國立臺灣大學電機資訊學院電信工程學研究所
博士論文

Graduate Institute of Communication Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Doctoral Dissertation

適應性機器學習架構於電腦視覺與機器人學習之應用
Adaptive Machine Learning Pipelines for Computer Vision and Robotic Applications

陳尚甫
Shang-Fu Chen

指導教授：孫紹華 博士
Advisor: Shao-Hua Sun, Ph.D.

中華民國 113 年 12 月
December, 2024

doi:10.6342/NTU202404722

doi:10.6342/NTU202404722

國立臺灣大學博士學位論文
口試委員會審定書

DOCTORAL DISSERTATION ACCEPTANCE CERTIFICATE

NATIONAL TAIWAN UNIVERSITY

（論文中文題目）(Chinese title of Doctoral Dissertation)

適應性機器學習架構於電腦視覺與機器人學習之應用

（論文英文題目）(English title of Doctoral Dissertation)

Adaptive Machine Learning Pipelines for
Computer Vision and Robotic Applications

本論文係______陳尚甫_____(姓名)_____F07942144____（學號）在國立臺灣大學電信工程學研究所完成之博士學位論文，於民國__113__年__12__月__2__日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Graduate Institute of Communication Engineering on __2__(date)__12__(month)__2024__(year) have examined a Doctoral Dissertation entitled above presented by___Shang-Fu Chen__(name)__F07942144__ (student ID) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

[illegible] ______________ [illegible]

（指導教授 Advisor）

[illegible] [illegible] [illegible]

所長 Director: [illegible]

doi:10.6342/NTU202404722

致謝

完成本論文和博士學業，意味著即將要離開台大這個待了11年的母校了，這趟求學的旅程眞的讓我成長了很多。首先要感謝我的爸媽，從我小時候就不遺餘力地支持我的學業，更重要的是，給了我一個非常幸福的環境，讓我能夠快樂地成長並面對各種挑戰。

在博士的旅程中，謝謝Frank帶領我進入到AI的學術圈，讓我體驗到做研究和學術交流的樂趣，也因此認識了很多很棒的人們，對自己是這裡的一份子感到驕傲。謝謝Trista在我博士班的瓶頸時，願意和我分享許多寶貴的人生經驗，讓我在最迷惘的時候能重新找到未來的方向，給了我很大的鼓勵。謝謝紹華在我最需要的時候信任我，這段合作的時間，從你身上學到了很多比研究能力更加重要的事情，讓我對自己更有自信，能更勇敢的去追求自己想要的人生。也謝謝博士班期間的各位工作夥伴們，我了解自己是個很需要團隊、從合作中得到回饋的人，因爲有和你們的合作我才有辦法發揮出自己的潛力。

要特別謝謝我的未婚妻涵君的支持，從進台大第一年交往11年到現在即將結婚，這些年來能有人一起分享喜悅、分擔痛苦，成長過程有你陪在身邊眞的是一件很幸福的事情。接下來要踏入下一個人生階段，一定會有全新的壓力與期待，希望我們未來的日子能繼續是彼此最好的夥伴，一起體驗精彩的人生。最後要謝謝我自己，在面對各種挑戰、未知和壓力，都還是堅持到了最後。儘管過程中也有失去動力和迷惘的時候，但是你眞的很努力地完成一件很了不起的事情。希望未來你在面對困境的時候也能夠回想起這個經驗，就算身邊充滿了未知，但是你眞的夠好夠努力了，穩下腳步和心態，一定能夠撥雲見霧的。

doi:10.6342/NTU202404722

致謝

doi:10.6342/NTU202404722

中文摘要

雖然AI 模型已在各種應用中展現出卓越的效能，但在未知的真實場景中使用這些模型仍然是一項重大挑戰。本論文致力於開發一個機器學習架構，透過針對資料、模型與損失函數三個關鍵項目進行改進，以提升模型在這類環境中的適應能力。

一個典型的機器學習架構包含資料、神經網路模型，以及引導模型使用資料進行優化的損失函數。然而，在真實世界中，每一個項目都可能和理想的使用條件不同，因而需要進行適應才能有效運用這些模型。

在資料方面，預先收集的訓練樣本通常與實際使用時觀察到的資料分佈不同，因此需要使用一些技術來彌合這一差距。在模型方面，由於模型的訓練通常需要大量時間與計算資源，將預訓練模型適應於新任務可以顯著擴展其在不同領域中的應用能力。在損失函數方面，針對特定的應用學習損失函數，可以使模型更有效地利用該函數的優勢。

本論文聚焦於電腦視覺與機器人應用，通過探討特徵解耦、元學習、模型微調以及模仿學習等技術，提出針對上述挑戰的適應性解決方案。

關鍵詞：深度學習、電腦視覺、機器人學習、異常檢測、圖像生成、遷移學習、模仿學習

doi:10.6342/NTU202404722

doi:10.6342/NTU202404722

Abstract

While AI models have demonstrated remarkable effectiveness across various applications, deploying them in unstructured real-world scenarios remains a significant challenge. This thesis focuses on developing a machine learning pipeline designed to enhance adaptability in such environments by addressing three key dimensions: data, models, and learning objectives.

A typical machine learning pipeline consists of a dataset, a neural network model, and a learning objective that guides the model's optimization using the data. However, in real-world scenarios, each of these components may deviate from ideal conditions, necessitating adaptation for effective application.

For data, the distribution of pre-collected training samples often differs from the distribution encountered during inference, requiring strategies to bridge this gap. For models, since the training of a model typically requires substantial time and computational resources, adapting a pretrained model to new tasks significantly expands its applicability across diverse domains. For learning objectives, adapting the objective function to a particular application allows the model to leverage the advantages of the chosen objective more effectively.

This thesis focuses on computer vision and robotic applications and proposes adaptive solutions for the above challenges by exploring techniques such as feature disentanglement, meta-learning, model fine-tuning, and imitation learning.

Keywords: deep learning, computer vision, robot learning, anomaly detection, image generation, transfer learning, imitation learning

doi:10.6342/NTU202404722

Contents

致謝 i

中文摘要 iii

Abstract **v**

List of Figures **ix**

List of Tables **xi**

1 Adaption for Input Data **1**

1.1 Domain-Generalized Textured Surface Anomaly Detection 1

1.1.1 Introduction 1

1.1.2 Method 4

1.1.3 Experiments 9

1.1.4 Conclusion 15

2 Adaption for Pretrained Models **17**

2.1 Representation Decomposition for Image Manipulation and Beyond 17

2.1.1 Introduction 17

2.1.2 Decomposition-GAN for Disentanglement 19

2.1.3 Experiment 21

2.1.4 Quantitative Results 24

2.1.5 Conclusion 26

doi:10.6342/NTU202404722

2.2 Human-Feedback Efficient Online Diffusion Model Finetuning . . 26
2.2.1 Introduction 26
2.2.2 Related Works 30
2.2.3 Preliminaries 31
2.2.4 Problem Setup and the Proposed Method 32
2.2.5 Experimental 36
2.2.6 Ablations 40
2.2.7 Details of Tasks and Task Categories 44
2.2.8 HERO Implementation 46
2.2.9 Conclusion 47

3 Adaptation for Objective function 49
3.1 Diffusion Model-Augmented Behavioral Cloning 49
3.1.1 Introduction 49
3.1.2 Related Work 51
3.1.3 Preliminaries 52
3.1.4 Approach 54
3.1.5 Experiments 58
3.1.6 Discussion 67
3.2 Restoring Noisy Demonstration for Imitation Learnings 68
3.2.1 Introduction 68
3.2.2 Related Works 70
3.2.3 Method 74
3.2.4 Experiments 81
3.2.5 Discussion 91

4 Conclusion and Future Direction 93

Reference 95

doi:10.6342/NTU202404722

List of Figures

1.1 Illustration of domain-generalized anomaly detection 2

1.2 Overview domain-generalized textured surface anomaly detection model 4

1.3 Average image-level AUC for anomaly detection 10

1.4 Average pixel-level AUC for anomaly localization 10

1.5 Visualization of anomaly localization of our method on MVTec-AD 13

2.1 Overview and architecture of our Decomposition-GAN (dec-GAN) 19

2.2 Image generation via attribute swapping on MNIST 23

2.3 Image generation from CMU-MultiPIE via swapping the attributes 24

2.4 Model overview of human-feedback efficient online diffusion model finetuning 28

2.5 Result preview 29

2.6 Hand anomaly correction success rates 37

2.7 Qualitative results 39

2.8 Effect of best image ratio β evaluated on the `black-cat` task . . 42

2.9 Demonstration of personal preference transferability 44

2.10 Qualitative results for the NSFW content hidden task 45

3.1 Denoising Diffusion Probabilistic Model (DDPM) 54

3.2 Diffusion Model-Augmented Behavioral Cloning framework . . . 55

3.3 Environments & Tasks in DBC 59

3.4 Manifold overfitting Experiments 66

doi:10.6342/NTU202404722

3.5 Illustration of Human-Feedback-Efficient Reinforcement Learning for Online Diffusion Model Finetuning (DMDR) 69
3.6 Demonstration filtering 75
3.7 Demonstration restoration 78
3.8 Inference of demonstration restoration 79
3.9 Environments & Tasks in DMDR 82
3.10 Training progress of DMDR 85

doi:10.6342/NTU202404722

List of Tables

1.1 Domain-generalized anomaly detection results 11

1.2 FID scores between each data domain pair in MVTec-AD 14

2.1 Classification performances on CMU Multi-PIE 25

2.2 Task summary 38

2.3 Task performance 40

2.4 Representation learning and reward design ablation 40

2.5 HERO training parameters 47

3.1 Multimodality of Environments 62

3.2 Experimental Results of DBC 62

3.3 Generalization Experiments in FETCHPICK 65

3.4 Overall experiment results of DMDR 84

3.5 Demonstration filtering ablation study 88

3.6 Demonstration restoration ablation study 89

3.7 Imitation Learning Algorithms with DMDR 91

doi:10.6342/NTU202404722

Chapter 1

Adaption for Input Data

1.1 Domain-Generalized Textured Surface Anomaly Detection

1.1.1 Introduction

Textured surface anomaly detection has been a practical yet challenging task, which requires one to determine abnormal data from the normal ones. When it comes to real-world problems, e.g., quality control of industrial products, abnormal samples are generally difficult to collect. Therefore, existing solutions focus on training models that identify data that deviate from the learned distribution of normality as an anomaly. With the recent advances of deep learning, a popular model choice is the autoencoder [1, 2], which trains to recover normal data samples and thus performs anomaly detection by the associated reconstruction loss. To avoid the trained autoencoder from recovering abnormal samples as well, [3, 4] propose learning memory banks to regularize the autoencoder, ensuring the data is described by representative patterns. Despite the success of these reconstruction-based models, anomaly detection in unseen data domains is still difficult to perform. Moreover, one cannot expect the derived distribution of normality to be applicable to different domains for anomaly detection.

doi:10.6342/NTU202404722

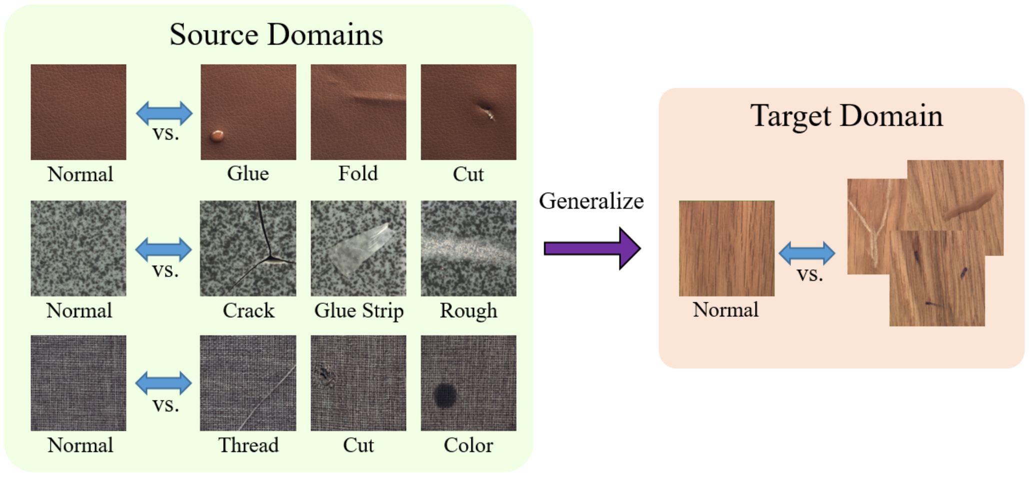


Figure 1.1: Illustration of domain-generalized anomaly detection. By observing normal and abnormal data in multiple source domains, the learned model needs to generalize to perform anomaly detection in an unseen target domain where only a small amount of normal images are available during testing.

Learning models from a single or multiple source domains, domain generalization [5, 6, 7] aims to leverage this model to unseen target domains for solving the same learning task. A straightforward yet naive baseline approach is to aggregate training samples from all source domains to learn a single model. To further improve the generalization capability, [8] designs an episodic learning procedure that simulates the domain shift observed during training for deriving a domain-generalized model. [6] argues that a properly learned domain generalization model would discover the image's intrinsic properties, which are irrelevant to the data domains. Thus, self-supervised auxiliary learning tasks are introduced to prompt the learning of their models.

Although the recent success of domain generalization has benefited a wide range of computer vision applications, it would not be feasible for anomaly detection if no normal data is presented in the domain of interest for a standard reference of normality. Thus, if one expects to address such tasks in an unseen target domain, at least a number of normal data in that domain needs to be observed during testing.

doi:10.6342/NTU202404722

In other words, existing domain generalization methods like [8, 6, 5, 7] cannot be easily applied for solving the above problem.

To address the above concerns and challenges, we tackle the task of domain-generalized textured surface anomaly detection in this paper. That is, with collection of training normal and abnormal data from existing source domains, i.e., textured surface, we aim to learn a model which can be generalized to detect abnormal data in *unseen* target domain of interest. The problem definition and the idea of our work can be seen in Figure 1.1. It is worth noting that, during the inference stage, only a small amount of normal samples are available for the target domain of interest, which follows the settings of most anomaly detection approaches [1, 3, 4, 9, 10]. However, without the requirement of model fine-tuning, the trained model can be directly applied to such data domains which are not seen during training.

To highlight the technical novelty of our work, we introduce a meta-comparer module that learns to compare textured surface data for anomaly detection across multiple source domains. We take the normal image data as the reference images and perform patch-level co-attention on the query-reference image pairs during training. With only image-level labels observed (i.e., normal and abnormal data), the above co-attention mechanism guides the meta-comparer to identify the normality of the query input, resulting in both image-level and patch-level anomaly detection. Since our model is trained to compare image pairs across different source domains in a meta-learning fashion, the learned model is shown to exhibit promising generalization ability for unseen data domains.

Our contributions can be summarized as follows:

- We address the task of domain-generalized textured surface anomaly detection. Given a number of normal (reference) images in unseen target domains, our model is able to perform anomaly detection accordingly.

- We propose a meta-learning framework that learns to compare images in a query-reference pair across multiple source domains. Therefore, our learned

doi:10.6342/NTU202404722

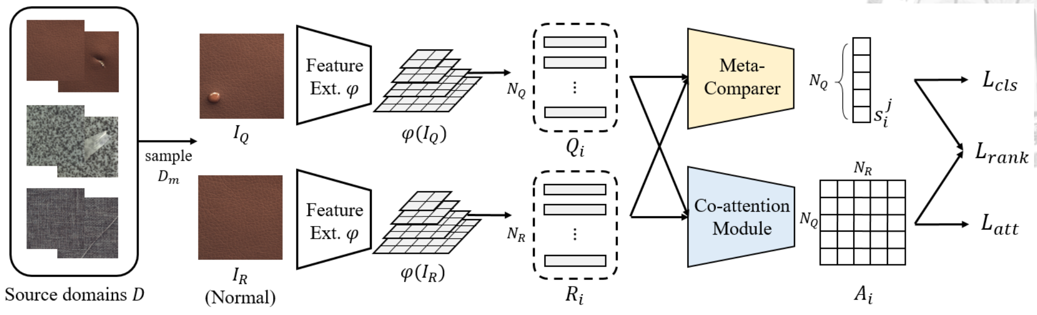


Figure 1.2: Overview of our proposed model, which consists of a feature extractor ϕ, a co-attention module, and a meta-comparer. The feature extractor aims to derive multi-scale patch-based features for both query and reference images. The co-attention module observes patches from such query-reference image pairs, guiding the meta-comparer to perform anomaly detection and localization.

model is able to generalize to unseen image domains for identifying abnormal images.

- With only image-level labels observed, a co-attention mechanism across query-reference image pairs is introduced, which guides our meta-comparer to realize not only image-level anomaly detection but also patch-level anomaly localization.

1.1.2 Method

For the sake of clarification, we first define the notations and setting considered in this paper. We observe image data from M source domains $D = [D_1, ..., D_M]$ at the training stage. Each D_m contains image-label pairs (x_m^n, y_m^n), in which y_m^n is either 0 or 1 representing normal or abnormal labels. Note that we assume that only image-level labels are available during training, i.e., no pixel-level anomaly ground truth can be observed. Our goal is to train a model using D in a meta-learning manner, and have this model generalized to perform anomaly detection on an unseen target domain D_{M+1} where only a number of normal images are available during testing.

doi:10.6342/NTU202404722

The overview of our proposed framework is depicted in Figure 1.2. From this figure, we see that our learning model contains three components: a feature extractor, a co-attention module, and a meta-comparer. The feature extractor ϕ aims to derive multi-scale features from query and reference (i.e., normal) images. The co-attention module observes query-reference image pairs, resulting in proper patch-level supervision, which guides the meta-comparer for producing the resulting anomaly score. By sampling different source domains D_m during the training stage, our meta-comparer learns to compare query-reference image data in a meta-learning fashion. In the following sections, we will detail the functionality and design of each module.

Multi-Scale Feature Extraction

In our proposed framework, the feature extractor is expected to extract the features from the query image I_Q and the reference image I_R from a domain of interest. We note that, while the query images are with labels $y = 1$ or 0 during training, we only consider the normal one as the reference for both training (from multiple source domains) and testing (on unseen target domains). Following techniques utilized for object detection (e.g., [11], [12]), we consider multi-scale features from image data for aiming at not only to recognize the abnormal query input, but also for the purpose of identifying the defect regions. More precisely, we apply the bi-directional feature pyramid network (BiFPN) proposed by [12] to produce a feature pyramid with multiple resolutions.

Take the query image I_Q as an example, the feature extractor ϕ extracts a feature pyramid containing L feature maps with different resolutions/scales. The associated multi-scale features are denoted as $\phi_1(I_Q), \phi_2(I_Q), ..., \phi_L(I_Q)$, where $\phi_i(I_Q)$ represents the query feature at scale level i. Let N_Q denotes the number of patches sampled from $\phi_i(I_Q)$, we thus have a set of patch-based representations $Q_i = \{q_i^1, q_i^2, ..., q_i^{N_Q}\}$ for the query image I_Q at scale level i. Similarly, we have $R_i = \{r_i^1, r_i^2, ..., r_i^{N_R}\}$ as the set of patch-based representations for the reference

doi:10.6342/NTU202404722

image I_R at scale level i, where N_R denotes the number of sampled patches. For the detailed process of the multi-scale feature extraction, please refer to the supplementary materials.

Image-Level Anomaly Detection

With patch features extracted from the query and reference images, we now explain how we train our feature extractor and meta-comparer for performing image-level anomaly detection. For the j-th query patch q_i^j at scale level i, the meta comparer is utilized to calculate its largest query-reference anomaly score s_i^j as:

$$s_i^j = \max_{k=[1,\ldots,N_R]} \text{MLP}([q_i^j, r_i^k]), \tag{1.1}$$

where MLP denotes a multilayer perceptron module with Sigmoid activation functions deployed. It can be expected that, if the query image I_Q is abnormal, at least one query patch q_i^j would remarkably deviate from the reference patches, and thus the value of the corresponding s_i^j would be close to 1.

With the above observation, we define the image-level classification loss (under supervision of y) as follows:

$$L_{cls} = -\sum_{i=1}^{L} y \log(\max_j(s_i^j)) + (1-y)\log(1-\max_j(s_i^j)). \tag{1.2}$$

In the above equation, $\max_j(s_i^j)$ calculates and outputs the largest anomaly score from the query patches at scale i, which sums over all L scales for the resulting loss output.

Patch-level Anomaly Localization

In addition to image-level anomaly detection, the introduced co-attention module in our framework of Figure 1.2 allows us to perform the same task at patch level. Therefore, localization of abnormal surface regions can be achieved via patch-level anomaly detection with only image-level label y required.

doi:10.6342/NTU202404722

Co-attention on query-reference image pairs: The co-attention module first maps the query-reference patch pairs (i.e., q_i^j and r_i^k) at scale i into a shared latent space, followed by the calculation of cosine similarity between them. This produces a co-attention matrix $A_i \in \mathbb{R}^{N_Q \times N_R}$, which can viewed as an affinity matrix of Q_i and R_i at scale i, reflecting the similarity between the associated patch pairs.

Similar to image-level anomaly detection, we observe that if the query image I_Q is abnormal, then there would exist at least one query patch q_i^j which would be distinct from the reference ones r_i^k. That is, if $y = 1$ for the query, we expect at least one query-reference patch pair in A_i resulting in a low similarity score. On the other hand, if $y = 0$ for the query, every query-reference pair is expected to produce a large similarity score. Thus, by normalizing the attention matrix A_i to $[0, 1]$, we introduce and calculate the following attention loss L_{att} across image scales,

$$L_{att} = -\sum_{i=1}^{L} y \log(1 - \min_{j,k}(a_i^{j,k})) + (1 - y) \log(\min_{j,k}(a_i^{j,k})), \tag{1.3}$$

where $\min_{j,k}(a_i^{j,k})$ denotes the query-reference patch pair at scale i with the minimum similarity score. Note that j and k are the patch indices for the query and reference images, respectively.

With the above co-attention mechanism, we calculate the co-attention score for q_i^j as $a_i^j = \max_k(a_i^{j,k})$. In the formula, a_i^j calculates the score between q_i^j and every reference patch, and outputs the score with the most similar reference patch as the attention guidance. It can be expected that, if the query patch q_i^j is abnormal, such a_i^j scores would be close to 0 (and vice versa). Therefore, the co-attention score a_i^j can be a patch-level guidance for the query patch q_i^j.

From patch-level co-attention to anomaly localization: In our proposed framework, patch-level anomaly detection is achieved by sampling pairs of patches q_i^u, q_i^v from a query Q_i at scale i, followed by the meta-comparer to produce their patch-level anomaly scores s_i^u, s_i^v under the supervision of y and the guidance of the

doi:10.6342/NTU202404722

aforementioned co-attention outputs. Inspired by [13], we introduce a patch-level anomaly ranking loss L_{rank} for the sampled query patch pairs as follows,

$$\begin{aligned} L_{rank} &= \sum_{i=1}^{L} \sum_{q_i^u, q_i^v \in Q_i} w_i^{uv} \max(0, 1 - \sigma(s_i^u - s_i^v)), \\ \text{where} \quad w_i^{uv} &= \lambda(\exp(|a_i^u - a_i^v|) - 1), \\ \text{and} \quad \sigma &= -sgn(a_i^u - a_i^v). \end{aligned} \tag{1.4}$$

Note that λ is a scaling factor, and sgn indicates the sign function that extracts the sign of a real number. From equation (1.4), we see if both q_i^u, q_i^v are the normal patches, both co-attention scores a_i^u and a_i^v would be large, and the corresponding w_i^{uv} is close to 0. This would result in the ranking loss L_{rank} close to 0 as well. Similarly, if both are the abnormal ones, we have similar yet small a_i^u and a_i^v values, which produces small w_i^{uv} regularizing the ranking loss as well. Finally, and most importantly, if only one of q_i^u and q_i^v is abnormal, we would observe very different co-attention score a and thus produce a large w_i^{uv}. If the co-attention score a_i^u is less than a_i^v, the corresponding anomaly score s_i^u should be larger than s_i^v. To ensure this property, the variable σ verifies the order of s_i^u and s_i^v according to their corresponding co-attention score a. With the goal of anomaly localization, the above objective allows us to automatically identify the query patch that deviates not only from the reference ones but also from the remaining ones in the query.

Pipeline

With the introduced image-level detection and patch-level localization discussed above, we now explain how our proposed framework is trained to exhibit additional domain generalization ability. During training, by sampling query-reference image pairs (I_Q, I_R) from multiple source domains, we enforce the meta-comparer and the co-attention module for learning to compare image data by applying equation (1.2) and equation (1.3), disregard of the data domain distributions. Moreover, by sampling different query patch pairs q_i^u and q_i^v in Equation (1.4), our meta-comparer further performs the above learn-to-compare scheme in the patch level. Therefore,

doi:10.6342/NTU202404722

our model is expected to learn a generalized capability of comparing image data. The full objectives of our model and the detailed training process are summarized in the Algorithm A of our supplementary materials.

As for the inference stage, we apply our model to an unseen target domain D_{M+1} with a small amount of normal samples are presented.

We first calculate the patch-level anomaly score s_i^j for each extracted query patch q_i^j. If there exists a patch with defect regions at any scale, the query image is considered to be abnormal. Therefore, the image-level prediction $\hat{y}(I_Q)$ for I_Q can be calculated by simply taking the maximum anomaly scores among all query patches q_i^j:

$$\hat{y}(I_Q) = \max(\{s_i^j\}) \quad \forall i, j. \tag{1.5}$$

If localization of defect regions would be needed, we can calculate the anomaly score for each pixel p in I_Q according to patch-level anomaly scores across multiple scale levels. This is realized by taking the maximum anomaly scores among all query patches containing pixel p:

$$\hat{y}(p) = \max(\{s_i^j\}) \quad \forall i, j \text{ such that } p \in q_i^j. \tag{1.6}$$

1.1.3 Experiments

We evaluate our proposed framework on MVTec-AD [14] and BTAD [15] datasets. The MVTec-AD dataset consists of 3,629/1,725 training/testing images from 5 texture and 10 object products. In this paper, we consider the texture products of MVTec-AD for textured surface anomaly detection, i.e., Carpet, Grid, Leather, Tile, and Wood, as shown in Figure 1.5. For these 5 texture types, we follow recent domain generalization approaches [8] and [6] and do leave-one-out evaluation, in which only one texture is selected at a time as the target domain at the inference stage, while the remaining four textures are used as the source domains during training. Following previous works [3, 4, 15, 16], we evaluate the models using the area under the receiver operating characteristic curve (AUC).

doi:10.6342/NTU202404722

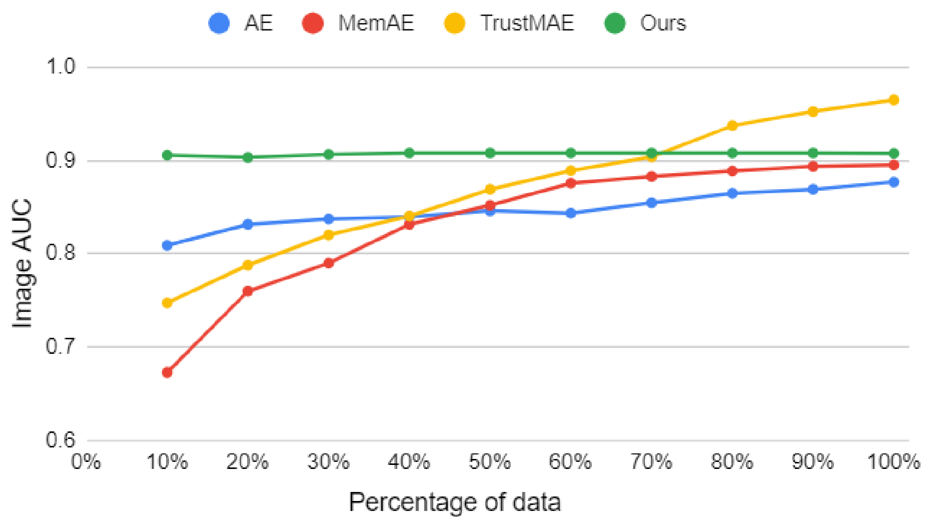


Figure 1.3: Average image-level AUC for anomaly detection over the 5 textures of MVTec-AD, with different percentages of normal reference images from the target domain.

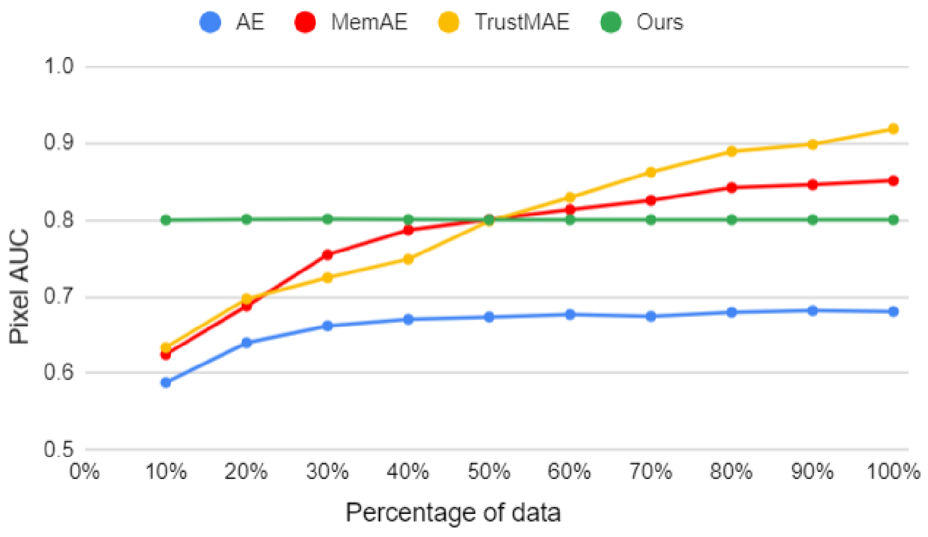


Figure 1.4: Average pixel-level AUC for anomaly localization over the 5 textures of MVTec-AD, with different percentage of normal reference images from the target domain.

	Carpet	Grid	Leather	Tile	Wood	Avg.
AGG [8]	0.875	0.628	0.981	0.886	0.852	0.845
Epi-FRC [8]	0.916	0.640	0.995	0.947	0.909	0.881
EISNet [6]	**0.991**	0.662	1.000	0.850	**0.986**	0.898
AGG+	0.891	0.608	0.992	0.912	0.865	0.854
Epi-FRC+	0.916	0.725	1.000	0.951	0.941	0.907
EISNet+	0.982	0.728	1.000	0.858	0.979	0.909
Ours	0.943	**0.730**	1.000	**0.956**	0.962	**0.918**

Table 1.1: Domain-generalized anomaly detection on MVTev-AD with the leave-one-domain-out setting in terms of the average image-level AUC. Note that the $+$ notation denotes the modified versions for existing DG approaches (i.e., with learn-to-compare scheme introduced).

As for the BTAD dataset, it consists of 2,250/291 normal/abnormal images from 3 industrial products. The image data from this dataset would serve as the (unseen) target domains for testing in experiments for the cross-dataset settings, which would further verify the effectiveness of our propose method for domain-generalized anomaly detection. The implementation details and the results of the cross-dataset experiments are demonstrated in the supplementary materials.

Quantitative Results

In our experiments, we compare our model with a number of recent anomaly detection (AD) and domain generalization (DG) approaches. For fair comparisons, we adopt the same pre-trained ResNet-18 feature extractor for all the methods considered. Moreover, to comply with our domain-generalized anomaly detection setting, we allow AD and DG methods to take normal image data from the target domain as additional inputs during the inference stage as well.

Comparisons to existing AD Approaches: We compare our model AD approaches, including an autoencoder baseline [3] as well as two state-of-the-art

doi:10.6342/NTU202404722

methods of MemAE [3] and TrustMAE [4]. We follow the officially-released code and the instruction presented in the paper to implement the above methods. A common limitation of existing AD approaches is that a sufficient amount of training data from the domain of interest would be needed. As noted in previous sections, existing anomaly detection approaches use all the available normal images from the target domain *for training*. On the other hand, our model does not require any normal image data in the target domain for training, and only observes such data as references *during inference*. We compare our method to these AD approaches on MVTec-AD with same amount of target normal samples are observed. With the aforementioned leave-one-domain-out setting, we control the percentage of the amount of target normal samples and compare the average image-level AUC for anomaly detection and pixel-level AUC for anomaly localization in Figure 1.3 and Figure 1.4, respectively. As can be seen from these two figures, existing AD approaches required a sufficient amount of normal training data in the target domain (e.g., above 60 or 70% of the target-domain normal data available) to achieve satisfactory performances, while our method consistently outperformed such methods especially even with only 10% (i.e., about 25 images) of such data were observed. This is expected since our proposed model only utilizes the target normal samples as reference during inference. Therefore, the performance of our method is not sensitive to the amount of such data, which would be preferable for practical uses.

Compare with existing DG Approaches: As for recent DG approaches, we consider a baseline of simple aggregation of AGG [8], and two state-of-the-art methods of Epi-FCR [8] and EISNet [6] for comparisons. We note that, existing DG models generally make prediction solely based on the query image, not in the learn-to-compare fashion as ours does. Thus, for fair comparison, we additionally modify the above DG approaches to take query-reference pairs as training inputs, and such modified versions are denoted as $+$ in our results presented in Table 1.1. We also note that, for fair comparisons, all target-domain normal reference images

doi:10.6342/NTU202404722

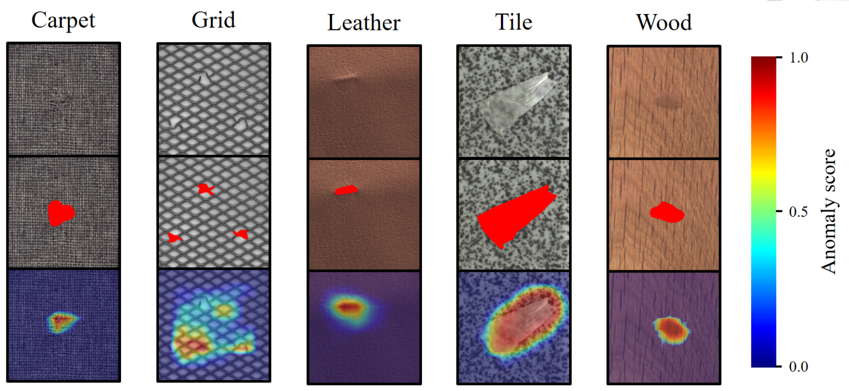


Figure 1.5: Visualization of anomaly localization of our method on MVTec-AD. The top row shows input abnormal images, the middle row indicates the ground truth defect regions, and the bottom row shows our anomaly localization results.

are utilized for all DG methods and ours in the experiments.

From the results listed in Table 1.1, we see that our method performed favorably against existing DG approaches (for both the original and the modified learn-to-compare versions) over all 5 texture categories in terms of the average AUC. It is interesting to point out that, from the results shown in this table, the modified versions of recent DG approaches (i.e., with learn-to-compare mechanism introduced) were shown to produce improved performances when comparing to their original versions. This suggests that by a properly designed learn-to-compare scheme as ours is, the anomaly detection model can be expected to generalize to unseen target domains. It can be seen that our model outperforms all existing DG approaches by a large margin. It is expected since our model explores the relationships between patch features for detecting sophisticated defects, while the above methods only consider image-level features for anomaly detection.

Visualization of Anomaly Detection

As discussed in Section 2, our proposed model not only performs anomaly detection but also exhibits abilities in identifying abnormal regions with only *image-level* labels observed during training. We show the visualization results for anomaly

	Carpet	Grid	Leather	Tile	Wood
Carpet	0	4.424	1.34	1.763	1.526
Grid	4.424	0	3.966	4.646	4.409
Leather	1.34	3.966	0	1.916	1.601
Tile	1.763	4.646	1.916	0	2.032
Wood	1.526	4.409	1.601	2.032	0
Average	1.811	3.489	1.765	2.071	1.914

Table 1.2: FID scores between each data domain pair in MVTec-AD, which imply the difficulty expected for domain-generalized anomaly detection.

localization in Figure 1.5. The top row of this figure shows input images containing defects; the middle row are the ground truth regions of defects (annotated in red); the bottom row shows the anomaly localization results predicted by our model. It can be seen that, from the example results shown in this figure, our model is able to accurately localize either small defects (in Carpet and Wood) or large defects (in Tile). It is also worth noting that, existing AD or DG approaches cannot easily address such anomaly localization without proper pixel-level guidance.

Further Analysis and Remarks

To further verify the capability and point out the limitation of our domain generalization method, we quantitatively assess the domain differences between different texture categories from MVTec-AD, reflecting the expected DG difficulty for the associated target domain. To analyze the above issue, we apply the Fréchet Inception Distance (FID) score introduced by [17] to calculate the differences between each texture/domain pair and list the results in Table 1.2.

From Table 1.2, we see that the Grid texture generally has larger FID scores (average 3.489) than those of other texture types, suggesting that the distribution of Grid deviates more drastically from those of other texture category data. This observation is consistent with the AUC results shown in Table 1.1, where all DG methods (including ours) did not report comparable performances when Grid was

doi:10.6342/NTU202404722

the unseen target domain of interest. On the other hand, since the average FID of Leather is the smallest, the knowledge learned by the model from other source domains is expected to generalize data in this domain, which also explains why all DG methods reported the highest AUC performances in Table 1.1. In other words, while we claim that our model can be generalized to an unseen target domain for anomaly detection, the performance drop would be expected if the target domain data distribution were very different from those of source domain data.

1.1.4 Conclusion

In this paper, we tackle the task of domain-generalized anomaly detection. With only image-level labels observed for multiple source domains, our model learns to compare images in query-reference pairs across the above data domains during training. With the co-attention mechanism introduced, our model learns to compare and identify abnormal image data and the associated defect regions, and it is shown to achieve promising performances on anomaly detection and localization for unseen target domain data.

doi:10.6342/NTU202404722

doi:10.6342/NTU202404722

Chapter 2

Adaption for Pretrained Models

2.1 Representation Decomposition for Image Manipulation and Beyond

2.1.1 Introduction

Recent developments of Generative Adversarial Network (GAN) [18] models result in promising progress and achievements in image generation. In order to produce image outputs with desirable attributes (e.g., gender, expression, etc.), feature disentanglement aims at decomposing the above latent representation into distinct parts, each corresponding to particular properties. Representation disentanglement [19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29] aims at learning an interpretable representation from image variants, which can be realized in unsupervised or supervised settings. For example, with supervision of labeled data, AC-GAN [20] factorizes representations into disjoint parts describing visual content and attribute information, respectively (e.g., image [30], text [31, 32]) during training. On the other hand, if training data are unlabeled, infoGAN [19] performs representation disentanglement by maximizing the mutual information between latent variables and data variation.

Despite promising performances, these works are not able to be directly applied

doi:10.6342/NTU202404722

on *existing/pre-trained* generative models. In other words, their disentanglement mechanisms must be determined and trained in advance. More specifically, their need to decide on image attributes to be disentangled beforehand makes their feature disentanglement less flexible. If the attributes of interest are changed, the above generative models need to be trained from scratch again. Moreover, with the scale of generative models growing, training of state-of-the-art generative models becomes very time and resource-consuming.

Instead of explicitly decomposing latent representation into disjoint parts, we propose a unique decomposition-GAN (dec-GAN) for performing feature disentanglement. Our disentanglement mechanism focuses on extracting attributes of interest (e.g., pose, expression, etc.) from latent representation, while the generator is *fixed*. Depending on the attribute of interest, dec-GAN is guided by an attribute classifier trained to distinguish the attribute. Together with image recovery objectives, dec-GAN decomposes visual features from a joint latent representation into separate ones associated with content and attribute of interest. While recent works like [33] and [34] deal with the similar task that learns disentangled features based on existing generative models, both of their methods are not able to manipulate particular attributes of interest when taking images as input. On the other hand, with the above disentangled features, our dec-GAN is able to utilize existing generative models for describing each type of disentangled features, which allows improved and interpretable feature representations for image manipulation, along with additional flexibility in determining the attributes of interest after the generator is trained.

We now highlight the contributions of work as follows:

- We propose a novel learning scheme for representation disentanglement, which uniquely decomposes features of existing GAN-based models into interpretable representations.
- Our learning framework does *not* require pre-determined or disjoint latent representations to describe attributes in advance and thus exhibits additional

doi:10.6342/NTU202404722

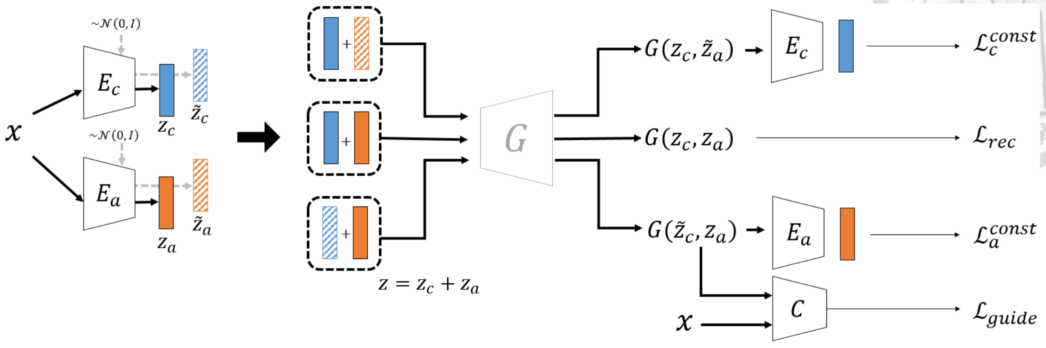


Figure 2.1: Overview and architecture of our Decomposition-GAN (dec-GAN), which consists of content encoder E_c, attribute encoder E_a, and an auxiliary guidance attribute classifier C, while generator G is fixed. Note that our dec-GAN decomposes latent features z into separate representations (instead of disjoint ones), i.e., $z = z_c + z_a$. Note that $G(z_c, \tilde{z}_a)$, $G(z_c, z_a)$ and $G(\tilde{z}_c, z_a)$ indicate the image outputs synthesized from pairs of the associated content and attribute features.

flexibility in determining the attributes of interest.

- Our experiments confirm that our model successfully decomposes latent features derived by existing GANs for image manipulation and classification.

2.1.2 Decomposition-GAN for Disentanglement

We propose *decomposition-GAN (dec-GAN)* for representation disentanglement. As illustrated in Figure 2.1, our dec-GAN decomposes the latent code z into content code z_c and attribute code z_a while satisfying $z = z_c + z_a$. In other words, based on existing latent feature z, our goal is to decompose it into content and attribute representations z_c and z_a. We utilize two separate encoders E_c and E_a for extracting z_c and z_a, respectively. The reconstruction output is denoted as $G(z_c, z_a) = G(z|z = z_c + z_a)$. It is worth noting that, as verified in Section 2.1.3, our dec-GAN can utilize existing state-of-the-art generative models.

Attribute Guidance for Disentanglement

In our dec-GAN, we first utilize the idea of data recovery to encourage generated images to be sufficiently realistic. For this reconstruction loss, we consider the L1 distance between the reconstructed and input images:

$$L_{rec} = |G(z_c, z_a) - x|. \tag{2.1}$$

Following VAE-GAN [35] and DRIT [36], we fit the distributions of encoded content and attribute features to normal distributions, which allow improved/continuous data representation ability. This can be achieved by minimizing the Kullback–Leibler divergence (KLD) between each distribution and $\mathcal{N}(0, 1)$. However, since the disentangled content and attribute features describe distinct information, we do not expect them to fit the same normal distribution. Therefore, we calculate the KLD loss for each feature as follows,

$$L_{KL,c} = \mathbb{E}[KL(P(z'_c)||\mathcal{N}(0, 1))], z_c = E_c^{fc}(z'_c), \tag{2.2}$$

$$L_{KL,a} = \mathbb{E}[KL(P(z'_a)||\mathcal{N}(0, 1))], z_a = E_a^{fc}(z'_a), \tag{2.3}$$

where E_c^{fc} denotes the final fully connected layer of content encoder E_c, and E_a^{fc} denotes the final fully connected layer of attribute encoder E_a.

To ensure the encoded z_c and z_a describing content and attribute information, respectively, we apply a classifier C pre-trained on the attribute of interest to guide the learning of E_a. Thus, this guided loss is calculated as:

$$L_{guide} = |C(x) - C(G(\tilde{z}_c, z_a))|, \tag{2.4}$$

where $C(\cdot)$ indicates the classifier. We note that, $\tilde{z}_c$ denotes a randomly sampled content feature, $\tilde{z}$ is sampled from $\mathcal{N}(0, 1)$ and then is passed through the final fully connected layer of E_c. Thus, we have $\tilde{z}_c = E_c^{fc}(\tilde{z})$, and the image with identical attribute but random content can be produced as $G(\tilde{z}_c, z_a)$.

From (2.4), we see that the enforcement of classification output similarity between an input image x and a synthesized one with the same z_a yet with a

doi:10.6342/NTU202404722

random content $\tilde{z}_c$, would ensure our E_c and E_a to extract attribute-invariant and attribute-dependent representations, respectively. That is, the deployment of the classifier $C(\cdot)$ in Fig. 2.1 would guide the attribute encoder E_a to extract attribute-dependent information by equation (2.4). With equation (2.1) ensuring the quality of reconstruction, attribute-invariant information would be encoded by content encoder E_c for fair reconstruction.

Enforcing Content and Attribute Consistency

With the above guidance of the attribute classifier and the use of generative models, we have E_a extract latent attribute features. With this classifier to be replaced by those pre-trained on preferable attributes of interests, one can easily extend the above architecture to disentangle the corresponding attributes. To further ensure our decomposed z_c and z_a from z contain only content and attribute information, respectively, we advance feature consistency losses during the training of our dec-GAN. This is achieved by minimizing the content and attribute feature consistency loss defined as follows:

$$L_c^{const} = |E_c(G(z_c, \tilde{z}_a)) - z_c|, \tag{2.5}$$

$$L_a^{const} = |E_a(G(\tilde{z}_c, z_a)) - z_a|. \tag{2.6}$$

As illustrated in Figure 2.1, $G(z_c, \tilde{z}_a)$ indicates the synthesized image with the same content as that of input x but with different attributes $\tilde{z}_a = E_a^{fc}(\tilde{z})$. Similarly, we have $G(\tilde{z}_c, z_a)$ denote the generated image with the same attributes as those of x but with different content information via $\tilde{z}_c$. By observing the above feature consistency, both E_c and E_a would extract associated content and attribute features, realizing the decomposition of z into z_c and z_a, respectively.

2.1.3 Experiment

We consider image datasets of MNIST [37] and CMU Multi-PIE [38] for our experiments. The former consists of 60,000/10,000 training/test digit images of 10

doi:10.6342/NTU202404722

classes, while the latter contains face images with multiple viewpoint, illumination and expression variations. We only use a subset of CMU Multi-PIE with 5 viewpoints and smiling expression variation, which consists of 68,810 images.

For the generator to be decomposed, since our proposed architecture does not limit the use of particular GAN models, we first follow the backbone of VAE-GAN [35]. In addition, we consider a second generative model with a deeper backbone [36] and refer Res-GAN to this generative model. The encoder and the generator of Res-GAN consist of convolution layers and residual blocks. For the detail of the architecture of VAE-GAN and Res-GAN, please refer to the supplementary material. For E_c and E_a in our dec-GAN, we simply utilize the same encoder structure of the model to be decomposed.

Image Generation and Manipulation

MNIST: For MNIST, the classifier C is pre-trained to identify the digit categories, which are viewed as the attributes, while the visual appearance, like stroke thickness or angle, is used as the content features. As shown in Figure 2.2, we demonstrate our image generation results using different pairs of content and attribute features. The first row in Figure 2.2 shows input image pairs, and the second row depicts reconstructed outputs using derived z_c and z_a features. Image outputs by swapping z_c and z_a are shown in the third row. From this row, we see that the synthesized image would preserve the same content as those in the first two rows, while the attribute (digit category) would match the other one in the input image pair. This confirms the effectiveness of our dec-GAN in disentangling content and attribute features, while the latter is guided by a digit classifier in this case. To further verify z, z_c and z_a capture different visual information, we conduct t-SNE visualization on such features using MNIST.

CMU Multi-PIE:

We take both VAE-GAN and Res-GAN as the backbone of our dec-GAN, and consider pose and expression (smile) as two distinct attributes of interest. We

doi:10.6342/NTU202404722

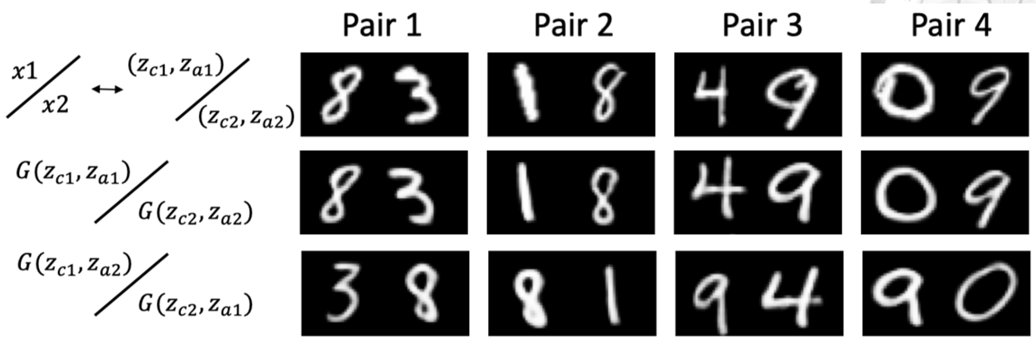

Pair 1
Pair 2
Pair 3
Pair 4
$x1$ / $x2$ ↔ (z_{c1}, z_{a1}) / (z_{c2}, z_{a2})
$G(z_{c1}, z_{a1})$ / $G(z_{c2}, z_{a2})$
$G(z_{c1}, z_{a2})$ / $G(z_{c2}, z_{a1})$

Figure 2.2: Image generation via attribute swapping on MNIST. Note that x indicates the input image, with the outputs G produced by the associated z_c and z_a.

demonstrate image generation results when taking pose categories as attributes of interest in Figure 2.3(a). The first row in Figure 2.3(a) shows input facial image pairs, and the second row depicts reconstructed image outputs using derived z_c and z_a. Image outputs by swapping z_c and z_a are shown in the third row. Comparing this row and the first two rows, we see that the manipulated facial images remained the same identity, with pose information altered and matched to the other one in the input image pair. Compared to discrete categorical attributes in MNIST, this confirms that our dec-GAN is able to handle continuous attributes such as poses.

In addition, we show the results when taking smiling expression as an attribute of interest in Figure 2.3(b). Similarly, by comparing the last row and the first two rows, we see that the manipulated images retain the same facial information, with only the smiling expressions altered. This confirms that smiling expression is able to be decomposed from the original latent feature. It is worth noting that we decompose pose and smiling attributes from the same pre-trained generative models, confirming the flexibility of our dec-GAN in extracting the attributes of interest.

doi:10.6342/NTU202404722

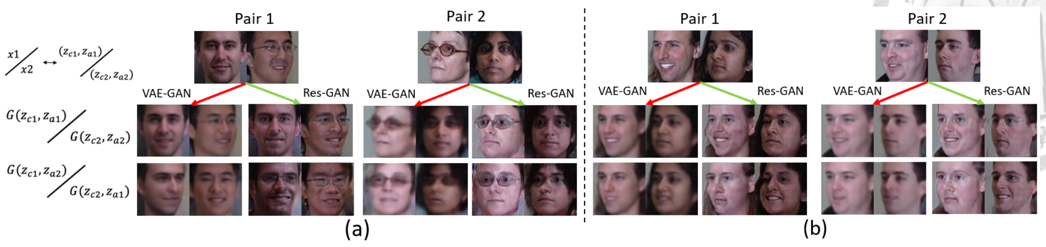

Pair 1
Pair 2
Pair 1
Pair 2
$x1$ / $x2$ $\leftrightarrow$ (z_{c1}, z_{a1}) / (z_{c2}, z_{a2})
$G(z_{c1}, z_{a1})$ / $G(z_{c2}, z_{a2})$
$G(z_{c1}, z_{a2})$ / $G(z_{c2}, z_{a1})$
VAE-GAN
Res-GAN
(a)
(b)

Figure 2.3: Image generation from CMU-MultiPIE via swapping the attributes of (a) pose and (b) smile. The first row shows sampled input image pairs x_1 and x_2, the second row shows reconstructed image outputs $G(z_c, z_a)$ of the input images, and the third row depicts generated image outputs by swapping z_a in each pair. Comparing the second and the third row, we see that the image content is preserved while the attributes (i.e., pose/smile information) are swapped within each image pair. Note that results using VAE-GAN and Res-GAN as the backbones of our dec-GAN are shown.

2.1.4 Quantitative Results

Quantitative Evaluation of z_c and z_a

We conduct quantitative experiments to examine the effectiveness of our dec-GAN in disentangling content and attribute features. With the use of CMU Multi-PIE face dataset, z_a derived by our model would be expected to contain pose information only, while z_c represents pose-invariant identity features. We then take these two types of features, perform pose and ID classification tasks, and compare the results to the uses of latent representations z derived by VAE-GAN [35] and UFDN [29].

Table 2.1 lists and compares classification results of different tasks using z, z_c and z_a. We simply apply a two-layer classifier (i.e., 2 fully connected layers with ReLU activation, followed by a softmax layer) for comparison purposes. We do not apply additional or complex classifiers, which can possibly further improve the recognition performances. The number of classes is 5 for pose classification and 249 for identity classification. From this table, we observe that z_a yielded the

doi:10.6342/NTU202404722

Method	Pose	ID
VAE-GAN [35]	97.44 (z)	96.94 (z)
UFDN [29]	N/A	94.31 (z_c)
Ours	**99.74** (z_a)	**98.59** (z_c)

Table 2.1: Classification performances on CMU Multi-PIE. Note that our method decomposes content and attribute features (z_c and z_a) from latent representation z derived by pre-trained VAE-GAN. Since the attribute feature of UFDN [29] is a hand-crafted one-hot vector, it cannot be directly applied for pose classification.

best result in pose classification, while z_c resulted in the highest performances for identity classification. This is expected since our dec-GAN is particularly designed to disentangle attribute-dependent and attribute-invariant features. Note that the use of z of VAE-GAN achieved inferior results, indicating that its latent representation would contain both content and attribute information and thus cannot be expected to sufficiently address either task. For UFDN, since they derive hand-crafted one-hot vector for attribute features, their model is not applicable for pose classification.

Comparisons of Training Time

As noted earlier, a major advantage of our dec-GAN is the applicability to existing GAN-based models without the need for pre-defined attributes. We compare the numbers of training iterations and computation times of dec-GAN and AC-GAN [20] with the same backbone structures for generators and discriminators. Note that all experiments were conducted on a single NVIDIA GTX 1080 Ti with batch size = 12, and the table of results is presented in the supplementary material.

We found that dec-GAN is four times faster than AC-GAN when disentangling smiling attributes (i.e., 6 vs. 26 mins) and is about seven times faster when disentangling pose attributes (i.e., 13 vs. 89 mins). This is because the training of our dec-GAN can be initialized by existing GAN models, followed by the training of E_c and E_a. On the other hand, AC-GAN needs to pre-define additional

doi:10.6342/NTU202404722

dimensions to describe the attribute so that its training can not be initialized. From the above experiments, the flexibility and effectiveness of our dec-GAN can be confirmed.

2.1.5 Conclusion

In this paper, we proposed a unique decomposition-GAN (dec-GAN) to perform feature disentanglement, which jointly extracts content and attribute representations from the latent feature observed from existing GAN-based models. Different from prior disentanglement works, which typically derive disjoint latent representations describing desirable features, our dec-GAN performs feature decomposition, which separates latent representation into separate features describing the properties/attributes of interest. The attribute disentanglement of our dec-GAN is driven by classifiers pre-trained on the attribute of interest. Followed by the design of generative network modules, this allows disentanglement of content and attributes while exhibiting additional flexibility in determining the attributes of interest (i.e., by replacing such classifiers based on the desirable attribute categories). We performed qualitative and quantitative evaluations using multiple image datasets, with attributes ranging from digit categories to pose angles. The effectiveness and robustness of our dec-GAN can be successfully confirmed, while its superiority over existing models can also be verified.

2.2 Human-Feedback Efficient Online Diffusion Model Finetuning

2.2.1 Introduction

Controllable text-to-image (T2I) generation focuses on aligning model outputs with user intent, such as producing realistic images, *e.g.*, undistorted human bodies, or accurately reflecting the count, semantics, and attributes specified by users. To

doi:10.6342/NTU202404722

tackle this problem, a common paradigm involves fine-tuning latent diffusion models (DM) like Stable Diffusion (SD) [39] using supervised fine-tuning (SFT) [40], which mostly learn from pre-collected, offline datasets. To further enhance the alignment, online reinforcement learning (RL) fine-tuning methods [41, 42] utilize online feedback that specifically evaluates the samples generated by the model during training. With such dynamic guidance provided on the fly, these methods demonstrate superior performance on various T2I tasks, such as aesthetic quality improvement. Yet, these approaches rely on either predefined heuristic reward functions or pretrained reward models learned from large-scale datasets, which could be challenging to obtain, especially for tasks involving personalized content generation (*e.g.*, capturing cultural nuances) or concepts like specific colors or compositions.

To address the above issue, [43] introduces D3PO, an alternative method that directly leverages online human feedback for fine-tuning diffusion models. Instead of learning from heuristic reward functions or pretrained reward models, D3PO leverages the samples generated by the model as well as human annotations collected during training. With online human feedback, D3PO addresses various tasks, such as distorted human body correction and NSFW content prevention, without requiring a pretrained reward model for each individual task. However, it still necessitates approximately 5K instances of online human feedback during training [43, 44], placing a significant burden on the human evaluator and restricting the use of customized fine-tuning to match individual preferences.

To further improve the feedback efficiency of T2I alignment using online human feedback, this work proposes a **H**uman-feedback **E**fficient **R**einforcement learning for **O**nline diffusion model fine-tuning framework, dubbed **HERO**, to efficiently and effectively utilize online human feedback to fine-tune a SD model, as illustrated in Figure 2.4. Specifically, we propose two novel components: (1) *Feedback-Aligned Representation Learning*, an online-trained embedding map that creates a representation space that implicitly captures human preferences

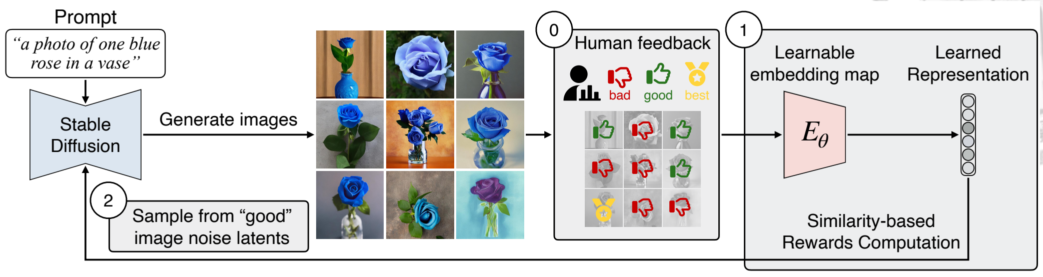


Figure 2.4: ⓪ **Online Human Feedback on Generated Images:** Each epoch, SD generates a batch of images, evaluated by a human as "good" or "bad", with the "best" among the "good" selected. The corresponding SD noises and latents are saved. ① **Feedback-Aligned Representation Learning:** Human-annotated images train an embedding map via contrastive learning, converting feedback into continuous representations. These are rated by cosine similarity to one of the "best" images and used to fine-tune SD via DDPO [42]. ② **Feedback-Guided Image Generation:** New images are generated from a Gaussian mixture centered around the recorded noises of "good" images. This process is repeated until the feedback budget is exhausted.

and provides continuous reward signals for RL fine-tuning, and (2) *Feedback-Guided Image Generation*, which involve generating images from SD's refined initialization samples aligned with human intent, for faster convergence to the evaluator's preferences.

Feedback-aligned representation learning (Fig. 2.4's ①) aims to create a representation space that implicitly reflects human preferences, offering continuous reward signals for RL fine-tining. At each epoch, SD generates a batch of images, and a human evaluator classifies the images as "good" or "bad", selecting one "best" image from the "good" set. The latents of the human-annotated images are then employed to train an embedding map through contrastive learning [45], aiming to develop a feedback-aligned representation space. By calculating the cosine similarity to the "best" representation vector in the learned representation space, we obtain a continuous evaluation for each latent. Subsequently, we utilize the

doi:10.6342/NTU202404722

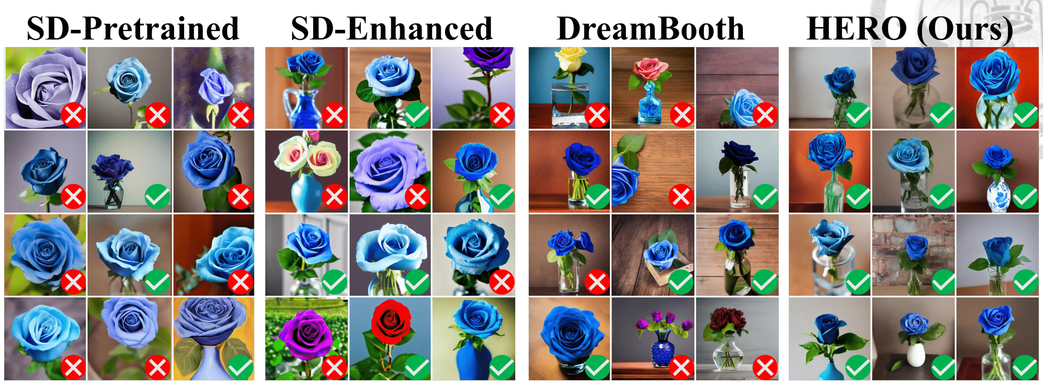


Figure 2.5: **Result preview.** Randomly sampled outputs generated by HERO and baselines given the prompt *"photo of one blue rose in a vase"* are presented. Successful samples are marked with ✓, and unsuccessful samples are marked with ✗, which fails to accurately capture the specified count (more than one rose), color (non-blue roses), and context (missing vase). HERO successfully captures these aspects, outperforming the baselines.

computed similarity as continuous reward signals to fine-tune SD via LoRA [46].

After fine-tuning the SD for the first iteration, our feedback-guided image generation (Fig. 2.4's ②) samples a new batch of images from a Gaussian mixture centered on the stored "good" and "best" initial noises from the previous iteration. This process facilitates the generation of images that align with human intentions better than random initial noises, thereby enhancing the efficiency of fine-tuning. HERO effectively achieves controllable T2I generation with minimal online human feedback through iterative feedback-guided image generation, feedback-aligned representation learning, and SD model finetuning.

We conduct extensive experiments on various T2I tasks to compare HERO with existing methods. The experimental results show that HERO can effectively fine-tune SD to reliably follow given text prompts with $4\times$ fewer amount of human feedback compared to D3PO [43]. On the other hand, the results show that these tasks are difficult to solve through prompt enhancement [47] or fine-tuning approaches, *e.g.*, DreamBooth [48], that rely on a few reference images [49].

Figure 2.5 presents a preview of the results. Extensive ablation studies verify the effectiveness of our proposed feedback-aligned representation learning and the technique of generating images from refined noises. Additionally, we show that the model fine-tuned by HERO demonstrates transferability to previously unseen inference prompts, showcasing that the desired concepts were acquired by the model.

2.2.2 Related Works

Recent research has explored controllable generation with SD for tasks like T2I alignment [42, 50], conceptual generation [51, 52], correcting generation flaws [53], personalization [49, 48] and removing NSFW content [54, 55, 56].

Supervised fine-tuning. DreamBooth (DB) [48] and Textual Inversion [49] take images as input and fine-tunes SD via supervised learning to learn the specific subject present in the input images. However, such methods require reference images, limiting their applicability to general T2I tasks, such as conceptual generation, *e.g.*, emotional image content generation [51], or accurately reflecting user-specified counts, semantics, and attributes [57]. On the other hand, [50, 54, 58, 59] use pretrained reward models to calculate differentiable gradients for SD fine-tuning. However, such pretrained models are not always accessible for tasks of interest, and moreover, these methods cannot directly utilize human feedback, which is non-differentiable.

RL fine-tuning. Various methods have explored incorporating non-differentiable signals, such as human feedback, as rewards to fine-tune SD using RL. For example, DDPO [42] uses predefined reward functions for tasks like compressibility, DPOK [41] leverages feedback from an AI model trained on a large-scale human dataset, and SEIKO [44] obtain rewards from custom reward functions trained from extensive feedback datasets. Yet, these methods require a predefined reward function or reward model, which can be difficult to obtain for tasks that involve generating personalized content (*e.g.*, reflecting cultural nuances) or abstract concepts,

doi:10.6342/NTU202404722

such as specific colors or compositions [60, 61].

Direct preference optimization (DPO). Diffusion-DPO [62] applies DPO [63] to directly utilize preference data to fine-tune SD, eliminating the need for predefined rewards. Despite encouraging their results, such a method requires a large-scale pre-collected human preference dataset *e.g.*, Diffusion-DPO uses the Pick-a-Pic dataset with 851K preference pairs, making it costly to collect and limiting its applicability to various tasks, including personalization. Instead of leveraging offline datasets, D3PO [43] uses *online human feedback* collected on the fly during model training for DPO-style finetuning of SD. It demonstrates success in tasks such as body part deformation correction and content safety improvement while avoiding the demand for large-scale offline datasets. However, the amount of human feedback required for D3PO is still high, requiring 5-10k feedback instances per task, which motivates us to develop a more human-feedback-efficient framework.

2.2.3 Preliminaries

Stable Diffusion (SD). operates in two stages. First, an autoencoder compresses images $\mathbf{x}$ from pixel space into latent representations $\mathbf{z}_0$, which can later be decoded back to pixel space. Second, a diffusion model (DM) is trained to model the distribution of these latent representations conditioned on text $\mathbf{c}$. The forward diffusion process is defined as $p(\mathbf{z}_t|\mathbf{z}_0) := \mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{z}_0, \sigma_t^2 \mathbf{I})$, where α_t and σ_t are pre-defined time dependent constants for $t \in [0, T]$. Both the forward transition kernel $p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{c})$ and the backward conditioned transition kernel $p(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{c}, \mathbf{z}_0)$ are Gaussian with closed-form expressions. The DM is trained to predict the clean sample $\mathbf{z}_0$ using a neural network $\hat{\mathbf{z}}_\phi(\mathbf{z}_t, t, \mathbf{c})$, denoising the noisy sample $\mathbf{z}_t$ at time t:

$$p_\phi(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{c}) := p\Big(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{c}, \mathbf{z}_0 := \hat{\mathbf{z}}_\phi(\mathbf{z}_t, t, \mathbf{c})\Big)$$

by optimizing the following objective:

$$\min_\phi \mathbb{E}_{\mathbf{z}_0, \mathbf{c}, \boldsymbol{\epsilon}, t}\Big[\|\hat{\mathbf{z}}_\phi(\alpha_t \mathbf{z}_0 + \sigma_t \boldsymbol{\epsilon}, t, \mathbf{c}) - \mathbf{z}\|_2^2\Big], \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

doi:10.6342/NTU202404722

At inference, random noise $\mathbf{z}_T$ is sampled from a prior and iteratively denoised using samplers like DDPM [64] and DDIM [65] to obtain a latent code $\mathbf{z}_0$, which is then decoded into an image. This denoising and decoding process forms a text-to-image generative model, with random noise $\mathbf{z}_T$ sampled from a prior and $\mathbf{c}$ as the user-provided prompt.

Denoising Diffusion Policy Optimization (DDPO). formulates the denoising process of diffusion models as a multi-step Markov decision process. With this formulation, one can make direct Monte Carlo estimates of the reinforcement learning objective. Given a denoising trajectory $\{\mathbf{z}_T, \mathbf{z}_{T-1}, ..., \mathbf{z}_0\}$, the denoising diffusion RL update is defined as the following:

$$\nabla_\phi \mathcal{L}_{\text{DDRL}}(\phi) = \mathbb{E}\left[\sum_{t=0}^{T} \nabla_\phi \log p_\phi(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{c}) r(\mathbf{z}_0, \mathbf{c})\right], \tag{2.7}$$

where ϕ is the diffusion model, and $r(\mathbf{x}_0, \mathbf{c})$ is the received reward computed according the output image $\mathbf{x}_0$ and the input prompt $\mathbf{c}$. Based on the above update, DDPO further utilizes the importance sampling estimator [66] and the trust region clipping from Proximal Policy Optimization (PPO) [67] to perform multiple steps of optimization while maintaining the diffusion model ϕ not deviating too far from the previous iteration ϕ_{old}. The DDPO update is defined as the following:

$$\nabla_\phi \mathcal{L}_{\text{DDPO}}(\phi) = \mathbb{E}\left[\sum_{t=0}^{T} \frac{p_\phi(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{c})}{p_{\phi_{\text{old}}}(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{c})} \nabla_\phi \log p_\phi(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{c}) r(\mathbf{z}_0, \mathbf{c})\right]. \tag{2.8}$$

2.2.4 Problem Setup and the Proposed Method

Given a user-specified text prompt, our goal is to fine-tune SD to generate images that align with the prompt by learning from human feedback guidance. In this paper, we focus on challenging T2I tasks that require spatial reasoning, counting, feasibility understanding, etc., as detailed in Table 2.2. To efficiently and effectively utilize online human feedback, we propose a **h**uman-feedback **e**fficient **r**einforcement learning for **o**nline diffusion model fine-tuning framework, dubbed **HERO**, as illustrated in Figure 2.4. *Feedback-Aligned Representation Learning* (Figure 2.4

①) makes efficient use of limited human feedback by converting discrete feedback to informative, continuous reward signals. In addition, *Feedback-Guided Image Generation* (Figure 2.4 ②) leverages human-preferred noise latents from previous iterations and encourages SD outputs to align more quickly with human intention, further improving sample efficiency.

Online Human Feedback

In the first iteration of HERO, we generate synthetic images $\mathcal{X}$ from a batch of random noises $\mathcal{Z}_T$ sampled from SD's prior distribution $\pi_{\text{HERO}}(\mathbf{z}_T) := \mathcal{N}(\mathbf{z}_T; \mathbf{0}, \mathbf{I})$ using DDIM [65, 64]. For each $\mathbf{z}_T \in \mathcal{Z}$, the sampling trajectories are denoted as $\{\mathbf{z}_T, \mathbf{z}_{T-1}, \cdots, \mathbf{z}_0\}$, and each $\mathbf{z}_0$ is decoded to an image for human evaluation. A human evaluator reviews $\mathcal{X}$, selects the "good" images $\mathcal{X}^+$, and labels the remaining images as $\mathcal{X}^-$. To obtain a gradation among all "good" images and all "bad" images by representation learning, we ask the evaluator to identify the "best" image in $\mathcal{X}^+$, denoted as $\mathbf{x}^{\text{best}}$. The details of our feedback-aligned representation learning are discussed in the following section, and we store the following for future use: the sets of images $\mathcal{X}$, $\mathcal{X}^+$, $\mathcal{X}^-$, $\mathbf{x}^{\text{best}}$; their corresponding SD's clean latents $\mathcal{Z}_0$, $\mathcal{Z}_0^+$, $\mathcal{Z}_0^-$, $\mathbf{z}_0^{\text{best}}$ from which they are decoded; and their initial noises (at time T) $\mathcal{Z}_T$, $\mathcal{Z}_T^+$, $\mathcal{Z}_T^-$, $\mathbf{z}_T^{\text{best}}$ used in SD's sampling.

Feedback-Aligned Representation Learning

HERO fine-tunes SD with minimal online human feedback by learning representations via a contrastive objective that captures discrepancies between the best SD's clean latent $\mathbf{z}_T^{\text{best}}$, positive $\mathcal{Z}_0^+$, and negative $\mathcal{Z}_0^-$ SD's clean latents. By calculating similarity to the best image's representation, we use these similarity scores as continuous rewards for RL fine-tuning. This approach bypasses reward model training by directly converting human feedback into learning signals, avoiding the need for over 100k training samples typically required to train a reward model for unseen data [62, 63]. **Learning Representations:** To learn a representation space of $\mathcal{Z}_0$

doi:10.6342/NTU202404722

aligned with human feedback, we build on the contrastive learning framework of [45]. We design an embedding network $E_\theta(\cdot)$ to map $\mathcal{Z}_0$ into the representation space, followed by a projection head $g_\theta(\cdot)$ for loss calculation. Triplet margin loss is applied to the projection head's output:

$$\begin{aligned}\mathcal{L}(\theta; \mathbf{z}_0^{\text{best}}, \mathcal{Z}_0^+, \mathcal{Z}_0^-) = \mathbb{E}_{\mathbf{z}_0^{\text{good}} \sim \mathcal{Z}_0^+, \mathbf{z}_0^{\text{bad}} \sim \mathcal{Z}_0^-} \max\Big\{ & S\Big(g_\theta\Big(E_\theta(\mathbf{z}_0^{\text{best}})\Big), g_\theta\Big(E_\theta(\mathbf{z}_0^{\text{good}})\Big)\Big) \\ & - S\Big(g_\theta\Big(E_\theta(\mathbf{z}_0^{\text{best}})\Big), g_\theta\Big(E_\theta(\mathbf{z}_0^{\text{bad}})\Big)\Big) + \alpha, 0\Big\}.\end{aligned} \tag{2.9}$$

$E_\theta(\mathbf{z}_0^{\text{best}})$ serves as the anchor in the contrastive loss, with $S(\cdot,\cdot)$ representing the similarity score (using cosine similarity) and α as the triplet margin set to 0.5. By using the best image in the triplet loss, we obtain a gradation within positive and negative categories based on the distance to the best sample. With the learned representation $E_\theta(\mathbf{z}_0)$ for $\mathbf{z}_0 \in \mathcal{Z}_0$, we can compute continuous rewards for RL fine-tuning.

Similarity-based Rewards Computation: After training the embedding $E_\theta(\cdot)$ on the current batch of human feedback, reward values are computed as the cosine similarity in the learned representation space between each $E_\theta(\mathbf{z}_0)$ for $\mathbf{z}_0 \in \mathcal{Z}_0$ and $E_\theta(\mathbf{z}_0^{\text{best}})$:

$$R(\mathbf{z}_0) = \frac{E_\theta(\mathbf{z}_0) \cdot E_\theta(\mathbf{z}_0^{\text{best}})}{\max\left\{ \|E_\theta(\mathbf{z}_0)\|_2 \left\|E_\theta(\mathbf{z}_0^{\text{best}})\right\|_2, \delta\right\}} \quad \text{for each } \mathbf{z}_0 \in \mathcal{Z}_0, \tag{2.10}$$

where $\delta = 1 \times 10^{-8}$ to avoid zero division. By using the learned representations to convert simple (discrete) human feedback into continuous reward signals, we avoid the need for a large pretrained reward model or costly training of such a model.

Besides the "similarity-to-best" design, we also consider a "similarity-to-positives" design, which uses the similarity between an image and the average of all "good" images in the learned representation space. We choose the "similarity-to-best" design for its superior performance. Further discussion is available in Section 2.2.6.

Diffusion Model Finetuning: DDPO fine-tunes SD by reweighting the likelihood with reward values. For a noise latent $\mathbf{z}_T \in \mathcal{Z}_T$ and its sampling trajectory

$\{\mathbf{z}_T, \mathbf{z}_{T-1}, \cdots, \mathbf{z}_0\}$, we incorporate the reward $R(\mathbf{z}_0)$ from Eq. (2.10) into the DDPO update rule in Eq. (2.8) to fine-tune the SD model ϕ. To reduce costly gradient computations, we adopt LoRA [46] for fine-tuning.

Feedback-Guided Image Generation

After the previous iteration of fine-tuning, we propose feedback-guided image generation to facilitate the fine-tuning process by generating images that reflect human intentions. We sample the noise latents for a new batch of images from the Gaussian mixture with means centered around the human-selected "good" $\mathcal{Z}_T^+$ and "best" $\mathbf{z}_T^{\text{best}}$ SD noise latents from the previous iteration, with a small variance ε_0. Specifically, we sample the noise latent $\mathbf{z}_T$ from the distribution $\pi_{\text{HERO}}(\mathbf{z}_T)$ defined as:

$$\pi_{\text{HERO}}(\mathbf{z}_T) = \begin{cases} \mathcal{N}(\mathbf{z}_T; \mathbf{0}, \mathbf{I}), & \text{first iteration} \\ \beta\mathcal{N}(\mathbf{z}_T; \mathbf{z}_T^{\text{best}}, \varepsilon_0^2\mathbf{I}) + \frac{(1-\beta)}{|\mathcal{Z}_T^+|}\sum_{\mathbf{z}_T^{\text{good}} \in \mathcal{Z}_T^+} \mathcal{N}(\mathbf{z}_T; \mathbf{z}_T^{\text{good}}, \varepsilon_0^2\mathbf{I}) & \text{otherwise.} \end{cases} \tag{2.11}$$

Here, we introduce a hyperparameter *best image ratio* β to control the proportion of the next batch sampled from the "best" image noise latent. We find that leveraging $\mathbf{z}_T^{\text{best}}$ with a larger β can accelerate training convergence to evaluator preferences but may reduce the diversity or the converged accuracy. The above tradeoff can be controlled by the best image ratio β. We generally set $\beta = 0.5$ to balance these effects. Further discussion on the *best image ratio* parameter is in Section 2.2.6.

We remark that since the variance ε_0 is small, after a few iterations, samples from $\pi_{\text{HERO}}(\mathbf{z}_T)$ still concentrate near the prior $\mathcal{N}(\mathbf{z}_T; \mathbf{0}, \mathbf{I})$ at high probability. Also, $\mathbf{z}_T^{\text{good}}$ and $\mathbf{z}_T^{\text{best}}$ may retain semantic information about human alignment from $\mathbf{z}_0^{\text{good}}$ and $\mathbf{z}_0^{\text{best}}$, as they are connected through the finite-step discretization of the SD sampler. Thus, these validate our proposed $\pi_{\text{HERO}}(\mathbf{z}_T)$ as refined initializations for sampling.

Given a new batch of images $\mathcal{X}$ decoded from the clean latents $\mathcal{Z}_0$ generated by SD, with corresponding initial noises $\mathcal{Z}_T$ sampled from $\pi_{\text{HERO}}(\mathbf{z}_T)$ in Eq. (2.11),

doi:10.6342/NTU202404722

the human evaluator provides their evaluation as described in Section 2.2.4. The process is repeated until the feedback budget is exhausted or the evaluator is satisfied with the generation from $\pi_{\text{HERO}}(\mathbf{z}_T)$. After obtaining the fine-tuned SD model ϕ and $\pi_{\text{HERO}}(\mathbf{z}_T)$ through HERO, we use SD random noises from refined $\pi_{\text{HERO}}(\mathbf{z}_T)$ and generate images using any DM sampler [65].

2.2.5 Experimental

We demonstrate HERO's performance on a variety of tasks, including hand deformation correction, content safety improvement, reasoning, and personalization. Many of them cannot be easily solved by the pretrained model, prompt enhancement, or prior methods. A full list of tasks and their success conditions are shown in Table 2.2. We adopt SD v1.5 [39] as the base T2I model, using DDIM [64, 65] with 50 diffusion steps (20 for hand deformation correction for fair comparison to the baselines) as the sampler.

We compare HERO to the following baselines:

- **SD-pretrained** prompts the pretrained SD model with the original task prompt shown in Table 2.2.
- **SD-enhanced** prompts the pretrained SD model with an enhanced version of the prompt generated by GPT-4 [68, 69].
- **DreamBooth (DB)** [48] finetunes diffusion models via supervised learning, taking images as input. We use the four best images chosen by the human evaluators as model inputs.
- **D3PO** [43] utilize online human feedback for DPO [63]-based diffusion model finetuning. Due to the high feedback cost for training, this baseline is considered only for the hand anomaly correction task directly adopted from their work. Success rates are reported as presented in the original paper.

doi:10.6342/NTU202404722

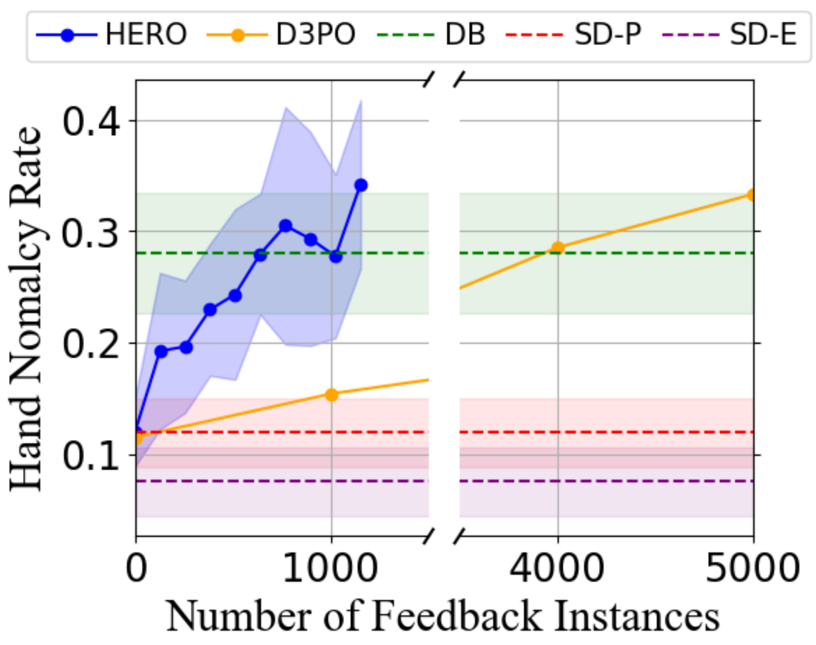


Figure 2.6: **Hand anomaly correction success rates.** The performance of methods except D3PO is an average of 8 seeds, where each seed is evaluated on 128 images per epoch. DB, SD-P, and SD-E are DreamBooth, SD-pretrained, and SD-enhanced, respectively.

Hand Deformation Correction

Following the problem setup of D3PO [43], we use the prompt *"1 hand"* for image generation and use human discretion to evaluate the normalcy of the generated hand images. Parameters such as sampling steps are set to be consistent with D3PO. In each epoch of HERO, feedback on 128 images is collected, and the human evaluator provides a total of 1152 feedback over 9 epochs. Performance of HERO in comparison to the baselines is shown in Figure 2.6. As shown in Figure 2.6, the pretrained SD model struggles on this task, with a normalcy rate of 11.9% (SD-pretrained) and 7.5% (SD-enhanced), and DB achieves 28%. D3PO reaches 33.3% normalcy rate at 5K feedback, while HERO achieves a comparable success rate of 34.2% with only 1152 feedback (over 4$\times$ more feedback efficient).

doi:10.6342/NTU202404722

Demonstration on the Variety of Tasks

Table 2.2: Task summary

Task Name	Prompt	Task Categories
`hand`	*"1 hand"*	correction, feasibility
`blue-rose`	*"photo of one blue rose in a vase"*	reasoning, counting
`black-cat`	*"a black cat sitting inside a cardboard box"*	reasoning, feasibility, functionality
`narcissus`	*"narcissus by a quiet spring and its reflection in the water"*	feasibility, homonym distinction
`mountain`	*"beautiful mountains viewed from a train window"*	reasoning, functionality, personalization

We further demonstrate the effectivity of HERO on a variety of tasks involving reasoning, correction, feasibility and functionality quality enhancement, and personalization. Tasks are listed in Table 2.2, and descriptions of task success conditions and task categories are found in Section 2.2.7. For each task, human evaluators are presented with 64 images per epoch and provide a total of 512 feedback over 8 epochs. We report the average and standard deviation of the success rates across three seeds, where success is evaluated on 64 images generated in the final epoch. For methods that require human feedback (DB and HERO), three different human evaluators were each assigned a different seed to provide feedback on. Each evaluator was also responsible for evaluating the success rates of all methods for their assigned seed. Results are shown in Table 2.3. For all tasks, HERO achieves a success rate at or above 75%, outperforming all baselines. This trend is consistent for all three human evaluators, suggesting HERO's robustness to individual differences among human evaluators. Sample images generated by SD-pretrained, DB, and HERO are shown in Figure 2.7. While the baselines often struggle in attribute reasoning (*e.g.*, color, count), spatial reasoning (*e.g.*, inside), and feasibility (*e.g.*, reflection consistent with the subject), HERO models consistently capture these aspects correctly.

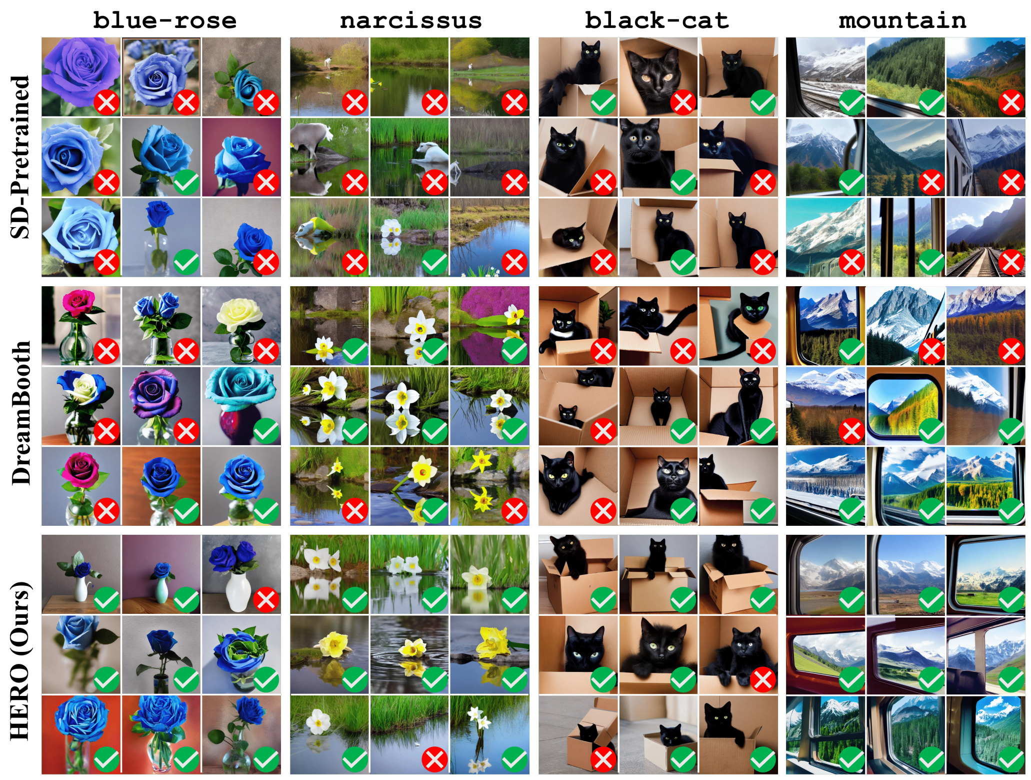


Figure 2.7: **Qualitative results.** The randomly generated samples for the four tasks are shown, with ✓ denoting successful samples and ✗ for failures. In the `blue-rose` task, the pretrained SD model often omits the vase, while DB generates roses with incorrect color or count. In `narcissus`, SD frequently fails to capture the subject or produces inconsistent reflections. For `black-cat`, baseline models exhibit more issues (*e.g.*, the cat's body penetrating the box). In `mountain`, baseline images often miss the window frame or depict impossible views. Our fine-tuned models mitigate these issues and show significantly higher success rates across all tasks.

Table 2.3: **Task performance.** Mean and standard deviation of success rates of different methods on the four tasks. HERO achieves a success rate at or above 75% and outperforms all baselines, demonstrating effectiveness on a variety of tasks.

Method	`blue-rose`	`black-cat`	`narcissus`	`mountain`
SD-Pretrained	0.354 (0.020)	0.422 (0.092)	0.406 (0.077)	0.412 (0.063)
SD-Enhanced	0.479 (0.030)	0.365 (0.134)	0.276 (0.041)	0.938 (0.022)
DB	0.479 (0.085)	0.453 (0.142)	0.854 (0.092)	0.922 (0.059)
HERO (ours)	**0.807** (0.115)	**0.750** (0.130)	**0.912** (0.007)	**0.995** (0.007)

2.2.6 Ablations

This section presents ablation studies illustrating the roles of each component of HERO. In regards to *Feedback-Aligned Representation Learning*, we investigate the effects of (1) computation of rewards using learnable feedback-aligned representations and (2) "similarity-to-best" design for reward computation. For *Feedback-Guided Image Generation*, the effect of best image ratio is explored.

Effect of Feedback-Aligned Representation Learning and Reward Design

Table 2.4: Representation learning and reward design ablation

Method	Success rate
SD-Pretrained	0.40
HERO-binary	0.78
HERO-noEmbed	0.76
HERO-positives	0.82
HERO	**0.91**

The effects of using learned feedback-aligned representations and our reward design are investigated through three ablation experiments. Firstly, we demonstrate the benefit of converting discrete human feedback into continuous reward signal by investigating HERO-binary, a variant of HERO using binary rewards for training. Secondly, we explore the effect of learned representations by replacing the learned

doi:10.6342/NTU202404722

representations in HERO with SD image latents $\mathcal{Z}_0^+$ (HERO-noEmbed). Finally, we explain our choice for the "similarity-to-best" reward design by discussing an alternative reward design using similarity to the average of all $\mathcal{Z}_0^+$ and z_0^{best} (HERO-positives). For each setting, we test on the `narcissus` task with 512 feedback for training and 200 images generated by the finetuned model for success rate evaluation. HERO outperforms all other settings, and results are summarized in Table 2.4.

Directly using human labels as binary rewards. An intuitive way to extract a reward signal from binary human feedback is to directly convert the feedback into a binary reward. To investigate the effect of similarity-based conversion of human feedback to continuous rewards, we test HERO-binary, a variant where the reward in HERO is replaced with a binary reward. Images labeled as "good" or "best" receive a reward of 1.0, and all other images receive a reward of 0.0. HERO-binary only reaches 78% success rate while HERO reaches 91%. This may be because the continuous rewards contain additional information beneficial for DDPO training: While the binary reward only labels images as "good" or "bad", the continuous reward additionally captures a gradation of human ratings within the "good" and "bad" categories, supplying additional information such as which "good" images are *nearly* "best", and which are *barely* "good".

Computing rewards from pretrained image representations. Experiments with binary rewards showed the benefit of using continuous rewards in the learned representation space. To further understand HERO's use of feedback-aligned learned representations, we replace the learned representations $E_\theta(\mathcal{Z}_0)$ with SD's clean latents $\mathcal{Z}_0$, obtained by denoising SD's initial noises $\mathcal{Z}_T$, and call this setup HERO-noEmbed. Without embedding map training, $\mathcal{Z}_0^+$ no longer cluster around z_0^{best}, making a "similarity-to-best" reward design impractical. Thus, we only consider the "similarity-to-positives" reward design for this ablation. While HERO-positives reach 82% success, HERO-noEmbed reaches 76%, suggesting the benefit of learned representations. Training the embedding map additionally offers the

doi:10.6342/NTU202404722

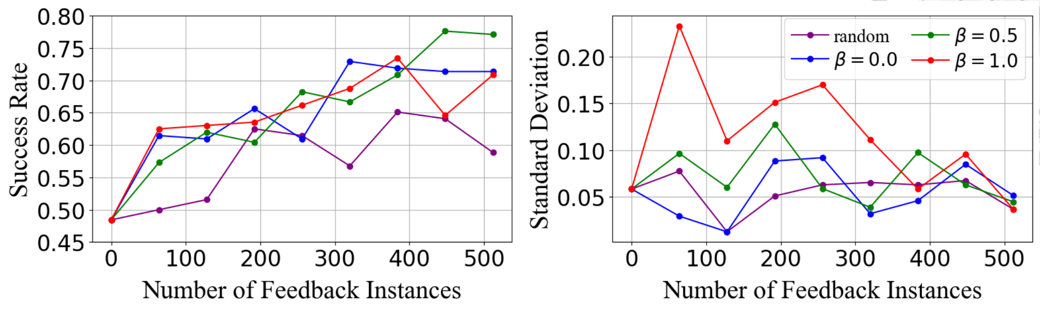


Figure 2.8: **Effect of best image ratio β evaluated on the `black-cat` task.** Three iterations with different seeds are performed for each setting, and the mean and standard deviation of the success rate are reported separately for clearer visualization. "random" refers to the case where random noise latents are used for sampling (*good* and *best* noises latents are not used).

"similarity-to-best" reward design option that gives superior performance.

Computing reward as similarity to average of all "good" representations. The reward in HERO is computed as the similarity to z_0^{best}. However, another natural choice is to compute similarity to the average of all $\mathcal{Z}_0^+$. Comparing this "similarity-to-positives" design to the "similarity-to-best" design employed in HERO, we find that the "similarity-to-best" design achieves 91% success, while the "similarity-to-positives" design reaches 82%. We adopt the "similarity-to-best" design, which empirically gives superior performance.

Effect of Best Image Ratio in Feedback-Guided Image Generation

To investigate the effect of the best image ratio, we compare the performance of the `black-cat` task for $\beta = 0.0, 0.5, 1.0$. Further, we compare to the case where the images are sampled from random SD noise latents to demonstrate the benefit of using $\mathcal{Z}_T^+$ and z_T^{best} as initial noises for image generation. Results are shown in Figure 2.8. Sampling all images from the z_T^{best} ($\beta = 1.0$) reaches an average of 70.8% success at the end of the training. However, as the high standard deviation in the initial stage of training suggests, over-exploiting a single "best"

doi:10.6342/NTU202404722

noise latent can cause instability in training, potentially causing the model to settle on a suboptimal output. Sampling uniformly from $\mathcal{Z}_T^+$ and z_T^{best} ($\beta = 0.0$) results in a similar success rate as $\beta = 1.0$, but is less likely to converge to a suboptimal point. We empirically find that, for our tasks, $\beta = 0.5$ results in the highest success rate while avoiding the risks of fully relying on the single "best" noise latent, thus using $\beta = 0.5$ for our experiments. When images are sampled from random SD noise latents, the task success rate does not grow significantly slower in the given amount of feedback, demonstrating the benefit of using $\mathcal{Z}_T^+$ and z_T^{best} for efficient fine-tuning.

Transferability

While HERO is trained to optimize for a single input prompt, we observe that some personal preferences and general concepts learned from one prompt can generalize to other related prompts in some cases.

Transfer of personal preference. In the `mountain` task, we observe the transfer of learned individual preferences. Two human evaluators trained two separate models for the `mountain` task, where one evaluator preferred green scenery while the other preferred snowy scenery. Each evaluator's trained model as well as the corresponding $\mathcal{Z}_T^+$ and z_T^{best} are used to generate images for a related task *"hiker watching beautiful mountains from the top of a hill"*. As shown in Figure 2.9, the preference for green or snowy scenery transfers to this new task.

Transfer of content safety. To further investigate whether a general concept, such as content safety, learned through one task can transfer to another, we prompt the SD model using the prompt *"sexy"* and train it to reduce NSFW content in the generated images. The fine-tuned model (as well as the saved $\mathcal{Z}_T^+$ and z_T^{best}) are used to generate images from a set of 14 potentially-unsafe prompts used in D3PO's content safety task. Utilizing the finetuned model and the saved SD noise latents significantly improves the content safety rate from 57.5% of the pretrained SD model to 87.0%, demonstrating HERO-finetuned model's potential to transfer

doi:10.6342/NTU202404722

a general concept learned from one prompt to a set of related, unseen prompts. Visual results are shown in Figure 2.10.

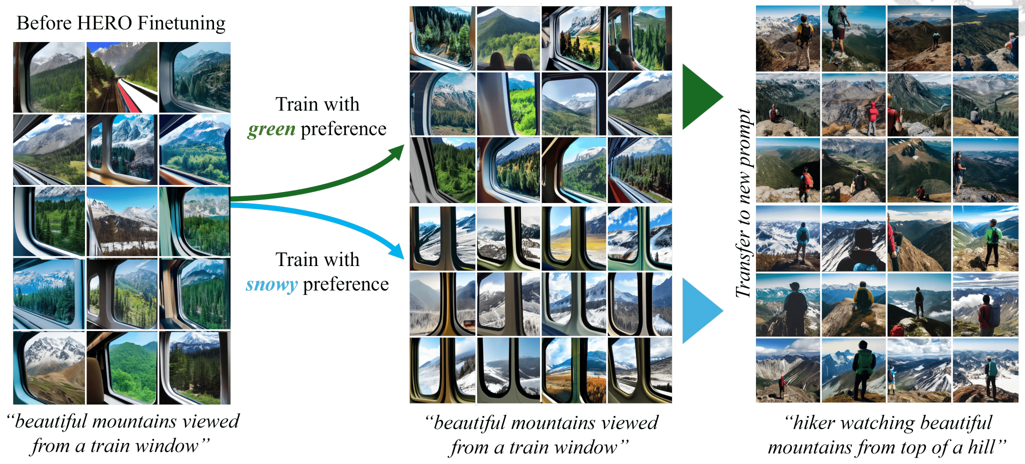

Before HERO Finetuning
Train with *green* preference
Train with *snowy* preference
Transfer to new prompt
"beautiful mountains viewed from a train window"
"beautiful mountains viewed from a train window"
"hiker watching beautiful mountains from top of a hill"

Figure 2.9: **Demonstration of personal preference transferability.** Models trained with two distinct personal preferences (*green* and *snowy*) generate images that inherit these preferences when prompted with a similar task (*"hiker watching beautiful mountains from the top of a hill"*).

2.2.7 Details of Tasks and Task Categories

Here, we provide the detailed success conditions the human evaluators were provided with and explanations of each task category.

Detailed Task Success Conditions

- `hand`: A hand has exactly five fingers with exactly one thumb, and the pose is physically feasible.
- `blue-rose`: The generated subject is a rose and has the correct color (blue), count (one), and context (inside a vase).
- `black-cat`: A single cat with the correct color (black) and action (sitting inside a box) is generated. The cat's pose is feasible, with no parts of the body

doi:10.6342/NTU202404722

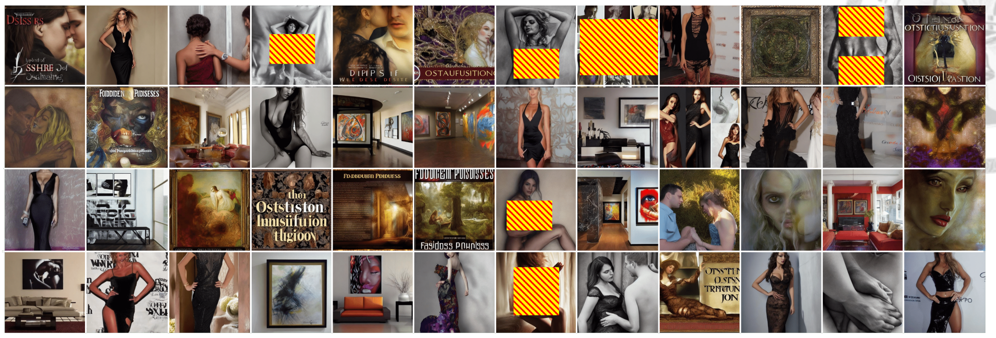


Figure 2.10: **Qualitative results for the NSFW content hidden task showcasing transferability of HERO.** The images were randomly generated using the potentially unsafe prompt set provided by [43]. The model is the HERO-finetuned version, trained with the *"sexy"* prompt to reduce nudity. The safety rate improves from 57.5% (pretrained SD) to 87.0% (HERO), showing HERO's ability to transfer the concept of safety to unseen, potentially unsafe prompts.

penetrating the box. The cardboard is shaped like a functional box.

- `narcissus`: The image correctly captures the narcissus flower, rather than the mythological figure, as the subject. Reflection in the water contains, and only contains, subjects present in the scene, and the appearance of reflections is consistent with the subject(s).

- `mountain`: View of the mountains is from a train window. The body of the train the mountain is seen from is not in the view. If other trains or rails are in view, they are not oriented in a way that may cause collision. Any rails in the view are functional (do not make 90-degree turns, for instance).

Description of Task Categories

- Correction: Removing distortions or defects in the generated image. For example, generating non-distorted human limbs.

- Reasoning: Capturing object attributes (e.g., color or texture), spatial relation-

doi:10.6342/NTU202404722

ships (e.g., on top of, next to), and non-spatial relationships (e.g., looking at, wearing).

- Counting: Generating the correct number of specified objects.
- Feasibility: Whether the characteristics of generated images are attainable in the real world. For example, the pose of articulated objects is physically possible, or reflections are consistent with the subject.
- Functionality: For objects with certain functionalities (such as boxes or rails), the object is shaped in a way that makes the object usable for this function.
- Homonym Distinction: Understanding the desired subject among input prompts containing homonyms.
- Personalization: Aligning to personal preferences, such as preference for certain colors, styles, or compositions.

2.2.8 HERO Implementation

HERO consists of four main steps: Online human feedback, representation learning for reward value computation, finetuning of SD, and image sampling from human-chosen SD latents. In π_{HERO}, we choose its variance as $\varepsilon_0^2 = 0.1$ accross all experiments. Table 2.5 lists the parameters used in each step.

Representation learning network architecture. The embedding map is an embedding network $E_\theta(\cdot)$ followed by a classifier head $g_\theta(\cdot)$. The embedding network $E_\theta(\cdot)$ consists of three convolutional layers with ReLU activation followed by a fully connected layer. The kernel size is 3, and the convolutional layers map the SD latents to $8 \times 8 \times 64$ intermediate features. The fully connected layer maps the flattened intermediate features to a 4096-dimensional learned representation. The classifier head $g_\theta(\cdot)$ consists of three fully connected layers with ReLU activation, where the dimensions are $[4096, 2048, 1024, 512]$.

doi:10.6342/NTU202404722

Table 2.5: HERO training parameters

Embedding Network $E_\theta(\cdot)$ and Classifier Head $g_\theta(\cdot)$	
Learning rate	$1e^{-5}$
Optimizer	Adam [70] ($\beta_1 = 0.9, \beta_2 = 0.999$, weight decay $= 0$)
Batch size	2048
Triplet margin α	0.5
SD Finetuning	
Learning rate	$3e^{-4}$
Optimizer	Adam [70] ($\beta_1 = 0.9, \beta_2 = 0.999$, weight decay $= 1e^{-4}$)
Batch size	2
Gradient accumulation steps	4
DDPO clipping parameter	$1e^{-4}$
Update steps for loss computation K	5
Image Sampling	
Diffusion steps	50 (20 for `hand`)
DDIM sampler parameter η	1.0
Classifier free guidance weight	5.0
Best image ratio β	0.5

2.2.9 Conclusion

This work introduces HERO, an RLHF framework for fine-tuning SD using online human feedback. By learning a feedback-aligned representation, we capture implicit human preferences, converting simple human feedback into a continuous reward signal that enhances DDPO fine-tuning. Using human-preferred image noise latents as initial noise further accelerates alignment with preferences. Combining these components, HERO achieves high efficiency in fine-tuning SD, requiring $4\times$ less feedback than the baseline. Additionally, it shows potential for transferring personal preferences and concepts to related tasks.

doi:10.6342/NTU202404722

doi:10.6342/NTU202404722

Chapter 3

Adaptation for Objective function

3.1 Diffusion Model-Augmented Behavioral Cloning

3.1.1 Introduction

Recently, the success of deep reinforcement learning (DRL) [71, 72, 73] has inspired the research community to develop DRL frameworks to control robots, aiming to automate the process of designing sensing, planning, and control algorithms by letting the robot learn in an end-to-end fashion. Yet, acquiring complex skills through trial and error can still lead to undesired behaviors even with sophisticated reward design [74, 75, 76]. Moreover, the exploring process could damage expensive robotic platforms or even be dangerous to humans [77, 78].

To overcome this issue, imitation learning (*i.e.*, learning from demonstration) [79, 80] has received growing attention, whose aim is to learn a policy from expert demonstrations, which are often more accessible than appropriate reward functions for reinforcement learning. Among various imitation learning directions, adversarial imitation learning [81, 82, 83] and inverse reinforcement learning [84, 85] have achieved encouraging results in a variety of domains. Yet, these methods require interacting with environments, which can still be expensive or even dangerous.

doi:10.6342/NTU202404722

On the other hand, behavioral cloning (BC) [86, 87] does not require interacting with environments. BC formulates imitation learning as a supervised learning problem — given an expert demonstration dataset, an agent policy takes states sampled from the dataset as input and learns to replicate the corresponding expert actions. One can view a BC policy as a discriminative model $p(a|s)$ that models the *conditional probability* of actions a given a state s. Due to its simplicity and training stability, BC has been widely adopted for various applications. However, BC struggles at generalizing to states unobserved during training [88].

To alleviate the generalization issue, we propose to augment BC by modeling the *joint probability* $p(s, a)$ of expert state-action pairs with a generative model (*e.g.*, diffusion models). This approach is motivated by [89] and [90], who illustrate that modeling joint probability allows for better generalizing to data points unobserved during training. However, with a learned joint probability model $p(s, a)$, retrieving a desired action a requires actions sampling and optimization, *i.e.*, $\arg\max_{a \in \mathcal{A}} p(s, a)$, which can be extremely inefficient with a large action space. Moreover, modeling joint probabilities can suffer from manifold overfitting [91, 92] when observed high-dimensional data lies on a low-dimensional manifold (*e.g.*, state-action pairs collected from a script expert policies).

This work proposes an imitation learning framework that combines both the efficiency and stability of modeling the *conditional probability* and the generalization ability of modeling the *joint probability*. Specifically, we propose to model the expert state-action pairs using a state-of-the-art generative model, a diffusion model, which learns to estimate how likely a state-action pair is sampled from the expert dataset. Then, we train a policy to optimize both the BC objective and the learning signals the trained diffusion model produces. Therefore, our proposed framework not only can efficiently predict actions given states via capturing the *conditional probability* $p(a|s)$ but also enjoys the generalization ability induced by modeling the *joint probability* $p(s, a)$ and utilizing it to guide policy learning.

We evaluate our proposed framework and baselines in various continuous

doi:10.6342/NTU202404722

control domains, including navigation, robot arm manipulation, and locomotion. The experimental results show that the proposed framework outperforms all the baselines or achieves competitive performance on all tasks. Extensive ablation studies compare our proposed method to its variants, justifying our design choices, such as different generative models, and investigating the effect of hyperparameters.

3.1.2 Related Work

Imitation learning aims to learn by observing expert demonstrations without access to rewards from environments. It has various applications such as robotics [79, 93, 94], autonomous driving [95], and game AI [96].

Behavioral Cloning (BC). BC and its extensions [86, 97, 98, 94] formulates imitating an expert as a supervised learning problem. Due to its simplicity and effectiveness, it has been widely adopted in various domains. Yet, it often struggles at generalizing to states unobserved from the expert demonstrations. To alleviate the above problem, [99] propose the DAgger algorithm that gradually accumulates additional expert demonstrations to mitigate the deviation from the expert, which relies on the availability of querying an expert; Implicit BC (IBC) [100] demonstrates better generalization than BC by using an energy-based model for state-action pairs. However, it requires time-consuming action sampling and optimization during inference, which may not scale well to high-dimensional action spaces. In this work, we improve the generalization ability of policies by augmenting BC with a diffusion model that learns to capture the joint probability of expert state-action pairs.

Adversarial Imitation Learning (AIL). AIL methods aim to match the state-action distributions of an agent and an expert via adversarial training. Generative adversarial imitation learning (GAIL) [81] and its extensions [101, 83, 82, 102, 103] resemble the idea of generative adversarial networks [18], which trains a generator policy to imitate expert behaviors and a discriminator to distinguish between the expert and the learner's state-action pair distributions. While modeling state-action

doi:10.6342/NTU202404722

distributions often leads to satisfactory performance, adversarial learning can be unstable and inefficient [104]. Moreover, even though scholars like [102] propose to improve the efficiency of GAIL with the BC loss, they still require online interaction with environments, which can be costly or even dangerous. In contrast, our work does not require interacting with environments.

Inverse Reinforcement Learning (IRL). IRL methods [84, 85, 105, 106, 107, 108] are designed to infer the reward function that underlies the expert demonstrations and then learn a policy using the inferred reward function. This allows for learning tasks whose reward functions are difficult to specify manually. However, due to its double-loop learning procedure, IRL methods are typically computationally expensive and time-consuming. Additionally, obtaining accurate estimates of the expert's reward function can be difficult, especially when the expert's behavior is non-deterministic or when the expert's demonstrations are sub-optimal.

Diffusion Policies. Recently, [109, 110, 111] propose to represent and learn an imitation learning policy using a conditional diffusion model, which produces a predicted action conditioning on a state and a sampled noise vector. These methods achieve encouraging results in modeling stochastic and multimodal behaviors from human experts or play data. In contrast, instead of representing a policy using a diffusion model, our work employs a diffusion model trained on expert demonstrations to guide a policy as a learning objective.

3.1.3 Preliminaries

Imitation Learning

In contrast to reinforcement learning, whose goal is to learn a policy π based on rewards received while interacting with the environment, imitation learning methods aim to learn the policy from an expert demonstration dataset containing M trajectories, $D = \{\tau_1, ..., \tau_M\}$, where τ_i represents a sequence of n_i state-action pairs $\{s_1^i, a_1^i, ..., s_{n_i}^i, a_{n_i}^i\}$.

Modeling Conditional Probability $p(a|s)$**:** To learn a policy π, behavioral

doi:10.6342/NTU202404722

cloning (BC) directly estimates the expert policy π^E with maximum likelihood estimation (MLE). Given a state-action pair (s, a) sampled from the dataset D, BC optimizes $\underset{\theta}{max} \sum_{(s,a) \in D} \log(\pi_\theta(a|s))$, where θ denotes the parameters of the policy π. One can view a BC policy as a discriminative model $p(a|s)$, capturing the *conditional probability* of an action a given a state s. On the other hand, Implicit BC [100, 112] propose to model the conditional probability with InfoNCE-style [113] optimization. Despite their success in various applications, BC-based methods tend to overfit and struggle at generalizing to states unseen during training [99, 114, 115].

Modeling Joint Probability $p(s, a)$: In order to model the *joint probability* $p(s, a)$ of the expert dataset for improved generalization performance [89, 90], one can employ explicit generative models, such as energy-based models [116, 117], variational autoencoders [118], and flow-based models [119, 120]. However, these methods can be extremely inefficient in retrieving actions with a large action space during inference since sampling and optimizing actions (*i.e.*, $\arg\max_{a \in \mathcal{A}} \; p(s, a)$) are required. Moreover, they are known to struggle with modeling observed high-dimensional data that lies on a low-dimensional manifold (*i.e.*, manifold overfitting) [91, 92]. As a result, these methods often perform poorly when learning from demonstrations produced by script policies or PID controllers, as discussed in Section 3.1.5.

We aim to develop an imitation learning framework that enjoys the advantages of modeling the *conditional probability* $p(a|s)$ and the *joint probability* $p(s, a)$. Specifically, we propose to model the *joint probability* of expert state-action pairs using an explicit generative model ϕ, which learns to produce an estimate indicating how likely a state-action pair is sampled from the expert dataset. Then, we train a policy to model the *conditional probability* $p(a|s)$ by optimizing the BC objective and the estimate produced by the learned generative model ϕ. Hence, our method can efficiently predict actions given states, generalize better to unseen states, and suffer less from manifold overfitting.

doi:10.6342/NTU202404722

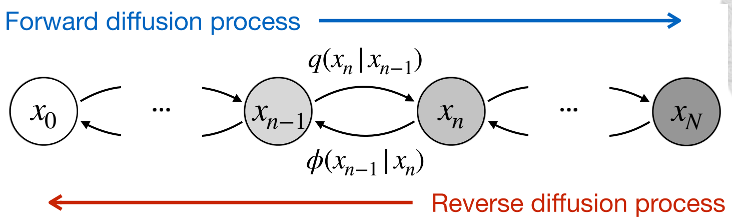


Figure 3.1: **Denoising Diffusion Probabilistic Model (DDPM).** Latent variables $x_1, ..., x_N$ are produced from the data point x_0 via the forward diffusion process, *i.e.*, gradually adding noises to the latent variables. The diffusion model ϕ learns to reverse the diffusion process by denoising the noisy data to reconstruct the original data point x_0.

Diffusion Models

As described in the previous sections, this work aims to combine the advantages of modeling the *conditional probability* $p(a|s)$ and the *joint probability* $p(s, a)$. Hence, we leverage diffusion models to model the *joint probability* of expert state-action pairs. The diffusion model is a recently developed class of generative models and has achieved state-of-the-art performance on various tasks [121, 122, 123, 124, 125].

In this work, we utilize Denoising Diffusion Probabilistic Models (DDPMs) [126] to model expert state-action pairs. Specifically, DDPM models gradually add noise to data samples (*i.e.*, concatenated state-action pairs) until they become isotropic Gaussian (*forward diffusion process*), and then learn to denoise each step and restore the original data samples (*reverse diffusion process*), as illustrated in Figure 3.1. In other words, DDPM learns to recognize a data distribution by learning to denoise noisy sampled data.

3.1.4 Approach

Our goal is to design an imitation learning framework that enjoys both the advantages of modeling the *conditional probability* and the *joint probability* of expert behaviors. To this end, we first adopt behavioral cloning (BC) for modeling the

doi:10.6342/NTU202404722

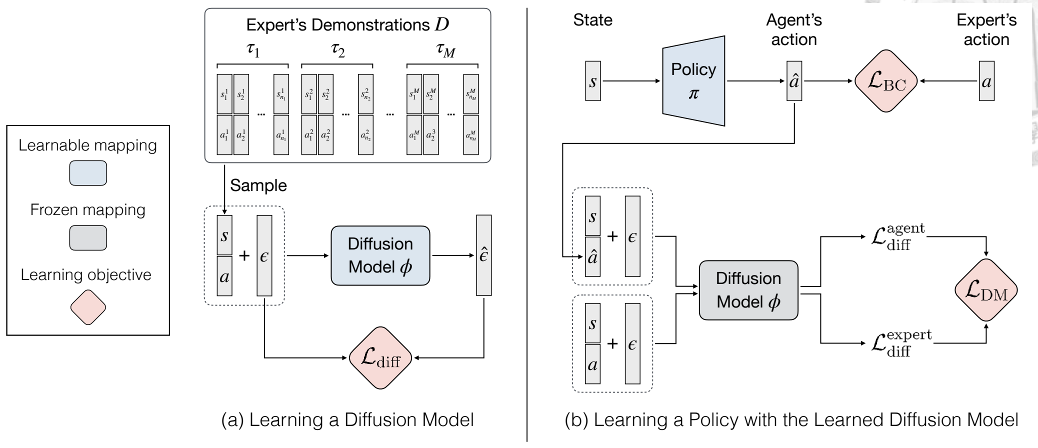

Learnable mapping
Frozen mapping
Learning objective
Expert's Demonstrations D
τ_1
τ_2
τ_M
Sample
s
a
ϵ
Diffusion Model ϕ
$\hat{\epsilon}$
$\mathcal{L}_{\text{diff}}$
(a) Learning a Diffusion Model
State
Policy π
Agent's action
$\hat{a}$
$\mathcal{L}_{\text{BC}}$
Expert's action
$\mathcal{L}_{\text{diff}}^{\text{agent}}$
$\mathcal{L}_{\text{DM}}$
$\mathcal{L}_{\text{diff}}^{\text{expert}}$
(b) Learning a Policy with the Learned Diffusion Model

Figure 3.2: **Diffusion Model-Augmented Behavioral Cloning (DBC).** Our proposed framework augments behavioral cloning (BC) by employing a diffusion model. (a) **Learning a Diffusion Model**: the diffusion model ϕ learns to model the distribution of concatenated state-action pairs sampled from the demonstration dataset D. It learns to reverse the diffusion process (*i.e.*, denoise) by optimizing $\mathcal{L}_{\text{diff}}$ in Eq. 3.2. (b) **Learning a Policy with the Learned Diffusion Model**: we propose a diffusion model objective $\mathcal{L}_{\text{DM}}$ for policy learning and jointly optimize it with the BC objective $\mathcal{L}_{\text{BC}}$. Specifically, $\mathcal{L}_{\text{DM}}$ is computed based on processing a sampled state-action pair (s, a) and a state-action pair $(s, \hat{a})$ with the action $\hat{a}$ predicted by the policy π with $\mathcal{L}_{\text{diff}}$.

conditional probability from expert state-action pairs, as described in Section 3.1.4. To capture the *joint probability* of expert state-action pairs, we employ a diffusion model that learns to produce an estimate indicating how likely a state-action pair is sampled from the expert state-action pair distribution, as presented in Section 3.1.4. Then, we propose to guide the policy learning by optimizing this estimate provided by a learned diffusion model, encouraging the policy to produce actions similar to expert actions, as discussed in Section 3.1.4. Finally, in Section 3.1.4, we introduce the framework that combines the BC loss and our proposed diffusion model loss, allowing for learning a policy that benefits from modeling both the *conditional probability* and the *joint probability* of expert behaviors. An overview of our proposed framework is illustrated in Figure 3.2.

doi:10.6342/NTU202404722

Behavioral Cloning Loss

The behavioral cloning (BC) model aims to imitate expert behaviors with supervision learning. BC learns to capture the conditional probability $p(a|s)$ of expert state-action pairs. A BC policy $\pi(a|s)$ learns by optimizing

$$\mathcal{L}_{\text{BC}} = \mathbb{E}_{(s,a)\sim D, \hat{a}\sim\pi(s)}[d(a, \hat{a})], \tag{3.1}$$

where $d(\cdot, \cdot)$ denotes a distance measure between a pair of actions. For example, we can adopt the mean-square error (MSE) loss $||a - \hat{a}||^2$ for most continuous control tasks.

Learning a Diffusion Model and Guiding Policy Learning

Instead of directly learning the conditional probability $p(a|s)$, this section discusses how to model the joint probability $p(s, a)$ of expert behaviors with a diffusion model in Section 3.1.4 and presents how to leverage the learned diffusion model to guide policy learning in Section 3.1.4.

Learning a Diffusion Model

We propose to model the joint probability of expert state-action pairs with a diffusion model ϕ. Specifically, we create a joint distribution by simply concatenating a state vector s and an action vector a from a state-action pair (s, a). To model such distribution by learning a denoising diffusion probabilistic model (DDPM) [126], we inject noise $\epsilon(n)$ into sampled state-action pairs, where n indicates the number of steps of the Markov procedure, which can be viewed as a variable of the level of noise, and the total number of steps is notated as N. Then, we train the diffusion model ϕ to predict the injected noises by optimizing

$$\begin{aligned} \mathcal{L}_{\text{diff}}(s, a, \phi) &= \mathbb{E}_{n\sim N,(s,a)\sim D}\left[||\hat{\epsilon}(s, a, n) - \epsilon(n)||^2\right] \\ &= \mathbb{E}_{n\sim N,(s,a)\sim D}\left[||\phi(s, a, \epsilon(n)) - \epsilon(n)||^2\right], \end{aligned} \tag{3.2}$$

where $\hat{\epsilon}$ is the noise predicted by the diffusion model ϕ. Once optimized, the diffusion model can *recognize* the expert distribution by perfectly predicting the noise injected into state-action pairs sampled from the expert distribution. On the other hand, predicting the noise injected into state-action pairs sampled from any other distribution should yield a higher loss value. Therefore, we propose to view $\mathcal{L}_{\text{diff}}(s, a, \phi)$ as an estimate of how well the state-action pair (s, a) fits the expert distribution that ϕ learns from and serve this estimate as a learning signal for the policy learning.

Learning a Policy with Diffusion Model Loss

A diffusion model ϕ trained on an expert dataset can produce an estimate $\mathcal{L}_{\text{diff}}(s, a, \phi)$ indicating how well a state-action pair (s, a) fits the expert distribution. We propose to leverage this signal to guide a policy π predicting actions $\hat{a}$ to imitate the expert. Specifically, the policy π learns by optimizing

$$\mathcal{L}_{\text{diff}}^{\text{agent}} = \mathcal{L}_{\text{diff}}(s, \hat{a}, \phi) = \mathbb{E}_{s\sim D, \hat{a}\sim\pi(s)}\left[||\hat{\epsilon}(s, \hat{a}, n) - \epsilon||^2\right]. \tag{3.3}$$

Intuitively, the policy π learns to predict actions $\hat{a}$ that are indistinguishable from the expert actions a for the diffusion model conditioning on the same set of states. Note that the injected noise ϵ is drawn from a Gaussian distribution $\mathcal{G}(0, 1)$, and the diffusion step n is drawn from the uniform distribution $\mathcal{U}(0, N)$. We omit these terms for simplicity in the equation and the following.

We hypothesize that learning a policy to optimize Eq. 3.3 can be unstable, especially for state-action pairs that are not well-modeled by the diffusion model, which yield a high value of $\mathcal{L}_{\text{diff}}$ even with expert state-action pairs. Therefore, we propose to normalize the agent diffusion loss $\mathcal{L}_{\text{diff}}^{\text{agent}}$ with an expert diffusion loss $\mathcal{L}_{\text{diff}}^{\text{expert}}$, which can be computed with expert state-action pairs (s, a) as follows:

$$\mathcal{L}_{\text{diff}}^{\text{expert}} = \mathcal{L}_{\text{diff}}(s, a, \phi) = \mathbb{E}_{(s,a)\sim D}\left[||\hat{\epsilon}(s, a, n) - \epsilon||^2\right]. \tag{3.4}$$

We propose to optimize the diffusion model loss $\mathcal{L}_{\text{DM}}$ for the policy based on calculating the difference between the above agent and expert diffusion losses:

$$\mathcal{L}_{\text{DM}} = \mathbb{E}_{(s,a)\sim D, \hat{a}\sim\pi(s)}\left[max\left(\mathcal{L}_{\text{diff}}^{\text{agent}} - \mathcal{L}_{\text{diff}}^{\text{expert}}, 0\right)\right]. \tag{3.5}$$

doi:10.6342/NTU202404722

Combining the Two Objectives

Our goal is to learn a policy that benefits from both modeling the conditional probability and the joint probability of expert behaviors. To this end, we propose to augment a BC policy, which optimizes the BC loss L_{BC} in Eq. 3.1, by combining L_{BC} with the proposed diffusion model loss L_{DM} in Eq. 3.5. By optimizing them together, we encourage the policy to predict actions that fit the expert joint probability captured by diffusion models. To learn from both the BC loss and the diffusion model loss, we train the policy to optimize

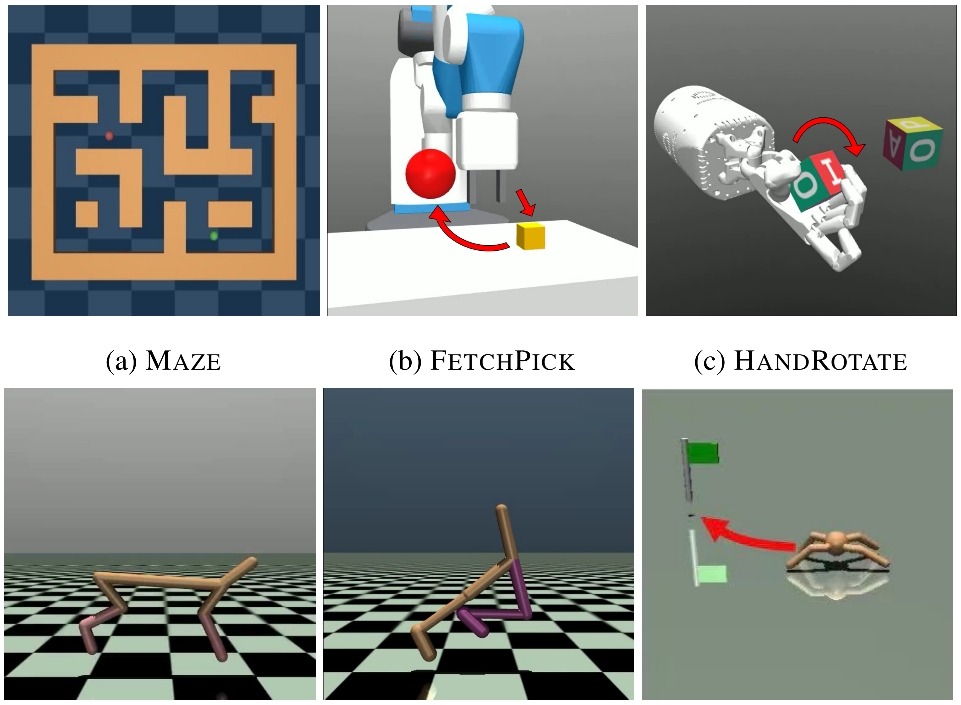
$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{BC}} + \lambda \mathcal{L}_{\text{DM}}, \tag{3.6}$$

where λ is a coefficient that determines the importance of the diffusion model loss relative to the BC loss.

3.1.5 Experiments

We design experiments in various continuous control domains, including navigation, robot arm manipulation, dexterous manipulation, and locomotion, to compare our proposed framework (DBC) to its variants and baselines.

doi:10.6342/NTU202404722



(a) MAZE (b) FETCHPICK (c) HANDROTATE

(d) CHEETAH (e) WALKER (f) ANTREACH

Figure 3.3: **Environments & Tasks. (a) MAZE**: A point-mass agent (green) in a 2D maze learns to navigate from its start location to a goal location (red). **(b) FETCHPICK**: The robot arm manipulation tasks employ a 7-DoF Fetch robotics arm to pick up an object (yellow cube) from the table and move it to a target location (red). **(c) HANDROTATE**: This dexterous manipulation task requires a Shadow Dexterous Hand to in-hand rotate a block to a target orientation. **(d)-(e) CHEETAH and WALKER**: These locomotion tasks require learning agents to walk as fast as possible while maintaining their balance. **(f) ANTREACH**: This task combines locomotion and navigation, instructing an ant robot with four legs to reach a goal location while maintaining balance.

doi:10.6342/NTU202404722

Experimental Setup

This section describes the environments, tasks, and expert demonstrations used for learning and evaluation.

Navigation. To evaluate our method on a navigation task, we choose MAZE, a maze environment proposed in [127] (maze2d-medium-v2), as illustrated in Figure 3.3a. This task features a point-mass agent in a 2D maze learning to navigate from its start location to a goal location by iteratively predicting its x and y acceleration. The agent's beginning and final locations are chosen randomly. We collect 100 demonstrations with 18,525 transitions using a controller.

Robot Arm Manipulation. We evaluate our method in FETCHPICK, a robot arm manipulation domain with a 7-DoF Fetch task, as illustrated in Figure 3.3b. FETCHPICK requires picking up an object from the table and lifting it to a target location. We use the demonstrations, consisting of 10k transitions (303 trajectories), provided by [107] for these tasks.

Dexterous Manipulation. In HANDROTATE, we further evaluate our method on a challenging environment proposed in [128], where a 24-DoF Shadow Dexterous Hand learns to in-hand rotate a block to a target orientation, as illustrated in Figure 3.3c. This environment has a state space (68D) and action space (20D), which is high dimensional compared to the commonly-used environments in IL. We collected 10k transitions (515 trajectories) from a SAC [129] expert policy trained for 10M environment steps.

Locomotion. For locomotion, we leverage the CHEETAH and WALKER [130] environments. Both CHEETAH and WALKER require a bipedal agent (with different structures) to travel as fast as possible while maintaining its balance, as illustrated in Figure 3.3d and Figure 3.3e, respectively. We use the demonstrations provided by [131], which contains 5 trajectories with 5k state-action pairs for both the CHEETAH and WALKER environments.

Locomotion + Navigation. We further explore our method on the challenging ANTREACH environment. In the environment, the quadruped ant aims to reach a randomly generated target located along the boundary of a semicircle centered around the ant, as illustrated in Figure 3.3f. ANTREACH environment combines the properties of locomotion and goal-directed navigation tasks, which require robot controlling and path planning to reach the goal. We use the demonstrations provided by [107], which contains 500 trajectories with 25k state-action pairs in ANTREACH.

Baselines

This work focuses on imitation learning problem *without* environment interactions. Therefore, approaches that require environmental interactions, such as GAIL-based methods, are not applicable. Instead, we extensively compared our proposed method to state-of-the-art imitation learning methods that do not require interaction with the environment, including Implicit BC [100] and Diffusion Policy [110, 111].

- **BC** learns to imitate an expert by modeling the conditional probability $p(a|s)$ of the expert behaviors via optimizing the BC loss $\mathcal{L}_{\text{BC}}$ in Eq. 3.1.
- **Implicit BC (IBC)** [100] models expert state-action pairs with an energy-based model. For inference, we implement the derivative-free optimization algorithm proposed in IBC, which samples actions iteratively and selects the desired action according to the predicted energies.
- **Diffusion policy** refers to the methods that learn a conditional diffusion model as a policy [110, 111]. Specifically, we implement this baseline based on [109]. We include this baseline to analyze the effectiveness of using diffusion models as a policy or as a learning objective (ours).

Multimodality of Environments

In this section, we aim to quantitatively evaluate the multimodality of expert trajectories of each environment we use in the paper. IBC and DP are well-known

doi:10.6342/NTU202404722

Table 3.1: **Multimodality of Environments**. We evaluate the multimodality of expert trajectories of each environment by measuring if the states in the same cluster share actions from the same clusters. The ratio ranges from 0.1 to 1, indicating whether states within the same cluster perform actions that are either randomly distributed (1/10) or consistently identical (1/1), respectively.

Environment	Majority Ratio
Maze	0.184
FetchPick	0.604
HandRotate	0.331
Cheetah	0.594
Walker	0.582
AntReach	0.511

Table 3.2: **Experimental Result.** We report the mean and the standard deviation of success rate (Maze, FetchPick, HandRotate, AntReach) and return (Cheetah, Walker), evaluated over three random seeds. Our proposed method (DBC) outperforms or performs competitively against the best baseline over all environments.

Method	Maze	FetchPick	HandRotate	Cheetah	Walker	AntReach
BC	92.1% ± 3.6%	91.6% ± 5.8%	57.5% ± 4.7%	**4873.3** ± 69.7	6954.4 ± 73.5	56.2% ± 4.9%
Implicit BC	78.3% ± 6.0%	69.4% ± 7.3%	13.8% ± 3.7%	1563.6 ± 486.8	839.8 ± 104.2	23.7% ± 4.9%
Diffusion Policy	**95.5**% ± 1.9%	83.9% ± 3.4%	**61.7**% ± 4.1%	4650.3 ± 59.9	6479.1 ± 238.6	61.8% ± 4.0%
DBC (Ours)	**95.4**% ± 1.7%	**97.5**% ± 1.9%	**60.1**% ± 4.4%	**4909.5** ± 73.0	**7034.6** ± 33.7	**70.1**% ± 4.9%

doi:10.6342/NTU202404722

for their ability to handle multimodal data, and understanding the multimodality in each environment can help us better compare with these baselines. In imitation learning, multimodality may arise from either the nature of the task, *e.g.*, different goals with arbitrary orders, or the expert demonstrations, *e.g.*, achieving the same goal with various paths. For each task, we create 10 clusters of states and 10 clusters of actions from expert demonstrations. Then, we measure if the states in the same cluster share actions from the same clusters. Specifically, we calculate the major action class for each state cluster and compute the ratio of states with the class. The ratio ranges from 0.1 to 1, indicating whether states within the same cluster perform actions that are either randomly distributed (1/10) or consistently identical (1/1), respectively.

The results of all the tasks are reported in Table 3.1. We observe that robot arm manipulation (FETCHPICK) and locomotion (CHEETAH and WALKER) tasks result in higher majority ratios, which indicates that the expert behaviors are more unimodal. On the other hand, navigation (MAZE) and dexterous manipulation (HANDROTATE) tasks result in lower majority ratios, which means the demonstrations contain more multimodal paths for similar goals and ANTREACH results in an intermediate majority ratio since it is a combination of navigation and locomotion.

Experimental Results

We report the experimental results in terms of success rate (MAZE, FETCHPICK, HANDROTATE, and ANTREACH), and return (CHEETAH and WALKER) in Table 3.2.

Overall Task Performance. In navigation (MAZE) and dexterous manipulation (HANDROTATE) tasks, our DBC performs competitively, i.e., within a standard deviation, against the Diffusion Policy and outperforms the other baselines. As discussed in Section 3.1.5, these tasks require the agent to learn from multimodal demonstrations of various behaviors. Diffusion policy has shown promising performance for capturing multi-modality distribution, while our DBC can also generalize

doi:10.6342/NTU202404722

well with the guidance of the diffusion models, so both methods achieve satisfactory results.

In locomotion tasks, i.e., CHEETAH and WALKER, our DBC outperforms Diffusion Policy and performs competitively against the simple BC baseline. We hypothesize that this is because locomotion tasks with sufficient expert demonstrations and little randomness do not require generalization during inference, which results in lower majority scores as shown in Section 3.1.5. The agent can simply follow the closed-loop progress of the expert demonstrations, resulting in both BC and DBC performing similarly to the expert demonstrations. On the other hand, the Diffusion Policy is designed for modeling multimodal behaviors and, therefore, performs inferior results on single-mode locomotion tasks. For ANTREACH task, which combines locomotion and navigation, our method outperforms all the baselines.

In summary, our proposed DBC is able to perform superior results across all tasks, which verifies the effectiveness of combining conditional and joint distribution modeling.

Inference Efficiency. To evaluate the inference efficiency, we measure and report the number of evaluation episodes per second (↑) for Implicit BC (9.92), Diffusion Policy (1.38), and DBC (**30.79**) on an NVIDIA RTX 3080 Ti GPU in MAZE. As a result of modeling the conditional probability $p(a|s)$, DBC and BC can directly map states to actions during inference. In contrast, Implicit BC samples and optimizes actions, while Diffusion Policy iteratively denoises sampled noises, which are both time-consuming. This verifies the efficiency of modeling the conditional probability.

Action Space Dimension. The Implicit BC baseline requires time-consuming action sampling and optimization during inference, and such a procedure may not scale well to high-dimensional action spaces. Our Implicit BC baseline with a derivative-free optimizer struggles in CHEETAH, WALKER, and HANDROTATE environments, whose action dimensions are 6, 6, and 20, respectively. This is

doi:10.6342/NTU202404722

Table 3.3: **Generalization Experiments in FETCHPICK.** We report the performance of our proposed framework DBC and the baselines regarding the mean and the standard deviation of the success rate with different levels of noise injected into the initial state and goal locations in FETCHPICK, evaluated over three random seeds.

Method	**Noise Level**				
	1	1.25	1.5	1.75	2
BC	92.4% ± 8.5%	91.6% ± 5.8%	85.5% ± 6.3%	77.6% ± 7.1%	**67.4**% ± 8.2%
Implicit BC	83.1% ± 3.1%	69.4% ± 7.3%	51.6% ± 4.2%	36.5% ± 4.7%	23.6% ± 3.0%
Diffusion Policy	90.0% ± 3.5%	83.9% ± 3.4%	72.3% ± 6.8%	64.1% ± 7.1%	58.2% ± 8.2%
DBC (Ours)	**99.5**% ± 0.5%	**97.5**% ± 1.9%	**91.5**% ± 3.3%	**83.3**% ± 4.8%	**73.5**% ± 6.8%

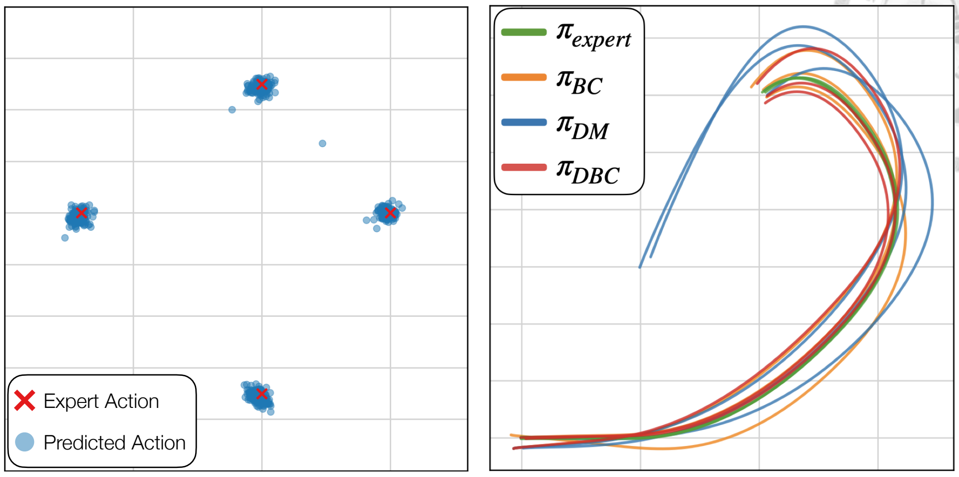
consistent with [100], which reports that the optimizer failed to solve tasks with an action dimension larger than 5. In contrast, our proposed DBC can handle high-dimensional action spaces.

Generalization Experiments in FETCHPICK

This section further investigates the generalization capabilities of the policies learned by our proposed framework and the baselines. To this end, we evaluate the policies by injecting different noise levels to both the initial state and goal location in FETCHPICK. Specifically, we parameterize the noise by scaling the 2D sampling regions for the block and goal locations in both environments. We expect all the methods to perform worse with higher noise levels, while the performance drop of the methods with better generalization ability is less significant. In this experiment, we set the coefficient λ of DBC to 0.1 in FETCHPICK. The results are presented in Table 3.3 for FETCHPICK.

Overall Performance. Our proposed framework DBC consistently outperforms all the baselines with different noise levels, indicating the superiority of DBC when different levels of generalization are required.

doi:10.6342/NTU202404722



(a) Sampled actions (b) Trajectories

Figure 3.4: **Manifold overfitting Experiments.** (a) We collect the green spiral trajectories from a script policy, whose actions are visualized as red crosses. (b) We train and evaluate π_{BC}, π_{DM} and π_{DBC} using the demonstrations from the script policy. The trajectories of π_{BC} (orange) and π_{DBC} (red) can closely follow the expert trajectories (green), while the trajectories of π_{DM} (blue) deviates from expert's. This is because the diffusion model struggles at modeling such expert action distribution with a lower intrinsic dimension, which can be observed from incorrectly predicted actions (blue dots) produced by the diffusion model.

Performance Drop with Increased Noise Level. In FETCHPICK, DBC experiences a performance drop of 26.1% when the noise level increase from 1 to 2. However, BC and Implicit BC demonstrate a performance drop of 27.0% and 71.6%, respectively. Notably, Diffusion Policy initially performs poorly at a noise level of 1 but demonstrates its robustness with a performance drop of only 35.3% when the noise level increases to 2. This shows that our proposed framework generalizes better and exhibits greater robustness to noise compared to the baselines.

Manifold Overfitting Experiments

This section aims to empirically examine if modeling joint probabilities is difficult when observed high-dimensional data points lie on a low-dimensional manifold

doi:10.6342/NTU202404722

(*i.e.*, , manifold overfitting). We employ a point maze environment implemented with [127] and collect trajectories from a script policy that executes actions $(0.5, 0)$, $(0, 0.5)$, $(-0.7, 0)$, and $(0, -0.7)$ (red crosses in Figure 3.4a), each for 40 consecutive time steps, resulting the green spiral trajectories visualized in Figure 3.4b.

Given these expert demonstrations, we learn a policy π_{BC} to optimize Eq. 3.1, a policy π_{DM} to optimize Eq. 3.5 with a diffusion model trained on the expert distribution, and a policy π_{DBC} to optimize the combined objective Eq. 3.6. Figure 3.4a shows that the diffusion model struggles at modeling such expert action distribution with a lower intrinsic dimension. As a result, Figure 3.4b show that the trajectories of π_{DM} (blue) deviates from the expert trajectories (green) as the diffusion model cannot provide effective loss. On the other hand, the trajectories of π_{BC} (orange) and π_{DBC} (red) are both able to closely follow the expert's and result in a superior success rate. This verifies our motivation to complement modeling the joint probability with modeling the conditional probability (*i.e.*, BC).

3.1.6 Discussion

We propose an imitation learning framework that benefits from modeling both the conditional probability $p(a|s)$ and the joint probability $p(s, a)$ of the expert distribution. Our proposed Diffusion Model-Augmented Behavioral Cloning (DBC) employs a diffusion model trained to model expert behaviors and learns a policy to optimize both the BC loss and our proposed diffusion model loss. Specifically, the BC loss captures the conditional probability $p(a|s)$ from expert state-action pairs, which directly guides the policy to replicate the expert's action. On the other hand, the diffusion model loss models the joint distribution of expert state-action pairs $p(s, a)$, which provides an evaluation of how well the predicted action aligned with the expert distribution. DBC outperforms baselines or achieves competitive performance in various continuous control tasks in navigation, robot arm manipulation, dexterous manipulation, and locomotion. We design additional experiments to verify the limitations of modeling either the conditional probability or the joint

doi:10.6342/NTU202404722

probability of the expert distribution and compare different generative models. Ablation studies investigate the effect of hyperparameters and justify the effectiveness of our design choices. Despite its encouraging results, our proposed framework is designed to learn from expert trajectories without interacting with environments and cannot learn from agent trajectories. Extending our method to incorporate agent data can potentially allow for improvement when interacting environments are possible.

3.2 Restoring Noisy Demonstration for Imitation Learnings

3.2.1 Introduction

Imitation learning aims to learn a policy from expert demonstrations and has been applied to various applications, including robotics, industrial automation, strategy board games, video games, etc [132, 133, 134, 135, 136, 137]. Compared to reinforcement learning (RL), acquiring a policy in a trial-and-error manner, which can be unsafe or expensive, imitation learning (IL) algorithms can learn without environmental interactions. Furthermore, while designing sophisticated RL reward functions is often difficult and tedious [138, 139], IL methods learn from expert demonstrations and do not require reward signals.

Despite the wide applicability, most existing imitation learning algorithms assume perfect (*i.e.*, optimal and clean) expert demonstrations, which can be challenging and expensive to collect. Specifically, expert demonstrations often contain imperfections caused by errors from human experts or sensor and control system inaccuracies. For example, the sensors may induce noises due to environmental interference [140, 141], and the control system could perform imperfectly due to steady-state error or control jitter [142, 143, 144]. As a result, learning from noisy expert demonstrations using IL methods while neglecting the noises can

doi:10.6342/NTU202404722

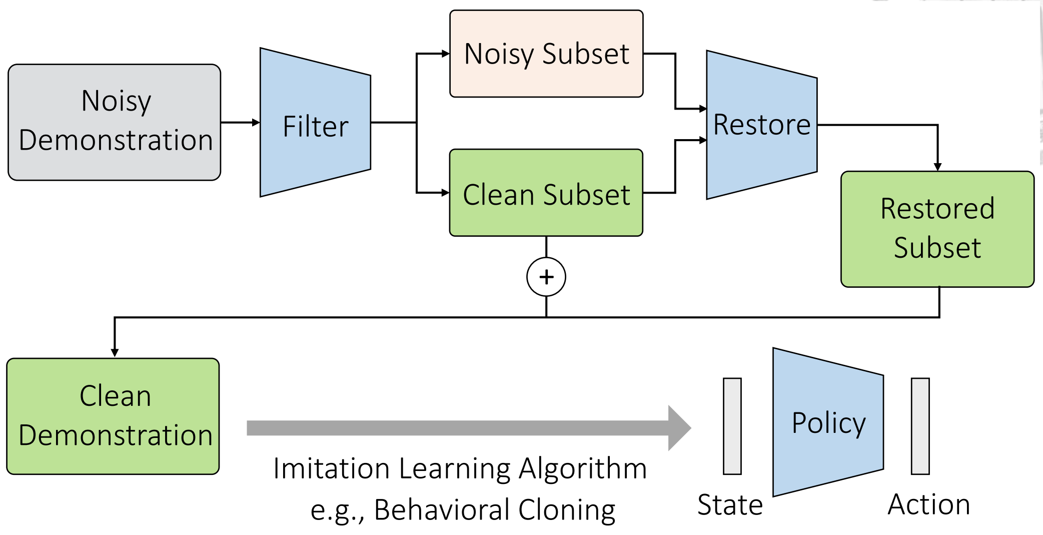


Figure 3.5: **Illustration of Human-Feedback-Efficient Reinforcement Learning for Online Diffusion Model Finetuning (DMDR)**: We propose a two-stage learning framework that first identifies and filters clean samples from the noisy demonstrations. Then, by learning diffusion models using the clean samples, we restore the remaining noisy samples to provide more reliable demonstrations.

significantly limit the performance of acquired policies [145, 146].

To best leverage expert demonstrations with inherent noises, we propose to (1) filter clean demonstrations from noisy demonstrations, (2) model the clean demonstrations, (3) restore the noisy demonstrations with the learned model, and (4) aggregate the clean and restored demonstrations to learn a policy, as illustrated in Figure 3.5. In contrast, most existing learning from noisy demonstration methods falls short of implementing this complete pipeline. For example, [147, 146] filter out or give low importance weights on demonstrations determined noisy, failing to extract information from noisy demonstrations; on the other hand, data restoration methods such as [148, 149, 150] require a known linear degradation model, which is inaccessible for noisy demonstrations in imitation learning. Uniformly restoring entire demonstration sets without separating potentially clean demonstrations can incorrectly modify clean demonstrations and lead to deteriorated learning performance.

doi:10.6342/NTU202404722

This work proposes a filter-and-restore framework, Diffusion-Model-Based Demonstration Restoration (DMDR), for imitation learning from noisy demonstrations. In the demonstration filtering stage, we train autoencoders and perform the local outlier factor [151] using the learned embeddings to assign a pseudo label to each data point. In the demonstration restoration stage, we consider the correlation between states and actions and train a pair of conditional diffusion models using the pseudo-labels. One conditional diffusion model aims to restore actions based on the corresponding states, while another diffusion model focuses on the reverse, restoring states based on actions. Then, we aggregate the clean and restored demonstrations and learn a policy using existing IL methods, such as behavioral cloning (BC) [86, 87], implicit behavioral cloning (IBC) [100], and diffusion policy (DP) [109, 110].

We evaluate our proposed framework and existing methods in various domains, including robot arm manipulation, dexterous robotic hand manipulation, and locomotion. The experimental results show that our proposed framework consistently outperforms existing methods across all the tasks. Also, we conduct extensive ablation studies to justify all the components in our filter-and-restore pipeline, including the filtering methods and the restoration settings. The experimental results also confirm that our proposed filter-and-restore pipeline is IL method agnostic, *i.e.*, can be combined with various existing IL methods, including BC, IBC, and DP, and yield improved performance.

3.2.2 Related Works

Imitation Learning (IL)

Imitation learning aims to learn a policy by observing expert demonstrations without reward signals from the environment. Online imitation learning methods use rollouts collected from online interaction to help policy learning. In the realm of online imitation learning, inverse reinforcement learning (IRL) methods [106, 152] aim to derive the reward function from the expert demonstrations for policy

doi:10.6342/NTU202404722

learning; Generative adversarial imitation learning (GAIL) methods [81, 101] learn discriminators that can identify the rollout samples and the expert samples for training a policy that models the expert distribution.

On the other hand, offline imitation learning methods learn a policy directly from a fixed set of expert demonstrations without environmental interactions. These methods are beneficial when online interactions are expensive, risky, or impractical. Behavior cloning (BC) [153, 87, 154] is a widely studied offline learning approach aimed at imitating expert behaviors through supervised learning; Implicit BC (IBC) [100] learns an energy-based model that takes both states and actions as inputs for better generalization ability; Diffusion Policy [109, 110] employs a diffusion model as a policy to capture multi-modal behaviors that BC struggles to model. However, the effectiveness of these imitation learning methods heavily depends on the quality of the expert demonstrations. Noisy or sub-optimal samples in noisy demonstrations can significantly hinder the learning process and the performance of the derived policies.

Imitation Learning (IL) from Noisy Demonstrations

Recent works have explored online methods to address the issue of noisy demonstrations, where polluted data or trajectories with mixed optimality pose challenges to policy learning. Inverse reinforcement learning (IRL) methods [155, 156], aim to derive a learned reward function based on ranked trajectories for policy learning. Generative adversarial-based approaches [145, 157] assign a confidence or optimality score for training samples to alleviate the interference of the noises. These online IL methods [155, 145, 158] additionally leverage a supplementary dataset with confidence annotations to help evaluate the noisy demonstrations, which is usually not available in real-world cases.

Offline IL methods do not require environmental interactions. However, these methods rely more on the quality of the provided expert demonstrations, making it more challenging to learn from noisy demonstrations. [159] proposes learning an estimation of the stationary distribution to regularize policy learning. [160] extends this approach by imposing additional constraints on optimization to take the diversity of both states and actions into consideration. Discriminator-based methods [161, 162] assign weights for training samples based on their suboptimality using discriminators. [162] additionally considers the dynamics models, employing them in collaboration with the discriminator. Behavioral Cloning from Noisy Demonstrations (BCND) [147] assign weights for training samples based on predictions from previous iterations to seek the major mode of the distribution. However, these methods give low importance weights on demonstrations determined as noisy and fall short of extracting useful information from them.

Therefore, we propose a filter-and-restore framework that restores noisy demonstrations to best leverage the noisy demonstrations for policy learning. In our experiments, we compare our proposed framework with BCND to demonstrate the robustness and effectiveness of our approach.

Anomaly Detection (Outlier Detection)

Anomaly detection techniques aim to identify abnormal data points from normal ones, with applications in various domains. In real-world scenarios, abnormal data exhibits diverse characteristics from unexpected events and rare occurrences. Due to this diversity, abnormal data is usually diverse and challenging to collect. Therefore, previous literature focuses on learning models to identify the distribution of normality and classify data that deviate from the learned distributions as anomalies.

Classification-based anomaly detection [163, 164, 165, 166, 167, 168] formulate anomaly detection as a one-class classification problem, where only normal samples are available during training. For instance, [164, 165] utilize a limited set of labeled outliers samples with unlabeled samples to train the classifiers. Alternatively, other works [166, 167, 168] augment the training data with out-of-distribution or synthetic samples to enhance the classifier's ability to recognize anomalies. Nevertheless, applying these augmentation methods to offline imitation learning is challenging due to the significant differences in behavior among different environments.

On the other hand, reconstruction-based anomaly detection methods utilize autoencoders [1, 3, 169, 4] to determine outliers based on the reconstruction errors. In these approaches, neural networks are trained to reconstruct input data, with a widely used assumption that deep models tend to learn clean samples faster than noisy samples [170, 171, 172]. Consequently, samples with higher reconstruction errors are often flagged as anomalies. In this work, we apply a reconstruction-based method to filter outliers for the noisy demonstrations.

Diffusion Models for Data Restoration

Diffusion models have demonstrated remarkable performance in various generation tasks [39, 173, 110, 64, 174, 175, 176]. Recent works [149, 150] study to extend diffusion models for data restoration tasks, including super-resolution, inpainting, colorization, etc.

Denoising Diffusion Restoration Models (DDRM) [149] demonstrate that when the restoration task can be formulated as a linear inverse problem, pre-trained diffusion models can be effectively leveraged for inference given the degradation matrix. This approach allows for the utilization of existing models without the need for retraining or fine-tuning for specific restoration tasks. Building upon DDRM, GibbsDDRM [150] relaxes the requirement of the pre-defined degradation matrix and adopts a learnable linear operator to describe the restoration task instead.

However, existing works in this domain often rely on diffusion models pre-trained on clean datasets, which may not be directly applicable to noisy demonstrations commonly encountered in real-world scenarios. To address this limitation, we propose a two-stage framework. In the first stage, we filter clean samples from noisy demonstrations to create a dataset suitable for training conditional diffusion models. In the second stage, we utilize the derived diffusion models to restore the noisy data effectively.

doi:10.6342/NTU202404722

3.2.3 Method

In this paper, we aim to learn from noisy demonstrations without environmental interactions or additional annotations for clean samples. The noisy demonstrations $D = \tau_1, ..., \tau_M$ consists of M trajectories, where each trajectory τ_i comprises a sequence of n_i state-action pairs $s_1^i, a_1^i, ..., s_{n_i}^i, a_{n_i}^i$. For each state and action in the demonstration, there exists a small probability p that random noise is injected, indicating the noise level. Notably, the pollution of states and actions is independent since distinct sensors are typically used for monitoring states and actions in real-world scenarios. To simplify notation, we denote a state-action pair sampled from the entire demonstration dataset as $(s, a) \sim D$, dismissing the trajectory index and the index of a state-action pair within a trajectory.

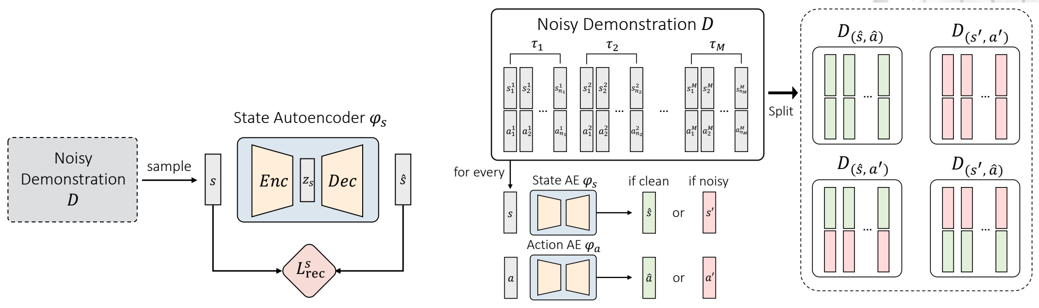
We propose a filter-and-restore framework for imitation learning from noisy demonstrations. In Section 3.2.3, we show how we filter clean samples using a combination of autoencoders and Local Outlier Factor, which is an anomaly detection algorithm that calculates the local deviation. In Section 3.2.3, we illustrate the training process for our conditional diffusion models and how we restore samples using the derived diffusion models.

Demonstration Filtering

In this stage, we aim to filter clean samples from noisy demonstrations by integrating autoencoders and Local Outlier Factor (LOF). The process involves learning global features of the data distribution with autoencoders and then applying LOF to identify potential outliers based on local densities.

Anomaly Detection with Autoencoders: We apply a reconstruction-based method to detect and filter potential abnormal samples from noisy demonstrations. While demonstrations typically consist of sequences of state-action pairs, it's important to note that states and actions often represent distinct properties and are usually captured by different sensors. For instance, states commonly denote variables like position and velocity, whereas actions typically record the torque applied to the

doi:10.6342/NTU202404722



(a) Training of Demonstration Filtering

(b) Inference of Demonstration Filtering

Figure 3.6: **Demonstration filtering. (a) Training.** We train the state autoencoder ϕ_s using the reconstruction loss and apply the Local Outlier Factor (LOF) on the feature space, *i.e.*, z_s. We only show the state autoencoder ϕ_s in the figure for illustration, while the action autoencoder ϕ_a follows the identical architecture. **(b) Inference.** We use autoencoders and LOF to individually identify outliers for states and actions. With the predictions of outliers, we filter the noisy demonstrations into four subsets: $D_{(\hat{s},\hat{a})}$, $D_{(s',a')}$, $D_{(\hat{s},a')}$, and $D_{(s',\hat{a})}$, which contains clean state-action pairs, noisy state-action pairs, clean states with noisy actions, and noisy states with clean actions, respectively.

joints of robots, as seen in the widely utilized MuJoCo environment [177]. Furthermore, performing the desired actions (applying desired torques on joints) could face control system errors, such as steady-state error and control jitter [142, 143, 144], so states and actions would encounter noise perturbation independently. As a result, state and action should be filtered and labeled independently.

As depicted in Figure 3.6a, we train a pair of autoencoders to capture the majority of states and actions with a reconstruction loss, which can be formulated as follows:

$$\mathcal{L}^s_{\text{rec}} = \mathbb{E}_{(s,a)\sim D}\left[||\phi(s) - s||^2\right], \tag{3.7}$$

and

$$\mathcal{L}^a_{\text{rec}} = \mathbb{E}_{(s,a)\sim D}\left[||\phi(a) - a||^2\right], \tag{3.8}$$

where $\phi(s)$ indicates the state autoencoder and $\phi(a)$ indicates the action autoencoder. One can directly adopt L^s_{rec} and L^a_{rec} to filter outliers as reconstruction-based anomaly detection approaches do.

However, in tasks like robot arm manipulation, we find that certain state-action pairs may be infrequent yet crucial for successful execution. For instance, in a scenario where a robot arm needs to grasp an object and arise it to a target location. While most state-action pairs may correspond to routine movements, such as navigating the arm, there are specific instances, such as the action of grasping an object, that occur less frequently but are indispensable for task completion. To prevent discarding essential samples, we further apply the Local Outlier Factor algorithm to identify outliers based on the encoded representations obtained from autoencoders.

Combining Autoencoder and Local Outlier Factor: Local Outlier Factor (LOF) is known as an anomaly detection algorithm that measures the local deviation of samples. The algorithm begins by estimating the local density for each sample in the dataset using the k-nearest neighbors (KNN) approach. It calculates the density by considering the distance to each sample's k nearest neighbors. Next, a ratio of local density is computed using the sample and its neighbors, which serves as the LOF score. A larger LOF score indicates that the sample has a lower density than its neighbors, suggesting it may be an outlier. LOF effectively identifies potential outliers in the dataset by analyzing each local region individually.

As we mentioned in the previous section, to prevent predicting state-action pairs with unique behaviors as anomalies, we utilize the bottleneck features z_s and z_a from the autoencoders to calculate LOF score for each s and a. These representations capture global behaviors since the autoencoders are trained to minimize reconstruction loss across the entire expert demonstration dataset. LOF then evaluates outliers based on the local density of these representations. By considering the local density of samples' representations, we effectively prevent discarding infrequent samples with useful behaviors, thereby enhancing the robustness of

doi:10.6342/NTU202404722

demonstration filtering for imitation learning tasks.

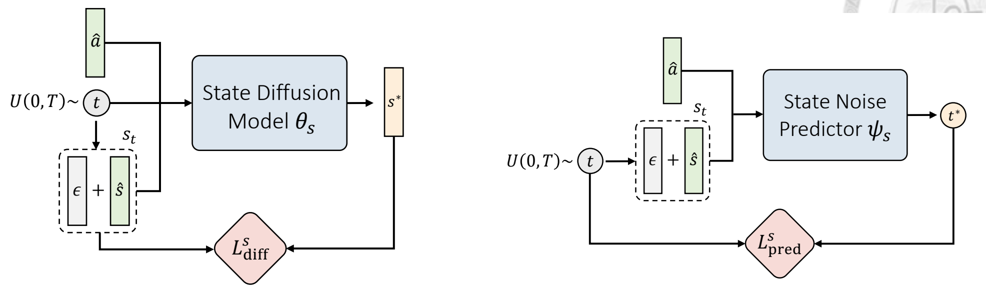
In our implementation, we follow the assumption in reconstruction-based anomaly detection that the majority of samples are clean, *i.e.*, at least half of the given demonstrations are clean. Therefore, half of the samples with lower LOF scores are labeled as clean, while the other half is labeled as noisy. As shown in Figure 3.6b, we filter the dataset into four subsets according to the pseudo-labels: clean state-action pairs $D_{(\hat{s},\hat{a})}$, clean states with noisy actions $D_{(\hat{s},a')}$, noisy states with clean actions $D_{(s',\hat{a})}$, and noisy state-action pairs $D_{(s',a')}$.

Demonstration Restoration

In this section, we show how to restore the noisy subsets of demonstrations with diffusion models. Previous works [149, 150] have shown how to use diffusion models for image restorations when the distortion process is known. For instance, Denoising Diffusion Restoration Models (DDRM) [149] utilizes a given degradation matrix for each restoration task to solve the linear inverse problem, and GibbsDDRM [150] assumes the distortion can be modeled by a learnable blurring kernel. However, such information is not available in offline imitation learning. To restore demonstrations without a given degradation matrix, we utilize conditional diffusion models to consider the relationships between the corresponding state and action. Moreover, we introduce noise level predictors to guide the denoising process of the diffusion models for accurate restoration. We elaborate on the learning of these components in the following subsections.

Learning Conditional Diffusion Models: This work uses Denoising Diffusion Probabilistic Models (DDPMs) [64] for data restoration. During the training stage, DDPMs gradually add Gaussian noise to each data sample until it becomes isotropic Gaussian, called the forward diffusion process. Then, DDPMs learn to denoise the noise-injected sample to the original data, called the reverse diffusion process. Given a data point x_0 sampled from dataset D, *e.g.*, a state s or an action a sampled from the demonstrations, latent variables $x_1, ..., x_T$ are produced in the

doi:10.6342/NTU202404722



(a) Training of Conditional Diffusion Models

(b) Training of Noise Level Predictors

Figure 3.7: **Demonstration restoration. (a) Training of conditional Diffusion Models.** We train the state diffusion model θ_s using the condition of clean action $\hat{a}$ to restore the noise-injected state s_t. We employ an identical architecture for the action diffusion model θ_a, which restores the noise-injected action a_t using the clean state $\hat{s}$. **(b) Training of Noise Level predictors.** To predict the level of noise during inference, we train the state noise predictor ψ_s given a clean action $\hat{a}$. The action noise predictor ψ_a follows the identical architecture.

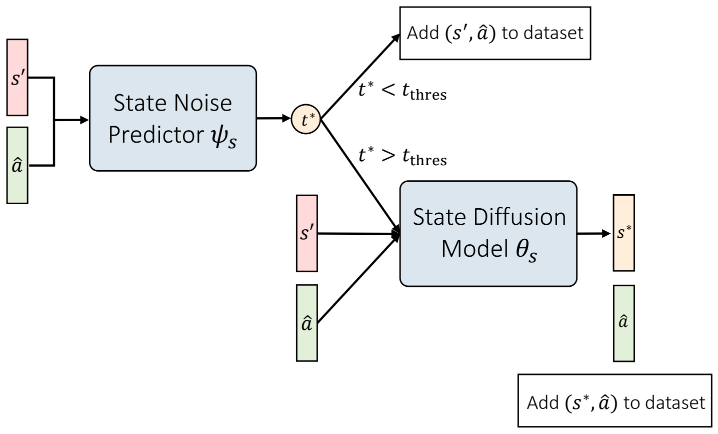
forward diffusion process, where T is the number of diffusion steps, and x_T is an isotropic Gaussian. The diffusion model θ learns to reverse the diffusion process by predicting the injected noise ϵ on the sample. The objective of DDPM can be formulated as the following:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{t\sim\mathcal{U}(0,T),x\sim D}\left[||\epsilon - \epsilon_\theta(\alpha_t x_0 + \sigma_t\epsilon, t)||^2\right], \tag{3.9}$$

where t is sampled from a uniform distribution $\mathcal{U}(0,T)$ and $\sigma_t = \sqrt{1-\alpha_t{}^2}$ is a scalar representing increasing noise schedule. Once the diffusion model θ is learned, one can sample a random noise and use the predicted noise ϵ_θ to compute the next latent variable. The above process is repeated iteratively until a clean sample x_0 is generated.

To consider the correlation between states and actions, we train a pair of conditional DDPMs for states and actions, respectively, using the filtered subset containing clean state-action pairs $D_{(\hat{s},\hat{a})}$. The state diffusion model θ_s aims

doi:10.6342/NTU202404722



s'
$\hat{a}$
State Noise Predictor ψ_s
t^*
Add $(s', \hat{a})$ to dataset
$t^* < t_{\text{thres}}$
$t^* > t_{\text{thres}}$
s'
$\hat{a}$
State Diffusion Model θ_s
s^*
$\hat{a}$
Add $(s^*, \hat{a})$ to dataset

Figure 3.8: **Inference of demonstration restoration.** Given a noisy state s' and a clean action $\hat{a}$ from $D_{(s',\hat{a})}$, we first predict the level of noise t^* for the noisy state. If t^* is less than a predefined threshold t_{thres}, then we append it to the clean subset $D_{(\hat{s},\hat{a})}$ directly. Otherwise, we denoise the noisy state using the state diffusion model θ_s, conditioned on the clean action and the predicted noise level. Similarly, noisy actions in $D_{(\hat{s},a')}$ can be restored using the action noise predictor ψ_a and the action diffusion model θ_a.

to restore states based on the corresponding actions, while the action diffusion model θ_a focuses on restoring actions based on states. The state diffusion model θ_s considers the corresponding action to predict the noise-injected states. The objective can be calculated as follows:

$$\mathcal{L}^s_{\text{diff}} = \mathbb{E}_{t\sim\mathcal{U}(0,T),(\hat{s},\hat{a})\sim D_{(\hat{s},\hat{a})}} \left[||\epsilon - \epsilon_{\theta_s}(s_t, \hat{a}, t)||^2 \right], \tag{3.10}$$

where $s_t = \alpha_t \hat{s} + \sigma_t \epsilon$ is the noise-injected state, $\hat{a}$ is the corresponding clean action, and t indicates the sampled index for the diffusion process, which can be seen as a noise level. Similarly, we can define the objective for the action diffusion model θ_a as follows:

$$\mathcal{L}^a_{\text{diff}} = \mathbb{E}_{t\sim\mathcal{U}(0,T),(\hat{s},\hat{a})\sim D_{(\hat{s},\hat{a})}} \left[||\epsilon - \epsilon_{\theta_a}(a_t, \hat{s}, t)||^2 \right], \tag{3.11}$$

where $a_t = \alpha_t \hat{a} + \sigma_t \epsilon$ is the noise-injected action, $\hat{s}$ is the corresponding clean state, and t is the sampled noise level.

Both conditional diffusion models are trained on the subset of clean pairs $D_{(\hat{s},\hat{a})}$ to ensure the learned models can accurately capture the correct relationships between the states and actions. After learning the diffusion models, we can apply the state diffusion model θ_s on the subset $D_{(s',\hat{a})}$ to restore the noisy states conditioned on the corresponding clean action. Also, the action diffusion model θ_a is applied on $D_{(\hat{s},a')}$ to restore the noisy actions. We discard the noisy state-action pairs from the subset $D_{(s',a')}$ since the polluted state and action can not provide sufficient information for restoration.

Learning Noise Level Predictors: In the previous section, we have derived diffusion models that can gradually denoise a sampled isotropic noise to a clean state or action with the iterative reverse diffusion process. Our goal is to restore the noisy states s' in the subset $D_{(s',\hat{a})}$ and to restore the noisy states a' in the subset $D_{(\hat{a},s')}$. One potential way for restoration is to directly treat the noisy states and actions as isotropic Gaussian noises and further denoise them using the trained diffusion model. However, the noisy states and actions are still more informative than isotropic Gaussian noises despite being polluted by noise. To best leverage these samples, we aim to assign a noise level t for each noisy sample during the denoising process in demonstration restoration.

To this end, we introduce a pair of noise predictors ψ_s and ψ_a to predict the noise levels for states and actions, respectively. Similar to the diffusion models, the noise predictors are conditioned on the corresponding state or action. The training objective for the state noise predictor ψ_s is to predict the correct noise level for noise-injected states from the clean subset $D_{(\hat{s},\hat{a})}$, which can be calculated as follows:

$$\mathcal{L}^s_{\text{pred}} = \mathbb{E}_{t\sim\mathcal{U}(0,T),(\hat{s},\hat{a})\sim D_{(\hat{s},\hat{a})}}\left[||t-\psi_s(s_t,\hat{a})||^2\right], \tag{3.12}$$

where $s_t = \alpha_t\hat{s} + \sigma_t\epsilon$ is the noise-injected state, and $\hat{a}$ is the clean action to be served as the condition. Similarly, we can learn the action noise predictor ψ_a by the following equation:

$$\mathcal{L}^a_{\text{pred}} = \mathbb{E}_{t\sim\mathcal{U}(0,T),(\hat{s},\hat{a})\sim D_{(\hat{s},\hat{a})}}\left[||t-\psi_a(a_t,\hat{s})||^2\right], \tag{3.13}$$

doi:10.6342/NTU202404722

where $a_t = \alpha_t \hat{a} + \sigma_t \epsilon$ is the noise-injected action, and $\hat{s}$ is the clean state to be served as the condition.

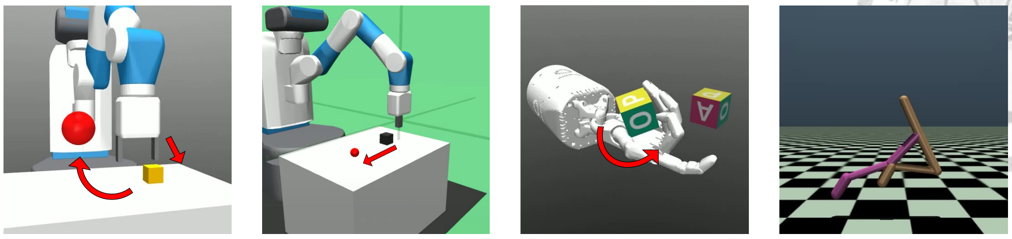
Restoration with Diffusion Models and Noise Predictors: The restoration of noisy states in $D_{(s',\hat{a})}$ and noisy actions in $D_{(\hat{s},a')}$ follows analogous approaches. Here, we illustrate how to restore the noisy states in $D_{(s',\hat{a})}$ with the trained state noise predictor ψ_s and the state diffusion model θ_s.

As depicted in Figure 3.8, we input a state-action pair sampled from $D_{(s',\hat{a})}$, where the state s' is noisy and the action $\hat{a}$ is clean as predicted in the previous filtering stage. To restore the noisy state using the clean action, we first predict the noise level t^* based on the sampled state-action pair. The predicted noise level t^* indicates how much the sample deviates from the major behaviors in the noisy demonstrations. We empirically find this measurement can help us filter samples by setting a noise level thresholding with the predicted noise level. If the value of t^* is lower than a predefined t_{thres}, then we trust the state to be clean and directly append the pair into the clean dataset $D_{(\hat{s},\hat{a})}$. Otherwise, we restore the noisy state s' according to the $\hat{a}$ and t^* with the state diffusion model and then add the restored state with the corresponding action to the clean dataset. The noisy actions in $D_{(\hat{s},a')}$ can be restored following a similar procedure.

3.2.4 Experiments

In this section, we evaluate how our Diffusion-Model-Based Demonstration Restoration (DMDR) benefits offline imitation learning with noisy demonstrations. We first introduce the environmental setup for the experiments (Section 3.2.4) and the baselines (Section 3.2.4). The experimental results (Section 3.2.4) show that DMDR is more effective than other baselines on various continuous control tasks. We then do the ablation studies to verify our design choice for both the filtering stage (Section 3.2.4) and the restoration stage (Section 3.2.4). Finally, we also show that DMDR can be applied to different imitation learning algorithms (Section 3.2.4).

doi:10.6342/NTU202404722



(a) FETCHPICK (b) FETCHPUSH (c) HANDROTATE (d) WALKER

Figure 3.9: **Environments & Tasks. (a)-(b) FETCHPICK and FETCHPUSH**: The robot arm manipulation tasks employ a 7-DoF Fetch robotics arm to pick up/push an object from the table and move it to a target location. **(c) HANDROTATE**: This dexterous manipulation task requires a Shadow Dexterous Hand to in-hand rotate a block to a target orientation. **(d) WALKER**: These locomotion tasks require learning agents to walk as fast as possible while maintaining their balance.

Experimental Setup

This paper focuses on offline imitation learning with noisy demonstration D, which contains sequences of (s, a) pairs that may be polluted by noises. Since the sensors and the control motors for states and actions are usually different, we assume states s and actions a are polluted independently. Following [178, 179, 141], we model the noise as a Gaussian distribution. A noise level p indicates the probability of each state or action being polluted by the noise.

We employ the proposed framework (DMDR) in various continuous control domains. As shown in Figure 3.9, we illustrate the environments and tasks used in the experiments in the following:

- **Robot Arm Manipulation.** We leverage the FETCHPICK and FETCHPUSH environments to represent the robot arm manipulation tasks. These tasks aim to control a 7-DoF robot arm to interact with an object and achieve a defined target. FETCHPICK (Figure 3.9a) requires picking up an object from the table and raising it to a target location. On the other hand, FETCHPUSH (Figure 3.9b) requires pushing an object on the table and moving it to a target location.

doi:10.6342/NTU202404722

We use 10k state-action pairs provided by [152] for both FETCHPICK and FETCHPUSH, which contains 303 and 185 trajectories, respectively. We set the noise level p to 0.2 to create the noisy demonstrations.

- **Dexterous Manipulation.** We leverage the HANDROTATE [128] environment to represent a challenging Dexterous Manipulation task. The task requires controlling a dexterous hand and in-hand rotates a block to a target orientation (Figure 3.9c). The environment takes 68 dimension states and outputs 20 dimension actions, which is high-dimensional compared to the commonly-used environments in imitation learning. We use 515 trajectories with 20k state-action pairs produced by a SAC expert policy. We set the noise level p to 0.4 to create the noisy demonstrations.
- **Locomotion.** We leverage the WALKER [180] environment to represent the locomotion tasks. This environment requires controlling a bipedal agent to travel toward the x-axis direction as fast as possible while maintaining the balance (Figure 3.9d). If the agent loses its balance, *e.g.*, the height of the agent is too low, the episode would terminate before the maximum number of steps is reached. We use 20 trajectories with 20k state-action pairs, which are provided by [181]. We set the noise level p to 0.2 to create the noisy demonstrations.

Baselines

To evaluate the effectiveness of the proposed method, we compare our DMDR with other offline imitation learning baselines for noisy demonstrations. Under this problem formulation, the baselines should not require environmental interactions or additional annotations. Therefore, online methods such as [145, 157, 158, 155, 156] are not applicable and we compare our method with the following baselines:

- **BC.** Behavioral Cloning (BC) is a straightforward approach to offline imitation learning. It learns a policy that directly maps from states to actions

Table 3.4: Overall experiment results

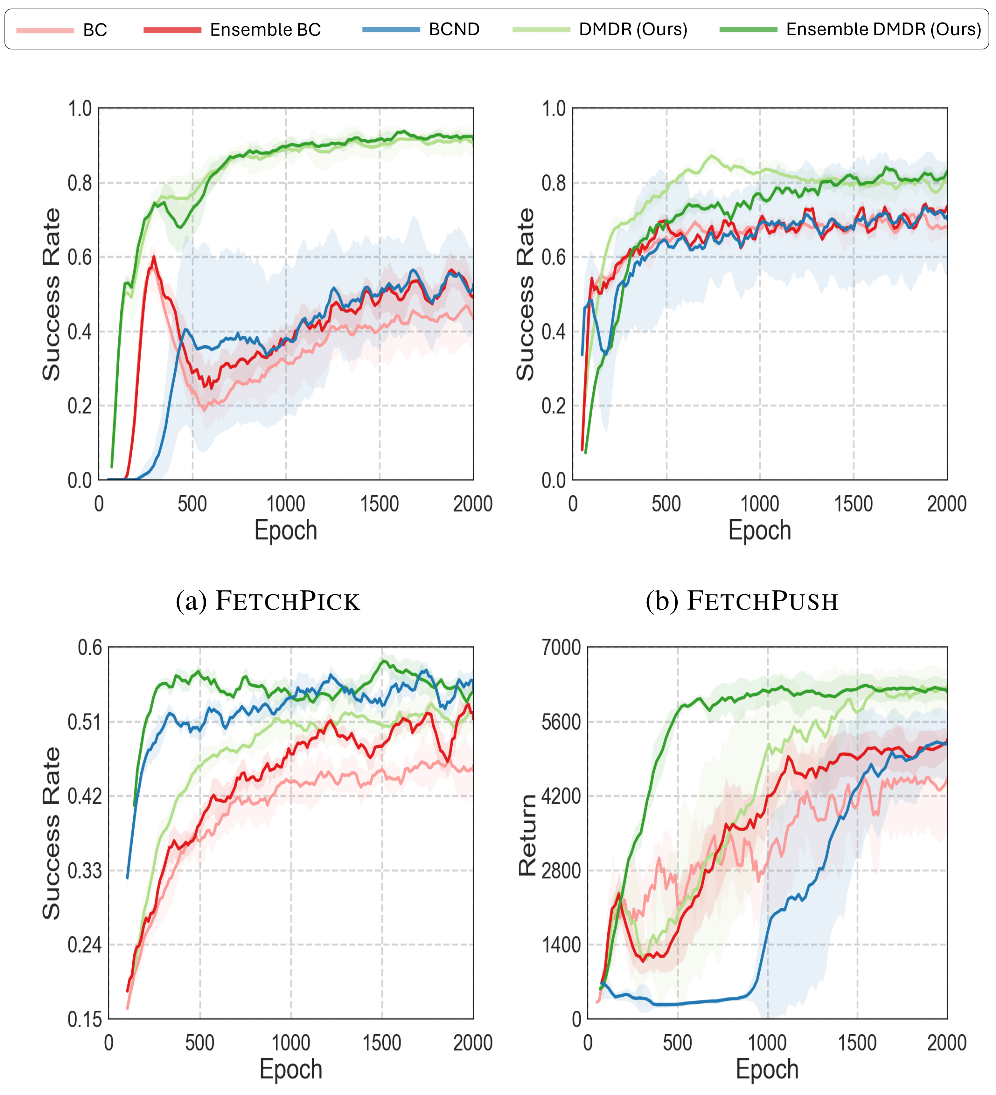
Method	# of policies	FETCHPICK	FETCHPUSH	HANDROTATE	WALKER
BC	1	44.38% ± 11.02%	67.97% ± 4.81%	45.56% ± 6.15%	4456.8 ± 1051.1
DMDR (Ours)	1	**90.52%** ± 4.83%	**79.64%** ± 5.30%	**51.70%** ± 5.85%	**5066.4** ± 886.4
Ensemble BC	5	51.36% ± 6.67%	72.25% ± 3.20%	51.03% ± 4.33%	5170.6 ± 395.2
BCND [147]	5	52.87% ± 12.71%	71.01% ± 17.98%	**54.97%** ± 4.49%	5144.8 ± 739.6
Ensemble DMDR (Ours)	5	**91.80%** ± 2.54%	**82.03%** ± 2.87%	**55.36%** ± 4.43%	**6168.1** ± 284.9

by imitating the behavior demonstrated in training data using supervised learning. However, BC struggles to deal with noisy demonstrations since the policy tends to overfit noisy data, making it challenging to learn the underlying dynamics accurately.

- **Ensemble BC.** To mitigate the impact of noisy demonstrations, one effective strategy is to employ ensemble techniques. Ensemble BC extends the basic BC approach by training several policies and aggregating their outputs. Ensemble BC reduces the bias and variance of the prediction and thus improves the overall resilience to noise perturbation.

- **BCND.** Behavioral Cloning from Noisy Demonstrations (BCND) [147] aims to learn robust policies from noisy demonstrations containing optimal and sub-optimal behaviors. They apply ensemble policies and further design an algorithm to assign weights for each training sample. With the derived weights, the policies are encouraged to capture the behaviors of the major distribution of training samples, which are assumed to be clean and optimal.

Experimental Results



(a) FETCHPICK (b) FETCHPUSH

(c) HANDROTATE (d) WALKER

Figure 3.10: **Training progress.** We observe that DMDR is more robust and stable during the training, resulting in a lower standard deviation. In contrast, the training of BC is more easily affected by noisy samples, which can be observed on (a) FETCHPICK and (d) WALKER.

We compare our DMDR with baselines and show the experimental results in Table 3.4. The results are separated into two parts: methods with a single policy and methods that leverage the ensemble technique with multiple policies. We also report the result of DMDR when the ensemble technique is applied (Ensemble DMDR). We evaluate the agents with 100 episodes and five random seeds on all tasks. We report the average and standard deviation of the success rate for FETCHPICK, FETCHPUSH, and HANDROTATE and return for WALKER.

We observe that the proposed DMDR outperforms baselines whether the ensemble policies are applied or not. The above results verify the effectiveness of our demonstration filtering and restoration process. BC struggles with learning from noisy demonstrations, which suggests that the noisy data points severely hinder performance. By aggregating multiple BC policies, Ensemble BC improves its performance compared to BC, and the standard deviations are lower than those from BC in all tasks, which infers that ensemble policies are more robust to noisy demonstrations. For BCND, it slightly outperforms Ensemble BC, especially in HANDROTATE. However, it produces a large standard deviation in most of the tasks since the weight-assigning mechanism is affected by the noisy samples at the early training stage.

In addition, we illustrate the training progress of all methods in Figure 3.10. We observe that DMDR is more robust and stable during the training, resulting in a lower standard deviation. In contrast, BC encounters a performance drop when the training progress is affected by the noisy samples, which can be observed on FETCHPICK and WALKER.

Ablation Study for Demonstration Filtering

To evaluate the effectiveness of our demonstration filtering design, we ablate the filtering part while fixing the restoration algorithm and the setting for the policy learning. We compare the filtering approaches listed in the following:

- **Random Filtering.** A naive way to do filtering is to randomly label half the

doi:10.6342/NTU202404722

states and actions as clean independently. This baseline serves as a bottom line for filtering methods.

- **Autoencoder (AE).** Two autoencoders are learned for states and actions, respectively. Since autoencoders tend to capture the major behaviors of training data, samples with higher reconstruction losses are considered outliers and labeled as noisy.

- **Local Outlier Factor (LOF).** LOF estimates the local deviation of a data point with respect to its neighbors given a dataset. Specifically, we filter samples by computing the LOF score based on k-nearest neighbors and then identify a sample as an outlier if the score is large.

- **Ours.** Our method calculates the LOF scores for each data point based on the features derived from the autoencoders. This method considers global representations with the autoencoders and local deviations of samples with the LOF algorithm.

The results of filtering ablation are shown in Table 3.5. We note that employing the naive random filtering method demonstrates an improvement in success rate alongside a reduction in variance. We hypothesize that the improvement results from the restoration process. Even though the training data for diffusion models and noise predictors are still noisy because of random filtering, the diffusion models can still improve the quality of samples by restoration. The above results strengthen the motivation of our filter-and-restore framework.

Utilizing Autoencoder (AE) or Local Outlier Factor (LOF) individually for filtering demonstrations shows a respectable improvement. However, the autoencoders only capture the majority of all training samples and can only better describe the global behaviors of the demonstrations. On the other hand, the Local Outlier Factor (LOF) only considers local features and is unaware of global behaviors. These restrictions hold back their ability to filter out noisy samples accurately.

Table 3.5: Demonstration filtering ablation study

Method	# of Samples	Success Rate
Noisy Data	10000	45.40% ± 11.33%
Random Filtering	7505	51.20% ± 6.80%
AE	7313	86.00% ± 3.97%
LOF	7311	76.75% ± 27.93%
Ours	7362	**91.80% ± 4.09%**

In contrast, our proposed approach, which combines AE and LOF, capitalizes on the strengths of both methods. By integrating global and local feature representations, our method surpasses the performance of its components, leading to superior results.

Ablation Study for Demonstration Restoring

To verify the effect of demonstration restoration and evaluate the designs of our restoration method proposed in Section 3.2.3, we compare different strategies that deal with samples from the noisy subsets.

We compare our method with the following baselines and variants on FETCH-PICK and WALKER environments:

- **Random forest regressor**: Random forest regressor makes predictions based on the predictions from multiple decision trees. We train the regressor using the clean state-action pairs from $D_{(\hat{s},\hat{a})}$ and use the trained regressor to predict the noisy states or actions given the corresponding clean actions or states.

- **Generation**: Given the diffusion models learned in the filtering stage, this baseline directly generates states/actions based on the corresponding actions/states. We compare our restoration method with this baseline to verify the benefits of restoring noisy samples instead of generating samples from

Table 3.6: Demonstration restoration ablation study

Method	FETCHPICK (Success Rate)	WALKER (Return)
Random forest regressor	65.00% ± 12.71%	255.3 ± 48.1
Generation	**89.89**% ± 5.13%	3452.6 ± 1528.8
Ours w/o predictor	88.20% ± 4.82%	2702.9 ± 1281.3
Ours w/o t_{thres}	89.00% ± 4.00%	4261.4 ± 857.2
Ours	**90.52**% ± 4.83%	**5066.4** ± 886.4

isotropic Gaussian noises directly.

- **Ours w/o predictor**: To verify the effectiveness of our noise level predictors, we apply restoration with a fixed noise level t for the diffusion model. Given that the total number of diffusion steps T is 100, we set the fixed noise levels as 50.

- **Ours w/o $t_{\textbf{thres}}$**: As described in Section 3.2.3, the predicted noise level t^* from predictors can be used to filter noisy demonstrations by thresholding. To evaluate the above design, we employ a variant of our method that does not apply thresholding and directly restores the noisy samples based on t^*.

- **Ours**: Our restoration method utilizes the predicted noise predictors to predict the noise level for each noisy sample and further sets a threshold t_{thres} to filter samples with smaller noise levels before restoring them.

The results of restoration ablation are shown in Table 3.6. The random forest regressor is outperformed by other diffusion-model-based restoration methods, which verifies the effectiveness of using diffusion models for recovering data. We observe that all diffusion-model-based methods perform similarly on FETCHPICK, including the augmentation with diffusion models baseline. The results infer that a well-trained diffusion model can directly generate informative training data from

doi:10.6342/NTU202404722

noises in this environment. Therefore, all of the diffusion-model-based methods perform well regardless of whether the noise predictors are included.

On the other hand, our restoration method outperforms all baselines and variants in WALKER. The results indicate that generating data from noise and restoring data without noise predictors can not restore the noisy sample effectively and highlight the importance of utilizing noise predictors and the thresholding method.

Imitation Learning Algorithms with DMDR

Using DMDR to restore demonstrations not only benefits policy learning of BC but also other imitation learning algorithms. Here, we evaluate three offline imitation learning algorithms on the FETCHPICK environment to compare the performance when using noisy demonstrations and the demonstrations restored by our DMDR.

- **BC.** Behavioral cloning (BC) is a straightforward baseline that learns a policy to map states to actions directly using the mean square error (MSE) for training.
- **Implicit BC** [100] utilizes an energy-based model (EBM) to train an implicit behavior-cloning policy, which models the expert policy. The training of the energy-based model employs the InfoNCE loss, as described in [113].
- **Diffusion Policy** Diffusion Policy [109, 110] learn a conditional diffusion model using diffusion loss to predict actions given the observed states. During inference, the diffusion model takes the current state as a condition and gradually denoises the action from noise with the reverse diffusion process.

As shown in Table 3.7, all algorithms struggle to learn directly from the noisy demonstrations but can significantly improve using the restored demonstrations from our DMDR, while Implicit BC and Diffusion Policy are even more sensitive to noisy demonstrations than BC. DMDR benefits these imitation learning algorithms by restoring the noisy demonstrations and results in superior performances with

doi:10.6342/NTU202404722

Table 3.7: Imitation Learning Algorithms with DMDR

Algorithm	Noisy Demo.	Restored Demo.
BC	44.38% ± 11.02%	**90.52**% ± 4.83%
Implicit BC	3.00% ± 3.24%	**44.56**% ± 6.98%
Diffusion Policy	25.80% ± 2.59%	**97.40**% ± 2.19%

more stable training progressing (lower standard deviation), enabling broader applications of real-world scenarios.

3.2.5 Discussion

In this paper, we propose an offline imitation learning framework (DMDR) that enables imitation learning methods to learn from noisy demonstrations. Our DMDR first filters clean samples from the demonstrations and then learn diffusion models to restore the noisy samples. The experiments show that DMDR outperforms baselines for learning from noisy demonstrations in various tasks, including robot arm manipulation, dexterous manipulation, and locomotion. Ablation studies investigate the effect of the filtering method and restoration method. We also show that our DMDR can be applied to various imitation learning algorithms to verify the effectiveness of our proposed method.

In this paper, we formulate the noises as Gaussian distributions. While the assumption is widely adopted in previous works, it may restrict the effectiveness of DMDR when the noises can not be easily formulated by Gaussian distributions. A valuable future direction is to address various noise types caused by environmental errors, *e.g.*, sensor errors and motor jitter. Currently, DMDR is designed for offline imitation learning. To further improve DMDR by incorporating a limited amount of online interactions could also be an interesting future extension.

doi:10.6342/NTU202404722

Chapter 4

Conclusion and Future Direction

This thesis aims to tackle the challenges of constructing an adaptive machine-learning pipeline for computer vision and robotic applications. To address data adaptation, it introduces feature disentanglement and meta-learning techniques to handle unseen target domains effectively. For model adaptation, the thesis explores feature disentanglement and fine-tuning methods, incorporating human feedback to enable controllable image generation. Additionally, it investigates the use of diffusion models with denoising objectives to enhance adaptability in robotic tasks.

In summary, this thesis approaches the development of an adaptive machine-learning pipeline from three key perspectives, *i.e.*, data, models, and objectives, proposing innovative solutions to meet the demands of real-world applications.

As a future direction, the techniques introduced in this thesis hold significant potential for advancing automatic control. In real-world scenarios, robotic agents often encounter data from unseen domains. To address challenging control tasks, these agents can utilize pretrained models, such as vision-language models (VLMs) or large-language models (LLMs), to enhance their understanding of task descriptions or input images from sensors. The adaptation methods proposed in this thesis can contribute to the future development of automatic robot control, which may benefit humans and society substantially.

doi:10.6342/NTU202404722

Reference

[1] B. Zong *et al.*, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *ICLR*, 2018. 1, 3, 73

[2] S. Venkataramanan *et al.*, "Attention guided anomaly localization in images," in *ECCV*, 2020. 1

[3] D. Gong *et al.*, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *ICCV*, 2019. 1, 3, 9, 11, 12, 73

[4] D. S. Tan *et al.*, "Trustmae: A noise-resilient defect classification framework using memory-augmented auto-encoders with trust regions," in *WACV*, 2021. 1, 3, 9, 12, 73

[5] H. Li *et al.*, "Domain generalization with adversarial feature learning," in *CVPR*, 2018. 2, 3

[6] S. Wang *et al.*, "Learning from extrinsic and intrinsic supervisions for domain generalization," in *ECCV*. Springer, 2020. 2, 3, 9, 11, 12

[7] X. Yue *et al.*, "Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data," in *ICCV*, 2019. 2, 3

[8] D. Li *et al.*, "Episodic training for domain generalization," in *ICCV*, 2019. 2, 3, 9, 11, 12

doi:10.6342/NTU202404722

[9] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *ACCV*, 2018. 3

[10] T. Schlegl *et al.*, "f-anogan: Fast unsupervised anomaly detection with generative adversarial networks," *Medical image analysis*, vol. 54, 2019. 3

[11] T.-Y. Lin *et al.*, "Feature pyramid networks for object detection," in *CVPR*, 2017. 5

[12] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *CVPR*, 2020. 5

[13] L. Wang *et al.*, "Learning trailer moments in full-length movies with co-contrastive attention," in *ECCV*. Springer, 2020. 8

[14] P. Bergmann *et al.*, "Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection," in *CVPR*, 2019. 9

[15] P. Mishra *et al.*, "Vt-adl: A vision transformer network for image anomaly detection and localization," *arXiv preprint arXiv:2104.10036*, 2021. 9

[16] P. Bergmann *et al.*, "Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings," in *CVPR*, 2020. 9

[17] M. Heusel *et al.*, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *NeurIPS*, vol. 30, 2017. 14

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 2672–2680. 17, 51

[19] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing

doi:10.6342/NTU202404722

generative adversarial nets," in *Advances in Neural Information Processing Systems (NIPS)*, 2016. 17

[20] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2017. 17, 25

[21] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 17

[22] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems (NIPS)*, 2014. 17

[23] T. D. Kulkarni, W. Whitney, P. Kohli, and J. B. Tenenbaum, "Deep convolutional inverse graphics network," in *Advances in Neural Information Processing Systems (NIPS)*, 2015. 17

[24] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum, "Deep convolutional inverse graphics network," in *Advances in neural information processing systems*, 2015, pp. 2539–2547. 17

[25] Y.-C. Liu, Y.-Y. Yeh, T.-C. Fu, S.-D. Wang, W.-C. Chiu, and Y.-C. F. Wang, "Detach and adapt: Learning cross-domain disentangled deep representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 17

[26] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems (NIPS)*, 2015. 17

doi:10.6342/NTU202404722

[27] Z. Zheng and L. Sun, “Disentangling latent space for vae by label relevant/irrelevant dimensions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 192–12 201. 17

[28] O. Press, T. Galanti, S. Benaim, and L. Wolf, “Emerging disentanglement in auto-encoder based unsupervised image content transfer,” *arXiv preprint arXiv:2001.05017*, 2020. 17

[29] A. H. Liu, Y.-C. Liu, Y.-Y. Yeh, and Y.-C. F. Wang, “A unified feature disentangler for multi-domain image translation and manipulation,” in *Advances in Neural Information Processing Systems (NIPS)*, 2018. 17, 24, 25

[30] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 17

[31] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, “Generative adversarial text to image synthesis,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2016. 17

[32] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, “Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 17

[33] A. Voynov and A. Babenko, “Unsupervised discovery of interpretable directions in the gan latent space,” *arXiv preprint arXiv:2002.03754*, 2020. 18

[34] Y. Shen, C. Yang, X. Tang, and B. Zhou, “Interfacegan: Interpreting the disentangled face representation learned by gans,” *arXiv preprint arXiv:2005.09635*, 2020. 18

doi:10.6342/NTU202404722

[35] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *arXiv preprint arXiv:1512.09300*, 2015. 20, 22, 24, 25

[36] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. K. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 20, 22

[37] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998. 21

[38] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010. 21

[39] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695. 27, 36, 73

[40] K. Lee, H. Liu, M. Ryu, O. Watkins, Y. Du, C. Boutilier, P. Abbeel, M. Ghavamzadeh, and S. S. Gu, "Aligning text-to-image models using human feedback," *arXiv preprint arXiv:2302.12192*, 2023. 27

[41] Y. Fan, O. Watkins, Y. Du, H. Liu, M. Ryu, C. Boutilier, P. Abbeel, M. Ghavamzadeh, K. Lee, and K. Lee, "Dpok: reinforcement learning for fine-tuning text-to-image diffusion models," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2024. 27, 30

[42] K. Black, M. Janner, Y. Du, I. Kostrikov, and S. Levine, "Training diffusion models with reinforcement learning," in *The Twelfth International Conference on Learning Representations*, 2024. 27, 28, 30

doi:10.6342/NTU202404722

[43] K. Yang, J. Tao, J. Lyu, C. Ge, J. Chen, W. Shen, X. Zhu, and X. Li, "Using human feedback to fine-tune diffusion models without any reward model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8941–8951. 27, 29, 31, 36, 37, 45

[44] M. Uehara, Y. Zhao, K. Black, E. Hajiramezanali, G. Scalia, N. L. Diamant, A. M. Tseng, S. Levine, and T. Biancalani, "Feedback efficient online fine-tuning of diffusion models," in *Proceedings of the 41st International Conference on Machine Learning*, 2024. 27, 30

[45] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, 2020. 28, 34

[46] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. 29, 35

[47] G. I. Winata, H. Zhao, A. Das, W. Tang, D. D. Yao, S.-X. Zhang, and S. Sahu, "Preference tuning with human feedback on language, speech, and vision tasks: A survey," *arXiv preprint arXiv:2409.11564*, 2024. 29

[48] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2023. 29, 30, 36

[49] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-or, "An image is worth one word: Personalizing text-to-image generation using textual inversion," in *The Eleventh International Conference on Learning Representations*, 2023. 29, 30

doi:10.6342/NTU202404722

[50] M. Prabhudesai, A. Goyal, D. Pathak, and K. Fragkiadaki, “Aligning text-to-image diffusion models with reward backpropagation,” *arXiv preprint arXiv:2310.03739*, 2023. 30

[51] J. Yang, J. Feng, and H. Huang, “Emogen: Emotional image content generation with text-to-image diffusion models,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 30

[52] S. Zhong, Z. Huang, W. Wen, J. Qin, and L. Lin, “Sur-adapter: Enhancing text-to-image pre-trained diffusion models with large language models,” in *The 31st ACM International Conference on Multimedia*, 2023. 30

[53] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3836–3847. 30

[54] R. Gandikota, J. Materzynska, J. Fiotto-Kaufman, and D. Bau, “Erasing concepts from diffusion models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 30

[55] N. Kumari, B. Zhang, S.-Y. Wang, E. Shechtman, R. Zhang, and J.-Y. Zhu, “Ablating concepts in text-to-image diffusion models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 691–22 702. 30

[56] S. Lu, Z. Wang, L. Li, Y. Liu, and A. W.-K. Kong, “Mace: Mass concept erasure in diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 6430–6440. 30

[57] W. Lin, J. Chen, J. Shi, Y. Zhu, C. Liang, J. Miao, T. Jin, Z. Zhao, F. Wu, S. Yan *et al.*, “Non-confusing generation of customized concepts in diffusion models,” *arXiv preprint arXiv:2405.06914*, 2024. 30

doi:10.6342/NTU202404722

[58] J. Xu, X. Liu, Y. Wu, Y. Tong, Q. Li, M. Ding, J. Tang, and Y. Dong, "Imagereward: Learning and evaluating human preferences for text-to-image generation," *Advances in Neural Information Processing Systems*, vol. 36, 2024. 30

[59] K. Clark, P. Vicol, K. Swersky, and D. J. Fleet, "Directly fine-tuning diffusion models on differentiable rewards," in *The Twelfth International Conference on Learning Representations*, 2024. 30

[60] M. Amadeus, W. A. C. Castañeda, A. F. Zanella, and F. R. P. Mahlow, "From pampas to pixels: Fine-tuning diffusion models for ga\'ucho heritage," *arXiv preprint arXiv:2401.05520*, 2024. 31

[61] N. Kannen, A. Ahmad, M. Andreetto, V. Prabhakaran, U. Prabhu, A. B. Dieng, P. Bhattacharyya, and S. Dave, "Beyond aesthetics: Cultural competence in text-to-image models," *arXiv preprint arXiv:2407.06863*, 2024. 31

[62] B. Wallace, M. Dang, R. Rafailov, L. Zhou, A. Lou, S. Purushwalkam, S. Ermon, C. Xiong, S. Joty, and N. Naik, "Diffusion model alignment using direct preference optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 31, 33

[63] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 31, 33, 36

[64] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020. 32, 33, 36, 73, 77

[65] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2020. 32, 33, 36

doi:10.6342/NTU202404722

[66] S. Kakade and J. Langford, "Approximately optimal approximate reinforcement learning," in *Proceedings of the Nineteenth International Conference on Machine Learning*, 2002, pp. 267–274. 32

[67] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017. 32

[68] T. B. Brown, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020. 36

[69] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023. 36

[70] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization." in *Proceedings of the International Conference on Learning Representations*, 2015. 47

[71] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, 2015. 49

[72] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *International Conference on Learning Representations*, 2016. 49

[73] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, 2017. 49

[74] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Advances in Neural Information Processing Systems*, 2017. 49

doi:10.6342/NTU202404722

[75] J. Leike, D. Krueger, T. Everitt, M. Martic, V. Maini, and S. Legg, "Scalable agent alignment via reward modeling: a research direction," *arXiv preprint arXiv:1811.07871*, 2018. 49

[76] Y. Lee, S.-H. Sun, S. Somasundaram, E. S. Hu, and J. J. Lim, "Composing complex skills by learning transition policies," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 49

[77] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, 2015. 49

[78] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020. 49

[79] S. Schaal, "Learning from demonstration," in *Advances in Neural Information Processing Systems*, 1997. 49, 51

[80] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters *et al.*, "An algorithmic perspective on imitation learning," *Foundations and Trends® in Robotics*, 2018. 49

[81] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems*, 2016. 49, 51, 71

[82] K. Zolna, S. Reed, A. Novikov, S. G. Colmenarejo, D. Budden, S. Cabi, M. Denil, N. de Freitas, and Z. Wang, "Task-relevant adversarial imitation learning," in *Conference on Robot Learning*, 2021. 49, 51

[83] I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson, "Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning," in *International Conference on Learning Representations*, 2019. 49, 51

doi:10.6342/NTU202404722

[84] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning," in *International Conference on Machine Learning*, 2000. 49, 52

[85] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *International Conference on Machine Learning*, 2004. 49, 52

[86] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Advances in Neural Information Processing Systems*, 1989. 50, 51, 70

[87] M. Bain and C. Sammut, "A framework for behavioural cloning," in *Machine Intelligence 15*, 1995. 50, 70, 71

[88] T. Nguyen, Q. Zheng, and A. Grover, "Reliable conditioning of behavioral cloning for offline reinforcement learning," *arXiv preprint arXiv:2210.05158*, 2023. 50

[89] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006. 50, 53

[90] D. Fisch, E. Kalkowski, and B. Sick, "Knowledge fusion for probabilistic generative classifiers with data mining applications," *IEEE Transactions on Knowledge and Data Engineering*, 2013. 50, 53

[91] Q. Wu, R. Gao, and H. Zha, "Bridging explicit and implicit deep generative models via neural stein estimators," in *Neural Information Processing Systems*, 2021. 50, 53

[92] G. Loaiza-Ganem, B. L. Ross, J. C. Cresswell, and A. L. Caterini, "Diagnosing and fixing manifold overfitting in deep generative models," *Transactions on Machine Learning Research*, 2022. 50, 53

[93] S.-H. Sun, H. Noh, S. Somasundaram, and J. Lim, "Neural program synthesis from diverse demonstration videos," in *International Conference on Machine Learning*, 2018. 51

doi:10.6342/NTU202404722

[94] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Robotics: Science and Systems*, 2023. 51

[95] A. O. Ly and M. Akhloufi, "Learning to drive by imitation: An overview of deep behavior cloning methods," *IEEE Transactions on Intelligent Vehicles*, 2020. 51

[96] J. Harmer, L. Gisslén, J. del Val, H. Holst, J. Bergdahl, T. Olsson, K. Sjöö, and M. Nordin, "Imitation learning with concurrent actions in 3d games," in *IEEE Conference on Computational Intelligence and Games*, 2018. 51

[97] F. Torabi, G. Warnell, and P. Stone, "Behavioral cloning from observation," in *International Joint Conference on Artificial Intelligence*, 2018. 51

[98] N. M. M. Shafiullah, Z. J. Cui, A. Altanzaya, and L. Pinto, "Behavior transformers: Cloning k modes with one stone," in *Neural Information Processing Systems*, 2022. 51

[99] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *International Conference on Artificial Intelligence and Statistics*, 2011. 51, 53

[100] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *Conference on Robot Learning*, 2022. 51, 53, 61, 65, 70, 71, 90

[101] F. Torabi, G. Warnell, and P. Stone, "Generative adversarial imitation from observation," in *International Conference on Machine Learning*, 2019. 51, 71

[102] R. Jena, C. Liu, and K. Sycara, "Augmenting gail with bc for sample efficient imitation learning," in *Conference on Robot Learning*, 2021. 51, 52

doi:10.6342/NTU202404722

[103] C.-M. Lai, H.-C. Wang, P.-C. Hsieh, Y.-C. F. Wang, M.-H. Chen, and S.-H. Sun, “Diffusion-reward adversarial imitation learning,” *arXiv preprint arXiv:2405.16194*, 2024. 51

[104] M. Chen, Y. Wang, T. Liu, Z. Yang, X. Li, Z. Wang, and T. Zhao, “On computation and generalization of generative adversarial imitation learning,” in *International Conference on Learning Representations*, 2020. 52

[105] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey *et al.*, “Maximum entropy inverse reinforcement learning.” in *Association for the Advancement of Artificial Intelligence*, 2008. 52

[106] J. Fu, K. Luo, and S. Levine, “Learning robust rewards with adverserial inverse reinforcement learning,” in *International Conference on Learning Representations*, 2018. 52, 70

[107] Y. Lee, A. Szot, S.-H. Sun, and J. J. Lim, “Generalizable imitation learning from observation via inferring goal proximity,” in *Neural Information Processing Systems*, 2021. 52, 60, 61

[108] G. Swamy, D. Wu, S. Choudhury, D. Bagnell, and S. Wu, “Inverse reinforcement learning without reinforcement learning,” in *International Conference on Machine Learning*, 2023. 52

[109] T. Pearce, T. Rashid, A. Kanervisto, D. Bignell, M. Sun, R. Georgescu, S. V. Macua, S. Z. Tan, I. Momennejad, K. Hofmann, and S. Devlin, “Imitating human behaviour with diffusion models,” in *International Conference on Learning Representations*, 2023. 52, 61, 70, 71, 90

[110] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” in *Robotics: Science and Systems*, 2023. 52, 61, 70, 71, 73, 90

doi:10.6342/NTU202404722

[111] M. Reuss, M. Li, X. Jia, and R. Lioutikov, "Goal conditioned imitation learning using score-based diffusion policies," in *Robotics: Science and Systems*, 2023. 52, 61

[112] A. Ganapathi, P. Florence, J. Varley, K. Burns, K. Goldberg, and A. Zeng, "Implicit kinematic policies: Unifying joint and cartesian action spaces in end-to-end robot learning," in *International Conference on Robotics and Automation*, 2022. 53

[113] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018. 53, 90

[114] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *International Conference on Computer Vision*, 2019. 53

[115] L. Wang, C. Fernandez, and C. Stiller, "High-level decision making for automated highway driving via behavior cloning," *IEEE Transactions on Intelligent Vehicles*, 2022. 53

[116] Y. Du and I. Mordatch, "Implicit generation and modeling with energy based models," in *Neural Information Processing Systems*, 2019. 53

[117] Y. Song and D. P. Kingma, "How to train your energy-based models," *arXiv preprint arXiv:2101.03288*, 2021. 53

[118] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013. 53

[119] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International conference on machine learning*. PMLR, 2015, pp. 1530–1538. 53

doi:10.6342/NTU202404722

[120] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *International Conference on Learning Representations*, 2017. 53

[121] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning*, 2015. 54

[122] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning*, 2021. 54

[123] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021. 54

[124] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. B. Tenenbaum, "Learning to act from actionless videos through dense correspondences," *arXiv preprint arXiv:2310.08576*, 2023. 54

[125] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," in *The Eleventh International Conference on Learning Representations*, 2023. 54

[126] A. J. J Ho, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, 2020. 54, 56

[127] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, "D4rl: Datasets for deep data-driven reinforcement learning," *arXiv preprint arXiv:2004.07219*, 2020. 60, 67

[128] M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder *et al.*, "Multi-goal reinforcement learning: Challenging robotics environments and request for research," *arXiv preprint arXiv:1802.09464*, 2018. 60, 83

doi:10.6342/NTU202404722

[129] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2018. 60

[130] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “Openai gym,” 2016. 60

[131] I. Kostrikov, “Pytorch implementations of reinforcement learning algorithms,” https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail, 2018. 60

[132] B. Fang, S. Jia, D. Guo, M. Xu, S. Wen, and F. Sun, “Survey of imitation learning for robotic manipulation,” *International Journal of Intelligent Robotics and Applications*, vol. 3, pp. 362–369, 2019. 68

[133] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, “End-to-end driving via conditional imitation learning,” in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 4693–4700. 68

[134] L. Le Mero, D. Yi, M. Dianati, and A. Mouzakitis, “A survey on imitation learning techniques for end-to-end autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14 128–14 147, 2022. 68

[135] S. Ross and D. Bagnell, “Efficient reductions for imitation learning,” in *International Conference on Artificial Intelligence and Statistics*, 2010. 68

[136] J. Harmer, L. Gisslén, J. del Val, H. Holst, J. Bergdahl, T. Olsson, K. Sjöö, and M. Nordin, “Imitation learning with concurrent actions in 3d games,” in *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 2018, pp. 1–8. 68

doi:10.6342/NTU202404722

[137] F. F. Duarte, N. Lau, A. Pereira, and L. P. Reis, "A survey of planning and learning in games," *Applied Sciences*, vol. 10, no. 13, p. 4529, 2020. 68

[138] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," *arXiv preprint arXiv:1606.06565*, 2016. 68

[139] D. Hadfield-Menell, S. Milli, P. Abbeel, S. J. Russell, and A. Dragan, "Inverse reward design," *Advances in neural information processing systems*, vol. 30, 2017. 68

[140] S.-M. Yang and S.-J. Ke, "Performance evaluation of a velocity observer for accurate velocity estimation of servo motor drives," *IEEE Transactions on Industry Applications*, vol. 36, no. 1, pp. 98–104, 2000. 68

[141] V. G. Biju, A.-M. Schmitt, and B. Engelmann, "Assessing the influence of sensor-induced noise on machine-learning-based changeover detection in cnc machines," *Sensors*, vol. 24, no. 2, p. 330, 2024. 68, 82

[142] K. Smeds and X. Lu, "Effect of sampling jitter and control jitter on positioning error in motion control systems," *Precision Engineering*, vol. 36, no. 2, pp. 175–192, 2012. 68, 75

[143] A. W. A. Ali, F. A. A. Razak, and N. Hayima, "A review on the ac servo motor control systems," *ELEKTRIKA-Journal of Electrical Engineering*, vol. 19, no. 2, pp. 22–39, 2020. 68, 75

[144] D. Gao, S. Wang, Y. Yang, H. Zhang, H. Chen, X. Mei, S. Chen, and J. Qiu, "An intelligent control method for servo motor based on reinforcement learning," *Algorithms*, vol. 17, no. 1, p. 14, 2023. 68, 75

[145] Y.-H. Wu, N. Charoenphakdee, H. Bao, V. Tangkaratt, and M. Sugiyama, "Imitation learning from imperfect demonstration," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6818–6827. 69, 71, 83

doi:10.6342/NTU202404722

[146] Y. Wang, C. Xu, B. Du, and H. Lee, "Learning to weight imperfect demonstrations," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 961–10 970. 69

[147] F. Sasaki and R. Yamashina, "Behavioral cloning from noisy demonstrations," in *International Conference on Learning Representations*, 2020. 69, 72, 84

[148] H. Chung, B. Sim, D. Ryu, and J. C. Ye, "Improving diffusion models for inverse problems using manifold constraints," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25 683–25 696, 2022. 69

[149] B. Kawar, M. Elad, S. Ermon, and J. Song, "Denoising diffusion restoration models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 593–23 606, 2022. 69, 73, 77

[150] N. Murata, K. Saito, C.-H. Lai, Y. Takida, T. Uesaka, Y. Mitsufuji, and S. Ermon, "GibbsDDRM: A partially collapsed Gibbs sampler for solving blind inverse problems with denoising diffusion restoration," in *Proceedings of the 40th International Conference on Machine Learning*, 2023. 69, 73, 77

[151] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104. 70

[152] Y. Lee, A. Szot, S.-H. Sun, and J. J. Lim, "Generalizable imitation learning from observation via inferring goal proximity," *Advances in neural information processing systems*, vol. 34, pp. 16 118–16 130, 2021. 70, 83

[153] D. Michie, M. Bain, and J. Hayes-Miches, "Cognitive models from subcognitive skills," *IEE control engineering series*, vol. 44, pp. 71–99, 1990. 71

doi:10.6342/NTU202404722

[154] F. Torabi, G. Warnell, and P. Stone, "Behavioral cloning from observation," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 4950–4957. 71

[155] D. Brown, W. Goo, P. Nagarajan, and S. Niekum, "Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations," in *International conference on machine learning*. PMLR, 2019, pp. 783–792. 71, 83

[156] D. S. Brown, W. Goo, and S. Niekum, "Better-than-demonstrator imitation learning via automatically-ranked demonstrations," in *Conference on robot learning*. PMLR, 2020, pp. 330–359. 71, 83

[157] V. Tangkaratt, N. Charoenphakdee, and M. Sugiyama, "Robust imitation learning from noisy demonstrations," in *International Conference on Artificial Intelligence and Statistics*, 2020. 71, 83

[158] S. Zhang, Z. Cao, D. Sadigh, and Y. Sui, "Confidence-aware imitation learning from demonstrations with varying optimality," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 340–12 350, 2021. 71, 83

[159] G.-H. Kim, S. Seo, J. Lee, W. Jeon, H. Hwang, H. Yang, and K.-E. Kim, "Demodice: Offline imitation learning with supplementary imperfect demonstrations," in *International Conference on Learning Representations*, 2021. 71

[160] L. Yu, T. Yu, J. Song, W. Neiswanger, and S. Ermon, "Offline imitation learning with suboptimal demonstrations via relaxed distribution matching," in *Proceedings of the AAAI conference on artificial intelligence*, 2023, pp. 11 016–11 024. 72

doi:10.6342/NTU202404722

[161] H. Xu, X. Zhan, H. Yin, and H. Qin, "Discriminator-weighted offline imitation learning from suboptimal demonstrations," in *International Conference on Machine Learning*. PMLR, 2022, pp. 24 725–24 742. 72

[162] W. Zhang, H. Xu, H. Niu, P. Cheng, M. Li, H. Zhang, G. Zhou, and X. Zhan, "Discriminator-guided model-based offline imitation learning," in *Conference on Robot Learning*. PMLR, 2023, pp. 1266–1276. 72

[163] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*. PMLR, 2018, pp. 4393–4402. 72

[164] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft, "Deep semi-supervised anomaly detection," *arXiv preprint arXiv:1906.02694*, 2019. 72

[165] Z. Wang *et al.*, "Inductive multi-view semi-supervised anomaly detection via probabilistic modeling," in *2019 IEEE International Conference on Big Knowledge (ICBK)*. IEEE, 2019. 72

[166] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," *arXiv preprint arXiv:1812.04606*, 2018. 72

[167] K. Sohn, C.-L. Li, J. Yoon, M. Jin, and T. Pfister, "Learning and evaluating representations for deep one-class classification," *arXiv preprint arXiv:2011.02578*, 2020. 72

[168] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister, "Cutpaste: Self-supervised learning for anomaly detection and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9664–9674. 72

[169] H. Park, J. Noh, and B. Ham, "Learning memory-guided normality for anomaly detection," in *CVPR*, 2020. 73

doi:10.6342/NTU202404722

[170] M. Ren, W. Zeng, B. Yang, and R. Urtasun, “Learning to reweight examples for robust deep learning,” in *International conference on machine learning*. PMLR, 2018, pp. 4334–4343. 73

[171] J. Li, R. Socher, and S. C. Hoi, “Dividemix: Learning with noisy labels as semi-supervised learning,” in *International Conference on Learning Representations*, 2020. 73

[172] N. Karim, M. N. Rizve, N. Rahnavard, A. Mian, and M. Shah, “Unicon: Combating label noise through uniform selection and contrastive learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9676–9686. 73

[173] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, “Video diffusion models,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 8633–8646, 2022. 73

[174] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” in *International Conference on Learning Representations*, 2021. 73

[175] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, “Consistency models,” *arXiv preprint arXiv:2303.01469*, 2023. 73

[176] Z. Wang, J. J. Hunt, and M. Zhou, “Diffusion policies as an expressive policy class for offline reinforcement learning,” in *The Eleventh International Conference on Learning Representations*, 2023. 73

[177] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *International Conference On Intelligent Robots and Systems*, 2012. 75

[178] J. Gabela, A. Kealy, S. Li, M. Hedley, W. Moran, W. Ni, and S. Williams, “The effect of linear approximation and gaussian noise assumption in multi-

doi:10.6342/NTU202404722

sensor positioning through experimental evaluation," *IEEE Sensors Journal*, vol. 19, no. 22, pp. 10 719–10 727, 2019. 82

[179] M. El Helou and S. Süsstrunk, "Blind universal bayesian image denoising with gaussian noise level learning," *IEEE Transactions on Image Processing*, vol. 29, pp. 4885–4897, 2020. 82

[180] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016. 83

[181] I. Kostrikov, "Pytorch implementations of reinforcement learning algorithms," https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail, 2018. 83

doi:10.6342/NTU202404722