D²PPO: Diffusion Policy Policy Optimization with Dispersive Loss

Anonymous submission

Abstract

Diffusion policies excel at robotic manipulation by naturally modeling multimodal action distributions in high-dimensional spaces. Nevertheless, diffusion policies suffer from **diffusion representation collapse**: semantically similar observations are mapped to indistinguishable features, ultimately impairing their ability to handle subtle but critical variations required for complex robotic manipulation. To address this problem, we propose D²PPO (Diffusion Policy Policy Optimization with Dispersive Loss). D²PPO introduces dispersive loss regularization that combats representation collapse by treating all hidden representations within each batch as negative pairs. D²PPO compels the network to learn discriminative representations of similar observations, thereby enabling the policy to identify subtle yet crucial differences necessary for precise manipulation. In evaluation, we find that early-layer regularization benefits simple tasks, while late-layer regularization sharply enhances performance on complex manipulation tasks. On RoboMimic benchmarks, D²PPO achieves an average improvement of 22.7% in pre-training and 26.1% after fine-tuning, setting new SOTA results. In comparison with SOTA, results of real-world experiments on a Franka Emika Panda robot show the excitingly high success rate of our method. The superiority of our method is especially evident in complex tasks. **Code and supplementary materials are provided in the submitted package.**

Introduction

Diffusion models have recently emerged as a promising approach for learning robot control policies (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020; Peebles and Xie 2023). Through an iterative denoising mechanism, diffusion policies are able to model complex and multimodal action distributions (Lee et al. 2025; Geng et al. 2024; Li et al. 2023), making them well-suited for high-dimensional continuous control tasks (Chi et al. 2023).

Despite these advantages, we observe that diffusion policies still face significant challenges with low success rates when executing complex manipulation tasks. Consider a robotic arm performing a grasping task where the observations in two scenarios appear highly similar but require distinctly different actions. Standard diffusion policies typically fail to distinguish the subtle yet critical differences between similar observations, consequently generating identical actions that lead to task failure, as shown in Figure 1(a).

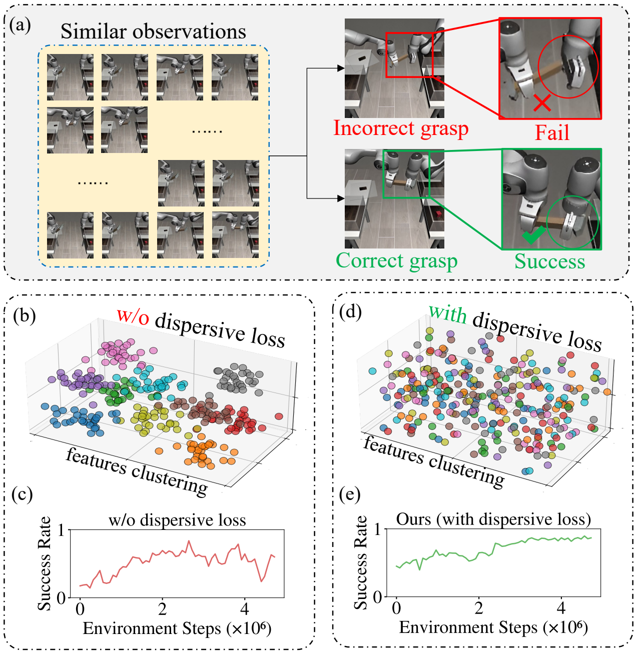

Figure 1: Effect of Dispersive Loss on Representation Quality and Policy Performance. (a) Shows how representation collapse leads to grasping failures: similar observations are encoded into nearly identical features, causing the policy to make incorrect decisions between successful (green) and failed (red) grasping attempts. (b) and (c) demonstrate training without dispersive loss: clustered feature representations with limited diversity and correspondingly slow, unstable learning curves with poor final performance. (d) and (e) show the effect of incorporating dispersive loss: well-distributed, diverse feature representations and significantly improved learning curves with faster convergence and higher success rates.

Through detailed analysis of these task failures, we identify the root cause as **diffusion representation collapse**. This phenomenon arises because diffusion policies rely primarily on reconstruction loss (Karras et al. 2022, 2023; Song et al. 2023). Such loss optimizes for denoising accuracy but neglects the quality and diversity of intermediate feature representations. Consequently, hidden layers produce nearly indistinguishable embeddings for semantically different observations. As illustrated in Figure 1(b), clustering analysis of hidden layer features reveals that numerous similar state representations are grouped into nearly identical clusters, indicating poor feature diversity. This representation collapse directly leads to poor performance, as shown in Figure 1(c). This representation collapse is particularly problematic for robotic manipulation, where subtle observational differences often require substantially different actions for successful task completion. When the learned representations fail to distinguish these nuanced differences, the policy cannot generate the context-sensitive actions required for precise manipulation tasks.

Motivated by advances in computer vision, we try to mitigate this problem through explicit representation regularization. As an explicit form of representation regularization, contrastive representation learning is the most widely used approach for enhancing the discriminability and robustness of feature embeddings (van den Oord, Li, and Vinyals 2018; Chen et al. 2020; He et al. 2020; Arora et al. 2019; Wang and Isola 2020). However, traditional contrastive learning typically requires constructing positive-negative sample pairs, additional pre-training stages (Oquab et al. 2023), auxiliary model components (Chen et al. 2025a), and access to external data for reliable positive pair generation (Yu et al. 2024).

Given these limitations, we propose dispersive loss regularization, a "**contrastive loss without positive pairs**" that encourages internal representations to spread out in the hidden space. Specifically, we integrate this dispersive regularization into the standard diffusion loss, maximizing feature dispersion within each batch and thereby enabling the network to distinguish the subtle variations essential for precise manipulation tasks. Importantly, our approach requires no extra pre-training, model parameters, or external data. As illustrated in Figure 1(d-e), dispersive loss promotes well-separated feature representations and leads to significantly improved learning performance.

Building upon this insight, we propose **D²PPO (Diffusion Policy Policy Optimization with Dispersive Loss)**, a two-stage training strategy: (1) pre-training with dispersive loss to encourage feature dispersion within each batch; (2) fine-tuning with PPO to maximize task success (Williams 1992; Schulman et al. 2017; Ren et al. 2024). This approach combines the generative expressiveness of diffusion models with the goal-directed precision of reinforcement learning, while ensuring that similar observations maintain distinct feature representations crucial for precise manipulation tasks. Detailed theoretical framework and mechanism understanding can be found in Appendix A.

This paper makes three key contributions:

- Through extensive experiments and analysis, we identify **diffusion representation collapse** in diffusion policies as the underlying cause of their inability to handle complex manipulation tasks.
- We propose D²PPO, which addresses diffusion representation collapse by introducing dispersive loss that eliminates the dependency on positive-negative sample pair construction in contrastive learning. We compare three dispersive variants (InfoNCE-L2, InfoNCE-Cosine, Hinge) and analyze their effects across different feature layers.
- We verify the impact of dispersive loss at different layers on task success rates across varying task complexities, discovering that more challenging tasks benefit increasingly from dispersive regularization. Building on this foundation, we further enhance the trained diffusion policies using the policy gradient algorithm for fine-tuning, achieving improved accuracy with **real robot validation**.

Related Work

Diffusion Policy for Robot Control. Diffusion models revolutionized generative modeling through DDPM (Ho, Jain, and Abbeel 2020), DDIM (Song, Meng, and Ermon 2020), Latent Diffusion Models (Rombach et al. 2022), and One step diffusion (Frans et al. 2024). This paradigm inspired Diffusion Policy (Chi et al. 2023), which adapted iterative denoising to robot control by modeling action distributions as denoising processes. Recent extensions include 3D Diffuser Actor (Ke, Gkanatsios, and Fragkiadaki 2024), 3D Diffusion Policy (Ze et al. 2024b), and humanoid manipulation applications (Ze et al. 2024a). However, direct application to robotics revealed performance limitations on complex manipulation tasks due to representation collapse, which our work addresses by enhancing diffusion policy representations through dispersive loss regularization to improve manipulation accuracy.

Policy Optimization for Diffusion-Based Control. Traditional policy gradient methods (TRPO (Schulman et al. 2015a), PPO (Schulman et al. 2015b, 2017)) required adaptation for diffusion policies' iterative denoising process. Recent approaches include DPPO (Ren et al. 2024) with two-layer MDP formulations, ReinFlow (Zhang et al. 2025; Hafner et al. 2023) for flow matching with online reinforcement learning, FDPP (Chen et al. 2025b) for human preference integration, and TrajHF (Li et al. 2025) for human feedback-driven trajectory generation. However, these online fine-tuning methods focus primarily on algorithmic design while neglecting pre-trained diffusion policy representations. Our D²PPO provides a superior starting point for fine-tuning by enhancing representations during pre-training.

Representation Learning as the Missing Link. Contrastive learning approaches (InfoNCE (van den Oord, Li, and Vinyals 2018), SimCLR (Chen et al. 2020), supervised contrastive learning (Khosla et al. 2020)) have shown that regularizing learned representations improves generalization. Related approaches in representation regularization include REPA (Yu et al. 2024) for alignment with external encoders like DINOv2 (Oquab et al. 2023), and various contrastive learning extensions building on InfoNCE foundations. Dispersive loss (Wang and He 2025) provides an elegant "contrastive loss without positive pairs" that encourages representational diversity. However, representation methods like dispersive loss have rarely been applied to diffusion policy learning. Our work bridges this insight with

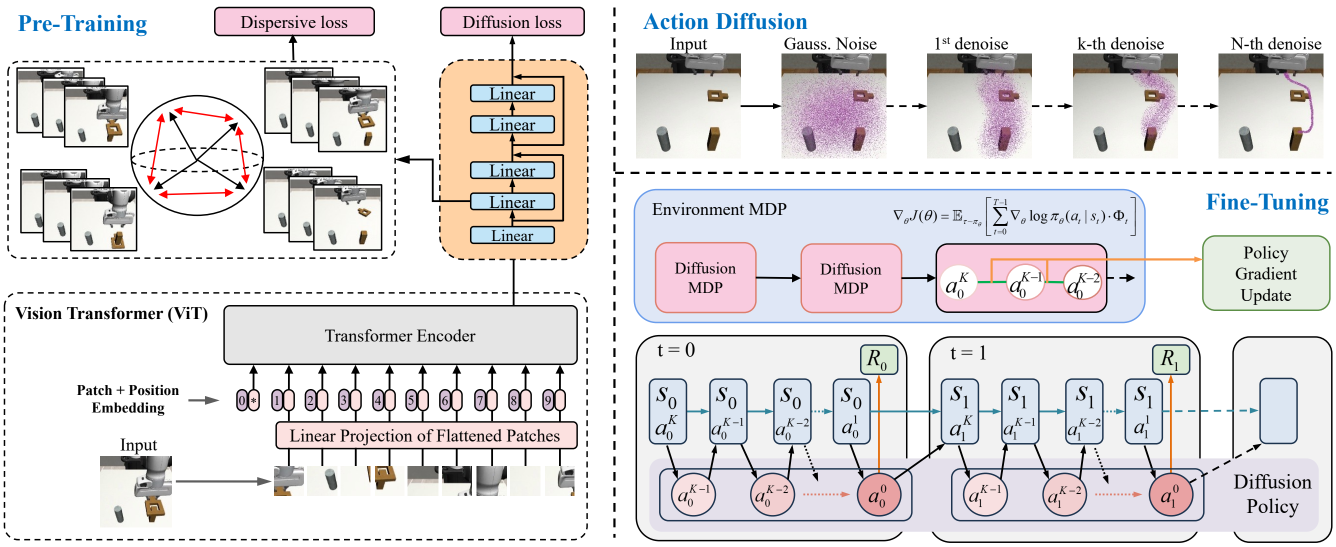

Figure 2: D²PPO Framework Overview. The complete two-stage training paradigm: **Left:** Pre-training stage with Vision Transformer (ViT) feature extraction and dispersive loss regularization to prevent representation collapse; **Top-right:** Action diffusion process showing iterative denoising from Gaussian noise to final actions; **Bottom-right:** Fine-tuning stage with policy gradient optimization using two-layer MDP formulation for environment interaction.

diffusion-based robot control, addressing representation collapse that limits complex task performance.

Preliminaries

Diffusion Policy: Diffusion Policy (Chi et al. 2023) models the action distribution using a diffusion process. Given an observation o_t, the policy generates actions through an iterative denoising process. The forward diffusion process adds Gaussian noise to the action:

$$q(a^1, \ldots, a^K | a^0) = \prod_{k=1}^{K} q(a^k | a^{k-1}) \tag{1}$$

where $q(a^k|a^{k-1}) = \mathcal{N}(a^k; \sqrt{1-\beta_k}a^{k-1}, \beta_k \mathrm{I})$, a^0 is the original clean action, a^k represents the noisy action at denoising timestep k, K is the total number of denoising steps, β_k is the noise schedule that controls the amount of noise added at each step, and I is the identity matrix.

The reverse process learns to predict the noise ϵ at each timestep:

$$p_\theta(a^{k-1}|a^k, o) = \mathcal{N}(a^{k-1}; \mu_\theta(a^k, k, o), \sigma_k^2) \tag{2}$$

where p_θ is the learned reverse process parameterized by θ, o is the observation, $\mu_\theta(a^k, k, o)$ is the predicted mean of the denoising distribution, σ_k^2 is the variance at timestep k, and

$$\mu_\theta(a^k, k, o) = \frac{1}{\sqrt{\alpha_k}}\left(a^k - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}}\epsilon_\theta(a^k, k, o)\right) \tag{3}$$

with $\alpha_k = 1 - \beta_k$, $\bar{\alpha}_k = \prod_{s=1}^{k}\alpha_s$, and $\epsilon_\theta(a^k, k, o)$ is the neural network that predicts the noise added at timestep k.

Dispersive Loss: The key insight of dispersive loss (Wang and He 2025) is to derive a "contrastive loss without positive pairs" by reformulating traditional contrastive learning objectives. Traditional contrastive learning optimizes the InfoNCE (van den Oord, Li, and Vinyals 2018) objective, which can be decomposed into two terms:

$$\mathcal{L}_{\text{InfoNCE}} = \frac{\mathcal{D}(z_i, z_i^+)}{\tau} + \log \sum_j \exp\left(-\frac{\mathcal{D}(z_i, z_j)}{\tau}\right) \tag{4}$$

where z_i and z_i^+ are positive pairs, z_j are negative samples, $\mathcal{D}(\cdot,\cdot)$ is a distance function, and τ is the temperature parameter. The first term enforces alignment between positive pairs, while the second term encourages dispersion among all samples.

The key insight of dispersive loss is to remove the positive pair alignment term entirely, leading to the dispersive objective that focuses solely on representation dispersion:

$$\mathcal{L}_{\text{Disp}} = \log \mathbb{E}_{i,j}\left[\exp\left(-\frac{\mathcal{D}(z_i, z_j)}{\tau}\right)\right] \tag{5}$$

Eq. (5) encourages all representations within a batch to be maximally dispersed in the hidden space, preventing representation collapse and promoting diversity without requiring explicit positive pair construction.

Method

D²PPO: Enhanced Diffusion Policy Policy Optimization using Dispersive Loss. D²PPO addresses representation collapse in diffusion policies through dispersive loss regularization. D²PPO adopts a two-stage training paradigm: dispersive pre-training followed by policy gradient optimization, as illustrated in Figure 2. Complete mathematical derivations are provided in Appendix B. Algorithm pseudocode is provided in Appendix C.

Stage 1: Enhanced Pre-training with Dispersive Loss. We augment the standard diffusion loss with dispersive regularization. Our approach employs a Vision Transformer encoder (ViT) (Vaswani et al. 2017; Dosovitskiy et al. 2021) for visual feature extraction and applies regularization to selected intermediate layers of the MLP denoising network. We define our pre-training objective as:

$$\mathcal{L}_{\text{D}^2\text{PPO}}^{\text{pre-train}} = \mathcal{L}_{\text{diff}} + \lambda \mathcal{L}_{\text{disp}} \tag{6}$$

This combined objective serves two purposes: the diffusion loss $\mathcal{L}_{\text{diff}}$ ensures accurate noise prediction for proper denoising, while the dispersive loss $\mathcal{L}_{\text{disp}}$ with weight λ encourages representational diversity in the learned features.

The dispersive loss is computed by averaging over all denoising timesteps:

$$\mathcal{L}_{\text{disp}} = \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_{\text{disp}}^{\text{variant}}(\mathbf{H}_k) \tag{7}$$

where K denotes the total number of denoising steps in the diffusion process, k is the current denoising timestep index, and $\mathcal{L}_{\text{disp}}^{\text{variant}}(\mathbf{H}_k)$ represents the specific dispersive loss variant (InfoNCE-L2, InfoNCE-Cosine, or Hinge) computed at timestep k. The notation $\mathbf{H}_k = \{h_{i,k}\}_{i=1}^{B}$ denotes the collection of intermediate feature representations from all B samples in the batch at timestep k.

We implement three main variants of dispersive loss, each derived from different contrastive learning methods by removing the positive alignment term:

1. InfoNCE-based Dispersive Loss with L2 Distance:

$$\mathcal{L}_{\text{disp}}^{\text{InfoNCE-L2}} = \log \mathbb{E}_{i,j} \left[\exp\left(-\frac{||h_i - h_j||_2^2}{\tau} \right) \right] \tag{8}$$

Eq. (8) uses squared ℓ_2 distance $\mathcal{D}(h_i, h_j) = ||h_i - h_j||_2^2$, which measures Euclidean distance in the representation space. This formulation encourages dispersion based on geometric distance between feature vectors.

2. InfoNCE-based Dispersive Loss with Cosine Distance:

$$\mathcal{L}_{\text{disp}}^{\text{InfoNCE-Cos}} = \log \mathbb{E}_{i,j} \left[\exp\left(-\frac{1 - \frac{h_i^T h_j}{||h_i||_2 \cdot ||h_j||_2}}{\tau} \right) \right] \tag{9}$$

Eq. (9) uses cosine dissimilarity $\mathcal{D}(h_i, h_j) = 1 - h_i^T h_j / ||h_i||_2 \cdot ||h_j||_2$, which captures angular differences between normalized representations. This formulation is scale-invariant and focuses on directional diversity.

3. Hinge Loss-based Dispersive Loss:

$$\mathcal{L}_{\text{disp}}^{\text{Hinge}} = \mathbb{E}_{i,j} \left[\max(0, \epsilon - \mathcal{D}(h_i, h_j))^2 \right] \tag{10}$$

Eq. (10) is derived from hinge loss by removing the positive pair term, focusing purely on enforcing a minimum margin ϵ between all representation pairs. This formulation directly penalizes representations that are closer than the margin threshold, providing explicit control over the minimum dispersion distance.

In our experiments, we evaluate all three variants across different network layers to determine optimal configurations for each task complexity level.

Stage 2: Dispersive loss-augmented diffusion policy optimization. In the optimization stage, D²PPO leverages the enhanced representations learned during pre-training to optimize diffusion policies through reinforcement learning. Our approach strategically focuses this stage on reward maximization while preserving the representational structure established during pre-training.

Overall Objective: We aim to optimize the diffusion policy π_θ to maximize expected return:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)] \tag{11}$$

For diffusion policies, the action probability involves the entire denoising chain from a_t^K (pure noise) to a_t^0 (final action):

$$\pi_\theta(a_t^0 | s_t) = \int p(a_t^K) \prod_{k=1}^{K} p_\theta(a_t^{k-1} | a_t^k, s_t) da_t^{1:K} \tag{12}$$

where the diffusion policy is a multi-step latent policy where we cannot directly obtain $\pi(a|s)$ as in standard policies, but must consider the joint probability across the entire denoising chain.

Using the chain rule, the log probability gradient becomes:

$$\nabla_\theta \log \pi_\theta(a_t^0 | s_t) = \sum_{k=1}^{K} \nabla_\theta \log p_\theta(a_t^{k-1} | a_t^k, s_t) \tag{13}$$

Since the previous step involves multi-step generation, we can only compute gradients for each conditional probability step and accumulate them using the chain rule. This step crucially transforms the entire policy gradient into differentiable losses over individual denoising steps, enabling gradient-based optimization.

To accelerate training, we employ importance sampling since computing gradients for all $k = 1, \ldots, K$ steps is computationally expensive:

$$\nabla_\theta J(\theta) \approx \frac{1}{|\mathcal{S}|} \sum_{k \in \mathcal{S}} \frac{K}{p(k)} \nabla_\theta \log p_\theta(a_t^{k-1} | a_t^k, s_t) \cdot \hat{A}_t^{(k)} \tag{14}$$

where $\mathcal{S}$ denotes the sampled subset of denoising steps, $|\mathcal{S}|$ is the subset size, $p(k)$ is the sampling probability for step k, and $\hat{A}_t^{(k)}$ is the step-conditioned advantage estimate. This formulation enables efficient policy gradient computation while maintaining the enhanced representational structure from pre-training.

Experiments

We evaluate D²PPO through a comprehensive three-stage experimental evaluation: (1) pre-training experiments that validate dispersive loss effectiveness, (2) fine-tuning experiments that demonstrate superior performance compared to existing algorithms, and (3) real robot experiments that showcase practical deployment capabilities. We conduct experiments on four representative tasks from the robomimic benchmark (Mandlekar et al. 2022; Todorov, Erez, and Tassa 2012): Lift, Can, Square, and Transport, spanning different manipulation complexities. Detailed experimental configurations, datasets, and supplementary experiments are provided in Appendix D.

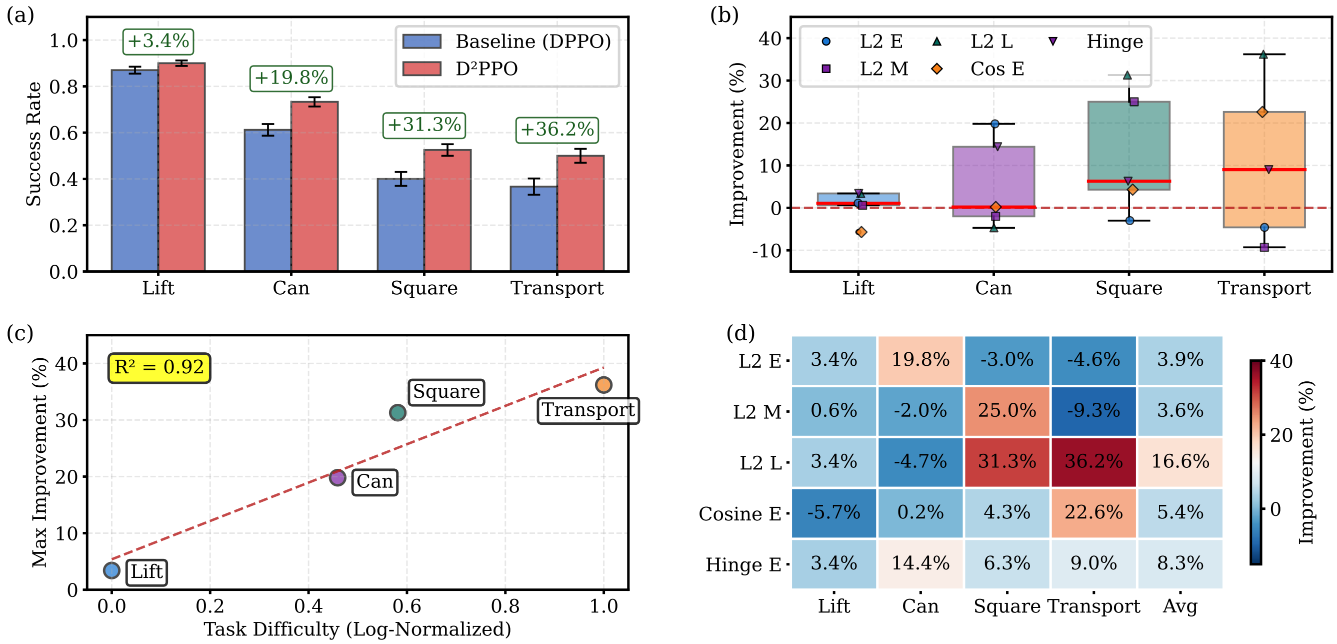

Figure 3: Comprehensive pre-training experimental results using D²PPO with dispersive loss across four robotic manipulation tasks. (a) Performance comparison showing baseline (DPPO (Ren et al. 2024)) versus D²PPO success rates with error bars, demonstrating consistent improvements across all tasks. (b) Distribution of improvement rates across five dispersive loss variants (InfoNCE L2 Early/Mid/Late, InfoNCE Cosine Early, Hinge Early). (c) Task difficulty correlation analysis showing the relationship between log-normalized task complexity and maximum improvement rates. (d) Method suitability matrix heatmap displaying improvement percentages for each dispersive loss variant across tasks, with color intensity indicating effectiveness.

Pre-training Experiments with Dispersive Loss

We first evaluate the effectiveness of dispersive loss regularization during the pre-training stage, examining multiple variants and configurations to validate the impact on representation learning and policy performance.

The four tasks represent increasing complexity levels: **Lift** (basic object manipulation), **Can** (cylindrical object grasping), **Square** (precise peg-in-hole placement), and **Transport** (multi-object coordination). We define task difficulty using log-normalized execution steps, providing objective complexity quantification as shown in Table 1:

Task	Steps	Difficulty	Complexity
Lift	108	0.000	Easiest
Can	224	0.459	Moderate
Square	272	0.581	High
Transport	529	1.000	Highest

Table 1: Task difficulty hierarchy based on average execution steps and log-normalized difficulty scores.

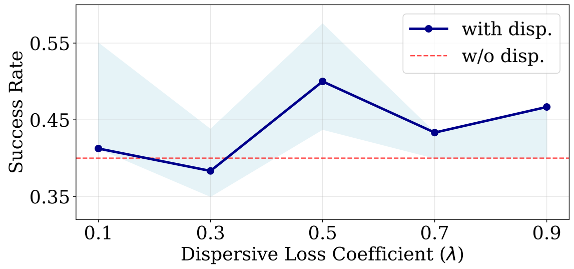

Figure 4: Effect of dispersive loss coefficient λ on Square task performance.

Figure 3 presents comprehensive pre-training experimental results across all five dispersive loss variants and four robotic manipulation tasks. Figure 3(a) shows D²PPO achieving consistent improvements across all tasks, with enhancement rates ranging from +3.4% (Lift) to +36.2% (Transport). Figure 3(b) reveals the distribution of improvement rates across five dispersive loss variants, with distinct markers for each algorithm type. Figure 3(c) demonstrates a strong positive correlation (R² = 0.92) between log-normalized task difficulty and maximum improvement rates, validating our hypothesis that representation quality becomes increasingly critical for complex robotic tasks. Figure 3(d) provides empirical guidance for layer selection through a heatmap showing dispersive loss effectiveness at different network layers, establishing that simple tasks achieve optimal performance with early-layer application while complex tasks require late-layer regularization.

These comprehensive pre-training results establish that D²PPO with dispersive loss achieves superior performance compared to DPPO, with an average improvement of 22.7%

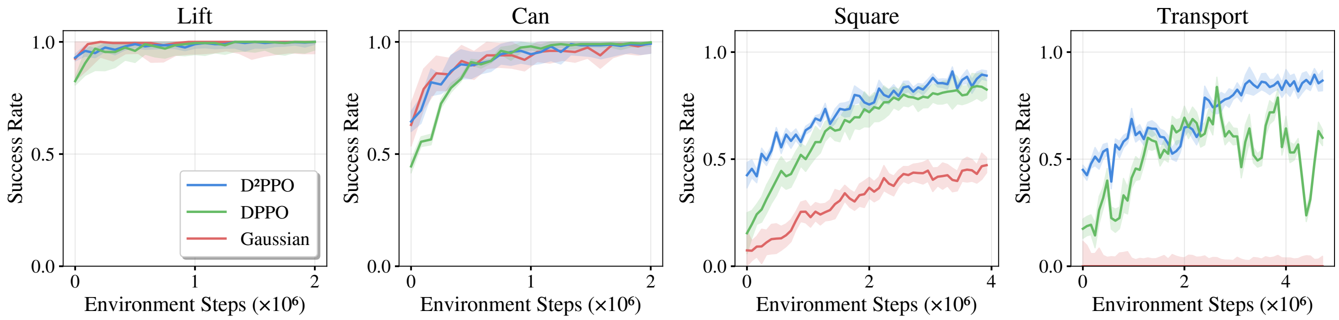

Figure 5: Policy gradient fine-tuning results across four robotic manipulation tasks. The learning curves demonstrate that D²PPO consistently achieves superior sample efficiency and final performance compared to baseline DPPO and Gaussian policies.

Table 2: Performance comparison on robomimic tasks. **Bold** indicates best performance, underline indicates second-best.

Method	Success Rate				Avg.	Median	Std.	Rank
	Lift	Can	Square	Transport				
LSTM-GMM (Mandlekar et al. 2022)	0.93	0.81	0.59	0.20	0.63	0.70	0.32	7
IBC (Florence et al. 2022)	0.15	0.01	0.00	0.00	0.04	0.01	0.07	9
BET (Shafiullah et al. 2022)	**1.00**	0.90	0.43	0.06	0.60	0.67	0.41	8
DiffusionPolicy-C (Chi et al. 2023)	0.97	0.96	0.82	0.46	0.80	0.89	0.23	4
DiffusionPolicy-T (Chi et al. 2023)	**1.00**	0.94	0.81	0.35	0.78	0.88	0.29	5
DPPO (Ren et al. 2024)	**1.00**	**1.00**	0.83	0.60	0.86	0.92	0.18	2
MTIL (10-step) (Zhou et al. 2025)	0.94	0.83	0.61	0.22	0.65	0.72	0.31	6
MTIL (Full History) (Zhou et al. 2025)	**1.00**	0.96	0.83	0.48	0.82	0.90	0.23	3
D²PPO (Ours)	**1.00**	**1.00**	**0.89**	**0.87**	**0.94**	**0.95**	**0.06**	**1**

across all tasks.

To further investigate the impact of the dispersive loss coefficient λ, we conduct detailed ablation studies on the Square task, which requires precise spatial coordination and is particularly sensitive to representation quality. Figure 4 reveals a non-monotonic relationship: performance peaks at $\lambda = 0.5$ (14.3% improvement over baseline), while $\lambda = 0.3$ degrades below baseline performance. This suggests insufficient regularization fails to encourage adequate representation diversity, while excessive values ($\lambda > 0.5$) may interfere with task-relevant learning, highlighting the importance of balanced dispersive regularization.

Fine-tuning Experiments

As the second phase of our evaluation, we validate the complete D²PPO method through comprehensive policy gradient fine-tuning experiments, demonstrating how enhanced representations translate into superior reinforcement learning performance and comparing against existing SOTA algorithms. For each task, we select the best-performing pretrained model weights from our dispersive loss variants and use them as initialization for policy gradient fine-tuning.

Figure 5 presents the fine-tuning learning curves across all four tasks, revealing several critical insights: **(1) Enhanced Sample Efficiency:** D²PPO demonstrates consistently faster convergence across almost all tasks, requiring significantly fewer environment interactions to achieve comparable performance levels. **(2) Superior Asymptotic Performance:** The final performance levels achieved by D²PPO consistently exceed baseline methods, with improvements being most dramatic in challenging manipulation scenarios. **(3) Training Stability:** D²PPO exhibits more stable learning dynamics with reduced variance, indicating that the enhanced representation foundation improves optimization robustness.

D²PPO consistently outperforms both DPPO and Gaussian-based algorithms across all four tasks. Notably, in complex manipulation scenarios, D²PPO achieves substantial improvements: Square task performance increases from 47% (Gaussian) $\rightarrow$ 83% (DPPO) $\rightarrow$ 89% (Ours), while Transport task shows progression from 0% (Gaussian) $\rightarrow$ 60% (DPPO) $\rightarrow$ 87% (Ours), demonstrating the critical importance of diffusion-based representations with dispersive regularization for complex manipulation.

Table 2 reveals several critical insights into the effectiveness of our two-stage D²PPO approach. Most significantly, D²PPO achieves SOTA performance across all tasks, ranking first overall with an average success rate of 0.94, representing an average improvement of 26.1% across the four tasks. These fine-tuning results provide definitive validation that our two-stage training paradigm successfully translates representation enhancements into practical performance improvements. Dispersive pre-training provides enhanced representations that improve policy gradient optimization, enabling more efficient learning and better performance across robotic manipulation tasks.

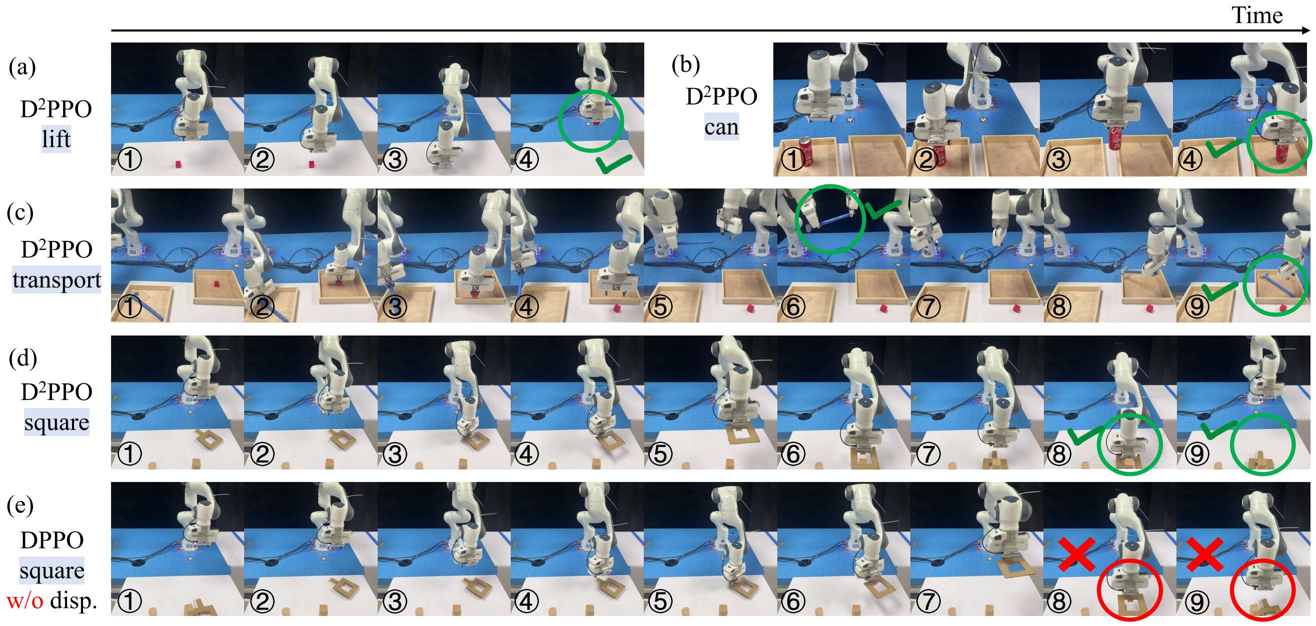

Figure 6: Real-world deployment results on RoboMimic tasks using D^{2}PPO. Each row shows the complete task execution trajectory over time. (a-d) D^{2}PPO with dispersive loss successfully completes all tasks: (a) Lift task with successful block grasping, (b) Can task with successful cylindrical object manipulation, (c) Transport task with successful object delivery to target area, and (d) Square task with precise peg-in-hole insertion (green checkmarks indicate successful completion with green circles highlighting key successful actions). (e) Baseline DPPO without dispersive loss fails in the Square task during final placement phases (red crosses and circles indicate failure points), demonstrating the effectiveness of dispersive representation learning for precise manipulation.

Real Robot Experiments

As the final validation of our approach, we demonstrate the practical deployment capability of D^{2}PPO through real robot experiments on the Franka Emika Panda robot. These experiments are conducted after completing the two-stage training process (dispersive pre-training followed by PPO fine-tuning) and showcase comprehensive evaluation across multiple RoboMimic tasks with direct comparison to baseline methods. On the most challenging transport task, we observe performance progression: 0% (Gaussian) → 45% (DPPO) → 70% (Ours). Quantitative experimental results are detailed in Appendix D.

Figure 6 presents qualitative results from real robot experiments. These results reveal several critical insights: **(1) Task Completion Success:** D^{2}PPO with dispersive loss successfully completes all four benchmark tasks, with each trajectory showing complete execution sequences from initial approach to final success confirmation. **(2) Precision in Complex Tasks:** The Square task, requiring precise peg-in-hole insertion, demonstrates D^{2}PPO's ability to handle fine-grained manipulation where spatial accuracy is critical. **(3) Baseline Comparison:** DPPO without dispersive loss fails to complete the Square task, with clear failure points visible in the final placement phases, highlighting the critical importance of dispersive representation learning for precise manipulation scenarios.

Conclusion

This paper addresses representation collapse in diffusion-based policy learning, where similar observations lead to indistinguishable features, causing failures in complex manipulation tasks. We introduce D^{2}PPO, a two-stage framework applying dispersive loss regularization during pre-training, followed by policy gradient optimization for fine-tuning.

Our contributions include: (1) systematic analysis identifying representation collapse as the underlying cause of poor performance in complex tasks; (2) D^{2}PPO method that combats representation collapse through dispersive regularization; (3) comprehensive validation demonstrating state-of-the-art performance with 94% average success rate, achieving 22.7% improvement in pre-training and 26.1% improvement after fine-tuning compared to baseline methods.

Experimental results reveal that representation quality becomes increasingly critical for complex tasks, with D^{2}PPO showing universal effectiveness across varying task complexities. D^{2}PPO requires no additional parameters and provides plug-and-play integration with existing diffusion policies. Real robot validation confirms practical deployment effectiveness, establishing dispersive loss as a powerful tool for enhancing diffusion-based robotic policies and opening new directions for representation-regularized policy learning. While our current evaluation focuses on manipulation tasks, future work could explore dispersive regularization in other robotic domains.

References

Arora, S.; Khandeparkar, H.; Khodak, M.; Plevrakis, O.; and Saunshi, N. 2019. A Theoretical Analysis of Contrastive Unsupervised Representation Learning. *arXiv preprint arXiv:1902.09229*.

Chen, J.; Tan, Z.; Zhou, G.; Pan, L.; and Zhou, K. 2025a. SARA: Structural and Adversarial Representation Alignment for Training-Efficient Diffusion Models. *arXiv preprint arXiv:2502.19669*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Chen, Y.; Jha, D. K.; Tomizuka, M.; and Romeres, D. 2025b. FDPP: Fine-tune Diffusion Policy with Human Preference. *arXiv preprint arXiv:2501.08259*.

Chi, C.; Xu, Z.; Feng, S.; Cousineau, E.; Du, Y.; Burchfiel, B.; Tedrake, R.; and Song, S. 2023. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 02783649241273668.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929*.

Florence, P.; Lynch, C.; Zeng, A.; Ramirez, O. A.; Wahid, A.; Downs, L.; Wong, A.; Lee, J.; Mordatch, I.; and Tompson, J. 2022. Implicit Behavioral Cloning. In *Conference on robot learning*, 158–168. PMLR.

Frans, K.; Hafner, D.; Levine, S.; and Abbeel, P. 2024. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*.

Geng, Z.; Yang, B.; Hang, T.; Li, C.; Gu, S.; Zhang, T.; Bao, J.; et al. 2024. Instructdiffusion: A generalist modeling interface for vision tasks. 12709–12720.

Hafner, D.; Tessler, C.; Micheli, V.; van der Pol, E.; and Strub, F. 2023. Generative Flow Networks as Entropy-Regularized RL. *arXiv preprint arXiv:2310.12934*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Karras, T.; Aittala, M.; Aila, T.; and Laine, S. 2022. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35: 26565–26577.

Karras, T.; Aittala, M.; Aila, T.; and Laine, S. 2023. Elucidating the Design Space of Diffusion-Based Generative Models. *Advances in Neural Information Processing Systems*.

Ke, T.-W.; Gkanatsios, N.; and Fragkiadaki, K. 2024. 3D Diffuser Actor: Policy Diffusion with 3D Scene Representations. *arXiv preprint arXiv:2402.10885*.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems*, volume 33, 18661–18673.

Lee, J.-Y.; Cha, B.; Kim, J.; and Ye, J. C. 2025. Aligning text to image in diffusion models is easier than you think. *arXiv preprint arXiv:2503.08250*.

Li, D.; Ren, J.; Wang, Y.; Wen, X.; Li, P.; Xu, L.; Zhan, K.; et al. 2025. Finetuning generative trajectory model with reinforcement learning from human feedback. *arXiv preprint arXiv:2503.10434*.

Li, X. L.; Thickstun, J.; Gulrajani, I.; Liang, P.; and Hashimoto, T. B. 2023. Diffusion-LM Improves Controllable Text Generation. *Advances in Neural Information Processing Systems*.

Lipman, Y.; Chen, R. T.; Ben-Hamu, H.; Nickel, M.; and Le, M. 2022. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.

Liu, X.; Gong, C.; and Liu, Q. 2022. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*.

Mandlekar, A.; Xu, D.; Wong, J.; Nasiriany, S.; Wang, C.; Kulkarni, R.; Fei-Fei, L.; Savarese, S.; Zhu, Y.; and Martín-Martín, R. 2022. What Matters in Learning from Offline Human Demonstrations for Robot Manipulation. In *Conference on Robot Learning*, 1678–1690. PMLR.

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; Assran, M.; Ballas, N.; Galuba, W.; Howes, R.; Huang, P.-Y.; Li, S.-W.; Misra, I.; Rabbat, M.; Sharma, V.; Synnaeve, G.; Xu, H.; Jégou, H.; Mairal, J.; Labatut, P.; Joulin, A.; and Bojanowski, P. 2023. DINOv2: Learning Robust Visual Features without Supervision. *arXiv preprint arXiv:2304.07193*.

Peebles, W.; and Xie, S. 2023. Scalable Diffusion Models with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4195–4205.

Ren, A. Z.; Lidard, J.; Ankile, L. L.; Simeonov, A.; Agrawal, P.; Majumdar, A.; Burchfiel, B.; Dai, H.; and Simchowitz, M. 2024. Diffusion Policy Policy Optimization. In *CoRL 2024 Workshop on Mastering Robot Manipulation in a World of Abundant Data*.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015a. Trust Region Policy Optimization. In *International Conference on Machine Learning*, 1889–1897.

Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; and Abbeel, P. 2015b. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*.

Shafiullah, N. M.; Cui, Z. J.; Altanzaya, A.; and Pinto, L. 2022. Behavior Transformers: Cloning k modes with one stone. In *Advances in neural information processing systems*, volume 35, 22955–22968.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y.; Dhariwal, P.; Chen, M.; and Sutskever, I. 2023. Consistency models.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. MuJoCo: A Physics Engine for Model-Predictive Control. 5026–5033.

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

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, volume 30.

Wang, R.; and He, K. 2025. Diffuse and Disperse: Image Generation with Representation Regularization. *arXiv preprint arXiv:2506.09027*.

Wang, T.; and Isola, P. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. *arXiv preprint arXiv:2005.10242*.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3): 229–256.

Yu, S.; Kwak, S.; Lee, J.; and Shin, J. 2024. Representation Alignment for Generation: Training Diffusion Transformers Is Easier Than You Think. *arXiv preprint arXiv:2410.06940*.

Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; and Deny, S. 2021. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. In *International Conference on Machine Learning*, 12310–12320.

Ze, Y.; Chen, Z.; Wang, W.; Chen, T.; He, X.; Yuan, Y.; Peng, X. B.; and Wu, J. 2024a. Generalizable Humanoid Manipulation with 3D Diffusion Policies. *arXiv preprint arXiv:2410.10803*.

Ze, Y.; Zhang, G.; Zhang, K.; Hu, C.; Wang, M.; and Xu, H. 2024b. 3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations. In *Robotics: Science and Systems (RSS)*.

Zhang, T.; Yu, C.; Su, S.; and Wang, Y. 2025. ReinFlow: Fine-tuning Flow Matching Policy with Online Reinforcement Learning. *arXiv preprint arXiv:2505.22094*.

Zhou, Y.; Lin, Y.; Peng, F.; Chen, J.; Zhou, Z.; Huang, K.; Yang, H.; and Yin, Z. 2025. MTIL: Encoding Full History with Mamba for Temporal Imitation Learning. *arXiv preprint arXiv:2505.12410*.

Reproducibility Checklist

Instructions for Authors:

This document outlines key aspects for assessing reproducibility. Please provide your input by editing this `.tex` file directly.

For each question (that applies), replace the "yes" text with your answer.

Example: If a question appears as

```
\question{Proofs of all novel claims
are included} {(yes/partial/no)}
yes
```

you would change it to:

```
\question{Proofs of all novel claims
are included} {(yes/partial/no)}
yes
```

Please make sure to:

- Replace ONLY the "yes" text and nothing else.
- Use one of the options listed for that question (e.g., **yes**, **no**, **partial**, or **NA**).
- **Not** modify any other part of the `\question` command or any other lines in this document.

You can `\input` this .tex file right before `\end{document}` of your main file or compile it as a stand-alone document. Check the instructions on your conference's website to see if you will be asked to provide this checklist with your paper or separately.

1. General Paper Structure

1.1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/partial/no/NA) yes

1.2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes/no) yes

1.3. Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes/no) yes

2. Theoretical Contributions

2.1. Does this paper make theoretical contributions? (yes/no) yes

If yes, please address the following points:

2.2. All assumptions and restrictions are stated clearly and formally (yes/partial/no) yes

2.3. All novel claims are stated formally (e.g., in theorem statements) (yes/partial/no) yes

2.4. Proofs of all novel claims are included (yes/partial/no) yes

2.5. Proof sketches or intuitions are given for complex and/or novel results (yes/partial/no) yes

2.6. Appropriate citations to theoretical tools used are given (yes/partial/no) yes

2.7. All theoretical claims are demonstrated empirically to hold (yes/partial/no/NA) yes

2.8. All experimental code used to eliminate or disprove claims is included (yes/no/NA) yes

3. Dataset Usage

3.1. Does this paper rely on one or more datasets? (yes/no) yes

If yes, please address the following points:

3.2. A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/NA) yes

3.3. All novel datasets introduced in this paper are included in a data appendix (yes/partial/no/NA) yes

3.4. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no/NA) yes

3.5. All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations (yes/no/NA) yes

3.6. All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available (yes/partial/no/NA) yes

3.7. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing (yes/partial/no/NA) yes

4. Computational Experiments

4.1. Does this paper include computational experiments? (yes/no) yes

If yes, please address the following points:

4.2. This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting (yes/partial/no/NA) yes

4.3. Any code required for pre-processing data is included in the appendix (yes/partial/no) yes

4.4. All source code required for conducting and analyzing the experiments is included in a code appendix (yes/partial/no) yes

4.5. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no) yes

4.6. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/partial/no) yes

4.7. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results (yes/partial/no/NA) yes

4.8. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks (yes/partial/no) yes

4.9. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics (yes/partial/no) yes

4.10. This paper states the number of algorithm runs used to compute each reported result (yes/no) yes

4.11. Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information (yes/no) yes

4.12. The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank) (yes/partial/no) yes

4.13. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments (yes/partial/no/NA) yes