestingly, for complex, "high-rank" games, SV-RL performs comparably. ME

naturally seeks solutions that balance low rank and a small reconstruction error (cf. Section 3.3.1). Such a balance on reconstruction error helps to maintain or only slightly degrade the performance for "high-rank" situation[1].

3.1.1 Contributions

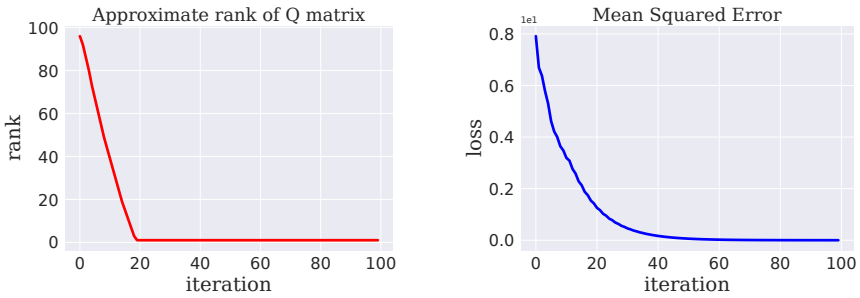
This thesis makes the following contributions in this chapter:

- We are the first to propose a framework that leverages matrix estimation as a general scheme to exploit the low-rank structures, from planning to deep reinforcement learning.
- We demonstrate the effectiveness of our approach on classical stochastic control tasks, where the low-rank structure allows for efficient planning with less computation.
- We extend our scheme to deep RL, which is naturally applicable for value-based techniques. Across a variety of methods, such as DQN, double DQN, and dueling DQN, experimental results on all Atari games show that SV-RL can consistently improve the performance of value-based methods, achieving higher scores for tasks when low-rank structures are confirmed to exist.

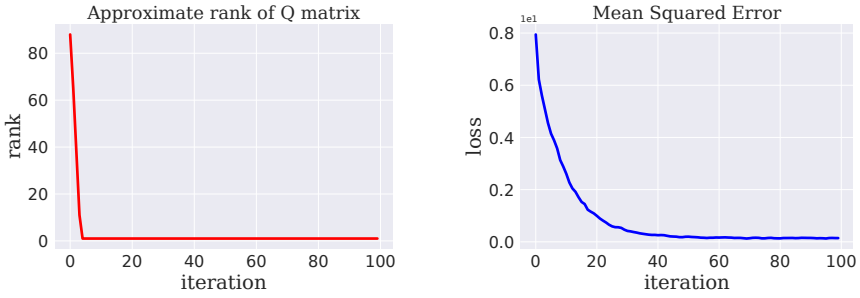
3.2 Warm-up: A Toy Example

To motivate our method, let us first investigate a toy example which helps to understand the structure within the Q-value function. We consider a simple deterministic MDP, with 1000 states, 100 actions and a deterministic state transition for each action. The reward $r(s, a)$ is randomly generated first for each (s, a) pair, and then fixed throughout. A discount factor $\gamma = 0.95$ is used. The deterministic nature imposes a strong relationship among connected states. In this case, our goal is to explore: (1) what kind of structures the Q function may contain; and (2) how to effectively exploit such structures.

[1]Code is available at: `https://github.com/YyzHarry/SV-RL`



(a) Vanilla value iteration (b) Vanilla value iteration



(c) Online reconstruction (d) Online reconstruction

Figure 3-1: The approximate rank and MSE of $Q^{(t)}$ during value iteration. (a) & (b) use vanilla value iteration; (c) & (d) use online reconstruction with only 50% observed data each iteration.

The Low-rank Structure Under this setting, Q-value could be viewed as a 1000×100 matrix. To probe the structure of the Q-value function, we perform the standard Q-value iteration as follows:

$$Q^{(t+1)}(s,a) = \sum_{s' \in \mathcal{S}} P(s'|s,a)\big[r(s,a) + \gamma \max_{a' \in \mathcal{A}} Q^{(t)}(s',a')\big], \quad \forall\, (s,a) \in \mathcal{S} \times \mathcal{A}, \tag{3.1}$$

where s' denotes the next state after taking action a at state s. We randomly initialize $Q^{(0)}$. In Fig. 3-1, we show the approximate rank of $Q^{(t)}$ and the mean-square error (MSE) between $Q^{(t)}$ and the optimal Q^*, during each value iteration. Here, the approximate rank is defined as the first k singular values (denoted by σ) that capture more than 99% variance of all singular values, i.e., $\sum_{i=1}^{k} \sigma_i^2 / \sum_j \sigma_j^2 \geq 0.99$. As illustrated in Fig. 3-1(a) and 3-1(b), the standard theory guarantees the convergence to Q^*; more interestingly, the converged Q^* is of low rank, and the approximate rank of $Q^{(t)}$ drops fast. These observations give a strong evidence for the intrinsic low dimensionality of this toy MDP. Naturally, an algorithm that leverages such structures

would be much desired.

Efficient Planning via Online Reconstruction with Matrix Estimation The previous results motivate us to exploit the structure in value function for efficient planning. The idea is simple:

If the eventual matrix is low-rank,
why not enforcing such a structure throughout the iterations?

In other words, with the existence of a global structure, we should be capable of exploiting it during intermediate updates and possibly also regularizing the results to be in the same low-rank space. In particular, at each iteration, instead of every (s, a) pair (i.e., Eq. (3.1)), we would like to only calculate $Q^{(t+1)}$ for *some* (s, a) pairs and then *exploit* the low-rank structure to recover the whole $Q^{(t+1)}$ matrix. We choose matrix estimation (ME) as our reconstruction oracle. The reconstructed matrix is often with low rank, and hence *regularizing* the Q matrix to be low-rank as well. We validate this framework in Fig. 3-1(c) and 3-1(d), where for each iteration, we only randomly sample 50% of the (s, a) pairs, calculate their corresponding $Q^{(t+1)}$ and reconstruct the whole $Q^{(t+1)}$ matrix with ME. As we enforce the low-rank structure at each iteration, the approximate rank reduces much faster as expected. Clearly, around 40 iterations, we obtain comparable results to the vanilla value iteration. Importantly, this comparable performance only needs to directly compute 50% of the whole Q matrix at each iteration. It is not hard to see that in general, each vanilla value iteration incurs a computation cost of $O(|\mathcal{S}|^2|\mathcal{A}|^2)$. The complexity of our method however only scales as $O(p|\mathcal{S}|^2|\mathcal{A}|^2) + O_{ME}$, where p is the percentage of pairs we sample and O_{ME} is the complexity of ME. In general, many ME methods employ SVD as a subroutine, whose complexity is bounded by $O(\min\{|\mathcal{S}|^2|\mathcal{A}|, |\mathcal{S}||\mathcal{A}|^2\})$ [55]. For low-rank matrices, faster methods can have a complexity of order linear in the dimensions [33]. In other words, our approach improves computational efficiency, especially for modern high-dimensional applications. This overall framework thus appears to be a successful technique: it exploits the low-rank behavior effectively and efficiently when the underlying task indeed possesses such a structure.

3.3 Structured Value-based Planning

Having developed the intuition underlying our methodology, we next provide a formal description in Sections 3.3.1 and 3.3.2. One natural question is whether such structures and our method are general in more realistic control tasks. Towards this end, we provide further empirical support in Section 3.3.3.

3.3.1 Matrix Estimation

ME considers about recovering a full data matrix, based on potentially incomplete and noisy observations of its elements. Formally, consider an unknown data matrix $X \in \mathbb{R}^{n\times m}$ and a set of observed entries Ω. If the observations are incomplete, it is often assumed that each entry of X is observed independently with probability $p \in (0, 1]$. In addition, the observation could be noisy, where the noise is assumed to be mean zero. Given such an observed set Ω, the goal of ME is to produce an estimator $\hat{M}$ so that $||\hat{M} - X|| \approx 0$, under an appropriate matrix norm such as the Frobenius norm.

The algorithms in this field are rich. In the past years, there has been a tremendous effort in advancing the theories and algorithms of matrix estimation within the community. Theoretically, the essential message is: exact or approximate recovery of the data matrix X is guaranteed if X contains some global structure [31, 32, 34]. In the literature, most attention has been focusing on the *low-rank* structure of a matrix. Correspondingly, there are many provable, practical algorithms to achieve the desired recovery. Early convex optimization methods [32] seek to minimize the nuclear norm, $||\hat{M}||_*$, of the estimator. For example, fast algorithms, such as the Soft-Impute algorithm [33] solves the following minimization problem:

$$\min_{\hat{M}\in\mathbb{R}^{n\times m}} \frac{1}{2} \sum_{(i,j)\in\Omega} \left(\hat{M}_{ij} - X_{ij}\right)^2 + \lambda||\hat{M}||_*. \tag{3.2}$$

Since the nuclear norm $||\cdot||_*$ is a convex relaxation of the rank, the convex optimization approaches favor solutions that are with small reconstruction errors and in the meantime being relatively low-rank, which are desirable for our applications. Apart from convex optimization, there are also spectral methods and even non-convex

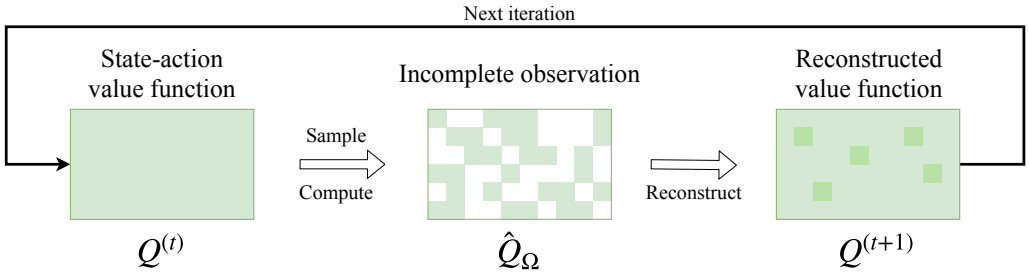


Figure 3-2: An illustration of the proposed SVP algorithm for leveraging low-rank structures.

optimization approaches [31, 53, 54]. In this thesis, we view ME as a principled reconstruction oracle to effectively exploit the low-rank structure. For faster computation, we mainly employ the Soft-Impute algorithm.

3.3.2 Our Approach: Structured Value-based Planning

We now formally describe our approach, which we refer as structured value-based planning (SVP). Fig. 3-2 illustrates our overall approach for solving MDP with a known model. The approach is based on the Q-value iteration. At the t-th iteration, instead of a full pass over all state-action pairs:

1. SVP first randomly selects a subset Ω of the state-action pairs. In particular, each state-action pair in $\mathcal{S} \times \mathcal{A}$ is observed (i.e., included in Ω) independently with probability p.

2. For each selected (s, a), the intermediate $\hat{Q}(s, a)$ is computed based on the Q-value iteration:

$$\hat{Q}(s,a) \leftarrow \sum_{s'} P(s'|s,a) \left(r(s,a) + \gamma \max_{a'} Q^{(t)}(s',a') \right), \quad \forall\, (s,a) \in \Omega.$$

3. The current iteration then ends by reconstructing the full Q matrix with matrix estimation, from the set of observations in Ω. That is, $Q^{(t+1)} = \mathrm{ME}\big(\{\hat{Q}(s,a)\}_{(s,a)\in\Omega}\big)$.

Overall, each iteration reduces the computation cost by roughly $1-p$ (cf. discussions in Section 3.2). In Appendix B.1, we provide the pseudo-code and additionally, a

short discussion on the technical difficulty for theoretical analysis. Nevertheless, we believe that the consistent empirical benefits, as will be demonstrated, offer a sound foundation for future analysis.

3.3.3 Empirical Evaluation on Stochastic Control Tasks

We empirically evaluate our approach on several classical stochastic control tasks, including the Inverted Pendulum, the Mountain Car, the Double Integrator, and the Cart-Pole. Our objective is to demonstrate, as in the toy example, that if the optimal Q^* has a low-rank structure, then the proposed SVP algorithm should be able to exploit the structure for efficient planning. We present the evaluation on Inverted Pendulum, and leave additional results on other planning tasks in Appendix B.2 and B.3.

Inverted Pendulum In this classical continuous task, our goal is to balance an inverted pendulum to its upright position, and the performance metric is the average angular deviation. The dynamics is described by the angle and the angular speed, i.e., $s = (\theta, \dot{\theta})$, and the action a is the torque applied. A reward function that penalizes control effort while favoring an upright pendulum is used. We discretize the task to have 2500 states and 1000 actions, leading to a 2500×1000 Q-value matrix. We follow [56] to handle the policy of continuous states by modelling their transitions using multi-linear interpolation. For different discretization scales, we provide further results in Appendix B.6.1.

The Low-rank Structure We first verify that the optimal Q^* indeed contains the desired low-rank structure. We run the vanilla value iteration until it converges. The converged Q matrix is found to have an approximate rank of 7. For further evidence, in Appendix B.3, we construct "low-rank" policies directly from the converged Q matrix, and show that the policies maintain the desired performance.

The SVP Policy Having verified the structure, we would expect our approach to be effective. To this end, we apply SVP with different observation probability p and fix the overall number of iterations to be the same as the vanilla Q-value iteration. Fig. 3-3 confirms the success of our approach. Fig. 3-3(a), 3-3(b) and 3-3(c) show the comparison between optimal policy and the final policy based on SVP. We further illustrate the performance metric, the average angular deviation, in Fig. 3-3(d).

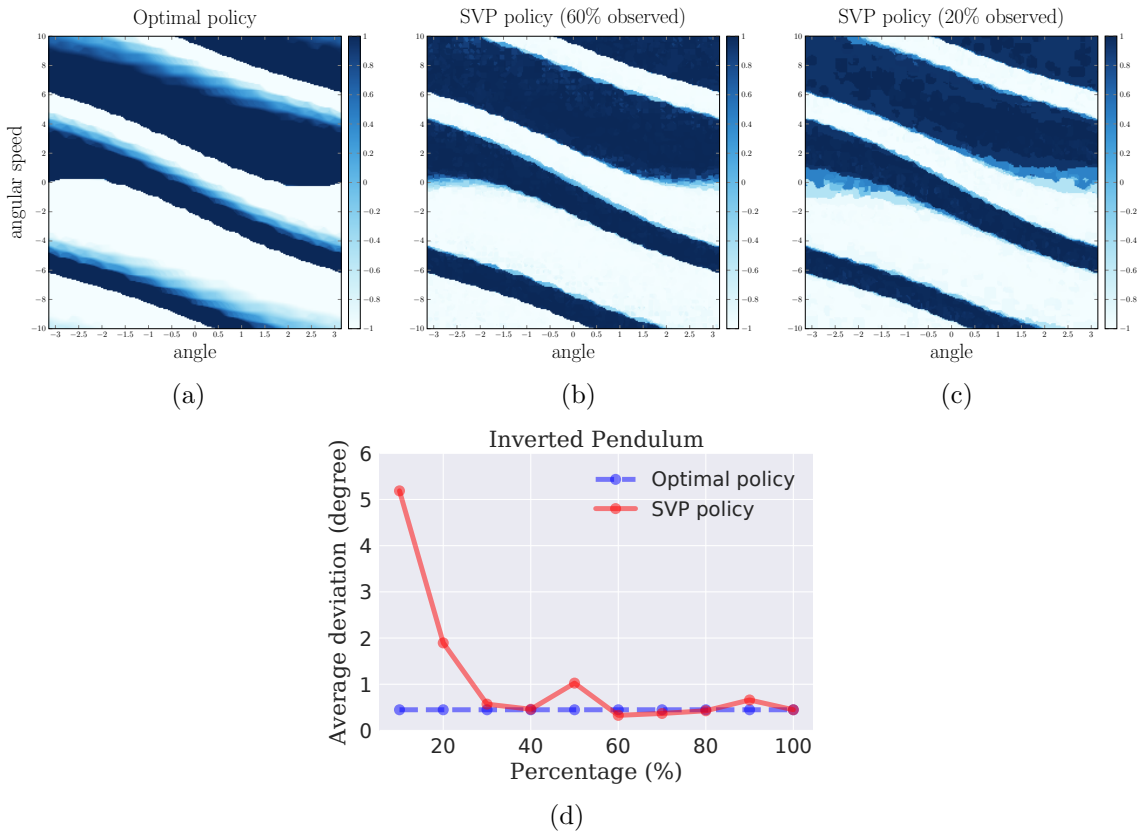


Figure 3-3: Performance comparison between optimal policy and the proposed SVP policy.

Overall, much fewer samples are needed for SVP to achieve a comparable performance to the optimal one.

3.4 Structured Value-based Deep Reinforcement Learning

So far, our focus has been on tabular MDPs where value iteration can be applied straightforwardly. However, the idea of exploiting structure is much more powerful: we propose a natural extension of our approach to deep RL. Our scheme again intends to exploit and regularize structures in the Q-value function with ME. As such, it can be seamlessly incorporated into value-based RL techniques that include a Q-network. We demonstrate this on Atari games, across various value-based RL techniques.

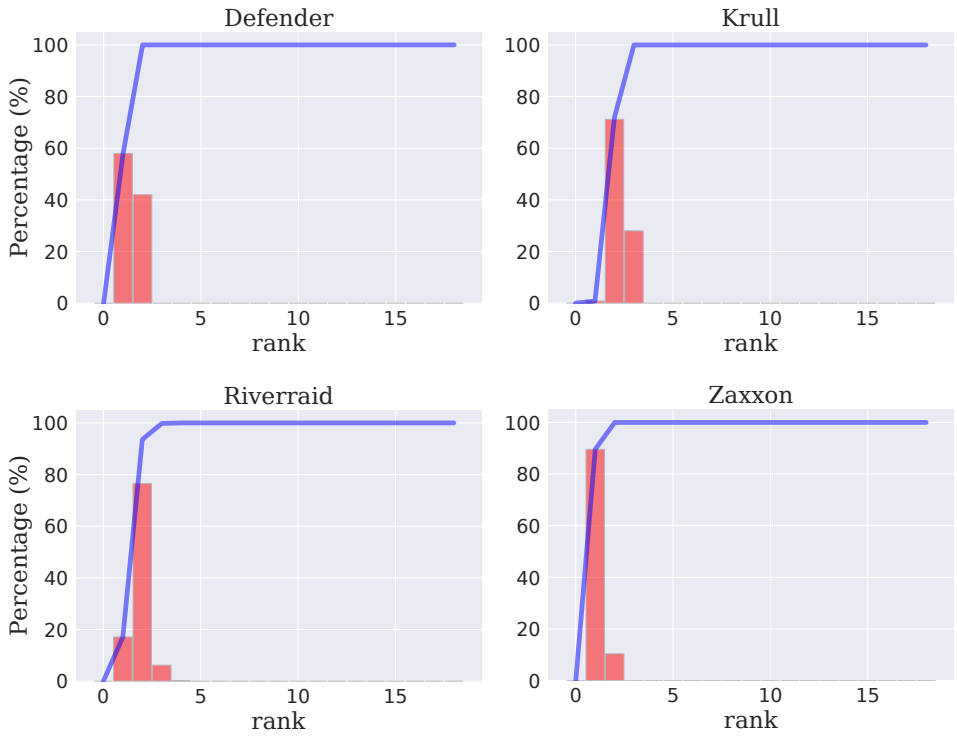
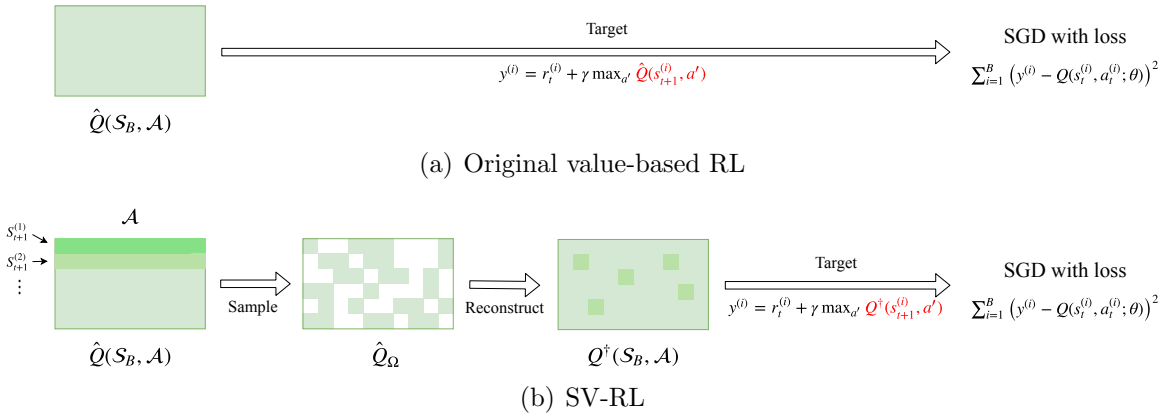


Figure 3-4: Approximate rank of different Atari games: histogram (red) and empirical CDF (blue) of the approximate rank of 10,000 randomly sampled data batch for the trained DQN.

3.4.1 Evidence of Structured Q-value Function

Before diving into deep RL, let us step back and review the process we took to develop our intuition. Previously, we start by treating the Q-value as a matrix. To exploit the structure, we first verify that certain MDPs have essentially a low-rank Q^*. We argue that if this is indeed the case, then enforcing the low-rank structures throughout the iterations, by leveraging ME, should lead to better algorithms.

A naive extension of the above reasoning to deep RL immediately fails. In particular, with images as states, the state space is effectively infinitely large, leading to a tall Q matrix with numerous number of rows (states). Verifying the low-rank structure for deep RL hence seems intractable. However, by definition, if a large matrix is low-rank, then almost any row is a linear combination of some other rows. That is, if we sample a small batch of the rows, the resulting matrix is most likely low-rank as well. To probe the structure of the deep Q function, it is, therefore, natural to understand the rank of a randomly sampled batch of states. In deep RL, our target for exploring



(a) Original value-based RL

(b) SV-RL

Figure 3-5: An illustration of the proposed SV-RL scheme, compared to the original value-based RL.

structures is no longer the optimal Q^*, which is never available. In fact, like SVP, the natural objective should be the converged values of the underlying algorithm, which in "deep" scenarios, are the eventually learned Q function.

With the above discussions, we now provide evidence for the low-rank structure of learned Q function on some Atari games. We train standard DQN on 4 games, with a batch size of 32. To be consistent, the 4 games all have 18 actions. After the training process, we randomly sample a batch of 32 states, evaluate with the learned Q network and finally synthesize them into a matrix. That is, a 32×18 data matrix with rows the batch of states, the columns the actions, and the entries the values from the learned Q network. Note that the rank of such a matrix is at most 18. The above process is repeated for 10,000 times, and the histogram and empirical CDF of the approximate rank is plotted in Fig. 3-4. Apparently, there is a strong evidence supporting a highly structured low-rank Q function for those games – the approximate ranks are uniformly small; in most cases, they are around or smaller than 3.

3.4.2 Our Approach: Structured Value-based RL

Having demonstrated the low-rank structure within some deep RL tasks, we naturally seek approaches that exploit the structure during the training process. We extend the same intuitions here: if eventually, the learned Q function is of low rank, then enforcing/regularizing the low rank structure for each iteration of the learning process should similarly lead to efficient learning and potentially better performance. In deep

RL, each iteration of the learning process is naturally the SGD step where one would update the Q network. Correspondingly, this suggests us to harness the structure within the batch of states. Following our previous success, we leverage ME to achieve this task.

We now formally describe our approach, referred as structured value-based RL (SV-RL). It exploits the structure for the sampled batch at each SGD step, and can be easily incorporated into any Q-value based RL methods that update the Q network via a similar step as in Q-learning. In particular, Q-value based methods have a common model update step via SGD, and we only exploit structure of the sampled batch at this step – the other details pertained to each specific method are left intact.

Precisely, when updating the model via SGD, Q-value based methods first sample a batch of B transitions, $\{(s_t^{(i)}, r_t^{(i)}, a_t^{(i)}, s_{t+1}^{(i)})\}_{i=1}^B$, and form the following updating targets:

$$y^{(i)} = r_t^{(i)} + \gamma \max_{a'} \hat{Q}(s_{t+1}^{(i)}, a'). \tag{3.3}$$

For example, in DQN, $\hat{Q}$ is the target network. The Q network is then updated by taking a gradient step for the loss function $\sum_{i=1}^B \left(y^{(i)} - Q(s_t^{(i)}, a_t^{(i)}; \theta)\right)^2$, with respect to the parameter θ.

To exploit the structure, we then consider reconstructing a matrix $Q^\dagger$ from $\hat{Q}$, via ME. The reconstructed $Q^\dagger$ will replace the role of $\hat{Q}$ in Eq. (3.3) to form the targets $y^{(i)}$ for the gradient step. In particular, the matrix $Q^\dagger$ has a dimension of $B \times |\mathcal{A}|$, where the rows represent the "next states" $\{s_{t+1}^{(i)}\}_{i=1}^B$ in the batch, the columns represent actions, and the entries are reconstructed values. Let $\mathcal{S}_B \triangleq \{s_{t+1}^{(i)}\}_{i=1}^B$. The SV-RL alters the SGD update step as illustrated in Algorithm 2 and Fig. 3-5.

Note the resemblance of the above procedure to that of SVP in Section 3.3.2. When the full Q matrix is available, in Section 3.3.2, we sub-sample the Q matrix and then reconstruct the entire matrix. When only a subset of the states (i.e., the batch) is available, naturally, we look at the corresponding sub-matrix of the entire Q matrix, and seek to exploit its structure.

Algorithm 2: Structured Value-based RL (SV-RL)

follow the chosen value-based RL method (e.g., DQN) as usual.
except that for model updates with gradient descent, **do**
    `/* exploit structure via matrix estimation*/`
    sample a set Ω of state-action pairs from $\mathcal{S}_B \times \mathcal{A}$. In particular, each state-action pair in $\mathcal{S}_B \times \mathcal{A}$ is observed (i.e., included in Ω) with probability p, independently.
    evaluate every state-action pair in Ω via $\hat{Q}$, where $\hat{Q}$ is the network that would be used to form the targets $\{y^{(i)}\}_{i=1}^{B}$ in the original value-based methods (cf. Eq. (3.3)).
    based on the evaluated values, reconstruct a matrix $Q^\dagger$ with ME, i.e.,

$$Q^\dagger = \mathrm{ME}\big(\{\hat{Q}(s,a)\}_{(s,a)\in\Omega}\big).$$

    `/* new targets with reconstructed` $Q^\dagger$ `for the gradient step*/`
    replace $\hat{Q}$ in Eq. (3.3) with $Q^\dagger$ to evaluate the targets $\{y^{(i)}\}_{i=1}^{B}$, i.e.,

$$\text{SV-RL Targets:} \quad y^{(i)} = r_t^{(i)} + \gamma \max_{a'} Q^\dagger(s_{t+1}^{(i)}, a'). \tag{3.4}$$

    update the Q network with the original targets replaced by the SV-RL targets.

3.4.3 Empirical Evaluation with Various Value-based Methods

Experimental Setup We conduct extensive experiments on Atari 2600. We apply SV-RL on three representative value-based RL techniques, i.e., DQN, double DQN and dueling DQN. We fix the total number of training iterations and set all the hyper-parameters to be the same. For each experiment, averaged results across multiple runs are reported. Additional details are provided in Appendix B.4.

Consistent Benefits for "Structured" Games We present representative results of SV-RL applied to the three value-based deep RL techniques in Fig. 3-6. These games are verified by method mentioned in Section 3.4.1 to be low-rank. Additional results on all Atari games are provided in Appendix B.5. The figure reveals the following results. First, games that possess structures indeed benefit from our approach, earning mean rewards that are strictly higher than the vanilla algorithms across time. More importantly, we observe consistent improvements across different value-based RL

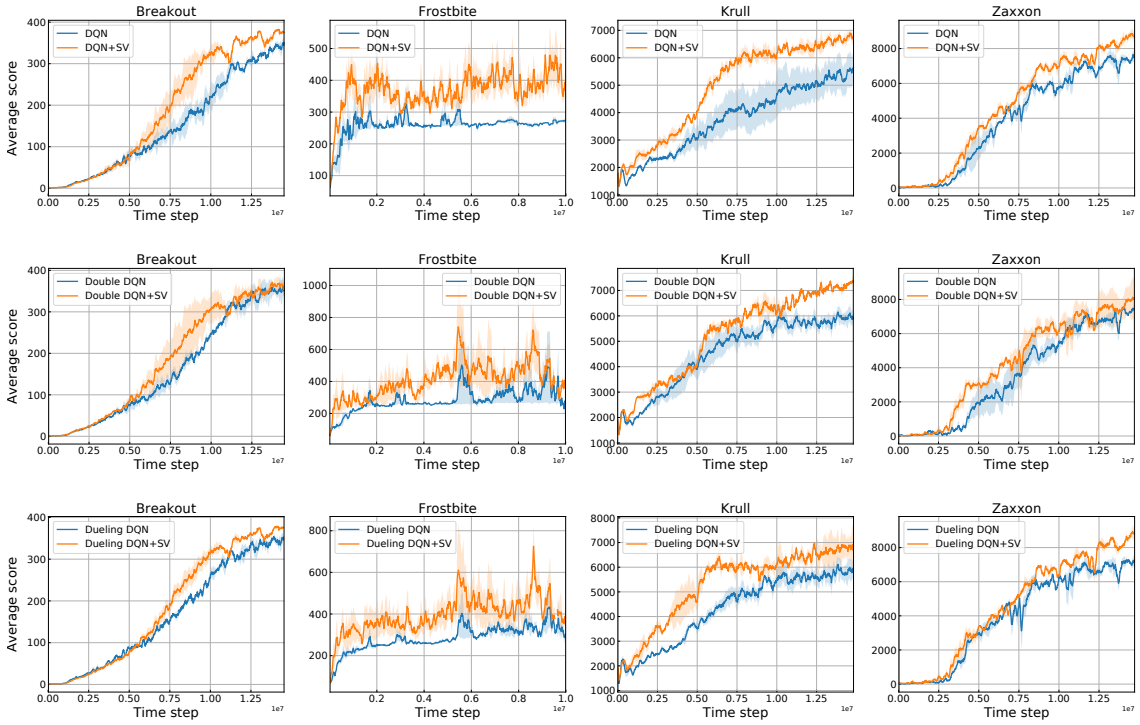


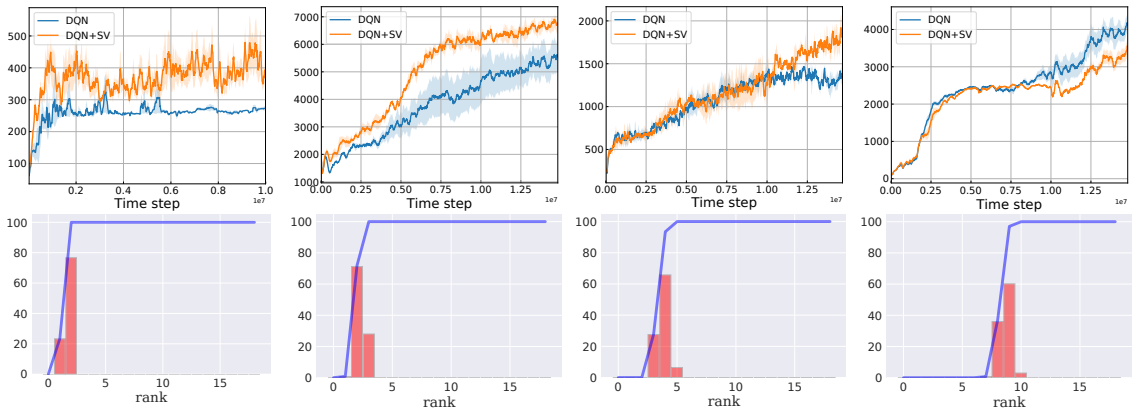
Figure 3-6: Results of SV-RL on various value-based deep RL techniques. **First row:** results on DQN. **Second row:** results on double DQN. **Third row:** results on dueling DQN.

techniques. This highlights the important role of the intrinsic structures, which are independent of the specific techniques, and justifies the effectiveness of our approach in consistently exploiting such structures.

Further Observations Interestingly however, the performance gains vary from games to games. Specifically, the majority of the games can have benefits, with few games performing similarly or slightly worse. Such observation motivates us to further diagnose SV-RL in the next section.

3.5 Diagnose and Interpret Performance in Deep RL

So far, we have demonstrated that games which possess structured Q-value functions can consistently benefit from SV-RL. Obviously however, not all tasks in deep RL would possess such structures. As such, we seek to diagnose and further interpret our



(a) Frostbite (better) (b) Krull (better) (c) Alien (slightly better) (d) Seaquest (worse)

Figure 3-7: Interpretation of deep RL results. We plot games where the SV-based method performs differently. More structured games (with lower rank) can achieve better performance with SV-RL.

approach at scale.

Diagnosis We select 4 representative examples (with 18 actions) from all tested games, in which SV-RL performs better on two tasks (i.e., FROSTBITE and KRULL), slightly better on one task (i.e., ALIEN), and slightly worse on the other (i.e., SEAQUEST). The intuitions we developed in Section 3.4 incentivize us to further check the approximate rank of each game. As shown in Fig. 3-7, in the two better cases, both games are verified to be approximately low-rank (~ 2), while the approximate rank in ALIEN is moderately high (~ 5), and even higher in SEAQUEST (~ 10). Note that all these games have an action space of size 18, thus the rank is at most 18.

Consistent Interpretations As our approach is designed to exploit structures, we would expect to attribute the differences in performance across games to the "strength" of their structured properties. Games with strong low-rank structures tend to have larger improvements with SV-RL (Fig. 3-7(a) and 3-7(b)), while moderate approximate rank tends to induce small improvements (Fig. 3-7(c)), and high approximate rank may induce similar or slightly worse performances (Fig. 3-7(d)). The empirical results are well aligned with our arguments: if the Q-function for the task contains low-rank structure, SV-RL can exploit such structure for better efficiency and performance; if not, SV-RL may introduce slight or no improvements over the vanilla algorithms. As mentioned, the ME solutions balance being low rank and having small reconstruction error, which helps to ensure a reasonable or only slightly degraded performance, even

for "high rank" games. We further observe consistent results on ranks *vs.* improvement across different games and RL techniques in Appendix B.5, verifying our arguments.

3.6 Related Work

Structures in Value Function Recent work in the control community starts to explore the structure of value function in control/planning tasks [15, 16, 18, 57]. These work focuses on decomposing the value function and subsequently operating on the reduced-order space. In spirit, we also explore the low-rank structure in value function. The central difference is that instead of decomposition, we focus on "completion". We seek to efficiently operate on the original space by looking at few elements and then leveraging the structure to infer the rest, which allows us to extend our approach to modern deep RL. In addition, while there are few attempts for basis function learning in high dimensional RL [58], functions are hard to generate in many cases and approaches based on basis functions typically do not get the same performance as DQN, and do not generalize well. In contrast, we provide a principled and systematic method, which can be applied to any framework that employs value-based methods or sub-modules. Finally, we remark that at a higher level, there are studies exploiting different structures within a task to design effective algorithms, such as exploring the low-dimensional system dynamics in predictive state representation [59] or exploring the so called low "Bellman rank" property [60].

Value-based Deep RL Deep RL has emerged as a promising technique, highlighted by key successes such as solving Atari games via deep Q-learning [22] and combining with Monte Carlo tree search [61, 62] to achieve superhuman performance in mastering Go, Chess and Shogi [19, 20]. Methods based on learning value functions are fundamental components in deep RL, exemplified by the deep Q-network [21, 22]. Over the years, there has been a large body of literature on its variants, such as double DQN [63], dueling DQN [28], IQN [64] and other techniques that aim to improve exploration [65, 66]. Our approach focuses on general value-based RL methods, rather than specific algorithms. As long as the method has a similar model update step as in Q-learning, our approach can leverage the structure to help with the task. We empirically demonstrate that deep RL tasks that have structured value functions

indeed benefit from our scheme.

Matrix Estimation ME is the primary tool we leverage to exploit the low-rank structure in value functions. Early work in the field is primarily motivated by recommendation systems. Since then, the techniques have been widely studied and applied to different domains [38, 51], and recently even in robust deep learning [1]. The field is relatively mature, with extensive algorithms and provable recovery guarantees for structured matrix [34, 36]. Because of the strong promise, we view ME as a principled reconstruction oracle to exploit the low-rank structures within matrices.

3.7 Summary & Discussion

In this chapter, we investigated the structures in value function, and proposed a complete framework to understand, validate, and leverage such structures in various tasks, from planning to deep reinforcement learning. The proposed SVP and SV-RL algorithms harness the strong low-rank structures in the Q function, showing consistent benefits for both planning tasks and value-based deep reinforcement learning techniques. Extensive experiments validated the significance of the proposed schemes, which can be easily embedded into existing planning and RL frameworks for further improvements.

Chapter 4

Conclusions and Future Work

Deep learning algorithms have recently been shown a great success, and adopted by many vision, speech and language applications. However, many real-world problems can exhibit intrinsic structure within tasks. The overall objective of this thesis is to study the intrinsic and meaningful structures that naturally arise in deep learning tasks, and propose corresponding algorithms to improve the performance by leveraging such structured properties. At a higher level, this investigation is motivated by the underlying system dynamics within learning tasks that intrinsically induces the learning properties, and the possibility to improve the learning performance given the structured representations. To this end, we have taken a close look and investigated two important deep learning algorithms, the *adversarial robustness* and the *value-based planning and deep reinforcement learning*. First, we study the problem as how to enhance the robustness of deep neural networks against adversarial attacks, through a structured (matrix) view of real-world images. Second, we investigate the problem as how to boost the efficiency and performance of value-based stochastic control and deep RL algorithms, when highly structured Q-value functions have been demonstrated across tasks.

In this thesis, we try to tackle these challenges by leveraging the Matrix Estimation (ME) technique. For each specific deep learning problem, we first verify the existence of strong *low-rank* structures in the learning objectives (e.g., images or Q-value functions). Motivated by the theoretical guarantees and appealing results in the ME field, we integrate ME into a generic framework for each learning task, with the aim to harness

the strong low-rank structures through the learning process. First, we propose ME-Net, an adversarial defense method that leverages ME to enforce the global structure in the images. ME-Net preprocesses images using a mask-and-reconstructed via ME, which destroys the adversarial structure of the noise while re-enforcing the global structure in the original images. We conduct comprehensive experiments on prevailing benchmarks such as MNIST, CIFAR-10, SVHN, and Tiny-ImageNet. We show that ME-Net consistently outperforms prior techniques when comparing ME-Net with state-of-the-art defense mechanisms, improving robustness against both black-box and white-box attacks. Second, we propose SVP and SV-RL, applicable for any value-based classical stochastic control and deep RL tasks, respectively. We demonstrate the effectiveness of SVP on several planning problems, where the low-rank structure allows for efficient planning with less computation. Furthermore, we extend our scheme to deep RL, which is naturally applicable for value-based techniques such as DQN, double DQN, and dueling DQN. Experimental results on all Atari games verify that SV-RL can consistently improve the performance of value-based methods, achieving higher scores for tasks when low-rank structures are confirmed to exist.

This thesis opens the door for several interesting extensions. For adversarial robustness, a meaningful line of variations can be considering more the intrinsic structures within images (or adversarial noises) when designing defense methods. Despite we focusing on the low-rank structures, other nice properties of natural images can also be explored, such as the frequency information. Analogously, in the context of control and deep RL tasks, we have exploited the low-rank structures in value-based methods (specifically, the Q-value matrix). What about policy-based algorithms? Can similar structured frameworks be derived also for various techniques that directly optimize policies? Overall, if the intrinsic structure does exist in RL problems, such desired pipeline for policy-based methods should be anticipated.

In addition, this thesis extends the applications of matrix estimation, and potentially serves as the first attempt on incorporating ME methods with deep learning framework. For matrix estimation / matrix completion community, over the years, the vast majority of applications have been focusing on the classical settings, such as recommendation systems, graphon estimation, community detection, time series, and so on. The results of this thesis connect the nice theory with modern applications,

e.g., deep neural networks. An interesting line of future work in ME applications can be the extension for other problems with (highly) structured properties that emerge in the deep learning era.

For future work on the application side of deep learning, our results demonstrate the possibility of leveraging theory-inspired approaches in deep learning. Throughout the thesis, we mainly focus on the low-rank structure; yet, there should be many other meaningful structures that naturally evolve in different learning tasks. Such new framework as combining theory models with deep learning algorithms can bring new insights for the audience and broadly benefit practitioners. Major challenges as well as opportunities mainly lie in the context of proper theory-inspired models that can be easily incorporated into the modern deep learning pipelines and frameworks for real-world applications. That is, for each specific problem, the potential of using well-studied theory models for deriving more intuitive and interpretable deep learning algorithms.

Appendix A

Supplementary Materials for Chapter 2

A.1 Training Details

Training settings. We summarize our training hyper-parameters in Table A.1. We follow the standard data augmentation scheme as in [11] to do zero-padding with 4 pixels on each side, and then random crop back to the original image size. We then randomly flip the images horizontally and normalize them into $[0, 1]$. Note that ME-Net's preprocessing is performed before the training process as in Algorithm 1.

Dataset	Model	Data Aug.	Optimizer	Momentum	Epochs	LR	LR decay
CIFAR-10	ResNet-18 Wide-ResNet	$\surd$	SGD	0.9	200	0.1	step (100, 150)
MNIST	LeNet	$\times$	SGD	0.9	200	0.01	step (100, 150)
SVHN	ResNet-18	$\surd$	SGD	0.9	200	0.01	step (100, 150)
Tiny-ImageNet	DenseNet-121	$\surd$	SGD	0.9	90	0.1	step (30, 60)

Table A.1: **Training details of ME-Net on different datasets.** Learning rate is decreased at selected epochs with a step factor of 0.1.

ME-Net details. As was mentioned in Section 2.2.3, one could either operate on the three RGB channels separately as independent matrices or jointly by concatenating them into one wide matrix. For the former approach, given an image, we can apply the same mask to each channel and then separately run ME to recover the matrix. For the latter approach, the RGB channels are first concatenated along the column

dimension to produce a wide matrix, i.e., if each channel is of size 32×32, then the concatenated matrix, [RGB], is of size 32×96. A mask is applied to the wide matrix and the whole matrix is then recovered. This approach is a common, simple method for estimating tensor data. Since this work focuses on structures of the image and channels within an image are closely related, we adopt the latter approach in this thesis.

In our experiments, we use the following method to generate masks with different observing probability: for each image, we select n masks in total with observing probability p ranging from $a \to b$. We use $n = 10$ for most experiments. To provide an example, "$p : 0.6 \to 0.8$" indicates that we select 10 masks in total with observing probability from 0.6 to 0.8 with an equal interval of 0.02, i.e., $0.6, 0.62, 0.64, \ldots$. Note that we only use this simple selection scheme for mask generation. We believe further improvement can be achieved with better designed selection schemes, potentially tailored to each image.

A.2 Additional Results on CIFAR-10

A.2.1 Black-box Attacks

We provide additional results of ME-Net against different black-box attacks on CIFAR-10. We first show the complete results using different kinds of black-box attacks, i.e., transfer-based (FGSM, PGD, CW), decision-based (Boundary) and score-based (SPSA) attacks. For CW attack, we follow the settings in [4] to use different confidence values κ. We report ME-Net results with different training settings on Table A.2. Here we use pure ME-Net as a preprocessing method without adversarial training. As shown, previous defenses only consider limited kinds of black-box attacks. We by contrast show extensive and also advanced experimental results.

Further, we define and apply another stronger black-box attack, where we provide the architecture and weights of our trained model to the black-box adversary to make it stronger. This kind of attack is also referred as "semi-black-box" or "gray-box" attack in some instances, while we still view it as a black-box one. This time the adversary is not aware of the preprocessing layer but has full access to the trained

Method		Clean	FGSM	PGD			CW		Boundary	SPSA
				7 steps	20 steps	40 steps	$\kappa = 20$	$\kappa = 50$		
Vanilla		93.4%	24.8%	7.6%	1.8%	0.0%	9.3%	8.9%	3.5%	1.4%
Madry		79.4%	67.0%	64.2%	–	–	78.7%	–	–	–
Thermometer		87.5%	–	77.7%	–	–	–	–	–	–
ME-Net	$p : 0.8 \to 1$	**94.9%**	**92.2%**	**91.8%**	**91.8%**	**91.3%**	**93.6%**	**93.6%**	**87.4%**	**93.0%**
	$p : 0.6 \to 0.8$	92.1%	85.1%	84.5%	83.4%	81.8%	89.2%	89.0%	81.8%	90.9%
	$p : 0.4 \to 0.6$	89.2%	75.7%	74.9%	73.0%	70.9%	82.0%	82.0%	77.5%	87.1%

Table A.2: **CIFAR-10 extensive black-box attack results.** Different kinds of strong black-box attacks are used, including transfer-, decision-, and score-based attacks.

network, and directly performs white-box attacks to the network. We show the results in Table A.3.

Method		FGSM	PGD			CW	
			7 steps	20 steps	40 steps	$\kappa = 20$	$\kappa = 50$
ME-Net	$p : 0.8 \to 1$	**85.1%**	**84.9%**	**84.0%**	**82.9%**	75.8%	75.2%
	$p : 0.6 \to 0.8$	83.2%	82.8%	81.7%	79.6%	81.5%	76.8%
	$p : 0.4 \to 0.6$	80.5%	80.2%	79.2%	76.4%	**84.0%**	**77.1%**

Table A.3: **CIFAR-10 additional black-box attack results where adversary has limited access to the trained network.** We provide the architecture and weights of our trained model to the black-box adversary to make it stronger.

A.2.2 White-box Attacks

Pure ME-Net

We first show the extensive white-box attack results with pure ME-Net in Table A.4. We use strongest white-box BPDA attack [5] with different attack steps. We select three preprocessing methods [6–8] as competitors. We re-implement the total variation minimization approach [8] and apply the same training settings as ME-Net on CIFAR-10. The experiments are performed under total perturbation ε of 8/255 (0.031). By comparison, ME-Net is demonstrated to be the first preprocessing method that is effective under strongest white-box attacks.

Further, we study the performance of ME-Net under different ε in Fig. A-1. Besides

Method		Type	Attack Steps			
			7	**20**	**40**	**100**
Vanilla		–	0.0%	0.0%	0.0%	0.0%
Thermometer		Prep.	–	–	0.0%*	0.0%*
PixelDefend		Prep.	–	–	–	9.0%*
TV Minimization		Prep.	14.7%	5.1%	2.7%	0.4%
ME-Net	$p : 0.8 \rightarrow 1$	Prep.	46.2%	33.2%	26.8%	23.5%
	$p : 0.7 \rightarrow 0.9$	Prep.	50.3%	40.4%	33.7%	29.5%
	$p : 0.6 \rightarrow 0.8$	Prep.	53.0%	45.6%	37.8%	35.1%
	$p : 0.5 \rightarrow 0.7$	Prep.	55.7%	47.3%	38.6%	35.9%
	$p : 0.4 \rightarrow 0.6$	Prep.	**59.8%**	**52.6%**	**45.5%**	**41.6%**

Table A.4: **CIFAR-10 extensive white-box attack results with pure ME-Net.** We use the strongest PGD or BPDA attacks in white-box setting with different attack steps. We compare ME-Net with other pure preprocessing methods [6–8]. We show that ME-Net is the first preprocessing method to be effective under white-box attacks. *Data from [5].

using $\varepsilon = 8$ which is commonly used in CIFAR-10 attack settings [4], we additionally provide more results including $\varepsilon = 2$ and 4 to study the performance of pure ME-Net under strongest BPDA white-box attacks.

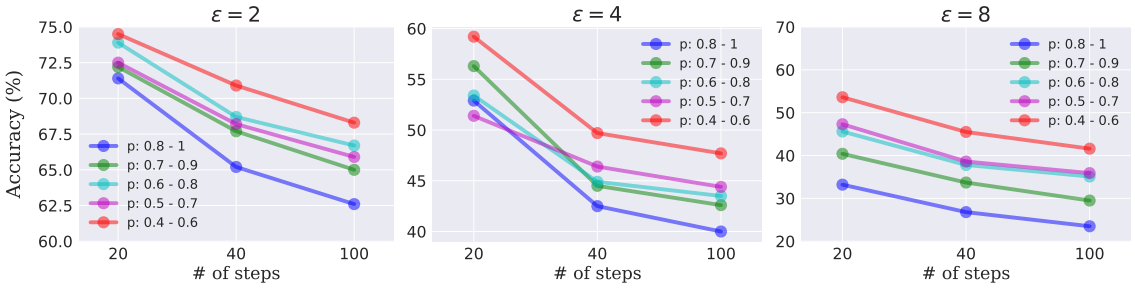


Figure A-1: **CIFAR-10 white-box attack results of pure ME-Net with different perturbation** ε. We report ME-Net results with different training settings under various attack steps.

Besides the strongest BPDA attack, we also design and apply another white-box attack to further study the effect of the preprocessing layer. We assume the adversary is aware of the preprocessing layer, but not use the backward gradient

approximation. Instead, it performs iterative attacks only for the network part after the preprocessing layer. This attack helps study how the preprocessing affects the network robustness against white-box adversary. The results in Table A.5 shows that pure ME-Net provides sufficient robustness if the white-box adversary does not attack the preprocessing layer.

Method		FGSM	PGD			CW	
			7 steps	20 steps	40 steps	$\kappa = 20$	$\kappa = 50$
ME-Net	$p : 0.8 \to 1$	**84.3%**	**83.7%**	**83.1%**	**82.5%**	77.0%	75.9%
	$p : 0.6 \to 0.8$	82.6%	82.1%	81.5%	80.3%	76.9%	76.4%
	$p : 0.4 \to 0.6$	79.1%	79.0%	78.3%	77.4%	**77.5%**	**77.2%**

Table A.5: **CIFAR-10 additional white-box attack results where the white-box adversary does not attack the preprocessing layer.** We remain the same attack setups as in the white-box BPDA attack, while only attacking the network part after the preprocessing layer of ME-Net.

Combining with Adversarial Training

We provide more advanced and extensive results of ME-Net when combining with adversarial training in Table A.6. As shown, preprocessing methods are not necessarily compatible with adversarial training, as they can perform worse than adversarial training alone [6]. Compared to current state-of-the-art [4], ME-Net achieves consistently better results under strongest white-box attacks.

A.3 Additional Results on MNIST

A.3.1 Black-box Attacks

In Table A.7, we report extensive results of ME-Net under different strong black-box attacks on MNIST. We follow [4] to use the same LeNet model and the same attack parameters with a total perturbation scale of 76.5/255 (0.3). We use a step size of 2.55/255 (0.01) for PGD attacks. We use the same settings as in CIFAR-10 for Boundary and SPSA attacks (i.e., 1000 steps for Boundary attack, and a batch size of 2048 for SPSA attack) to make them stronger. Note that we only use the *strongest*

Network	Method	Type	Clean	Attack Steps				
				7	**20**	**40**	**100**	**1000**
ResNet-18	Madry	Adv. train	79.4%	47.2%	45.6%	45.2%	45.1%	45.0%
	ME-Net $p : 0.8 \to 1$	Prep. + Adv. train	**85.5%**	57.4%	51.5%	49.3%	48.1%	47.4%
	ME-Net $p : 0.6 \to 0.8$	Prep. + Adv. train	84.8%	62.1%	53.0%	51.2%	50.0%	49.6%
	ME-Net $p : 0.4 \to 0.6$	Prep. + Adv. train	84.0%	**68.2%**	**57.5%**	**55.4%**	**53.5%**	**52.8%**
Wide-ResNet	Madry	Adv. train	87.3%	50.0%	47.1%	47.0%	46.9%	46.8%
	Thermometer	Prep. + Adv. train	89.9%	59.4%	34.9%	26.0%	18.4%	12.3%
	ME-Net $p : 0.6 \to 0.8$	Prep. + Adv. train	**91.0%**	69.7%	58.0%	54.9%	53.4%	52.9%
	ME-Net $p : 0.4 \to 0.6$	Prep. + Adv. train	88.7%	**74.1%**	**61.6%**	**57.4%**	**55.9%**	**55.1%**

Table A.6: **CIFAR-10 extensive white-box attack results.** We apply up to 1000 steps PGD or BPDA attacks in white-box setting to ensure the results are convergent. We use the released models in [4, 5] but change the attack steps up to 1000 for comparison. ME-Net shows significant advanced results by consistently outperforming the current state-of-the-art defense method [4].

transfer-based attacks, i.e., we use *white-box* attacks on the independently trained copy to generate black-box examples. As shown, ME-Net shows significantly more effective results against different strongest black-box attacks.

Method		Clean	FGSM	PGD		CW		Boundary	SPSA
				40 steps	100 steps	$\kappa = 20$	$\kappa = 50$		
Vanilla		98.8%	28.2%	0.1%	0.0%	14.1%	12.6%	3.7%	6.2%
Madry		98.5%	**96.8%**	**96.0%**	**95.7%**	96.4%	97.0%	–	–
Thermometer		99.0%	–	41.1%	–	–	–	–	–
ME-Net	$p : 0.8 \to 1$	**99.2%**	77.4%	73.9%	73.6%	**98.8%**	**98.7%**	**89.3%**	**98.1%**
	$p : 0.6 \to 0.8$	99.0%	87.1%	85.1%	84.9%	98.6%	98.4%	88.6%	97.5%
	$p : 0.4 \to 0.6$	98.4%	91.1%	90.7%	88.9%	98.4%	98.3%	88.0%	97.0%
	$p : 0.2 \to 0.4$	96.8%	**93.2%**	**92.8%**	**92.2%**	96.6%	96.5%	88.1%	96.1%

Table A.7: **MNIST extensive black-box attack results.** Different kinds of strong black-box attacks are used, including transfer-, decision-, and score-based attacks.

We further provide the architecture and weights of our trained model to the black-box adversary to make it stronger, and provide the results in Table A.8. As shown, ME-Net can still maintain high adversarial robustness against stronger black-box adversary under this setting.

Method		FGSM	PGD		CW	
			40 steps	100 steps	$\kappa = 20$	$\kappa = 50$
ME-Net	$p : 0.8 \to 1$	93.0%	91.9%	85.5%	**98.8%**	**98.7%**
	$p : 0.6 \to 0.8$	95.0%	94.2%	93.7%	98.3%	98.2%
	$p : 0.4 \to 0.6$	**96.2%**	**95.9%**	**95.3%**	98.3%	98.0%
	$p : 0.2 \to 0.4$	94.5%	94.2%	93.4%	96.5%	96.5%

Table A.8: **MNIST additional black-box attack results where adversary has limited access to the trained network.** We provide the architecture and weights of our trained model to the black-box adversary to make it stronger.

A.3.2 White-box Attacks

Table A.9 shows the extensive white-box attack results on MNIST. As discussed, we follow [4] to use 40 steps PGD during training when combining ME-Net with adversarial training. We apply up to 1000 steps strong BPDA-based PGD attack to ensure the results are convergent. For the competitor, we use the released model in [4], but change the attack steps to 1000 for comparison.

Method		Type	Clean	Attack Steps		
				40	**100**	**1000**
Madry		Adv. train	98.5%	93.2%	91.8%	**91.6%**
ME-Net	$p : 0.8 \to 1$	Prep.	**99.2%**	22.9%	21.8%	18.9%
	$p : 0.6 \to 0.8$	Prep.	99.0%	47.6%	42.4%	40.8%
	$p : 0.4 \to 0.6$	Prep.	98.4%	65.2%	62.1%	60.6%
	$p : 0.2 \to 0.4$	Prep.	96.8%	**86.5%**	**83.1%**	**82.6%**
ME-Net	$p : 0.8 \to 1$	Prep. + Adv. train	97.6%	87.8%	81.7%	78.0%
	$p : 0.6 \to 0.8$	Prep. + Adv. train	97.7%	90.5%	88.1%	86.5%
	$p : 0.4 \to 0.6$	Prep. + Adv. train	**98.8%**	92.1%	89.4%	88.2%
	$p : 0.2 \to 0.4$	Prep. + Adv. train	97.4%	**94.0%**	**91.8%**	**91.0%**

Table A.9: **MNIST extensive white-box attack results.** We apply up to 1000 steps PGD or BPDA attacks in white-box setting to ensure the results are convergent. We use the released models in [4] but change the attack steps up to 1000 for comparison. We show both pure ME-Net results and the results when combining with adversarial training.

A.4 Additional Results on SVHN

A.4.1 Black-box Attacks

Table A.10 shows extensive black-box attack results of ME-Net on SVHN. We use standard ResNet-18 as the network, and use a total perturbation of $\varepsilon = 8/255$ (0.031). We use the same strong black-box attacks as previously used (i.e., transfer-, decision-, and score-based attacks), and follow the same attack settings and parameters. As there are few results on SVHN dataset, we compare only with the vanilla model which uses the same network and training process as ME-Net. As shown, ME-Net provides significant adversarial robustness against various black-box attacks.

Method		Clean	FGSM	PGD			CW		Boundary	SPSA
				7 steps	20 steps	40 steps	$\kappa = 20$	$\kappa = 50$		
Vanilla		95.0%	31.2%	8.5%	1.8%	0.0%	20.4%	7.6%	4.5%	3.7%
ME-Net	$p : 0.8 \to 1$	**96.0%**	**91.8%**	**91.1%**	**90.9%**	**89.8%**	**95.5%**	**95.2%**	79.2%	**95.5%**
	$p : 0.6 \to 0.8$	95.5%	88.9%	88.7%	86.4%	86.2%	95.1%	94.9%	80.6%	94.6%
	$p : 0.4 \to 0.6$	94.0%	87.0%	86.4%	85.8%	84.4%	93.6%	93.4%	**85.3%**	93.8%
	$p : 0.2 \to 0.4$	88.3%	80.7%	76.4%	75.3%	74.2%	87.4%	87.4%	83.3%	87.6%

Table A.10: **SVHN extensive black-box attack results.** Different kinds of strong black-box attacks are used, including transfer-, decision-, and score-based attacks.

Again, we strengthen the black-box adversary by providing the network architecture and weights of our trained model. We then apply various attacks and report the results in Table A.11. ME-Net can still maintain high adversarial robustness under this setting.

Method		FGSM	PGD			CW	
			7 steps	20 steps	40 steps	$\kappa = 20$	$\kappa = 50$
ME-Net	$p : 0.8 \to 1$	83.8%	83.3%	81.3%	78.6%	**95.2%**	**95.0%**
	$p : 0.6 \to 0.8$	85.8%	85.7%	84.0%	82.1%	94.9%	94.8%
	$p : 0.4 \to 0.6$	**88.8%**	**88.6%**	**87.4%**	**86.8%**	93.5%	93.3%
	$p : 0.2 \to 0.4$	86.6%	86.3%	85.7%	85.5%	88.2%	88.2%

Table A.11: **SVHN additional black-box attack results where adversary has limited access to the trained network.** We provide the architecture and weights of our trained model to the black-box adversary to make it stronger.

A.4.2 White-box Attacks

For white-box attacks, we set attack parameters the same as in CIFAR-10, and use strongest white-box BPDA attack with different attack steps (up to 1000 for convergence). We show results of both pure ME-Net and adversarially trained one. We use 7 steps for adversarial training. Since in [4] the authors did not provide results on SVHN, we follow their methods to retrain a model. The training process and hyper-parameters are kept identical to ME-Net.

Table A.12 shows the extensive results under white-box attacks. ME-Net achieves significant adversarial robustness against the strongest white-box adversary, as it can consistently outperform [4] by a certain margin.

Method		**Type**	**Clean**	**Attack Steps**				
				7	**20**	**40**	**100**	**1000**
Madry		Adv. train	87.4%	52.5%	48.4%	47.9%	47.5%	47.1%
ME-Net	$p: 0.8 \to 1$	Prep.	**96.0%**	42.1%	27.2%	14.2%	8.0%	7.2%
	$p: 0.6 \to 0.8$	Prep.	95.5%	52.4%	39.6%	28.2%	17.1%	15.9%
	$p: 0.4 \to 0.6$	Prep.	94.0%	60.3%	48.7%	40.1%	27.4%	25.8%
	$p: 0.2 \to 0.4$	Prep.	88.3%	**74.7%**	**61.4%**	**52.7%**	**44.0%**	**43.4%**
ME-Net	$p: 0.8 \to 1$	Prep. + Adv. train	**93.5%**	62.2%	41.4%	37.5%	35.5%	34.3%
	$p: 0.6 \to 0.8$	Prep. + Adv. train	92.6%	72.1%	57.1%	49.6%	47.8%	46.5%
	$p: 0.4 \to 0.6$	Prep. + Adv. train	91.2%	79.9%	69.1%	64.2%	62.3%	61.7%
	$p: 0.2 \to 0.4$	Prep. + Adv. train	87.6%	**83.5%**	**75.8%**	**71.9%**	**69.8%**	**69.4%**

Table A.12: **SVHN extensive white-box attack results.** We apply up to 1000 steps PGD or BPDA attacks in white-box setting to ensure the results are convergent. We show results of both pure ME-Net and adversarially trained ones. ME-Net shows significantly better results as it consistently outperforms [4] by a certain margin.

A.5 Additional Results on Tiny-ImageNet

In this section, we extend our experiments to evaluate ME-Net on a larger and more complex dataset. We use Tiny-ImageNet, which is a subset of ImageNet and contains 200 classes. Each class has 500 images for training and 50 for testing. All images are 64×64 colored ones. Since ME-Net requires to train the model from scratch, due to the limited computing resources, we do not provide results on even larger dataset such

as ImageNet. However, we envision ME-Net to perform better on such larger datasets as it can leverage the global structures of those larger images.

A.5.1 Black-box Attacks

For black-box attacks on Tiny-ImageNet, we only report the Top-1 adversarial accuracy. We use standard DenseNet-121 [12] as our network, and set the attack parameters as having a total perturbation $\varepsilon = 8/255$ (0.031). We use the same black-box attacks as before and follow the same attack settings. The extensive results are shown in Table A.13.

Method		Clean	FGSM	PGD			CW		Boundary	SPSA
				7 steps	20 steps	40 steps	$\kappa = 20$	$\kappa = 50$		
Vanilla		66.4%	15.2%	1.3%	0.0%	0.0%	8.0%	7.7%	2.6%	1.2%
ME-Net	$p : 0.8 \to 1$	**67.7%**	**67.1%**	**66.3%**	**66.0%**	**65.8%**	**67.6%**	**67.4%**	**62.4%**	**67.4%**
	$p : 0.6 \to 0.8$	64.1%	63.6%	63.1%	63.1%	62.4%	63.8%	63.6%	61.9%	63.8%
	$p : 0.4 \to 0.6$	58.9%	54.8%	51.7%	51.6%	50.4%	58.2%	58.2%	58.9%	58.1%

Table A.13: **Tiny-ImageNet extensive black-box attack results.** Different kinds of strong black-box attacks are used, including transfer-, decision-, and score-based attacks.

Further, additional black-box attack results are provided in Table A.14, where the black-box adversary has limited access to ME-Net. The results again demonstrate the effectiveness of the preprocessing layer.

Method		FGSM	PGD			CW	
			7 steps	20 steps	40 steps	$\kappa = 20$	$\kappa = 50$
ME-Net	$p : 0.8 \to 1$	**66.5%**	**64.0%**	**62.6%**	59.1%	55.8%	56.0%
	$p : 0.6 \to 0.8$	61.1%	60.9%	60.7%	**59.2%**	57.6%	57.6%
	$p : 0.4 \to 0.6$	58.8%	58.2%	57.5%	56.9%	**58.3%**	**58.2%**

Table A.14: **Tiny-ImageNet additional black-box attack results where adversary has limited access to the trained network.** We provide the architecture and weights of our trained model to the black-box adversary to make it stronger.

A.5.2 White-box Attacks

In white-box settings, we set the attack hyper-parameters as follows: a total perturbation of 8/255 (0.031), a step size of 2/255 (0.01), and 7 steps PGD for adversarial training. We still use strongest BPDA attack with different attack steps up to 1000. We re-implement [4] to be the baseline, and keep all training process the same for ME-Net and [4]. Finally, we report both Top-1 and Top-5 adversarial accuracy in Table A.15, which demonstrates the significant adversarial robustness of ME-Net.

Metrics	Method	Type	Clean	Attack Steps				
				7	**20**	**40**	**100**	**1000**
Top-1	Madry	Adv. train	45.6%	23.3%	22.4%	22.4%	22.3%	22.1%
	ME-Net $p : 0.8 \to 1$	Prep. + Adv. train	53.9%	28.1%	25.7%	25.3%	25.0%	24.5%
	ME-Net $p : 0.6 \to 0.8$	Prep. + Adv. train	**57.0%**	33.7%	28.4%	27.3%	26.8%	26.3%
	ME-Net $p : 0.4 \to 0.6$	Prep. + Adv. train	55.6%	**38.8%**	**30.6%**	**29.4%**	**29.0%**	**28.5%**
Top-5	Madry	Adv. train	71.4%	47.5%	46.0%	45.9%	45.8%	45.0%
	ME-Net $p : 0.8 \to 1$	Prep. + Adv. train	77.4%	54.8%	52.2%	51.9%	51.2%	50.6%
	ME-Net $p : 0.6 \to 0.8$	Prep. + Adv. train	**80.3%**	62.1%	57.1%	56.7%	56.4%	55.1%
	ME-Net $p : 0.4 \to 0.6$	Prep. + Adv. train	78.8%	**66.7%**	**59.5%**	**58.5%**	**58.0%**	**56.9%**

Table A.15: **Tiny-ImageNet extensive white-box attack results.** We apply up to 1000 steps PGD or BPDA attacks in white-box setting to ensure the results are convergent. We select [4] as the baseline and keep the training process the same for both [4] and ME-Net. We show both Top-1 and Top-5 adversarial accuracy under different attack steps. ME-Net shows advanced results by outperforming [4] consistently in both Top-1 and Top-5 adversarial accuracy.

A.6 Additional Results of Different ME Methods

A.6.1 Black-box Attacks

We first provide additional experimental results using different ME methods against black-box attacks. We train different ME-Net models on CIFAR-10 using three ME methods, including the USVT approach, the Soft-Impute algorithm and the Nuclear Norm minimization algorithm. The training processes are identical for all models. For the black-box adversary, we use different transfer-based attacks and report the results in Table A.16.

Method	Complexity	Type	Clean	FGSM	PGD			CW	
					7 steps	20 steps	40 steps	$\kappa = 20$	$\kappa = 50$
Vanilla	–	–	93.4%	24.8%	7.6%	1.8%	0.0%	9.3%	8.9%
ME-Net - USVT	Low	Prep.	94.8%	90.5%	90.3%	89.4%	88.9%	**93.6%**	**93.6%**
ME-Net - Soft-Imp.	Medium	Prep.	**94.9%**	**92.2%**	**91.8%**	**91.8%**	**91.3%**	**93.6%**	93.5%
ME-Net - Nuc. Norm	High	Prep.	94.8%	92.0%	91.7%	91.4%	91.0%	93.3%	93.4%

Table A.16: **Comparison between different ME methods against black-box attacks.** We report the generalization and adversarial robustness of three ME-Net models using different ME methods on CIFAR-10. We apply transfer-based black-box attacks as the adversary.

A.6.2 White-box Attacks

We further report the white-box attack results of different ME-Net models in Table A.17. We use 7 steps PGD to adversarially train all ME-Net models with different ME methods on CIFAR-10. We apply up to 1000 steps strongest white-box BPDA attacks as the adversary. Compared to the previous state-of-the-art [4] on CIFAR-10, all the three ME-Net models can outperform them by a certain margin, while also achieving higher generalizations. The performance of different ME-Net models may vary slightly, where we can observe that more complex methods can lead to slightly better performance.

Method	Complexity	Type	Clean	Attack Steps				
				7	20	40	100	1000
Madry	–	Adv. train	79.4%	47.2%	45.6%	45.2%	45.1%	45.0%
ME-Net - USVT	Low	Prep. + Adv. train	**85.5%**	67.3%	55.8%	53.7%	52.6%	51.9%
ME-Net - Soft-Imp.	Medium	Prep. + Adv. train	**85.5%**	67.5%	56.5%	54.8%	53.0%	52.3%
ME-Net - Nuc. Norm	High	Prep. + Adv. train	85.0%	**68.2%**	**57.5%**	**55.4%**	**53.5%**	**52.8%**

Table A.17: **Comparison between different ME methods against white-box attacks.** We adversarially trained three ME-Net models using different ME methods on CIFAR-10, and compare the results with [4]. We apply up to 1000 steps PGD or BPDA white-box attacks as adversary.

A.7 Additional Studies of Attack Parameters

We present additional studies of attack parameters, including different random restarts and step sizes for further evaluations of ME-Net. Authors in [67] show that using

multiple random restarts and different step sizes can drastically affect the performance of PGD adversaries. We consider the same white-box BPDA-based PGD adversary as in Table 2.4, and report the results on CIFAR-10. Note that with n random restarts, given an image, we consider a classifier successful only if it was not fooled by any of these n attacks. In addition, this also significantly increases the computational overhead. We hence fix the number of attack steps as 100 (results are almost flattened; see for example Fig. 2-6), and select three step sizes and restart values. We again compare ME-Net with [4].

Method	**Step sizes**	**Random restarts**		
		10	20	50
Madry	2/255	43.4%	42.7%	41.7%
	4/255	43.8%	43.3%	41.9%
	8/255	44.0%	43.3%	41.9%
ME-Net	2/255	**48.7%**	**47.2%**	**44.8%**
	4/255	**49.7%**	**48.4%**	**45.2%**
	8/255	**50.8%**	**49.8%**	**46.0%**

Table A.18: **Results of white-box attacks with different random restarts and step sizes on CIFAR-10.** We compare ME-Net with [4] using three different step sizes and random restart values. We apply 100 steps PGD or BPDA white-box attacks as adversary.

As shown in Table A.18, with different step sizes, the performance of ME-Net varies slightly. Specifically, the smaller the step size (e.g., 2/255) is, the stronger the adversary becomes for both ME-Net and [4]. This is as expected, since a smaller step size enables a finer search for the adversarial perturbation.

ME-Net leverages randomness through masking, and it would be helpful to understand how random restarts, with a hard success criterion, affect the overall pipeline. As observed in Table A.18, more restarts can reduce the robust accuracy by a few percent. However, we note that ME-Net can still outperform [4] by a certain margin across different attack parameters. We remark that arguably, one could potentially always handle such drawbacks by introducing restarts during training as well, so as to maximally match the training and testing conditions. This introduces in unnecessary

overhead that might be less meaningful. We hence focus on other parameters such as the number of attack steps in the main chapter.

A.8 Additional Benefits by Majority Voting

It is common to apply an ensemble or vote scheme during the prediction stage to further improve accuracy. ME-Net naturally provides a majority voting scheme. As we apply masks with different observation probability p during training, an intuitive method is to also use multiple masks with the same p (rather than only one p) for each image during inference, and output a majority vote over predicted labels. One can even use more masks with different p within the training range. By such, the training procedure and model can remain unchanged while the inference overhead only gets increased by a small factor.

Attack Steps	**Method**	**MNIST**	**CIFAR-10**	**SVHN**	**Tiny-ImageNet**	
					Top-1	Top-5
40	Standard	94.0%	55.4%	71.9%	29.4%	58.5%
	Vote	**95.9%**	**59.3%**	**76.0%**	**33.8%**	**68.9%**
100	Standard	91.8%	53.5%	69.8%	29.0%	58.0%
	Vote	**94.2%**	**56.2%**	**73.1%**	**31.2%**	**65.4%**
1000	Standard	91.0%	52.8%	69.4%	28.5%	56.9%
	Vote	**92.6%**	**54.2%**	**71.4%**	**29.8%**	**59.5%**

Table A.19: **Comparison between majority vote and standard inference.** For each image, we apply 10 masks with same p used during training, and the model outputs a majority vote over predicted labels. The standard inference only uses one mask with the mean probability of those during training. We use 40, 100 and 1000 steps white-box BPDA attack and report the results on each dataset.

In Table A.19, we report the majority voting result of ME-Net on different datasets, where voting can consistently improve the adversarial robustness of the standard one by a certain margin. This is especially helpful in real-world settings where the defender can get more robust output without highly increasing the computational overhead. Note that by using majority vote, we can further improve the state-of-the-art white-box robustness.

A.9 Hyper-Parameters Study

A.9.1 Observation Probability p

As studied previously, by applying different masks with different observation probability p, the performance of ME-Net can change differently. We have already reported extensive quantitative results of different ME-Net models trained with different p. Here we present the qualitative results by visualizing the effect of different p on the original images. As illustrated in Fig. A-2, the first row shows the masked image with different p, and the second row shows the recovered image by ME. It can be observed that the global structure of the image is maintained even when p is small.

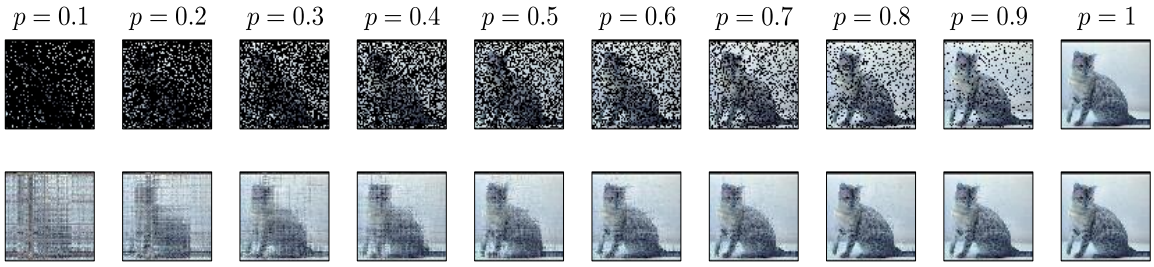


Figure A-2: **Visualization of ME result with different observation probability p. First row:** Images after applying masks with different observation probabilities. **Second row:** The recovered images by applying ME. We can observe that the global structure of the image is maintained even when p is small.

A.9.2 Number of Selected Masks

Another hyper-parameter of ME-Net is the number of selected masked images for each input image. In the main chapter, all experiments are carried out using 10 masks. We here provide the hyper-parameter study on how the number of masks affects the performance of ME-Net. We train ME-Net models on CIFAR-10 using different number of masks and keep other settings the same. In Table A.20, we show the results of both standard generalization and adversarial robustness. We use transfer-based 40 steps PGD as black-box adversary, and 1000 steps BPDA as white-box adversary. As expected, using more masks can lead to better performances. Due to the limited computation resources, we only try a maximum of 10 masks for each image. However, we expect ME-Net to perform even better with more sampled masks and better-tuned

hyper-parameters.

# of Masks	Method		Clean	Black-box	White-box
–	Vanilla		93.4%	0.0%	0.0%
1	ME-Net	$p : 0.9$	92.7%	82.3%	44.1%
		$p : 0.5$	79.8%	59.7%	47.4%
5	ME-Net	$p : 0.8 \rightarrow 1$	94.1%	87.8%	46.5%
		$p : 0.4 \rightarrow 0.6$	86.3%	68.5%	49.3%
10	ME-Net	$p : 0.8 \rightarrow 1$	**94.9%**	**91.3%**	47.4%
		$p : 0.4 \rightarrow 0.6$	89.2%	70.9%	**52.8%**

Table A.20: **Comparisons between different number of masked images used for each input image.** We report the generalization and adversarial robustness of ME-Net models trained with different number of masks on CIFAR-10. We apply transfer-based 40 steps PGD attack as black-box adversary, and 1000 steps PGD-based BPDA as white-box adversary.

A.10 Additional Visualization Results

We finally provide more visualization results of ME-Net applied to clean images, adversarial images, and their differences. We choose Tiny-ImageNet since it has a higher resolution. As shown in Fig. A-3, for vanilla model, the highly structured adversarial noises are distributed over the entire image, containing human imperceptible adversarial structure that is very likely to fool the network. In contrast, the redistributed noises in the reconstructed images from ME-Net mainly focus on the global structure of the images, which is well aligned with human perception. As such, we would expect ME-Net to be more robust against adversarial attacks.

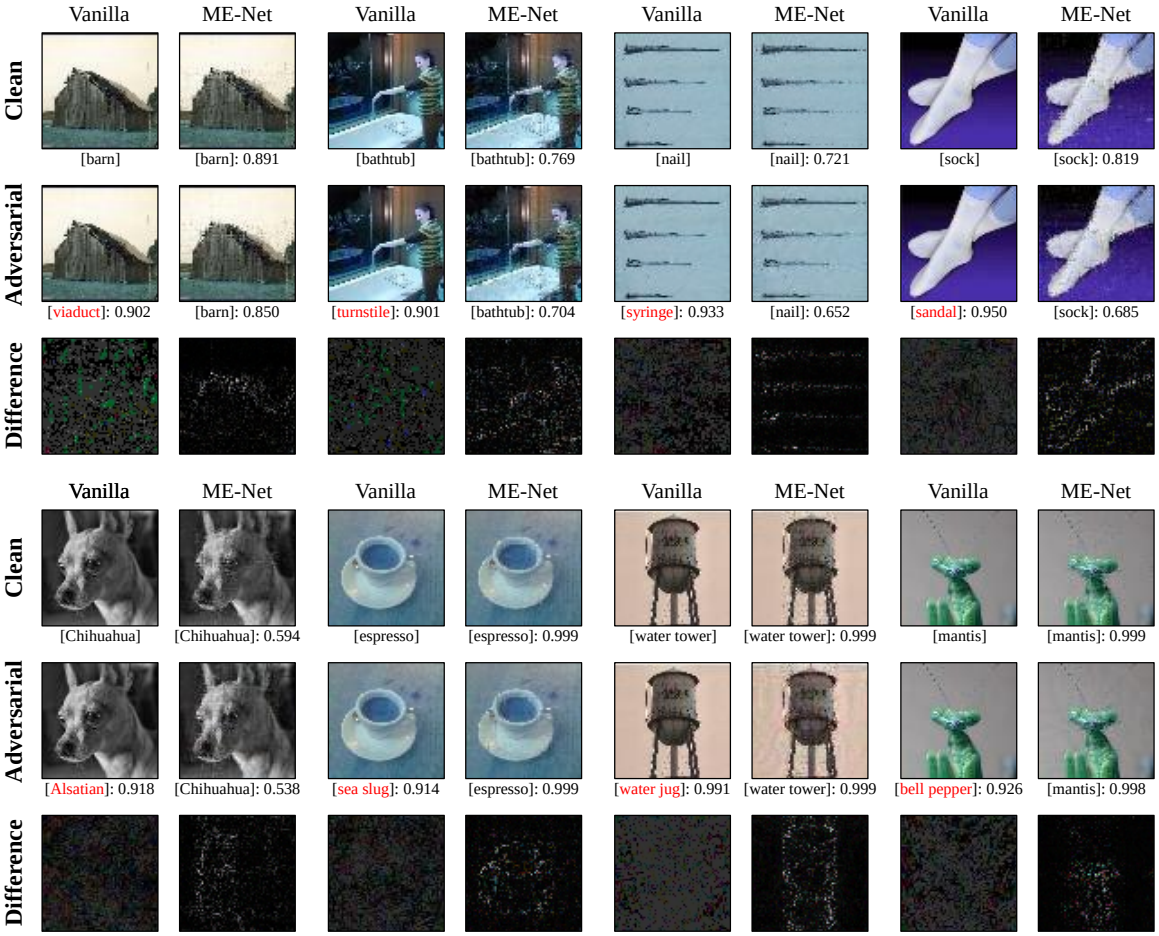


Figure A-3: **Visualization of ME-Net applied to clean images, adversarial images, and their differences on Tiny-ImageNet. First column** from top to bottom: the clean image, the adversarial example generated by PGD attacks, the difference between them (i.e., the adversarial noises). **Second column** from top to bottom: the reconstructed clean image by ME-Net, the reconstructed adversarial example by ME-Net after performing PGD attacks, the difference between them (i.e., the redistributed noises). Underlying each image is the predicted class and its probability. We multiply the difference images by a constant scaling factor to increase the visibility. The differences between the reconstructed clean image by ME-Net and the reconstructed adversarial example by ME-Net after performing PGD attacks, i.e., the new adversarial noises, are redistributed to the global structure.

Appendix B

Supplementary Materials for Chapter 3

B.1 Pseudo Code and Discussions for Structured Value-based Planning (SVP)

Algorithm 3: Structured Value-based Planning (SVP)

Input: initialized value function $Q^{(0)}(s,a)$;
prescribed observing probability p.
for $t = 1, 2, 3, \ldots$ **do**
  randomly sample a set Ω of observed entries from $\mathcal{S} \times \mathcal{A}$, each with probability p
  `/* update the randomly selected state-action pairs*/`
  for each state-action pair $(s,a) \in \Omega$ **do**

$$\hat{Q}(s,a) \leftarrow \sum_{s'} P(s'|s,a) \left(r(s,a) + \gamma \max_{a'} Q^{(t)}(s',a') \right)$$

  end for
  `/* reconstruct the` Q `matrix via matrix estimation*/`
  apply ME to the observed values $\{\hat{Q}(s,a)\}_{(s,a)\in\Omega}$ to reconstruct $Q^{(t+1)}$:

$$Q^{(t+1)} \leftarrow \mathrm{ME}\big(\{\hat{Q}(s,a)\}_{(s,a)\in\Omega}\big)$$

end for

While based on classical value iteration, we remark that a theoretical analysis,

even in the tabular case, is quite complex. (1) Although the field of ME is somewhat mature, the analysis has been largely focused on the "one-shot" problem: recover a static data matrix given one incomplete observation. Under the iterative scenario considered here, standard assumptions are easily broken and the analysis warrants potentially new machinery. (2) Furthermore, much of the effort in the ME community has been devoted to the Frobenius norm guarantees rather than the infinite norm as in value iteration. Non-trivial infinite norm bound has received less attention and often requires special techniques [68, 69]. Resolving the above burdens would be important future avenues in its own right for the ME community. Henceforth, this thesis focuses on empirical analysis and more importantly, *generalizing* the framework successfully to modern deep RL contexts. As we will demonstrate, the consistent empirical benefits offer a sounding foundation for future analysis.

B.2 Experimental Setups for Stochastic Control Tasks

Inverted Pendulum As stated earlier in Sec. 3.3.3, the goal is to balance the inverted pendulum on the upright equilibrium position. The physical dynamics of the system is described by the angle and the angular speed, i.e., $(\theta, \dot{\theta})$. Denote τ as the time interval between decisions, u as the torque input on the pendulum, the dynamics can be written as [24, 57]:

$$\theta := \theta + \dot{\theta}\ \tau, \tag{B.1}$$

$$\dot{\theta} := \dot{\theta} + \Big(\sin\theta - \dot{\theta} + u\Big)\ \tau. \tag{B.2}$$

A reward function that penalizes control effort while favoring an upright pendulum is used:

$$r(\theta, u) = -0.1u^2 + \exp\left(\cos\theta - 1\right). \tag{B.3}$$

In the simulation, the state space is $(-\pi, \pi]$ for θ and $[-10, 10]$ for $\dot{\theta}$. We limit the input torque in $[-1, 1]$ and set $\tau = 0.3$. We discretize each dimension of the state space into 50 values, and action space into 1000 values, which forms an Q-value function a

matrix of dimension 2500×1000. We follow [56] to handle the policy of continuous states by modelling their transitions using multi-linear interpolation.

Mountain Car We select another classical control problem, i.e., the Mountain Car [24], for further evaluations. In this problem, an under-powered car aims to drive up a steep hill [24]. The physical dynamics of the system is described by the position and the velocity, i.e., $(x, \dot{x})$. Denote u as the acceleration input on the car, the dynamics can be written as

$$x := x + \dot{x}, \tag{B.4}$$

$$\dot{x} := \dot{x} - 0.0025\cos(3x) + 0.001u. \tag{B.5}$$

The reward function is defined to encourage the car to get onto the top of the mountain at $x_0 = 0.5$:

$$r(x) = \begin{cases} 10, & x \geq x_0, \\ -1, & else. \end{cases} \tag{B.6}$$

We follow standard settings to restrict the state space as $[-0.07, 0.07]$ for x and $[-1.2, 0.6]$ for $\dot{x}$, and limit the input $u \in [-1, 1]$. Similarly, the whole state space is discretized into 2500 values, and the action space is discretized into 1000 values. The evaluation metric we are concerned about is the total time it takes to reach the top of the mountain, given a randomly and uniformly generated initial state.

Double Integrator We consider the Double Integrator system [70], as another classical control problem for evaluation. In this problem, a unit mass brick moving along the x-axis on a frictionless surface, with a control input which provides a horizontal force, u [26]. The task is to design a control system to regulate this brick to $\boldsymbol{x} = [0, 0]^T$. The physical dynamics of the system is described by the position and the velocity (i.e., $(x, \dot{x})$), and can be derived as

$$x := x + \dot{x}\ \tau, \tag{B.7}$$

$$\dot{x} := \dot{x} + u\ \tau. \tag{B.8}$$

Follow [26], we use the quadratic cost formulation to define the reward function, which regulates the brick to $\boldsymbol{x} = [0, 0]^T$:

$$r(x, \dot{x}) = -\frac{1}{2}\left(x^2 + \dot{x}^2\right). \tag{B.9}$$

We follow standard settings to restrict the state space as $[-3, 3]$ for both x and $\dot{x}$, limit the input $u \in [-1, 1]$ and set $\tau = 0.1$. The whole state space is discretized into 2500 values, and the action space is discretized into 1000 values. Similarly, we define the evaluation metric as the total time it takes to reach to $\boldsymbol{x} = [0, 0]^T$, given a randomly and uniformly generated initial state.

Cart-Pole Finally, we choose the Cart-Pole problem [71], a harder control problem with 4-dimensional state space. The problem consists a pole attached to a cart moving on a frictionless track. The cart can be controlled by means of a limited force within $10N$ that is possible to apply both to the left or to the right of the cart. The goal is to keep the pole on the upright equilibrium position. The physical dynamics of the system is described by the angle and the angular speed of the pole, and the position and the speed of the cart, i.e., $(\theta, \dot{\theta}, x, \dot{x})$. Denote τ as the time interval between decisions, u as the force input on the cart, the dynamics can be written as

$$\ddot{\theta} := \frac{g \sin\theta - \frac{u + ml\dot{\theta}^2 \sin\theta}{m_c + m} \cos\theta}{l\left(\frac{4}{3} - \frac{m\cos^2\theta}{m_c + m}\right)}, \tag{B.10}$$

$$\ddot{x} := \frac{u + ml\left(\dot{\theta}^2 \sin\theta - \ddot{\theta}\cos\theta\right)}{m_c + m}, \tag{B.11}$$

$$\theta := \theta + \dot{\theta}\ \tau, \tag{B.12}$$

$$\dot{\theta} := \dot{\theta} + \ddot{\theta}\ \tau, \tag{B.13}$$

$$x := x + \dot{x}\ \tau, \tag{B.14}$$

$$\dot{x} := \dot{x} + \ddot{x}\ \tau, \tag{B.15}$$

where $g = 9.8m/s^2$ corresponds to the gravity acceleration, $m_c = 1kg$ denotes the mass of the cart, $m = 0.1kg$ denotes the mass of the pole, $l = 0.5m$ is half of the pole length, and u corresponds to the force applied to the cart, which is limited by

$u \in [-10, 10]$.

A reward function that favors the pole in an upright position, i.e., characterized by keeping the pole in vertical position between $|\theta| \leq \frac{12\pi}{180}$, is expressed as

$$r(\theta) = \cos^4(15\theta). \tag{B.16}$$

In the simulation, the state space is $[-\frac{\pi}{2}, \frac{\pi}{2}]$ for θ, $[-3, 3]$ for $\dot{\theta}$, $[-2.4, 2.4]$ for x and $[-3.5, 3.5]$ for $\dot{x}$. We limit the input force in $[-10, 10]$ and set $\tau = 0.1$. We discretize each dimension of the state space into 10 values, and action space into 1000 values, which forms an Q-value function a matrix of dimension 10000×1000.

B.3 Additional Results for SVP

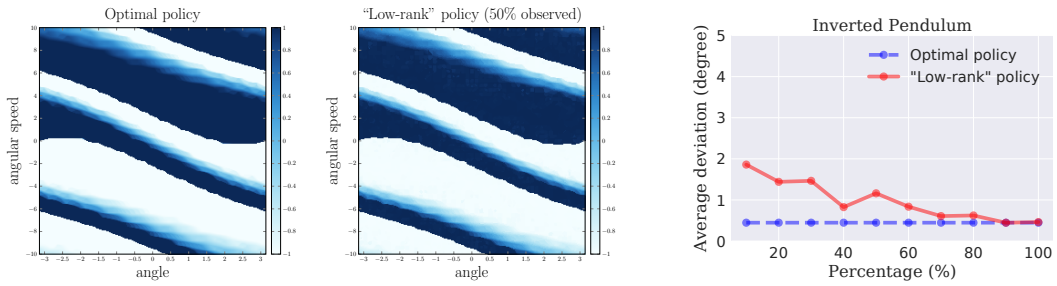
B.3.1 Inverted Pendulum

We further verify that the optimal Q^* indeed contains the desired low-rank structure. To this end, we construct "low-rank" policies directly from the converged Q matrix. In particular, for the converged Q matrix, we sub-sample a certain percentage of its entries, reconstruct the whole matrix via ME, and finally construct a corresponding policy. Fig. B-1 illustrates the results, where the policy heatmap as well as the performance (i.e., the angular error) of the "low-rank" policy is essentially identical to the optimal one. The results reveal the intrinsic strong low-rank structures lie in the Q-value function.

We provide additional results for the inverted pendulum problem. We show the policy trajectory (i.e., how the angle of the pendulum changes with time) and the input changes (i.e., how the input torque changes with time), for each policy.

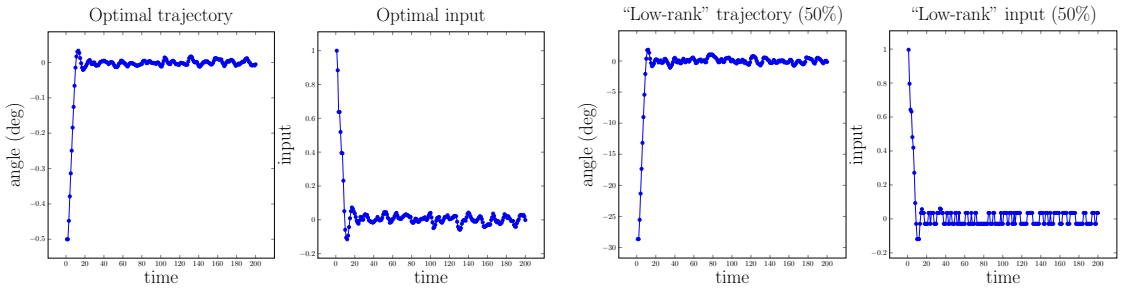
In Fig. B-2, we first show the comparison between the optimal policy and a "low-rank" policy. Recall that the low-rank policies are directly reconstructed from the converged Q matrix, with limited observation of a certain percentage of the entries in the converged Q matrix. As shown, the "low-rank" policy performs nearly identical to the optimal one, in terms of both the policy trajectory and the input torque changes. This again verifies the strong low-rank structure lies in the Q function.

Further, we show the policy trajectory and the input torque changes of the SVP



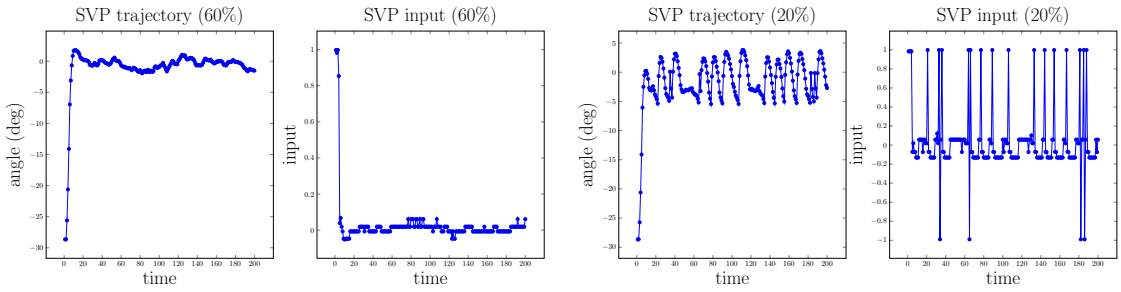
(a) (b) (c)

Figure B-1: Performance comparison between optimal policy and the reconstructed "low-rank" policy, on the Inverted Pendulum task.



(a) Optimal policy (b) "Low-rank" policy with 50% observed data

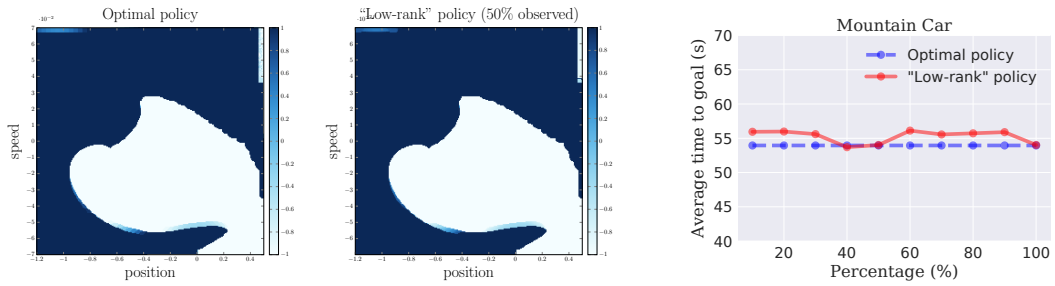
Figure B-2: Comparison of the policy trajectories and the input torques between the two schemes, on the Inverted Pendulum task.



(a) SVP policy with 60% observed data (b) SVP policy with 20% observed data

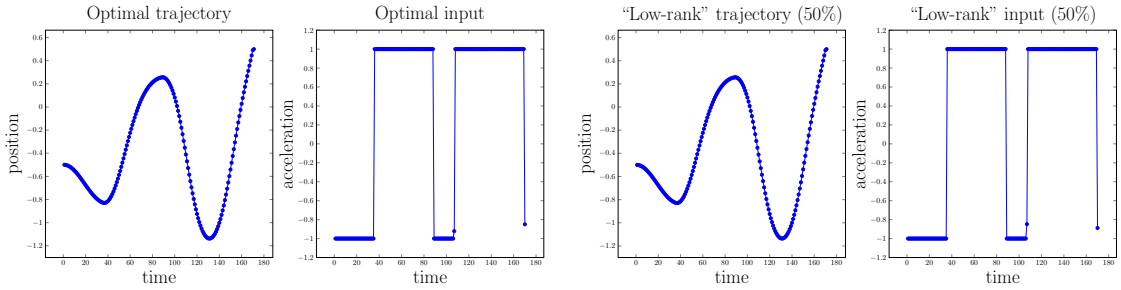
Figure B-3: The policy trajectories and the input torques of the proposed SVP scheme, on the Inverted Pendulum task.

policy. We vary the percentage of observed data for SVP, and present the policies with 20% and 60% for demonstration. As reported in Fig. B-3, the SVP policies are essentially identical to the optimal one. Interestingly, when we further decrease the observing percentage to 20%, the policy trajectory vibrates a little bit, but can still stabilize in the upright position with a small average angular deviation $\leq 5^\circ$.



(a) (b) (c)

Figure B-4: Performance comparison between optimal policy and the reconstructed "low-rank" policy, on the Mountain Car task.



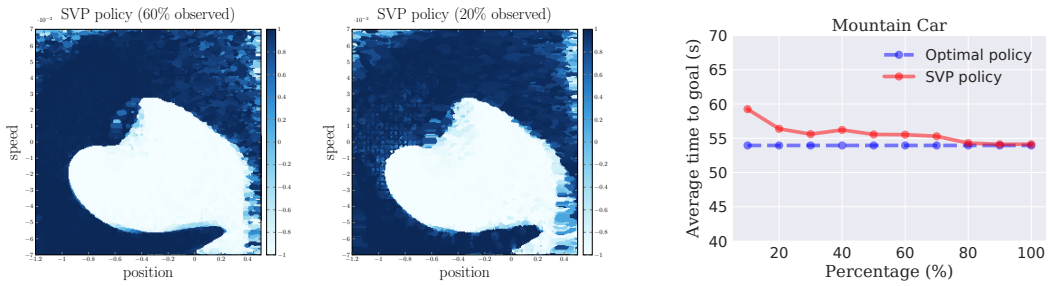
(a) Optimal policy (b) "Low-rank" policy with 50% observed data

Figure B-5: Comparison of the policy trajectories and the input changes between the two schemes, on the Mountain Car task.

B.3.2 Mountain Car

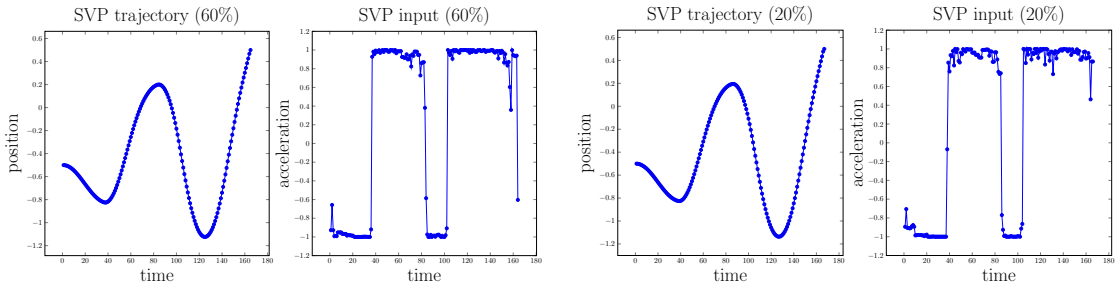
Similarly, we first verify the optimal Q^* contains the desired low-rank structure. We use the same approach to generate a "low-rank" policy based on the converged optimal value function. Fig. B-4(a) and B-4(b) show the policy heatmaps, where the reconstructed "low-rank" policy maintains visually identical to the optimal one. In Fig. B-4(c) and B-5, we quantitatively show the average time-to-goal, the policy trajectory and the input changes between the two schemes. Compared to the optimal one, even with limited sampled data, the reconstructed policy can maintain almost identical performance.

We further show the results of the SVP policy with different amount of observed data (i.e., 20% and 60%) in Fig. B-6 and B-7. Again, the SVP policies show consistently comparable results to the optimal policy, over various evaluation metrics. Interestingly, the converged Q matrix of vanilla value iteration is found to have an approximate



(a) (b) (c)

Figure B-6: Performance of the proposed SVP policy, with different amount of observed data, on the Mountain Car task.



(a) SVP policy with 60% observed data (b) SVP policy with 20% observed data

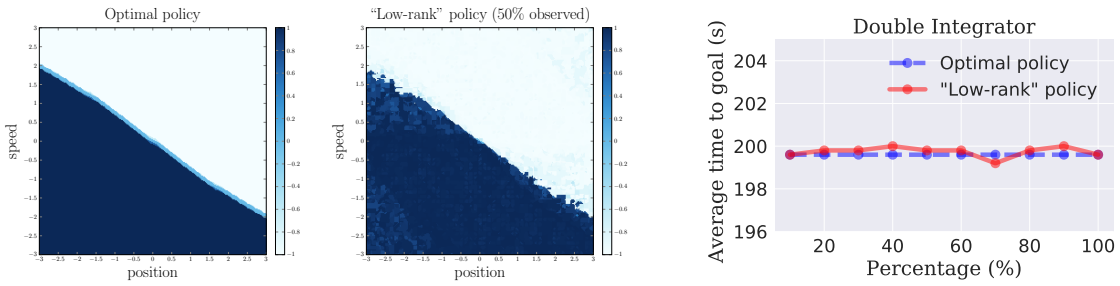
Figure B-7: The policy trajectories and the input changes of the proposed SVP scheme, on the Mountain Car task.

rank of 4 (the whole matrix is 2500×1000), thus the SVP can harness such strong low-rank structure for perfect recovery even with only 20% observed data.

B.3.3 Double Integrator

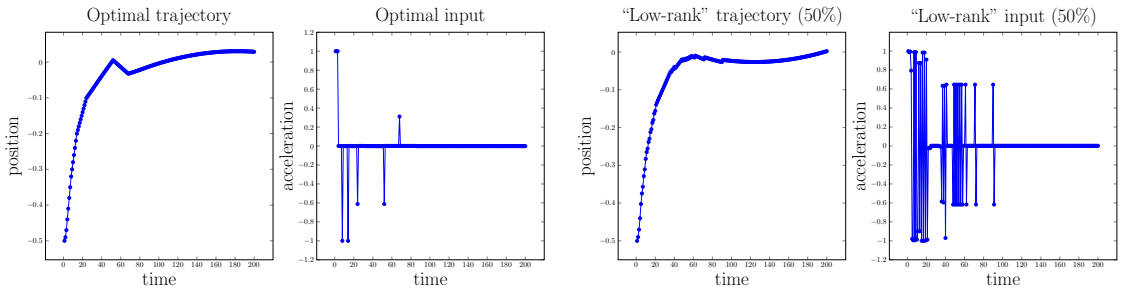
For the Double Integrator, We first use the same approach to generate a "low-rank" policy. Fig. B-8(a) and B-8(b) show that the reconstructed "low-rank" policy is visually identical to the optimal one. In Fig. B-8(c) and B-9, we quantitatively show the average time-to-goal, the policy trajectory and the input changes between the two schemes, where the reconstructed policy can achieve the same performance.

Further, we show the results of the SVP policy with different amount of observed data (i.e., 20% and 60%) in Fig. B-10 and B-11. As shown, the SVP policies show consistently decent results, which demonstrates that SVP can harness such strong low-rank structure even with only 20% observed data.



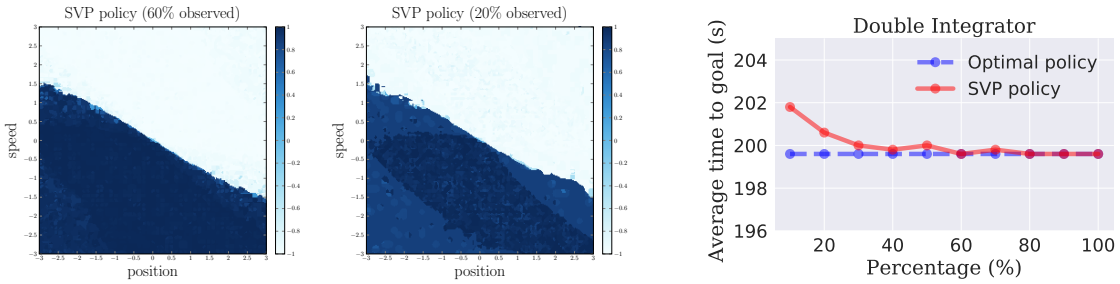
(a) (b) (c)

Figure B-8: Performance comparison between optimal policy and the reconstructed "low-rank" policy, on the Double Integrator task.



(a) Optimal policy (b) "Low-rank" policy with 50% observed data

Figure B-9: Comparison of the policy trajectories and the input changes between the two schemes, on the Double Integrator task.

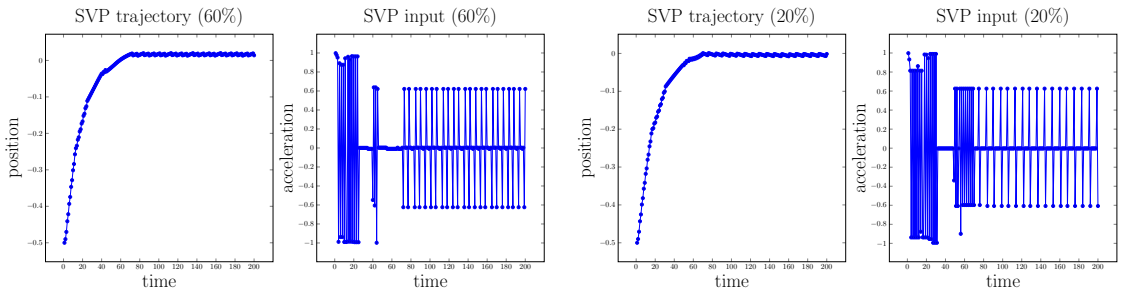


(a) (b) (c)

Figure B-10: Performance of the proposed SVP policy, with different amount of observed data, on the Double Integrator task.

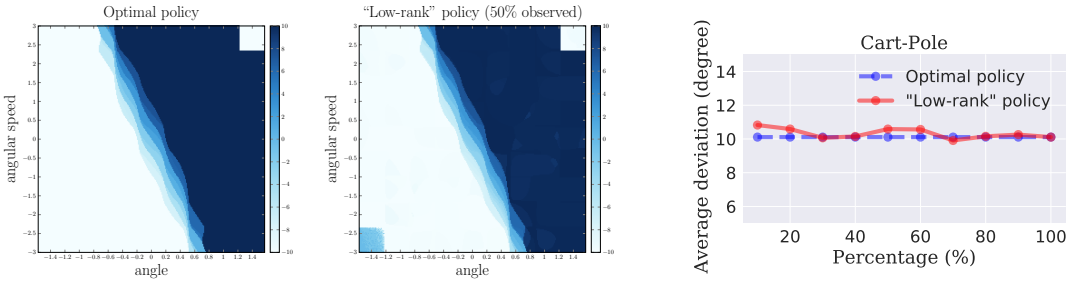
B.3.4 Cart-Pole

Finally, we evaluate SVP on the Cart-Pole system. Note that since the state space has a dimension of 4, the policy heatmap should contain 4 dims, but is hard to visualize. Since the metric we care is the angle deviation, we here only plot the first two dims (i.e., the $(\theta, \dot{\theta})$ tuple) with fixed x and $\dot{x}$, to visualize the policy heatmaps. We first use



(a) SVP policy with 60% observed data (b) SVP policy with 20% observed data

Figure B-11: The policy trajectories and the input changes of the proposed SVP scheme, on the Double Integrator task.

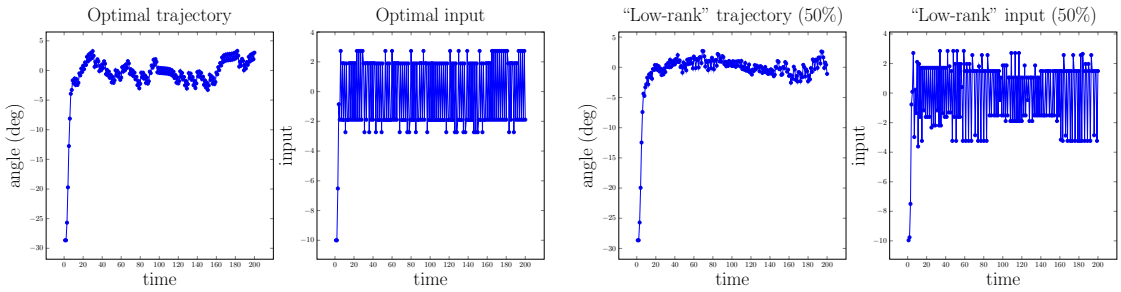


(a) (b) (c)

Figure B-12: Performance comparison between optimal policy and the reconstructed "low-rank" policy, on the Cart-Pole task.

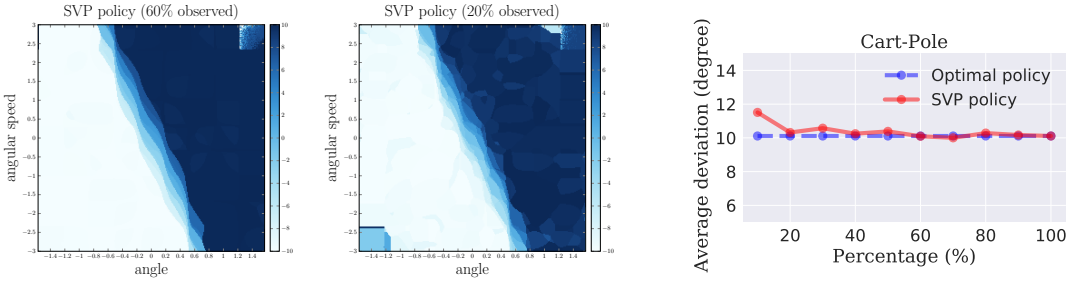
the same approach to generate a "low-rank" policy. Fig. B-12(a) and B-12(b) show the policy heatmaps, where the reconstructed "low-rank" policy is visually identical to the optimal one. In Fig. B-12(c) and B-13, we quantitatively show the average time-to-goal, the policy trajectory and the input changes between the two schemes. As demonstrated, the reconstructed policy can maintain almost identical performance with only small amount of sampled data.

We finally show the results of the SVP policy with different amount of observed data (i.e., 20% and 60%) in Fig. B-14 and B-15. Even for harder control tasks with higher dimensional state space, the SVP policies are still essentially identical to the optimal one. Across various stochastic control tasks, we demonstrate that SVP can consistently leverage strong low-rank structures for efficient planning.



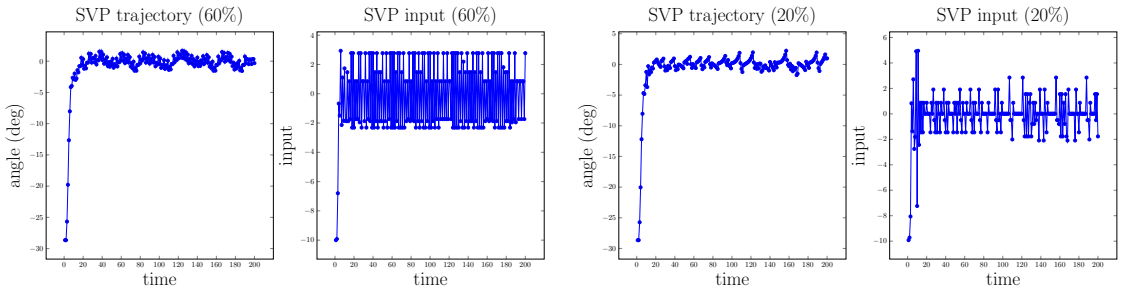
(a) Optimal policy (b) "Low-rank" policy with 50% observed data

Figure B-13: Comparison of the policy trajectories and the input changes between the two schemes, on the Cart-Pole task.



(a) (b) (c)

Figure B-14: Performance of the proposed SVP policy, with different amount of observed data, on the Cart-Pole task.



(a) SVP policy with 60% observed data (b) SVP policy with 20% observed data

Figure B-15: The policy trajectories and the input changes of the proposed SVP scheme, on the Cart-Pole task.

B.4 Training Details of Structured Value-based RL (SV-RL)

Training Details and Hyper-parameters The network architectures of DQN and dueling DQN used in our experiment are exactly the same as in the original

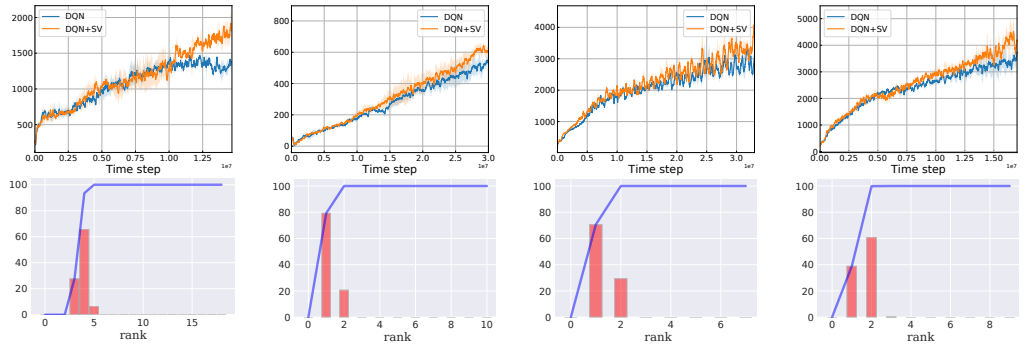
papers [21, 28, 63]. We train the network using the Adam optimizer [72]. In all experiments, we set the hyper-parameters as follows: learning rate $\alpha = 1e^{-5}$, discount coefficient $\gamma = 0.99$, and a minibatch size of 32. The number of steps between target network updates is set to $10,000$. We use a simple exploration policy as the ϵ-greedy policy with the ϵ decreasing linearly from 1 to 0.01 over $3e^5$ steps. For each experiment, we perform at least 3 independent runs and report the averaged results.

SV-RL Details To reconstruct the matrix $Q^\dagger$ formed by the current batch of states, we mainly employ the Soft-Impute algorithm [33] throughout our experiments. We set the sub-sample rate to $p = 0.9$ of the Q matrix, and use a linear scheduler to increase the sampling rate every $2e^6$ steps.

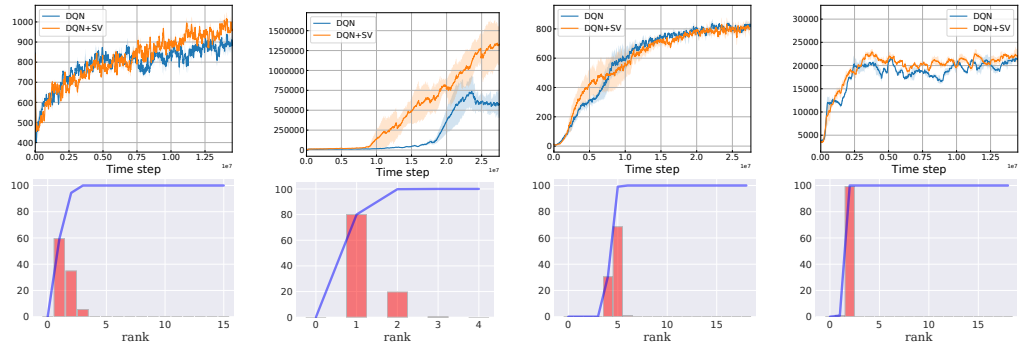
B.5 Additional Results for SV-RL

Experiments across Various Value-based RL We show more results across DQN, double DQN and dueling DQN in Fig. B-16, B-17, B-18, B-19, B-20 and B-21, respectively. For DQN, we complete **all 57 Atari games** using the proposed SV-RL, and verify that the majority of tasks contain low-rank structures (43 out of 57), where we can obtain consistent benefits from SV-RL. For each experiment, we associate the performance on the Atari game with its approximate rank. As mentioned in the main text, majority of the games benefit consistently from SV-RL. We note that roughly only **4** games, which have a significantly large rank, perform slightly worse than the vanilla DQN.

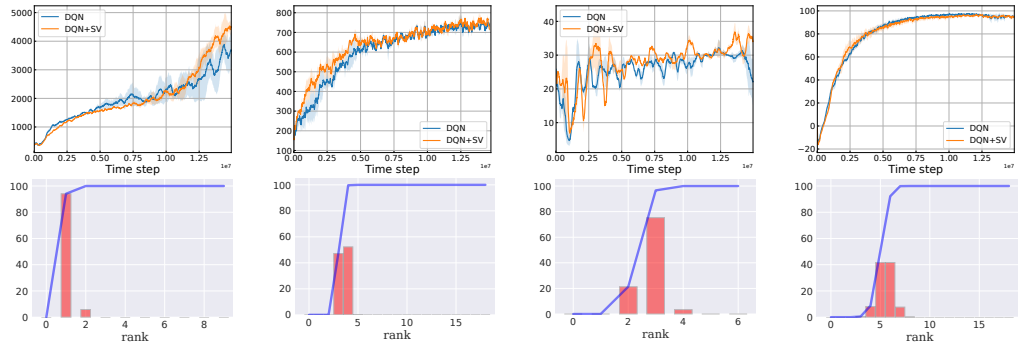
Consistency and Interpretation Across all the experiments we have done, we observe that when the game possesses structures (i.e., being approximately low-rank), SV-RL can consistently improve the performance of various value-based RL techniques. The superior performance is maintained through most of the experiments, verifying the ability of the proposed SV-RL to harness the structures for better efficiency and performance in value-based deep RL tasks. In the meantime, when the approximate rank is relatively higher (e.g., SEAQUEST), the performance of SV-RL can be similar or worse than the vanilla scheme, which also aligns well with our intuitions. Note that the majority of the games have an action space of size 18 (i.e., rank is at most 18), while some (e.g., PONG) only have 6 or less (i.e., rank is at most 6).



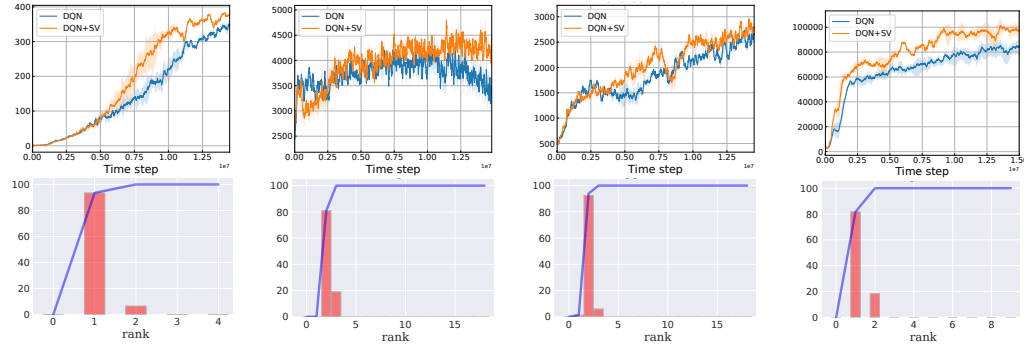
(a) Alien (better) (b) Amidar (better) (c) Assault (better) (d) Asterix (better)



(e) Asteroids (better) (f) Atlantis (better) (g) Bank Heist (similar) (h) Battle Zone (better)

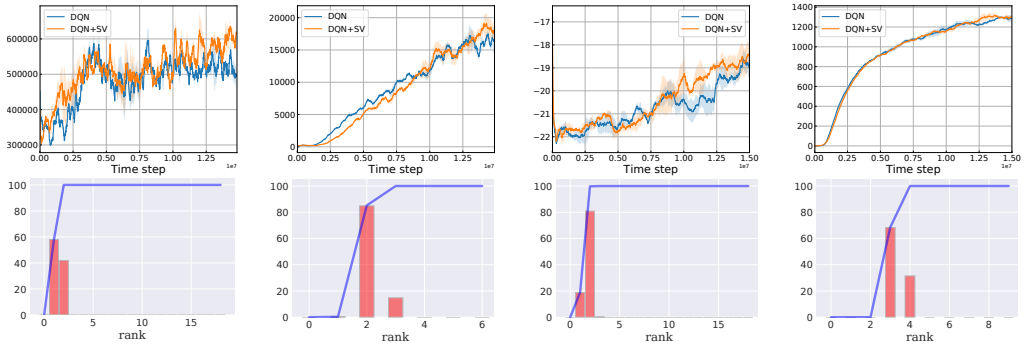


(i) Beam Rider (better) (j) Berzerk (better) (k) Bowling (similar) (l) Boxing (similar)

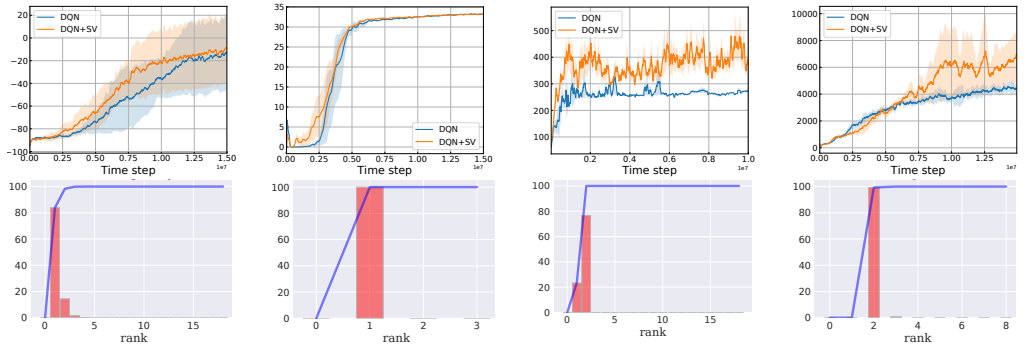


(m) Breakout (better) (n) Centipede (better) (o) Chopper (better) (p) Crazy Cli. (better)

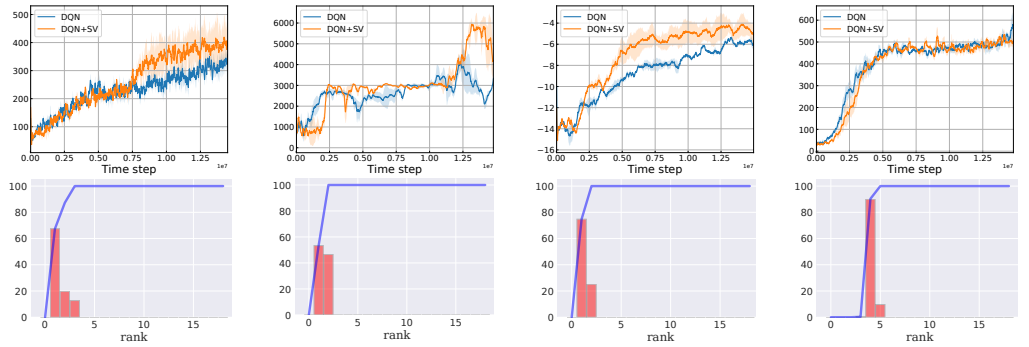
Figure B-16: Additional results of SV-RL on DQN (Part A).



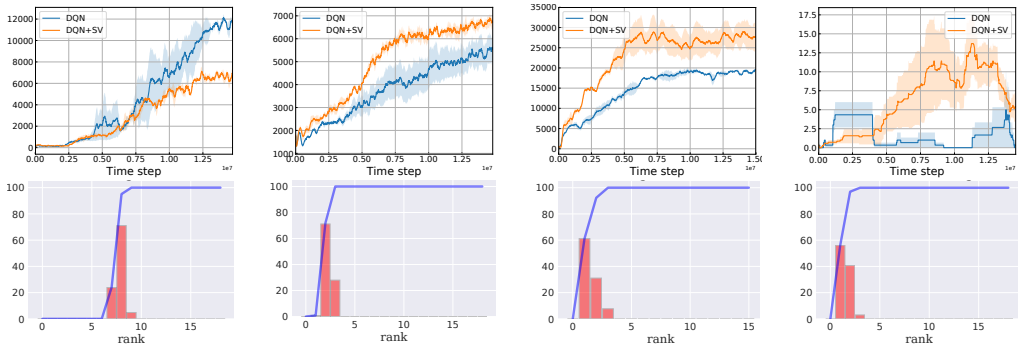
(a) Defender (better) (b) Demon Att. (better) (c) Double Dunk (better) (d) Enduro (similar)



(e) Fish Derby (better) (f) Freeway (similar) (g) Frostbite (better) (h) Gopher (better)

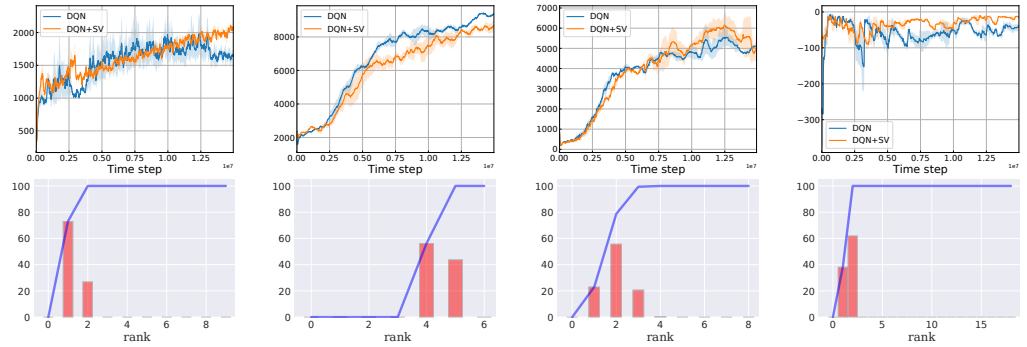


(i) Gravitar (better) (j) Hero (better) (k) IceHockey (better) (l) Jamesbond (similar)

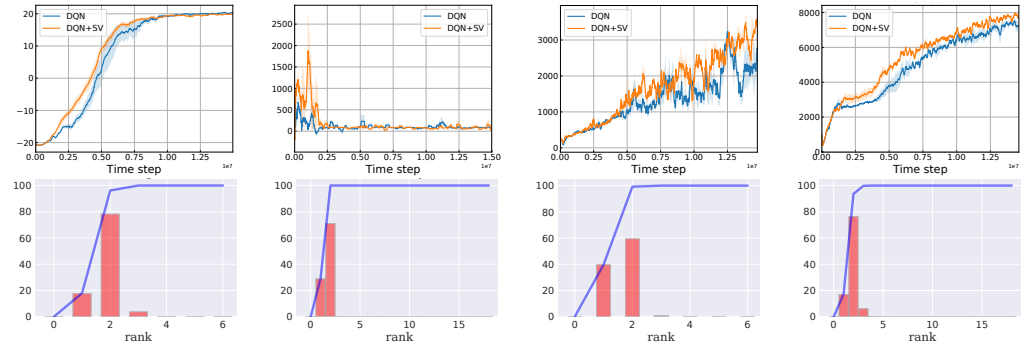


(m) Kangaroo (worse) (n) Krull (better) (o) KungFu (better) (p) Montezuma (better)

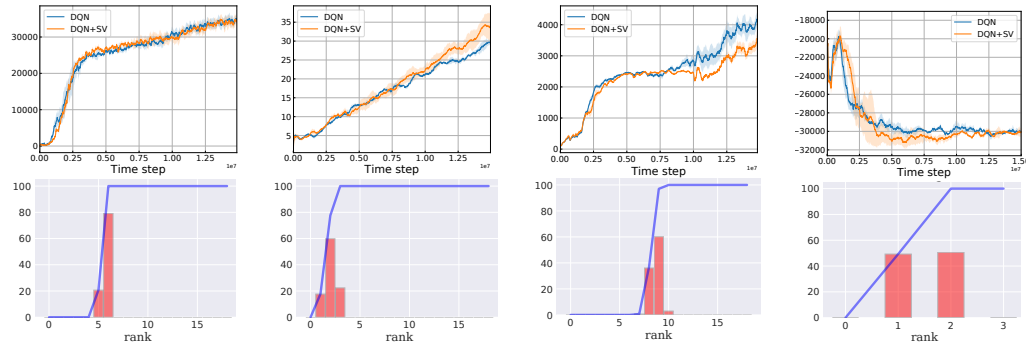
Figure B-17: Additional results of SV-RL on DQN (Part B).



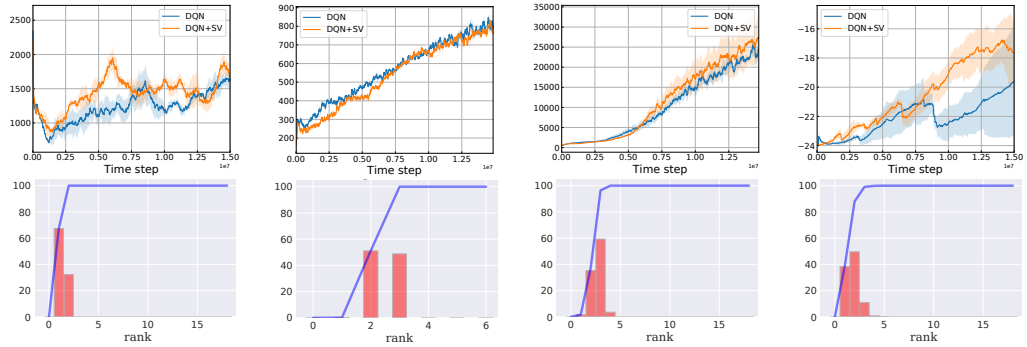
(a) MsPacman (better) (b) NameThisG. (worse) (c) Phoenix (better) (d) Pitfall (better)



(e) Pong (slightly better) (f) PrivateEye (better) (g) Qbert (better) (h) Riverraid (better)

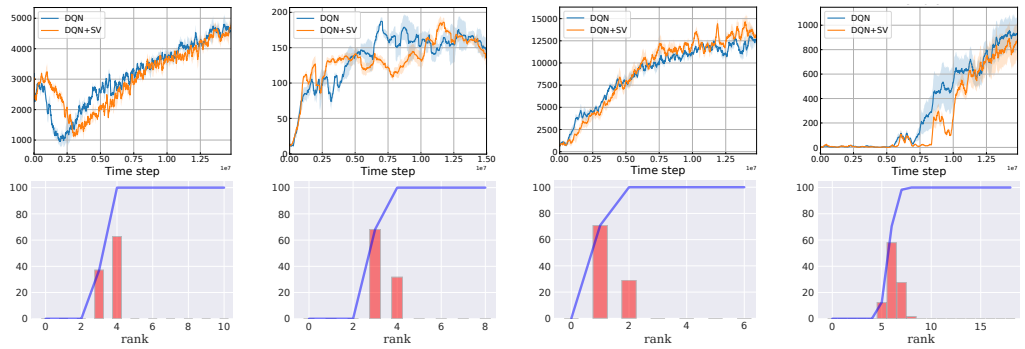


(i) Road Run. (similar) (j) Robotank (better) (k) Seaquest (worse) (l) Skiing (similar)

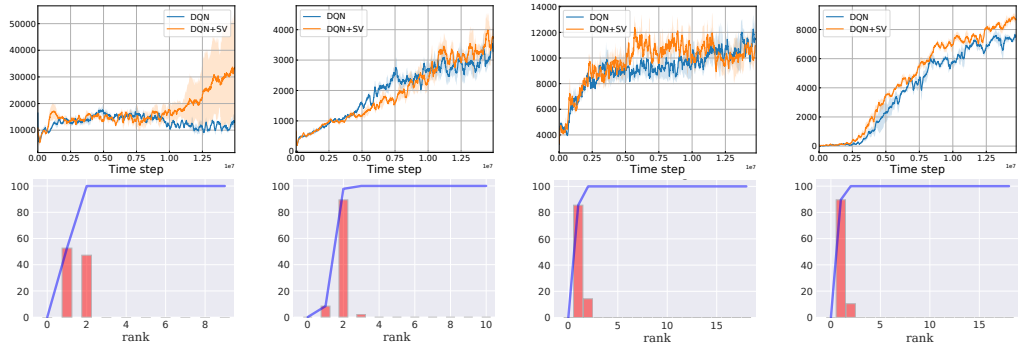


(m) Solaris (better) (n) SpaceInv. (similar) (o) Star Gunner (better) (p) Tennis (better)

Figure B-18: Additional results of SV-RL on DQN (Part C).

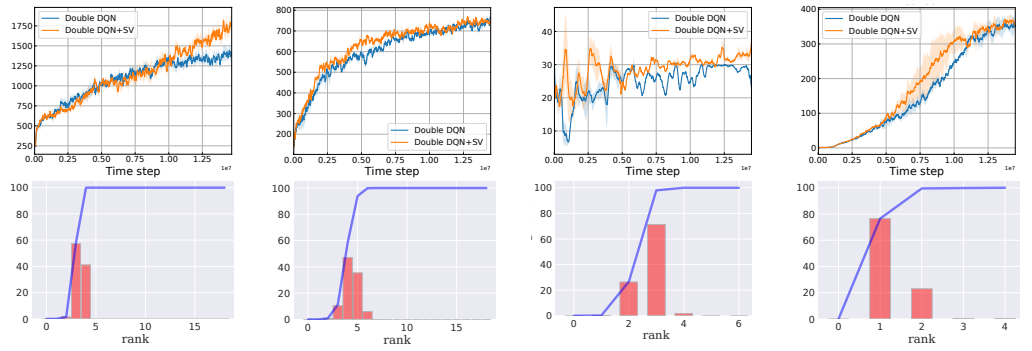


(a) Time Pilot (similar) (b) Tutankham (similar) (c) UpNDown (better) (d) Venture (worse)

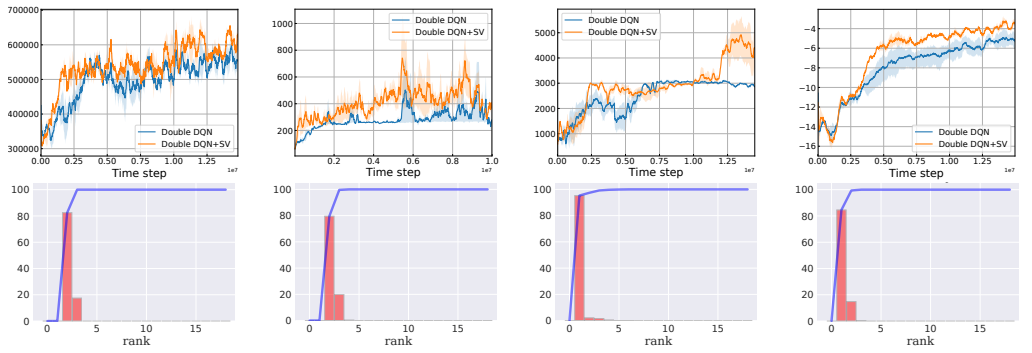


(e) Video Pin. (better) (f) WizardOW. (better) (g) YarsRevenge (better) (h) Zaxxon (better)

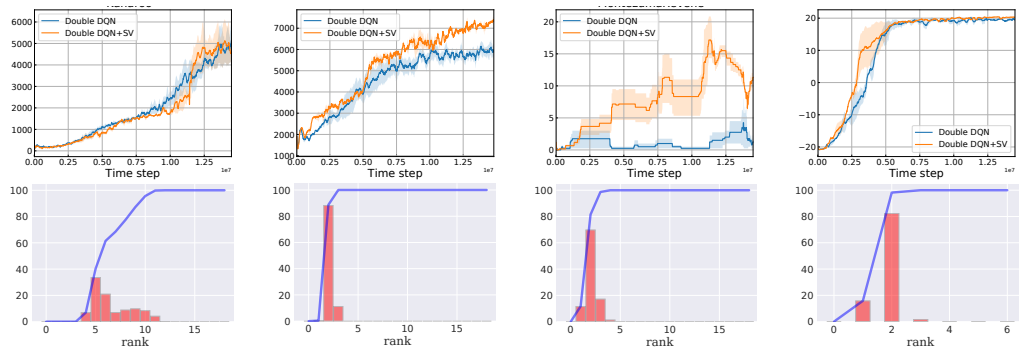
Figure B-19: Additional results of SV-RL on DQN (Part D).



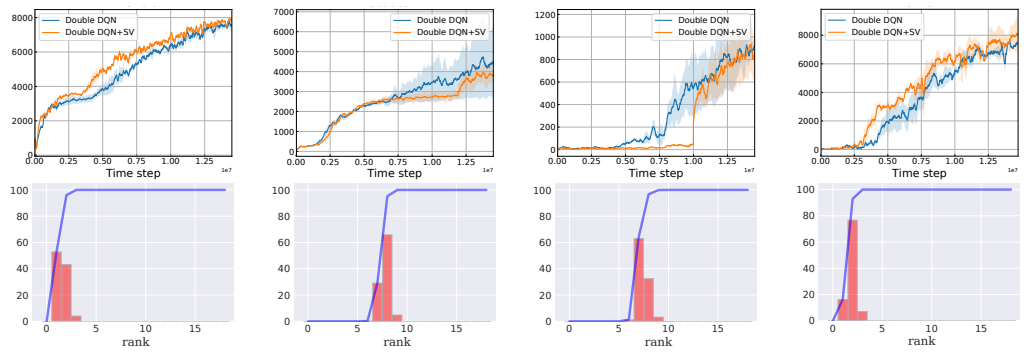
(a) Alien (slightly better) (b) Berzerk (similar) (c) Bowling (similar) (d) Breakout (better)



(e) Defender (better) (f) Frostbite (better) (g) Hero (better) (h) IceHockey (better)

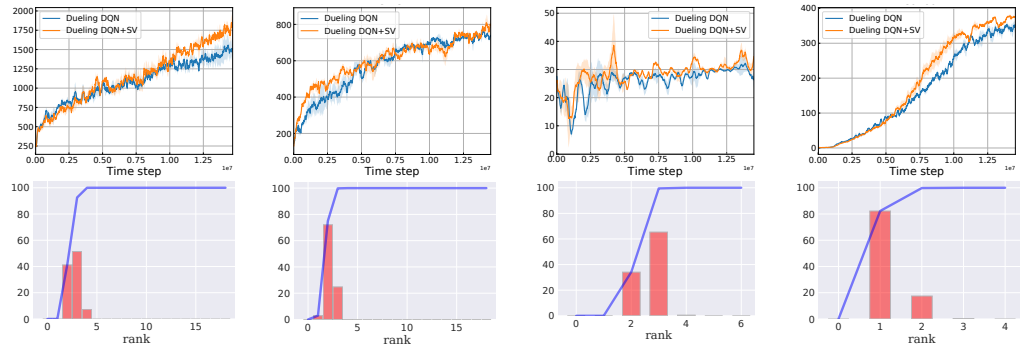


(i) Kangaroo (worse) (j) Krull (better) (k) Montezuma (better) (l) Pong (slightly better)

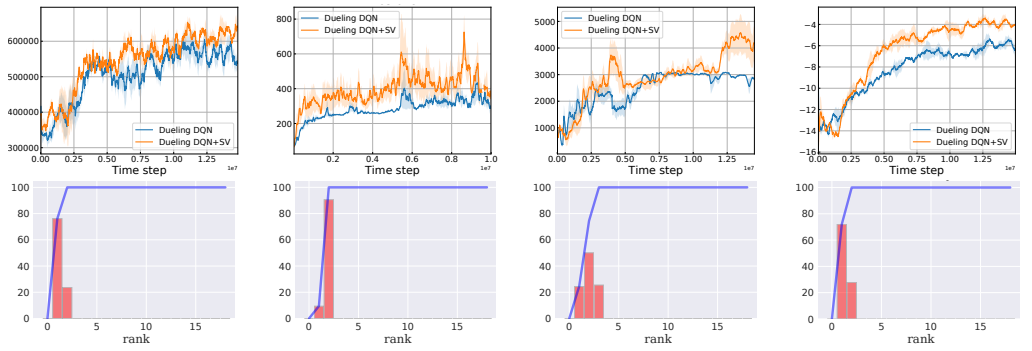


(m) Riverraid (better) (n) Seaquest (worse) (o) Venture (worse) (p) Zaxxon (better)

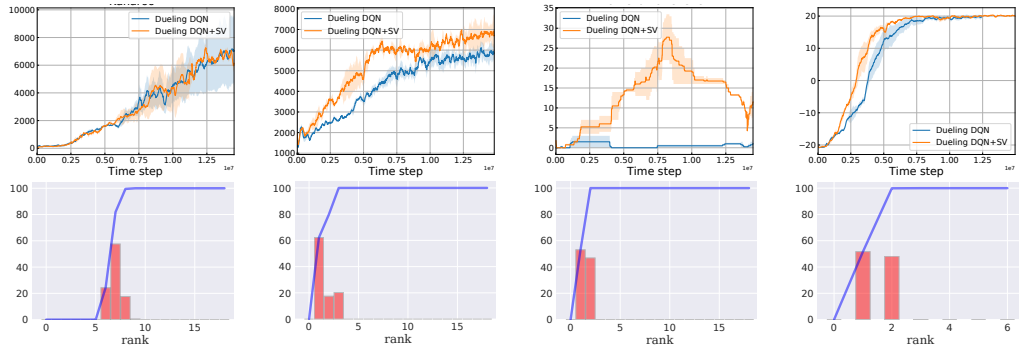
Figure B-20: Additional results of SV-RL on double DQN.



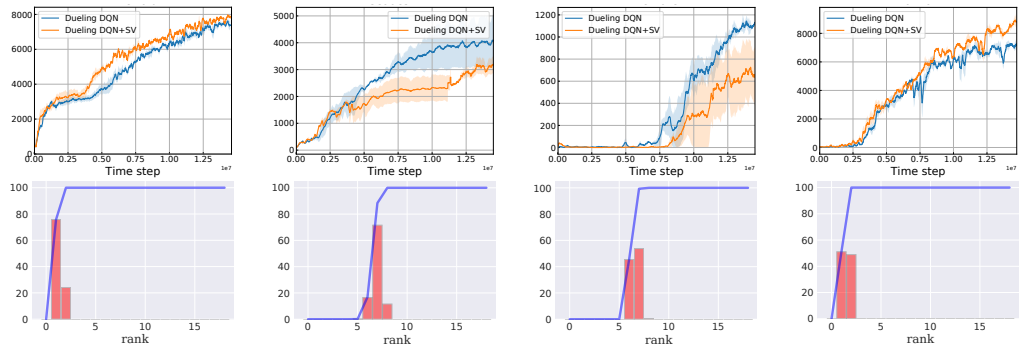
(a) Alien (slightly better) (b) Berzerk (similar) (c) Bowling (similar) (d) Breakout (better)



(e) Defender (better) (f) Frostbite (better) (g) Hero (better) (h) IceHockey (better)



(i) Kangaroo (worse) (j) Krull (better) (k) Montezuma (better) (l) Pong (slightly better)



(m) Riverraid (better) (n) Seaquest (worse) (o) Venture (worse) (p) Zaxxon (better)

Figure B-21: Additional results of SV-RL on dueling DQN.

B.6 Additional Empirical Study

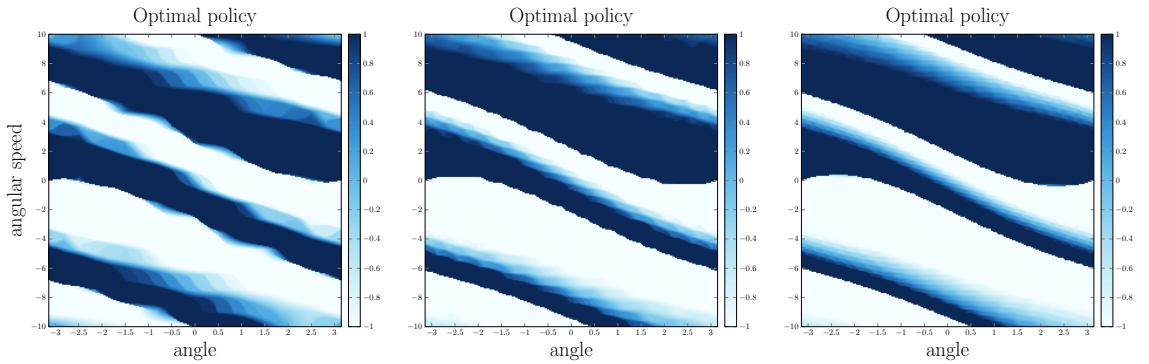
B.6.1 Discretization Scale on Control Tasks

We provide an additional study on the Inverted Pendulum problem with respect to the discretization scale. As described in Sec. 3.3, the dynamics is described by the angle and the angular speed as $s = (\theta, \dot{\theta})$, and the action a is the torque applied. To solve the task with value iteration, the state and action spaces need to be discretized into fine-grids. Previously, the two-dimensional state space was discretized into 50 equally spaced points for each dimension and the one-dimensional action space was evenly discretized into 1000 actions, leading to a 2500×1000 Q-value matrix. Here we choose three different discretization values for state-action pairs: (1) 400×100, (2) 2500×1000, and (3) 10000×4000, to provide different orders of discretization for both state and action values.

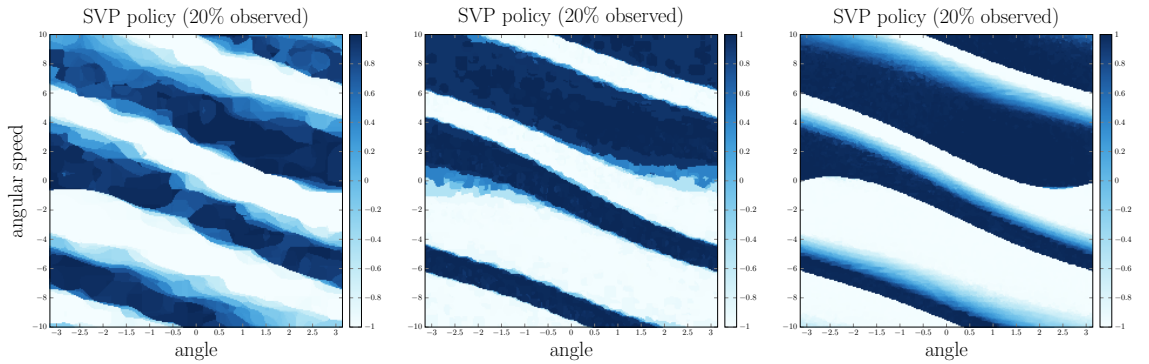
As Table B.1 reports, the approximate rank is consistently low when discretization varies, demonstrating the intrinsic low-rank property of the task. Fig. B-22 and Table B.1 further demonstrates the effectiveness of SVP: it can achieve almost the same policy as the optimal one even with only 20% observations. The results reveal that as long as the discretization is fine enough to represent the optimal policy for the task, we would expect the final Q matrix after value iteration to have similar rank.

Discretization Scale	**Approximate Rank**	**Average deviation (degree)**	
		Optimal Policy	SVP Policy
400×100	4	1.49	2.07
2500×1000	7	0.53	1.92
10000×4000	8	0.18	0.96

Table B.1: **Additional study on discretization scale.** We choose three different discretization value on the Inverted Pendulum task, i.e. 400 (states, 20 each dimension) $\times$ 100 (actions), 2500 (states, 50 each dimension) $\times$ 1000 (actions), and 10000 (states, 100 each dimension) $\times$ 4000 (actions). We report the approximate rank of the final Q matrix, as well as the performance metric (i.e., the average angular deviation) on the three different discretization scales.



(a) Optimal Policy (400×100) (b) Optimal Policy (2500×1000) (c) Optimal Policy (10000×4000)



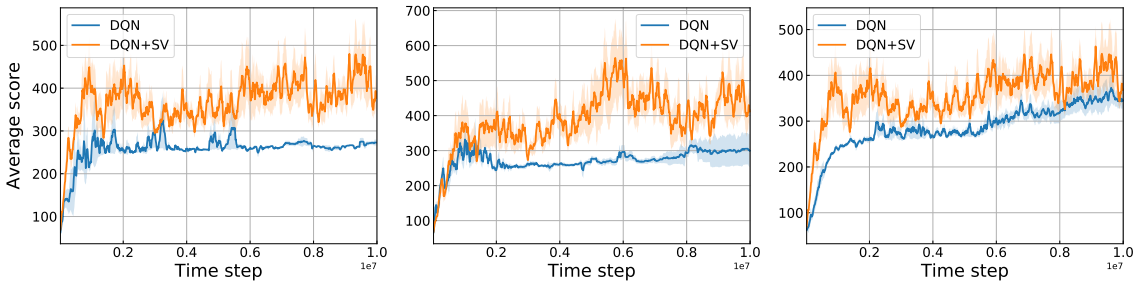
(d) SVP Policy (400×100) (e) SVP Policy (2500×1000) (f) SVP Policy (10000×4000)

Figure B-22: **Additional study on discretization scale.** We choose three different discretization value on the Inverted Pendulum task, i.e. 400 (states, 20 each dimension) $\times$ 100 (actions), 2500 (states, 50 each dimension) $\times$ 1000 (actions), and 10000 (states, 100 each dimension) $\times$ 4000 (actions). First row reports the optimal policy, second row reports the SVP policy with 20% observation probability.

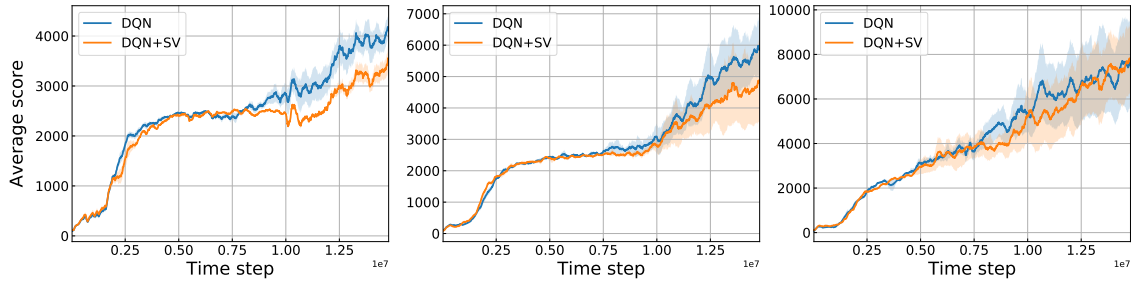
B.6.2 Batch Size on Deep RL Tasks

To further understand our approach, we provide another study on batch size for games of different rank properties. Two games from Fig. 3-7 are investigated; one with a small rank (Frostbite) and one with a high rank (Seaquest). Different batch sizes, 32, 64, and 128, are explored and we show the results in Fig. B-23.

Intuitively, for a learning task, the more complex the learning task is, the more data it would need to fully learn the characteristics. For a complex game with higher rank, a small batch size may not be sufficient to capture the game, leading the recovered matrix via ME to impose a structure that deviates from the original, more complex structure of the game. In contrast, with more data, i.e., a larger batch size, the ME oracle attempts to find the best rank structure that would effectively describe the rich



(a) Frostbite (batchsize = 32) (b) Frostbite (batchsize = 64) (c) Frostbite (batchsize = 128)



(d) Seaquest (batchsize = 32) (e) Seaquest (batchsize = 64) (f) Seaquest (batchsize = 128)

Figure B-23: **Additional study on batch size.** We select two games for illustration, one with a small rank (Frostbite) and one with a high rank (Seaquest). We vary the batch size with 32, 64, and 128, and report the performance with and without SV-RL.

observations and at the same time, balance the reconstruction error. Such a structure is more likely to be aligned with the underlying complex task. Indeed, this is what we observe in Fig. B-23. As expected, for Seaquest (high rank), the performance is worse than the vanilla DQN when the batch size is small. However, as the batch size increases, the performance gap becomes smaller, and eventually, the performance of SV-RL is the same when the batch size becomes 128. On the other hand, for games with low rank, one would expect that a small batch size would be enough to explore the underlying structure. Of course, a large batch size would not hurt since the game is intrinsically low-rank. In other words, our intuition would suggest SV-RL to perform better across different batch sizes. Again, we observe this phenomenon as expected in Fig. B-23. For Frostbite (low rank), under different batch sizes, vanilla DQN with SV-RL consistently outperforms vanilla DQN by a certain margin.

Bibliography

[1] Yuzhe Yang, Guo Zhang, Dina Katabi, and Zhi Xu. ME-Net: Towards effective adversarial robustness with matrix estimation. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[2] Yuzhe Yang, Guo Zhang, Zhi Xu, and Dina Katabi. Harnessing structures for value-based planning and reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2020.

[3] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[4] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[5] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018.

[6] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. 2018.

[7] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018.

[8] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

[9] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[10] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[12] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[13] Yuzhe Yang, Zhiwen Hu, Kaigui Bian, and Lingyang Song. ImgSensingNet: Uav vision guided aerial-ground air quality sensing system. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2019.

[14] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

[15] Alex A Gorodetsky, Sertac Karaman, and Youssef M Marzouk. Efficient high-dimensional stochastic optimal motion control using tensor-train decomposition. In *Robotics: Science and Systems*, 2015.

[16] Alex Gorodetsky, Sertac Karaman, and Youssef Marzouk. High-dimensional stochastic optimal control using continuous tensor decompositions. *The International Journal of Robotics Research*, 37(2-3):340–377, 2018.

[17] Alex Gorodetsky, Sertac Karaman, and Youssef Marzouk. A continuous analogue of the tensor-train decomposition. *Computer Methods in Applied Mechanics and Engineering*, 347:59–84, 2019.

[18] John Irvin Alora, Alex Gorodetsky, Sertac Karaman, Youssef Marzouk, and Nathan Lowry. Automated synthesis of low-rank control systems from sc-ltl specifications using tensor-train decompositions. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1131–1138. IEEE, 2016.

[19] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[20] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

[21] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[22] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[23] Yuzhe Yang, Zijie Zheng, Kaigui Bian, Lingyang Song, and Zhu Han. Real-time profiling of fine-grained air quality index distribution using uav sensing. *IEEE Internet of Things Journal*, 5(1):186–198, 2018.

[24] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[25] Yuzhe Yang, Zixuan Bai, Zhiwen Hu, Zijie Zheng, Kaigui Bian, and Lingyang Song. AQNet: Fine-grained 3d spatio-temporal air quality monitoring by aerial-ground wsn. In *IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 2018.

[26] Russ Tedrake. Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation. *Course Notes for MIT 6.832*, 2019.

[27] Shichao Yue, Yuzhe Yang, Hao Wang, Hariharan Rahul, and Dina Katabi. Bodycompass: Monitoring sleep posture with wireless signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(2), 2020.

[28] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*, 2015.

[29] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[30] Madeleine Udell and Alex Townsend. Why are big data matrices approximately low rank? *SIAM Journal on Mathematics of Data Science*, 1(1):144–160, 2019.

[31] Sourav Chatterjee et al. Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214, 2015.

[32] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

[33] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug):2287–2322, 2010.

[34] Yudong Chen and Yuejie Chi. Harnessing structures in big data via guaranteed low-rank matrix estimation. *arXiv preprint arXiv:1802.08397*, 2018.

[35] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 2010.

[36] Mark A Davenport and Justin Romberg. An overview of low-rank matrix recovery from incomplete observations. *arXiv preprint arXiv:1601.06422*, 2016.

[37] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *NIPS*, 2018.

[38] Christian Borgs, Jennifer Chayes, Christina E Lee, and Devavrat Shah. Thy friend is my friend: Iterative collaborative filtering for sparse matrix estimation. In *Advances in Neural Information Processing Systems*, pages 4715–4726, 2017.

[39] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[40] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[41] Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. *arXiv preprint arXiv:1802.05666*, 2018.

[42] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *arXiv preprint arXiv:1801.00553*, 2018.

[43] Minghao Guo, Yuzhe Yang, Rui Xu, Ziwei Liu, and Dahua Lin. When NAS meets robustness: In search of robust architectures against adversarial attacks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[44] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018.

[45] Jiefeng Chen, Xi Wu, Yingyu Liang, and Somesh Jha. Improving adversarial robustness by data-specific discretization. *CoRR*, abs/1805.07816, 2018.

[46] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[47] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018.

[48] Robert M. Bell and Yehuda Koren. Lessons from the netflix prize challenge. *SIGKDD Explor. Newsl.*, 9(2):75–79, December 2007.

[49] Edo M Airoldi, Thiago B Costa, and Stanley H Chan. Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. In *Advances in Neural Information Processing Systems*, pages 692–700, 2013.

[50] Emmanuel Abbe and Colin Sandon. Recovering communities in the general stochastic block model without knowing the parameters. In *Advances in neural information processing systems*, pages 676–684, 2015.

[51] Emmanuel Abbe and Colin Sandon. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 670–688. IEEE, 2015.

[52] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2013.

[53] Yudong Chen and Martin J Wainwright. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *arXiv preprint arXiv:1509.03025*, 2015.

[54] Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

[55] Lloyd N Trefethen and III David Bau. *Numerical linear algebra*. Society for Industrial and Applied Mathematics (SIAM), 1997.

[56] Simon J Julier and Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[57] Hao Yi Ong. Value function approximation via low-rank models. *arXiv preprint arXiv:1509.00061*, 2015.

[58] Yitao Liang, Marlos C Machado, Erik Talvitie, and Michael Bowling. State of the art control of atari games using shallow reinforcement learning. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 485–493. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

[59] Satinder Singh, Michael James, and Matthew Rudary. Predictive state representations: A new theory for modeling dynamical systems. *arXiv preprint arXiv:1207.4167*, 2012.

[60] Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1704–1713, 2017.

[61] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[62] Devavrat Shah, Qiaomin Xie, and Zhi Xu. On reinforcement learning using monte carlo tree search with supervised learning: Non-asymptotic analysis. *arXiv preprint arXiv:1902.05213*, 2019.

[63] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[64] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. *arXiv preprint arXiv:1806.06923*, 2018.

[65] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems*, pages 4026–4034, 2016.

[66] Georg Ostrovski, Marc G Bellemare, Aäron van den Oord, and Rémi Munos. Count-based exploration with neural density models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2721–2730. JMLR. org, 2017.

[67] Marius Mosbach, Maksym Andriushchenko, Thomas Trost, Matthias Hein, and Dietrich Klakow. Logit pairing methods can fool gradient-based attacks. 2018.

[68] Lijun Ding and Yudong Chen. The leave-one-out approach for matrix completion: Primal and dual analysis. *arXiv preprint arXiv:1803.07554*, 2018.

[69] Jianqing Fan, Weichen Wang, and Yiqiao Zhong. An ℓ_∞ eigenvector perturbation bound and its application to robust covariance estimation. *arXiv preprint arXiv:1603.03516*, 2016.

[70] Wei Ren and Randal W Beard. Consensus algorithms for double-integrator dynamics. *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*, pages 77–104, 2008.

[71] Andrew G Barto, Richard S Sutton, and Charles W Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, pages 834–846, 1983.

[72] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.