2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

MoE-DP: An MoE-Enhanced Diffusion Policy for Robust Long-Horizon Robotic Manipulation with Skill Decomposition and Failure Recovery

Baiye Cheng*[1,4], Tianhai Liang*[1], Suning Huang[2], Maanping Shao[1],
Feihong Zhang[1], Botian Xu[1], Zhengrong Xue[1,3], Huazhe Xu†[1,3]

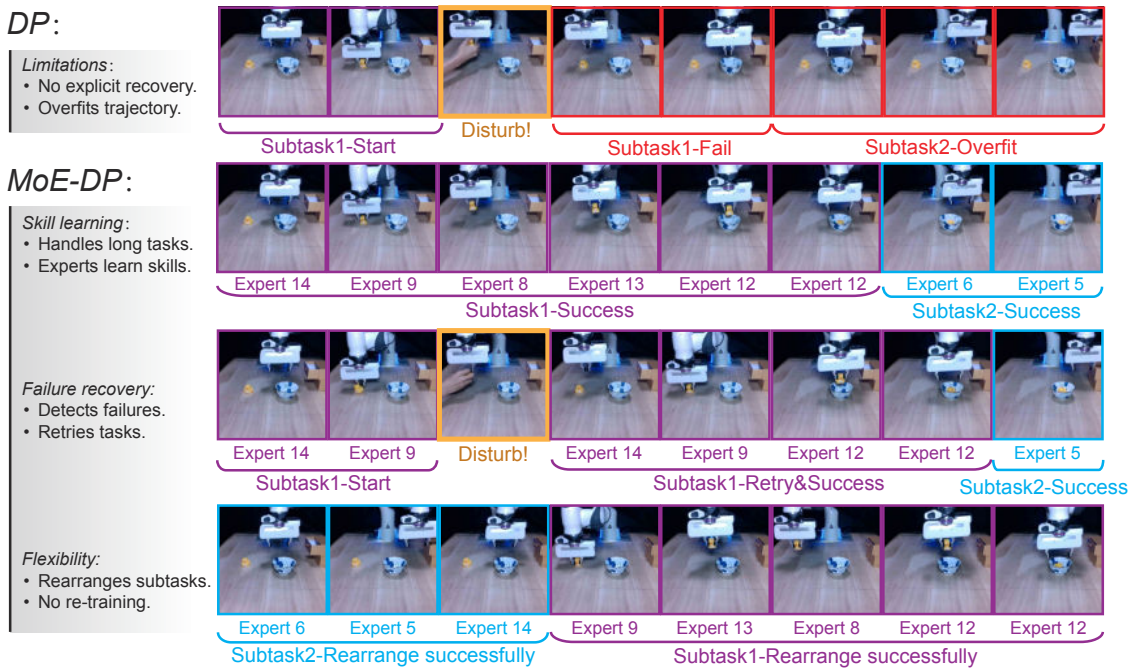

Fig. 1: **MoE-DP enables robust recovery, interpretable skill decomposition, and high-level control for long-horizon manipulation.** The baseline DP fails under disturbances such as object displacement. It lacks stage awareness, cannot recover from errors, overfits to subsequent trajectories, and cascades into further failures. MoE-DP learns an interpretable skill decomposition, with experts specializing in different skills, such as approaching, grasping, and placing. MoE-DP can detect failures and reactivate the correct expert to retry failed subtasks. Task order can be flexibly rearranged by controlling the sequence of expert activations without re-training, such as executing subtask 2 before subtask 1. Colored overlays indicate expert activations and the stage of subtasks.

***Abstract*— Diffusion policies have emerged as a powerful framework for robotic visuomotor control, yet they often lack the robustness to recover from subtask failures in long-horizon, multi-stage tasks and their learned representations of observations are often difficult to interpret. In this work, we propose the Mixture of Experts-Enhanced Diffusion Policy (MoE-DP), where the core idea is to insert a Mixture of Experts (MoE) layer between the visual encoder and the diffusion model. This layer decomposes the policy's knowledge into a set of specialized experts, which are dynamically activated to handle different phases of a task. We demonstrate through extensive experiments that MoE-DP exhibits a strong capability to recover from disturbances, significantly outperforming standard baselines in robustness. On a suite of 6 long-horizon simulation tasks, this leads to a 36% average relative improvement in success rate under disturbed conditions. This enhanced robustness is further validated in the real world, where MoE-DP also shows significant performance gains. We further show that MoE-DP learns an interpretable skill decomposition, where distinct experts correspond to semantic task primitives (e.g., approaching, grasping). This learned structure can be leveraged for inference-time control, allowing for the rearrangement of subtasks without any re-training. Our video and code are available at the https://moe-dp-website.github.io/MoE-DP-Website/.**

* Equal Contribution. † Corresponding Author.
[1] Tsinghua University. [2] Stanford University. [3] Shanghai Qi Zhi Institute.
[4] Huazhong University of Science and Technology.

I. Introduction

Learning visuomotor policies for robotic manipulation has become a prevailing paradigm, with recent approaches increasingly leveraging powerful generative models, such as diffusion frameworks [1]–[4], to map high-dimensional observations to control actions. While these methods have demonstrated significant success in short-term tasks, they reveal a critical limitation in long-term, multi-stage scenarios: a lack of stage-awareness. When an intermediate subtask fails—for instance, an unsuccessful grasp—the policy often

proceeds with the subsequent action sequence as if the failure never occurred, leading to cascading task failure. This brittleness stems from the underlying structure of the policy's learned representation, which is typically a high-dimensional and highly entangled 'black box.' This opacity not only prevents the policy from recovering from local errors but also fundamentally hinders our ability to interpret how observational features map to actions, making it difficult to analyze, debug, or extend the learned behavior.

A promising architectural paradigm to address these challenges is the Mixture of Experts (MoE) framework [5]–[7], which has seen widespread success in scaling large language models [8]–[12] and is now gaining traction within robotics [13]–[17]. The core principle of MoE is to decompose a complex problem into a set of simpler ones by routing inputs to specialized sub-networks, or 'experts.' This principle of modular decomposition offers a direct remedy to the limitations of conventional policies. By leveraging MoE, a single, entangled control policy can be broken down into a collection of distinct, interpretable skills, providing a clear path toward enhancing both robustness and interpretability.

In this work, we introduce the Mixture of Experts-Enhanced Diffusion Policy (MoE-DP), which integrates an MoE layer between the visual encoder and the diffusion model. MoE-DP decomposes long-horizon tasks into a set of specialized skills in an end-to-end manner. This approach enhances robustness when facing disturbance and improves interpretability by creating a clear mapping between experts and skills. Our experiments confirm that MoE-DP not only improves success rates under disturbance but also enables flexible inference-time control over learned skills.

Our contributions are threefold:

- We propose MoE-DP, a novel method that demonstrates strong robustness to environmental disturbances. Evaluated across a suite of 6 long-horizon simulation tasks, MoE-DP achieves a 36% average relative improvement in success rate under disturbed conditions where baseline methods typically fail.
- Through visualization, we demonstrate that MoE-DP achieves a meaningful and interpretable skill decomposition, where distinct experts are consistently activated for different semantic phases of a task (e.g., approaching, grasping, placing).
- We demonstrate that this learned decomposition enables inference-time control over the policy's behavior, allowing us to rearranged subtasks and generalize to new task structures without any re-training.

II. Related Work

A. Visual Imitation Learning

Numerous policy learning algorithms have been proposed for robotic manipulation [1], [13], [18]–[21]. Generative approaches like Diffusion Policies [1], [2], [22], [23] have proven particularly effective at modeling the complex, multi-modal action distributions required for these tasks. However, their learned representations are highly entangled and lack stage-awareness, which makes the policies prone to cascading errors after intermediate failures and obscures the mapping from observation to action. In this work, we directly addresses this gap by introducing an MoE structure, which explicitly disentangles the latent space into specialized modules, thereby enhancing both the policy's robustness and the interpretability of its learned skills.

B. Skill Decomposition with Mixture of Experts

The MoE architecture has proven highly effective in large language models [8]–[12], [24] and multi-task robotics [13]–[17]. Building on these successes, our work adapts this paradigm not for inter-task assignment, but for fine-grained *skill decomposition* within a single long-horizon task. Unlike traditional two-stage methods that discover skills separately before policy integration [25]–[27], we integrate an MoE layer directly into the policy for end-to-end learning. This yields a set of interpretable, task-optimized skills that can be flexibly controlled at inference time to alter the execution order of subtasks.

C. Inference-Time Control of Policy Behavior

Modifying a policy's behavior at inference time without re-training is a key challenge in robotics [28]–[33]. Common strategies rely on external modules, such as safety value functions [34], [35] or gradient-based guidance from dynamics models [36], which typically require separate training. In contrast, our work provides an **intrinsic control mechanism** by directly leveraging the policy's learned MoE structure. The resulting skill decomposition offers interpretable "handles" for high-level control, as each expert corresponds to a clear semantic skill, enabling dynamic control over the execution flow without requiring auxiliary models.

III. Method

To learn a more structured and human-interpretable representation, we introduce an MoE layer to dynamically process the latent features obtained from the observation encoder. This strategic integration transforms the policy's conditioning from a static feature vector into a dynamic, structured representation. This learned decomposition is the key to enabling two critical capabilities: robust recovery from intermediate task failures and flexible, high-level control over the policy's behavior at inference time.

A. Preliminaries

1) Visuomotor Diffusion Policy: A diffusion policy models the visuomotor mapping $p(A_t|O_t)$ as a conditional denoising process [37]–[39]. An encoder first processes a history of observations into a latent vector $\mathbf{z}_t = \text{Encoder}(O_{t-T_o+1:t})$. The policy then generates an action sequence by iteratively refining Gaussian noise over multiple steps using a learned denoising network $\epsilon_\theta(A_t^k, \mathbf{z}_t, k)$. The training objective is a mean-squared error loss to predict the noise added to ground-truth actions:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{A_{t:t+T_a},\epsilon,k}\left[\left\|\epsilon - \epsilon_\theta(A_t^0 + \sigma_k\epsilon \mid \mathbf{z}_t, k)\right\|^2\right] \quad (1)$$

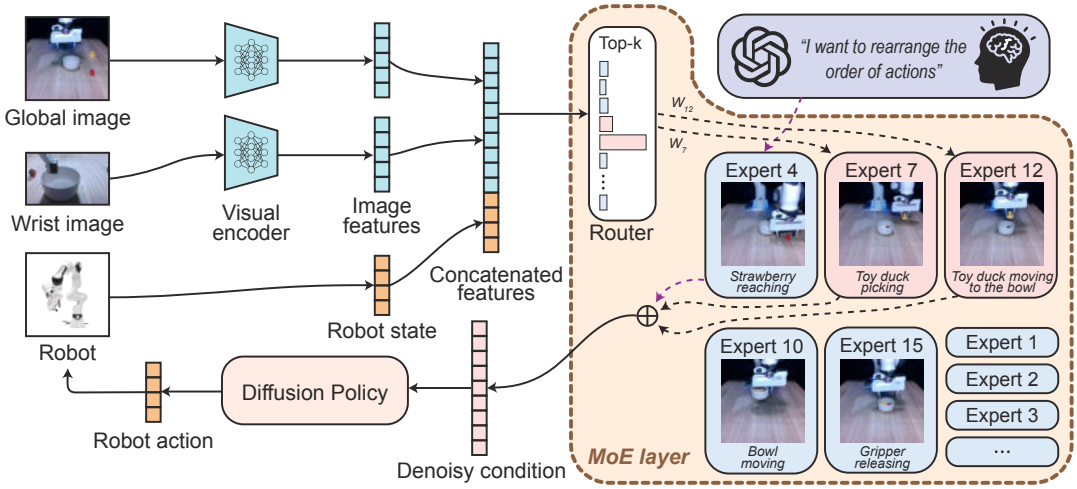

Fig. 2: **Overview of MoE-DP with high-level guidance.** In its **autonomous mode**, the system encodes observation inputs (images and robot state) into a feature vector, which is then fed to an MoE layer. The MoE's router automatically selects the appropriate expert for the current observation. The output of the selected expert then serves as a conditioning input for the Diffusion Policy during action generation. While the router typically operates autonomously, the architecture supports **high-level control**: an external agent, such as a human operator or a Vision-Language Model (VLM), can guide the policy by **overriding the router's default selection**. This capability enables flexible behaviors, such as reordering subtasks to generalize to novel sequences not seen during training.

2) Mixture of Experts: The Mixture of Experts (MoE) architecture enables conditional computation through a set of N "expert" networks (typically feed-forward networks) and a "router". For an input $\mathbf{x}$, the router computes a weighting distribution, $g(\mathbf{x})$, which is typically generated by applying a softmax function to the output of a linear layer. The final output is a weighted combination of the outputs from the activated experts (often only the Top-k):

$$\text{MoE}(\mathbf{x}) = \sum_{i \in \text{Top-k}(\mathbf{x})} g(\mathbf{x})_i \cdot E_i(\mathbf{x}) \tag{2}$$

This allows for a significant increase in model capacity without a proportional rise in computational cost, as individual experts learn specialized functions.

B. MoE-Enhanced Diffusion Policy (MoE-DP)

Our core contribution is the MoE-DP, which integrates an MoE [5], [6] layer into the Diffusion Policy [1] framework. The implementation is built upon the original Diffusion Policy architecture; it takes as input a global camera view, a wrist-mounted camera view, and the robot's proprioceptive state, processed by a standard ResNet18 [40] visual encoder that is trained from scratch. By interposing the MoE layer between this visual encoder and the diffusion model, MoE-DP encourages the policy to learn a decomposed skill representation. This architecture achieves two primary goals: enhancing the policy's robustness in multi-stage tasks by making it stage-aware, and improving the interpretability of its internal representations by encouraging specialization.

1) MoE Architecture: Our primary architectural modification is to insert an MoE layer to process the latent feature vector $\mathbf{z}_t$ from the visual encoder. Instead of directly conditioning the diffusion model on latent feature, the MoE layer dynamically routes $\mathbf{z}_t$ through a set of specialized expert networks to produce a refined conditioning feature. Each expert, $\{E_i\}_{i=1}^{N}$, is a multi-layer perceptron (MLP) designed to specialize in a specific phase of the manipulation task. A lightweight gating network, the **router**, computes routing weights for these N experts:

$$g_t = \text{Softmax}(W_g \mathbf{z}_t) \tag{3}$$

where W_g is a learned weight matrix. Instead of a dense combination, we employ a sparse, Top-k routing strategy. The final conditioning vector $\mathbf{z}'_t$ is a weighted combination of the outputs of only the top-k selected experts:

$$\mathbf{z}'_t = \sum_{i \in \text{Top-k}(g_t)} g_{t,i} \cdot E_i(\mathbf{z}_t) \tag{4}$$

This MoE-enhanced feature $\mathbf{z}'_t$ then serves as the condition for the diffusion model ϵ_θ. This design encourages different experts to specialize in distinct phases of a task (skill decomposition). The router's gating mechanism can thus re-activate an appropriate expert to retry a failed subtask, promoting stage-aware robustness.

2) Training Objective: A key challenge when training MoE models is to prevent *router collapse* [9], [41], where the gating network learns to always select only a few dominant experts, leaving others untrained. To ensure all experts are utilized and learn meaningful specializations, we introduce an auxiliary loss, $\mathcal{L}_{\text{aux}}$, which is critical for achieving balanced and specialized expert usage.

This auxiliary loss consists of two terms. The first is a **load-balancing loss** [9], [41], which encourages a more uniform load distribution across experts, preventing the router

from consistently selecting only a few experts while leaving others untrained. This loss is computed as the scaled dot-product between the fraction of tokens dispatched to each expert and the fraction of router probability for them:

$$\mathcal{L}_{\text{load}} = N \sum_{i=1}^{N} f_i \cdot P_i \tag{5}$$

where N is the total number of experts. Here, f_i is the fraction of samples in the batch $\mathcal{B}$ dispatched to expert i:

$$f_i = \frac{1}{B} \sum_{t \in \mathcal{B}} \mathbf{1}\{\text{argmax}_j(p_{t,j}) = i\} \tag{6}$$

and P_i is the average router probability for expert i across the batch:

$$P_i = \frac{1}{B} \sum_{t \in \mathcal{B}} p_{t,i} \tag{7}$$

where B is the batch size and $p_{t,i}$ is the router's softmax output probability for sample t to expert i.

The second term is an **entropy loss**, which promotes a more structured division of labor among the experts and enhances their specialization. By encouraging the router to make confident, low-entropy (i.e., high-probability) assignments for each individual sample, it sharpens the role of each expert. This specialization is crucial for downstream goals: it improves interpretability by creating a clearer mapping between experts and skills, and it enables **high-level behavioral control**. The loss is the mean entropy of the router's output distribution over the batch:

$$\mathcal{L}_{\text{entropy}} = -\frac{1}{B} \sum_{t=1}^{B} \sum_{i=1}^{N} p_{t,i} \log(p_{t,i} + \epsilon) \tag{8}$$

where ϵ is a small constant for numerical stability.

The training of MoE-DP model is guided by a composite objective function. This function combines the standard diffusion loss with auxiliary terms that encourage balanced and specialized expert utilization. The complete training objective is a weighted sum of these components:

$$\mathcal{L} = \mathcal{L}_{\text{diff}} + \underbrace{\lambda \mathcal{L}_{\text{load}} + \beta \mathcal{L}_{\text{entropy}}}_{\text{Auxiliary Loss}} \tag{9}$$

where $\mathcal{L}_{\text{diff}}$ is the standard diffusion denoising loss. The **auxiliary loss** term consists of the load-balancing loss ($\mathcal{L}_{\text{load}}$) and the entropy loss ($\mathcal{L}_{\text{entropy}}$), weighted by their respective coefficients λ and β. This combined objective guides the model to learn a set of specialized experts for different subtasks while accurately modeling the action distribution.

C. Skill Decomposition and Inference-Time Control of Behavior

Leveraging the auxiliary loss, MoE-DP model learns a disentangled and interpretable representation of skills, enabling effective decomposition of long-horizon, multi-stage tasks. This is achieved through the synergy of its two components. The load-balancing loss encourages a more uniform load distribution across experts, ensuring a wider range of experts are utilized during training. Meanwhile, the entropy loss incentivizes the router to produce low-entropy, high-confidence probability distributions. This ensures that for any given observation, the router's output is sharply peaked, meaning one expert is typically selected with a probability approaching 1, while all others receive near-zero probability.

Tab. 1: **Simulation task results.** We compare the performance of MoE-DP against the baseline across all simulation tasks under both nominal (✓) and disturbed (✗) execution conditions. The Disturbance (Object) column indicates the action of resetting the object to its initial position after it has been successfully grasped. Bolded numbers indicate the highest success rate for each condition.

Task	Disturbance (Object)	Method	
		DP	**MoE-DP**
Hammer Cleanup T0	✓	**86.7**	84.0
	✗(Hammer)	17.3	**33.3**
Kitchen T0	✓	**93.3**	80.7
	✗(Cube)	32.0	**48.7**
	✗(Pot)	18.7	**62.7**
Coffee Preparation T0	✓	**52.7**	51.3
	✗(Mug)	10.0	**16.0**
Mug Cleanup T0	✓	59.3	**62.0**
	✗(Mug)	32.7	**35.3**
Kitchen Cleanup T0	✓	52.7	**96.7**
	✗(Hammer)	12.0	**82.7**
Table Cleanup T0	✓	**100.0**	**100.0**
	✗(Cube)	2.0	**98.0**
Average	✓	74.1	**79.1**
	✗	17.8	**53.8**

And this emergent property is the foundation for two powerful capabilities of MoE-DP. First, it enables a clear and interpretable **skill decomposition**. Because a single, dominant expert is active during each phase of a manipulation task, a direct correspondence emerges between specific experts and distinct, semantic skills (e.g., approaching, grasping, placing). This transparency allows us to analyze and understand the policy's decision-making process in a way that is impossible with conventional architectures.

Second, this explicit expert-skill mapping enables direct **inference-time control of behavior**. Since the choice of expert dictates the subsequent action sequence, we can externally manipulate the router's output to command specific behaviors. This opens up possibilities for hierarchical control, where a high-level agent—such as a human operator or a VLM—can direct the policy's execution flow. A compelling application of this is generalizing beyond the training data's fixed task sequences. For example, consider a policy trained exclusively on demonstrations that execute subtask A before subtask B. While a standard policy would be confined to this order, MoE-DP allows a VLM, given the instruction "Do B then A," to simply alter the activation order of the corresponding experts. This allows the policy to generalize to novel task structures not seen in the training data. And

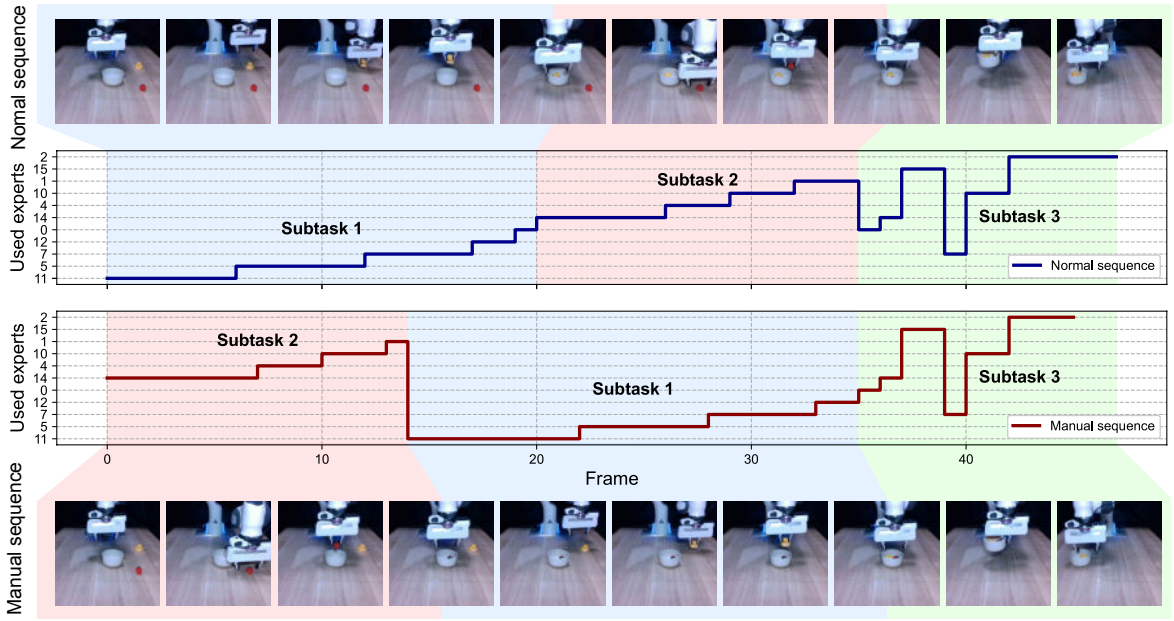

Fig. 3: **Inference-time control via compositional skill decomposition.** We demonstrate that MoE-DP learns modular and reusable skills that can be flexibly recombined to form novel behaviors without re-training. **(Top)** When executing a task in its demonstrated order, the policy decomposes the process into three distinct subtasks—Subtask 1 (picking the yellow duck), Subtask 2 (picking the strawberry), and Subtask 3 (moving the bowl)—each invoking a consistent sequence of expert activations. **(Bottom)** At inference time, we manually command a novel sequence by altering the subtask order (Subtask 2 followed by Subtask 1). The policy successfully executes this new task by reordering the learned skill modules. Crucially, the expert activation pattern for an individual subtask (e.g., Subtask 1) remains consistent across both scenarios, proving that MoE-DP learns truly compositional skills that enable generalization to new task structures.

we will talk this capability in our experiments part IV-C

IV. Experiments

We evaluate MoE-DP on a series of long-horizon manipulation tasks in both simulation and the real world. We use Diffusion Policy (DP) as our baseline. To ensure a fair comparison, the network capacity of MoE-DP does not differ significantly from the baseline. The experiments are designed to investigate three central questions:

- Does MoE-DP outperform a standard diffusion policy baseline in long-horizon, multi-stage tasks, especially under disturbed conditions? (Sec. IV-A, IV-B)
- Does the MoE-DP learn a meaningful skill decomposition, where distinct experts consistently activate for different subtasks or behaviors? (Sec. IV-C)
- Can the learned skill decomposition be leveraged to flexibly control the policy's behavior at inference time, for instance, to rearranged subtasks? (Sec. IV-D)
- How does the auxiliary loss contribute to the model's performance and skill decomposition? (Sec. IV-E)

A. Simulation Experiments

Experimental Setup. Inspired by the MimicGen [42] benchmark, we constructed a suite of 6 long-horizon tasks in a simulation environment. These tasks involve multi-stage, multi-object manipulation and are specifically designed to challenge a policy's stage-awareness and its ability to recover from subtask failures. We evaluate all policies under two conditions: (1) *nominal execution*, where tasks proceed without disturbance, and (2) *disturbed execution*, a condition designed to explicitly test the policy's robustness and recovery capabilities. In this setup, we programmatically induce a failure within a specific subtask. For instance, after the robot arm successfully grasps an object but before it can place it at the target location, we reset the object back to its initial position. This forces the policy to recognize the state discrepancy and re-attempt the subtask. For each simulation task, we collect 100 expert demonstrations. To accelerate training, we use a batch size of 128, while all other hyperparameters and training settings are kept consistent with the original Diffusion Policy implementation. We run 3 seeds for each experiment. For each seed, we train policies for 500 epochs, evaluating with 50 rollouts every 10 epochs. We report the maximum success rate achieved during training, averaged across the three seeds.

Results. As shown in Tab. 1, under *nominal execution* conditions, the performance of MoE-DP is comparable to the baseline. However, the advantage of MoE-DP becomes evident in the *disturbed execution* scenario, where MoE-DP's success rate is 36% higher than the baseline's. This result demonstrates MoE-DP's robustness in the face of disturbances. During the tuning process, we observed that due to

Tab. 2: **Real-world task results.** We compare the performance of MoE-DP against the baseline across all real-world tasks under both nominal (✓) and disturbed (✗) execution conditions. The Disturbance (Object) column indicates the action of resetting the object to its initial position after it has been successfully grasped. Bolded numbers indicate the highest success rate for each condition.

Task Name	Disturbance (Object)	Method	
		DP	**MoE_DP**
Pick two cube	✓	85.0	**90.0**
	✗(Green Cube)	5.0	**60.0**
	✗(Red Cube)	75.0	**90.0**
Duck place drawer close	✓	**90.0**	**90.0**
	✗(Cube)	5.0	**70.0**
Duck place bowl transport	✓	95.0	**100.0**
	✗(Duck)	20.0	**55.0**
	✗(Strawberry)	35.0	**80.0**
Average	✓	90.0	**93.3**
	✗	28.0	**71.0**

Tab. 3: **MoE hyperparameter settings.** Due to the unique characteristics of each task and its training data, optimal hyperparameter settings were found empirically. 'expert_num' denotes the total number of experts in the MoE architecture; 'top_k' specifies the number of experts activated by the router at each forward pass; λ is the weight for the load-balancing loss, and β is the weight for the entropy loss.

Task Name	#experts	top-k	λ	β
Simulation				
Hammer Cleanup T0	8	2	0.1	0.01
Kitchen T0	16	2	0.1	0.007
Coffee Preparation T0	16	4	0.1	0.01
Mug Cleanup T0	16	4	0.1	0.01
Kitchen Cleanup T0	16	2	0.1	0.01
Table Cleanup T0	16	2	0.1	0.01
Real-world				
Pick two cube	16	2	0.1	0.03
Duck place drawer close	16	2	0.1	0.04
Duck place bowl transport	16	2	0.1	0.03

the distinct characteristics of the data for each task, different tasks require specific MoE hyperparameter configurations to achieve optimal performance, as detailed in Tab. 3.

B. Real-World Experiments

Experimental Setup. We deploy MoE-DP on a Franka Emika robot, using two Intel RealSense D435i cameras (one stationary for a global view and one wrist-mounted) for visual feedback. We designed three long-horizon manipulation tasks to evaluate performance on physical hardware:

- **Duck place drawer close:** Place a rubber duck into a bowl and subsequently close a nearby drawer.
- **Pick two cube:** Pick and place two colored blocks into a designated container in a predefined order.
- **Duck place bowl transport:** Place both a duck and a strawberry into a bowl, and then transport the bowl to a target location.

These tasks require the policy to chain together fine-grained manipulation skills and reason about the temporal progression of the task.

Results. We evaluate the trained models over 20 test trials for each task. As shown in Tab. 2, the real-world results directly mirrored the simulation findings. While both policies performed comparably under nominal conditions, MoE-DP again demonstrated significantly higher robustness when subjected to manual disturbance, reliably recovering from induced failures where the baseline could not. This improved real-world reliability is a direct result of the structured behavior promoted by the MoE architecture, leading to more

Tab. 4: **Generalization to novel task sequences via inference-time control.** We quantitatively evaluate the policy's ability to execute subtasks in novel orders not seen during training. The execution flow was guided at inference time by either a human operator (`Human_Control`) or a VLM (`VLM_Control`). Success rates are reported over 10 trials for each task, demonstrating the flexibility of the learned compositional skills.

Task Name	Human_Control	VLM_Control
Pick two cube	7/10	6/10
Duck place drawer close	9/10	5/10
Duck place bowl transport	9/10	5/10

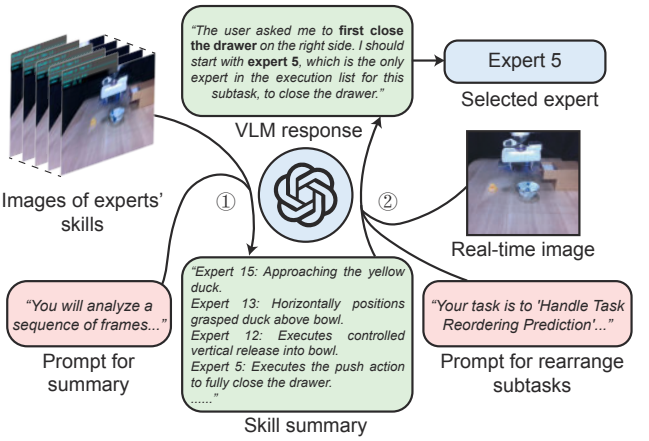

Fig. 4: **Overview of the VLM-based planning and control framework.** Our system leverages a VLM for high-level task planning in two stages. First, at the *skill summarization* (①) stage, the VLM builds a textual knowledge base of the robot's capabilities by analyzing annotated frames from a demonstration that follows the same execution sequence as the training data. Second, at the *task execution* stage (②), the VLM uses this knowledge, a high-level goal, and a real-time image to reason about the current task stage and predict the appropriate expert to activate. This hierarchical architecture enables the system to dynamically plan and rearrange the order of subtasks without any re-training, translating abstract goals into concrete robotic actions.

predictable and interpretable action sequences as different experts specialize in distinct task stages.

C. Analysis of Skill Decomposition

To verify that MoE-DP learns a meaningful skill decomposition, we visualize expert activations during the inference phase of our tasks. As visualized in the top panel of Fig. 3, MoE-DP, guided by the auxiliary loss, successfully decomposes long-horizon, multi-stage tasks. We observe a clear and consistent pattern where different experts are systematically activated for distinct execution stages. For example, during the subtask of placing a duck into a bowl, a clear division of labor emerges: one expert becomes primarily active for 'approaching the duck', a second expert handles the 'grasping' action, and a third takes over for the subsequent phase of 'moving it into the bowl'. This fine-grained specialization confirms that MoE-DP disentangles complex behaviors into interpretable, semantically consistent primitives, showcasing a robust and meaningful skill decomposition.

D. Inference-Time Control of Behavior

Additionally, we investigate whether the learned skill decomposition enables explicit, high-level control over the policy's execution flow. As qualitatively demonstrated in Fig. 3, MoE-DP allows for the flexible recomposition of learned skills. This control can be exerted at inference time by an external agent, such as a human operator or a VLM. We propose a hierarchical framework for VLM-based control, detailed in Fig. 4, where the VLM reasons about the task and directs the policy by selecting the appropriate expert sequence. This allows the robot to execute subtasks in novel orders (e.g., B $\rightarrow$ A $\rightarrow$ C) not seen in the training data, without any re-training. We quantitatively evaluate this capability in Tab. 4, reporting success rates over 10 trials for both human and VLM control. The results confirm the flexibility afforded by the compositional skills, though we note that the VLM's performance is lower than that of direct human control due to its limitations in spatial understanding and precise progress recognition. Ultimately, this demonstrates that MoE-DP not only decomposes tasks into reusable skills but also allows for their flexible recomposition, enabling generalization to new task structures.

E. Ablation Study on Auxiliary Loss

Finally, to isolate the contributions of the two components of the auxiliary loss, we conduct an ablation study with results summarized in Tab. 5. We evaluate four variants of the auxiliary loss: a version with no auxiliary loss (**N**), a version with only the load-balancing loss (**L**), a version with only the entropy loss (**E**), and our proposed model which combines the load-balancing and entropy losses (**LE**). We compare these variants on both task performance and the quality of the learned skill decomposition.

Our findings confirm that the combination of both the load-balancing and entropy losses yields the best overall task success rate. Regarding skill decomposition, we made several key observations. When using only the load-balancing loss, we found that while all experts were utilized during training, no clear expert specialization emerged. The router's output distribution remained relatively uniform, which prevents a clear mapping from experts to specific behaviors and hinders effective inference-time control. Conversely, when using only the entropy loss, the model suffered from severe router collapse, where the gating network quickly learned to activate only a small subset of experts, leaving the majority completely untrained. The full auxiliary loss, combining both terms, successfully addresses both issues. The load-balancing term ensures all experts participate in the training process, while the entropy term encourages each expert to develop a sharp, specialized function. This synergy is crucial for learning the robust and interpretable skill decomposition that is central to MoE-DP's success.

Tab. 5: **Ablation study on the auxiliary loss.** Bold numbers denote the best result in each row, and underlined numbers denote the second best. Across all tasks, the combination (LE) of the load-balance loss (L) and entropy loss (E) gives the best performance.

Task Ablation	Disturbance (Object)	Method			
		LE	L	E	N
Hammer Cleanup T0	✓	**84.0**	82.0	<u>83.0</u>	82.7
	✗(Hammer)	**33.3**	19.7	<u>26.5</u>	22.7
Kitchen Cleanup T0	✓	**96.7**	86.0	<u>93.0</u>	86.0
	✗(Hammer)	**82.7**	65.3	<u>74.7</u>	54.7
Table Cleanup T0	✓	**100.0**	96.7	98.3	<u>99.3</u>
	✗(Cube)	**98.0**	33.3	<u>64.7</u>	44.0

V. Conclusion

In this work, we addressed the critical challenges of robustness and interpretability in diffusion policies by introducing the MoE-DP. MoE-DP learns to decompose complex behaviors into a set of specialized and interpretable skills. This novel structure significantly enhances robustness, enabling reliable recovery from subtask failures where standard policies fail. Furthermore, MoE-DP learned decomposition provides a handle for inference-time control, allowing for the flexible recombination of skills to execute novel task sequences without re-training.

Limitations and Future Work. A primary limitation of MoE-DP is that, due to the varying data structures and characteristics of different tasks, identifying the optimal MoE architecture and related hyperparameters still requires empirical tuning. Future work could investigate adaptive or automated strategies for selecting MoE configurations, enabling more robust and generalizable robotic agents.

Acknowledgment

This work was supported by the Tsinghua Dushi fund.

REFERENCES

[1] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[2] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky, "π_0: A vision-language-action flow model for general robot control," 2024. [Online]. Available: https://arxiv.org/abs/2410.24164

[3] Y. Lu, Y. Tian, Z. Yuan, X. Wang, P. Hua, Z. Xue, and H. Xu, "H^3dp: Triply-hierarchical diffusion policy for visuomotor learning," 2025. [Online]. Available: https://arxiv.org/abs/2505.07819

[4] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," 2024. [Online]. Available: https://arxiv.org/abs/2403.03954

[5] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[6] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.

[7] W. Cai, J. Jiang, F. Wang, J. Tang, S. Kim, and J. Huang, "A survey on mixture of experts in large language models," *IEEE Transactions on Knowledge and Data Engineering*, p. 1–20, 2025. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2025.3554028

[8] D. Dai, C. Deng, C. Zhao, R. X. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu, Z. Xie, Y. K. Li, P. Huang, F. Luo, C. Ruan, Z. Sui, and W. Liang, "Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models," 2024. [Online]. Available: https://arxiv.org/abs/2401.06066

[9] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," 2022. [Online]. Available: https://arxiv.org/abs/2101.03961

[10] Qwen, :, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu, "Qwen2.5 technical report," 2025. [Online]. Available: https://arxiv.org/abs/2412.15115

[11] B. Zoph, I. Bello, S. Kumar, N. Du, Y. Huang, J. Dean, N. Shazeer, and W. Fedus, "St-moe: Designing stable and transferable sparse expert models," 2022. [Online]. Available: https://arxiv.org/abs/2202.08906

[12] X. Qu, D. Dong, X. Hu, T. Zhu, W. Sun, and Y. Cheng, "Llama-moe v2: Exploring sparsity of llama from perspective of mixture-of-experts with post-training," 2024. [Online]. Available: https://arxiv.org/abs/2411.15708

[13] S. Huang, Z. Zhang, T. Liang, Y. Xu, Z. Kou, C. Lu, G. Xu, Z. Xue, and H. Xu, "Mentor: Mixture-of-experts network with task-oriented perturbation for visual reinforcement learning," 2025. [Online]. Available: https://arxiv.org/abs/2410.14972

[14] Q. Chen, N. Gao, S. Huang, J. Low, T. Chen, J. Sun, and M. Schwager, "Grad-nav++: Vision-language model enabled visual drone navigation with gaussian radiance fields and differentiable dynamics," 2025. [Online]. Available: https://arxiv.org/abs/2506.14009

[15] R. Huang, S. Zhu, Y. Du, and H. Zhao, "Moe-loco: Mixture of experts for multitask locomotion," 2025. [Online]. Available: https://arxiv.org/abs/2503.08564

[16] M. Reuss, J. Pari, P. Agrawal, and R. Lioutikov, "Efficient diffusion transformer policies with mixture of expert denoisers for multitask learning," 2024. [Online]. Available: https://arxiv.org/abs/2412.12953

[17] Y. Li, Y. Lin, J. Cui, T. Liu, W. Liang, Y. Zhu, and S. Huang, "Clone: Closed-loop whole-body humanoid teleoperation for long-horizon tasks," 2025. [Online]. Available: https://arxiv.org/abs/2506.08931

[18] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," 2023. [Online]. Available: https://arxiv.org/abs/2304.13705

[19] S. Huang, B. Chen, H. Xu, and V. Sitzmann, "Dittogym: Learning to control soft shape-shifting robots," 2025. [Online]. Available: https://arxiv.org/abs/2401.13231

[20] S. Huang, Q. Chen, X. Zhang, J. Sun, and M. Schwager, "Particleformer: A 3d point cloud world model for multi-object, multi-material robotic manipulation," 2025. [Online]. Available: https://arxiv.org/abs/2506.23126

[21] Z. Yuan, T. Wei, L. Gu, P. Hua, T. Liang, Y. Chen, and H. Xu, "Hermes: Human-to-robot embodied learning from multi-source motion data for mobile dexterous manipulation," 2025. [Online]. Available: https://arxiv.org/abs/2508.20085

[22] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *International Conference on Machine Learning*, 2022.

[23] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," 2015. [Online]. Available: https://arxiv.org/abs/1503.03585

[24] L. Wang, H. Gao, C. Zhao, X. Sun, and D. Dai, "Auxiliary-loss-free load balancing strategy for mixture-of-experts," 2024. [Online]. Available: https://arxiv.org/abs/2408.15664

[25] Y. Zhu, P. Stone, and Y. Zhu, "Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation," 2022. [Online]. Available: https://arxiv.org/abs/2109.13841

[26] M. Xu, Z. Xu, C. Chi, M. Veloso, and S. Song, "Xskill: Cross embodiment skill discovery," 2023. [Online]. Available: https://arxiv.org/abs/2307.09955

[27] Z. Liang, Y. Mu, H. Ma, M. Tomizuka, M. Ding, and P. Luo, "Skilldiffuser: Interpretable hierarchical planning via skill abstractions in diffusion-based task execution," 2024. [Online]. Available: https://arxiv.org/abs/2312.11598

[28] M. Uehara, Y. Zhao, C. Wang, X. Li, A. Regev, S. Levine, and T. Biancalani, "Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review," 2025. [Online]. Available: https://arxiv.org/abs/2501.09685

[29] Y. Wang, L. Wang, Y. Du, B. Sundaralingam, X. Yang, Y.-W. Chao, C. Perez-D'Arpino, D. Fox, and J. Shah, "Inference-time policy steering through human interactions," 2025. [Online]. Available: https://arxiv.org/abs/2411.16627

[30] A. Ajay, Y. Du, A. Gupta, J. Tenenbaum, T. Jaakkola, and P. Agrawal, "Is conditional generative modeling all you need for decision-making?" 2023. [Online]. Available: https://arxiv.org/abs/2211.15657

[31] Y. Du, T. Lin, and I. Mordatch, "Model based planning with energy based models," 2021. [Online]. Available: https://arxiv.org/abs/1909.06878

[32] N. Gkanatsios, A. Jain, Z. Xian, Y. Zhang, C. Atkeson, and K. Fragkiadaki, "Energy-based models are zero-shot planners for compositional scene rearrangement," 2024. [Online]. Available: https://arxiv.org/abs/2304.14391

[33] M. Reuss, M. Li, X. Jia, and R. Lioutikov, "Goal-conditioned imitation learning using score-based diffusion policies," 2023. [Online]. Available: https://arxiv.org/abs/2304.02532

[34] H. Liu, Y. Zhang, V. Betala, E. Zhang, J. Liu, C. Ding, and Y. Zhu, "Multi-task interactive robot fleet learning with visual world models," 2024. [Online]. Available: https://arxiv.org/abs/2410.22689

[35] K. Nakamura, L. Peters, and A. Bajcsy, "Generalizing safety beyond collision-avoidance via latent-space reachability analysis," 2025. [Online]. Available: https://arxiv.org/abs/2502.00935

[36] M. Du and S. Song, "Dynaguide: Steering diffusion polices with active dynamic guidance," 2025. [Online]. Available: https://arxiv.org/abs/2506.13922

[37] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020. [Online]. Available: https://arxiv.org/abs/2006.11239

[38] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 681–688.

[39] S. Rissanen, M. Heinonen, and A. Solin, "Generative modelling with inverse heat dissipation," *arXiv preprint arXiv:2206.13397*, 2022.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[41] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "Gshard: Scaling giant models with conditional computation and automatic sharding," 2020. [Online]. Available: https://arxiv.org/abs/2006.16668

[42] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox, "Mimicgen: A data generation system for scalable robot learning using human demonstrations," 2023. [Online]. Available: https://arxiv.org/abs/2310.17596