ForceLDM: A Force-aware Latent Dynamics Model for Contact-Rich Manipulation

Peiyuan Tang[1,4*] **Yao Li**[2,3*] **Xiaodong Zhang**[5†] **Wuyang Zhang**[2,3,4]
Qin Xia[1] **Yanyong Zhang**[2] **Zijiang James Yang**[5,6†]

[1]School of Computer Science and Technology, Xi'an Jiaotong University
[2]School of Artificial Intelligence and Data Science, University of Science and Technology of China
[3]Suzhou Institute for Advanced Research, University of Science and Technology of China
[4]Suzhou Artificial Intelligence Laboratory
[5]School of Computer Science and Technology, University of Science and Technology of China
[6]Synkrotron, Inc.
[*]Equal contribution [†]Corresponding authors.

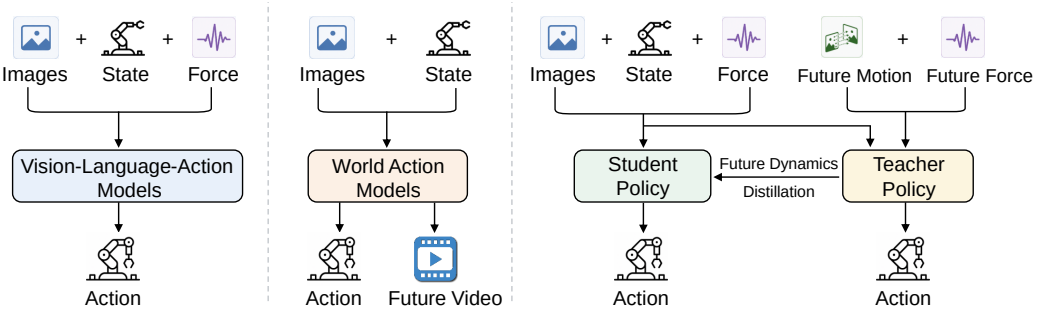
Abstract

Contact-rich manipulation requires fine-grained control and complex physical interactions with the environment, posing significant challenges for robot policy learning. Vision-only policies struggle with such tasks, as contact states and interaction forces are difficult to infer from visual observations alone. Existing methods mitigate this issue by incorporating force or tactile sensing. However, they often treat these signals as auxiliary observations without fully exploiting their role in modeling future interaction dynamics. To address this, we propose ForceLDM, a force-aware latent dynamics model that learns future contact and motion dynamics to guide action generation. Rather than directly predicting future observations, which can entangle task-relevant dynamics with irrelevant sensory details, ForceLDM learns a compact representation of future dynamics through knowledge distillation. Specifically, a teacher network with privileged future force/torque and optical flow signals extracts future dynamic features, which are then distilled into a student network trained only on current observations. This enables the student model to reason about upcoming contacts and physical dynamics without relying on future sensory inputs at deployment. To improve robustness and generalization, we further introduce a curriculum-based progressive noise injection strategy to mitigate over-reliance on future features. Experiments on five real-world contact-rich manipulation tasks demonstrate that ForceLDM outperforms state-of-the-art baselines, generalizes effectively to novel objects, and remains robust to environmental perturbations. Project page and code are available at `https://tpy001.github.io/ForceLDM/`.

1 Introduction

Humans are naturally capable of performing dexterous manipulation such as USB insertion and component assembly. However, these simple daily tasks remain highly challenging for robots. Contact-rich manipulation, particularly high-precision insertion, requires not only accurate motion planning and execution, but also the ability to perceive and respond to contact forces in real time [11, 44, 22]. Although vision-based policies have achieved remarkable progress in general-purpose robotic manipulation [17, 5, 4, 3], they still struggle with contact-rich tasks, as contact states and force magnitudes are often difficult to infer reliably from vision alone. This limitation has motivated

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Force-aware VLA (b) World Action Model (c) ForceLDM (Ours)

Figure 1: Comparison with prior methods. (a) Force-aware VLAs incorporate force feedback but lack explicit modeling of future interaction dynamics. (b) Video-based WAMs model future dynamics through visual prediction, but often lack force/tactile feedback and may fit task-irrelevant visual details. (c) ForceLDM learns compact, action-relevant future dynamics by distilling future motion and force cues from a privileged teascher.

recent studies [46, 21, 25, 52, 44] to incorporate physical feedback, such as force or torque sensing, to improve contact-rich manipulation performance.

Recent studies on force-aware contact-rich manipulation have primarily focused on designing more effective multimodal architectures to fully exploit force/torque feedback. For example, ForceVLA [46] employs a Mixture-of-Experts (MoE) architecture with modality-specific experts to adaptively fuse visual and force features. TA-VLA [52] investigates joint torque fusion in VLA models and shows that injecting torque signals into the action generation module improves performance. Other works [44, 54] explore fast-slow dual-system architectures that combine low-frequency vision with high-frequency force feedback for responsive control.

Despite these advances, most work still treats force/torque signals as additional inputs without modeling future interaction dynamics. This leads to two key limitations. First, high-dimensional visual observations provide rich global semantics, whereas low-dimensional force/torque signals are spatiotemporally sparse and only effective during the contact stage. Consequently, models tend to rely on visual cues while failing to fully exploit force feedback [32, 7, 41]. Second, most force-aware VLAs learn only a mapping from current observations to actions, which may lead to overfitting to expert behaviors and limit generalization to unseen environments.
A natural approach to addressing these limitations is to incorporate world models into VLAs, learning dynamics modeling through future observation prediction [13, 42, 51]. However, we find that integrating this idea into force-aware VLAs is non-trivial. On the one hand, most existing world models are based on video generation models. They can generate visually realistic videos, but may fail to capture physically consistent contact dynamics. Fine-tuning these models with force observations requires substantial computational resources and large amounts of synchronized multimodal data. On the other hand, predicting future observations often entangles task-relevant dynamics with task-irrelevant details, such as backgrounds, object appearances, and sensor noise. This may lead models to focus on irrelevant information rather than dynamics that are critical for control.

To address these challenges, we propose **ForceLDM**, a force-aware latent dynamics model for contact-rich manipulation. As illustrated in Figure 1, ForceLDM uses optical flow and force/torque signals as dynamics representations, providing direct information about motion and contact. To effectively model these dynamics, we propose a Future-Dynamics Distillation framework that distills future dynamics from a teacher to a student model. Specifically, a privileged teacher model extracts compact dynamics features from future observations by predicting actions. This encourages the teacher to selectively retain information that is most useful for control. A student policy is then trained to match these latent representations using a set of learnable query tokens. This allows the student to imagine future dynamics in the teacher's latent space without reconstructing future observations in full detail. To further improve generalization, we introduce curriculum-based Progressive Noise Injection, which gradually increases noise on the student's query tokens to reduce overfitting. Together, these designs enable ForceLDM to learn control-relevant future dynamics and improve generalization in contact-rich manipulation.

We evaluate ForceLDM on five real-world contact-rich manipulation tasks under both in-distribution and out-of-distribution settings. ForceLDM achieves stronger task performance and better generalization to unseen visual and physical variations than existing VLA and force-aware baselines. Further ablations and representation analyses show that the student queries encode predictive information about future force and motion dynamics, indicating that ForceLDM implicitly learns physical interaction dynamics rather than merely fitting expert actions. Our contributions are summarized as follows:

- We propose a future dynamics representation where force/torque reflects contact states and optical flow encodes motion dynamics, overcoming the limitations of purely visual representations in contact-rich manipulation.
- We introduce ForceLDM, a force-aware latent dynamics model that enables effective dynamics modeling from current observations, improving policy robustness and generalization to unseen objects and environmental variations.
- We empirically validate ForceLDM on five real-world contact-rich manipulation tasks, demonstrating its effectiveness in improving task performance, out-of-distribution generalization, and implicit dynamics learning.

2 Related Work

Vision-Language-Action Models. Recent studies have leveraged pretrained vision-language models to build general-purpose robotic policies that follow open-ended instructions and perform diverse manipulation tasks [58, 5, 24, 3]. These methods encode visual observations and language instructions into multimodal representations and map them to robot actions. Existing approaches adapt VLMs into robot policies by either autoregressively predicting discretized action tokens [17, 33] or directly modeling continuous action distributions via flow matching [5, 3]. Building on these foundations, recent studies further improve VLA models by enhancing high-level reasoning and planning [23, 26], strengthening 3D spatial perception [35, 20], incorporating memory mechanisms [37, 43], and developing dual-system architectures [6, 38]. Despite these advances, most VLA policies rely primarily on visual observations and robot states, which provide limited information about contact events, force directions and friction. To bridge this gap, we introduce ForceLDM, which fuses force/torque sensing with vision-language representations to improve precision and compliance in contact-rich manipulation.

Contact-Rich Manipulation with Physical Feedback. Several approaches have explored incorporating physical feedback into robotic policies. Forcemimic [25] and FoAR [11] introduce force-aware policies by integrating 6-DoF force/torque measurements, enabling robots to better perceive interaction dynamics. Other works [52, 31] exploit proprioceptive signals, such as joint efforts, to estimate external contact states without additional force sensors. Beyond force-based sensing, tactile-based approaches leverage high-resolution tactile images to capture localized contact geometry and surface interactions [14, 10, 15, 2]. To fuse these physical modalities with vision-language features, existing methods adopt feature concatenation [49, 52, 10], cross-attention [12, 57, 18], gating mechanisms [11, 50, 54, 19], and Mixture-of-Experts architectures [16, 46, 21]. Building on this, recent works [47, 54, 27, 41] incorporate world models to reason about future physical interactions by jointly predict future force or tactile signals and future videos. However, this multimodal prediction paradigm requires large-scale synchronized visual and force/tactile data and incurs substantial computational overhead, leading to high inference latency. In contrast, our method distills task-relevant future force and motion signals into the policy, enabling efficient inference by avoiding video generation.

World Modeling for VLA Policies. VLA policies learn a direct mapping from current visual-language observations to robot actions [58, 17, 5], whereas world models learn how future world states evolve through interaction [9, 56, 45]. Incorporating world modeling into VLA policies augments action supervision with future-state prediction, allowing action generation to account for future consequences rather than expert actions alone [13, 8, 51, 42]. Early world-action models realize this idea through future image or video prediction [56, 45], whereas latent world models replace expensive pixel-space rollouts with compact, task-relevant predictive representations. In contrast to approaches that derive action-oriented predictive representations primarily from future visual observations [53, 39, 28], ForceLDM distills privileged future force/torque and optical-flow

dynamics into the policy to capture contact physics that may be visually subtle or occluded, without requiring future inputs or explicit rollouts at inference time.

3 Method

We introduce **ForceLDM**, a latent dynamics learning framework that distills future force and motion dynamics into vision-language-action models, enabling them to better model physical interactions. In this section, we begin with the problem formulation, followed by an overview of the framework, and then describe the training and inference strategies.

3.1 Problem Formulation

We formulate contact-rich manipulation as a policy learning problem. At each time step t, the policy π_θ takes the multimodal observation $\mathbf{O}_t = (I_t, s_t, F_t, l)$ as input, and outputs an action chunk $\mathbf{A}_t = (\mathbf{a}_t, \ldots, \mathbf{a}_{t+H-1})$ consisting of H actions to be executed sequentially:

$$\mathbf{A}_t = \pi_\theta(I_t, s_t, F_t, l), \tag{1}$$

where I_t denotes multi-view images, s_t denotes the robot state (end-effector pose or joint positions), F_t denotes 6D wrench measurements, and l denotes the language instruction. To train the policy, we collect a dataset consisting of N expert demonstrations, denoted as $\mathcal{D} = \{\tau_i\}_{i=1}^{N}$, where each trajectory is defined as $\tau_i = \{(\mathbf{O}_t^i, \mathbf{a}_t^i)\}_{t=1}^{T_i}$. The policy π_θ is optimized via behavior cloning [1, 36] by minimizing the discrepancy between predicted actions and expert demonstrated actions:

$$\mathcal{L}_{\mathrm{BC}}(\theta) = \mathbb{E}_{(\mathbf{O}_t, \mathbf{A}_t)\sim\mathcal{D}} \left[\ell_{\mathrm{act}}\big(\mathbf{A}_t, \pi_\theta(\mathbf{O}_t)\big) \right], \tag{2}$$

where ℓ_{act} denotes the action prediction loss, commonly defined as the mean absolute error (MAE) or mean squared error (MSE).

3.2 Overall Architecture

We aim to improve contact-rich manipulation by learning future interaction dynamics. Instead of predicting all details of future observations, we adopt a teacher-student distillation framework to focus on information relevant to robot control. As illustrated in Figure 2, ForceLDM consists of a shared VLM backbone, a future-dynamics encoder, and two action experts. The VLM backbone encodes current images and language instructions to provide scene context. Robot states and force history are embedded as tokens and fed into the action experts. During training, the teacher takes future force and optical flow as inputs and learns compact representations of physical interactions through action prediction. These representations are then distilled into the student through representation alignment, allowing it to infer future interactions from current observations. To improve generalization, we further introduce progressive noise injection, which gradually adds noise to the student's future queries to prevent overfitting. At inference time, only the student policy is retained for deployment.

3.3 Future Dynamics Encoding

Prior approaches [13, 42, 48] often predict future visual observations, but RGB images are sensitive to changes in lighting and appearance and cannot fully capture physical interactions such as contact and resistance. We therefore use future force/torque measurements and optical flow as complementary sources of information. Force/torque measurements provide direct feedback on physical contact, while optical flow captures the motion of the robot and objects. Their combination provides more informative cues for robot control than RGB frames alone.

Future Force Encoding. We use an MLP to encode the 6D force/torque measurements over the next H steps into a compact force token:

$$\mathbf{f}_t = E_f\left(\mathbf{F}_{t:t+H}\right). \tag{3}$$

This token captures changes in contact forces over time and provides information about the robot's physical interactions with its environment.

Future Motion Encoding. Optical flow provides a dense representation of scene motion by describing the pixel-wise displacement between two frames. Given the current image I_t and a future image

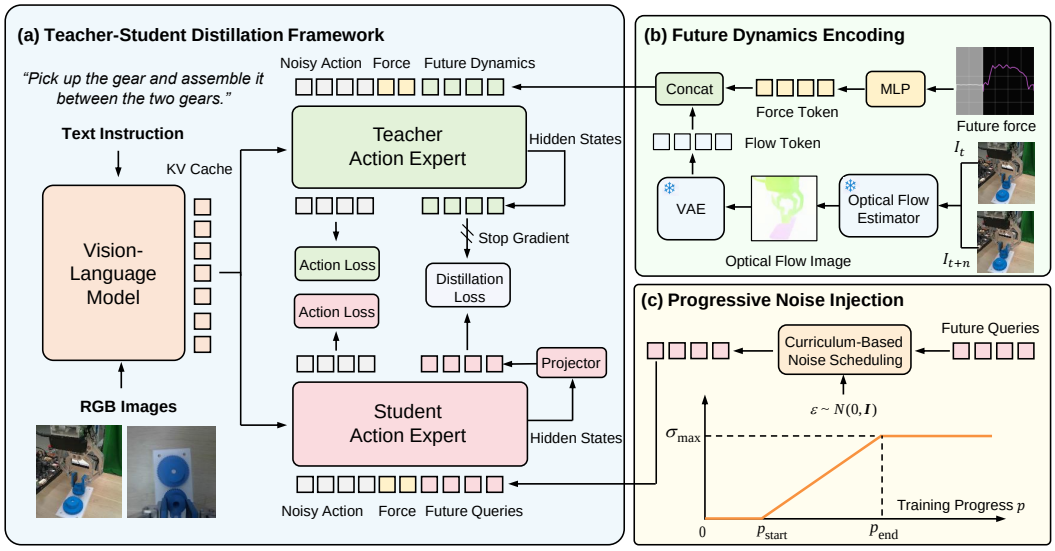


Figure 2: Overview of ForceLDM. (a) Both teacher and student generate actions from current multimodal observations and language instructions. The teacher additionally receives future force and motion tokens, and its future-aware representations are distilled into the student's future queries. (b) The future dynamics encoder converts future force/torque and optical flow into compact force and motion tokens. (c) Progressive noise injection perturbs the student's future queries with scheduled noise to improve generalization.

I_{t+H}, we use a pretrained optical-flow estimation model [40] to obtain a dense motion field. For each pixel p, the corresponding flow vector $\mathbf{u}(p) = (u_x(p), u_y(p))$ represents its horizontal and vertical displacement in image space, thereby capturing the image-space motion of the robot and manipulated objects. To align the optical flow with the image representation used in our framework, we colorize the two-channel flow field into a three-channel motion image $\mathbf{M}_{t\to t+H}$. Following FlowVLA [55], we map flow direction to hue and use the normalized, clipped magnitude to control the strength of the resulting direction color against a white background. We then compress $\mathbf{M}_{t\to t+H}$ using a pretrained image VAE [34] and employ N_m learnable motion queries Q_m to aggregate the encoded features into compact motion tokens:

$$\begin{aligned} \mathbf{Z}_m &= E_m\left(\mathbf{M}_{t\to t+H}\right), \\ \mathbf{m}_t &= \mathrm{CrossAttn}\left(\mathbf{Q}_m, \mathbf{Z}_m\right). \end{aligned} \tag{4}$$

where E_m denotes the pretrained VAE model.

3.4 Online Future-Dynamics Distillation

Directly predicting future observations may force the model to learn details irrelevant to control. It may also lead to overfitting to sensor noise in force/torque measurements and estimation errors in optical flow. To avoid these issues, we learn future dynamics through latent representation alignment, without explicitly reconstructing future observations in detail.

The student uses $N_m + 1$ learnable queries $\mathbf{Q} \in \mathbb{R}^{(N_m+1)\times D}$, comprising one force query and N_m motion queries, to predict the teacher's future-aware intermediate representations. We perform multi-layer distillation over a set of selected layers K. At each layer $l \in K$, let $\mathbf{h}_f^{(l)}$ and $\mathbf{y}_f^{(l)}$ denote the student and teacher force representations, while $\mathbf{h}_m^{(l,i)}$ and $\mathbf{y}_m^{(l,i)}$ denote their i-th motion representations. The distillation loss aligns both modalities using cosine similarity:

$$\mathcal{L}_{\text{distill}} = \frac{1}{|K|}\sum_{l\in K}\left[1 - \cos\left(g_f(\mathbf{h}_f^{(l)}), \mathrm{sg}(\mathbf{y}_f^{(l)})\right) + \frac{1}{N_m}\sum_{i=1}^{N_m}\left[1 - \cos\left(g_m(\mathbf{h}_m^{(l,i)}), \mathrm{sg}(\mathbf{y}_m^{(l,i)})\right)\right]\right]. \tag{5}$$

Here, g_f and g_m denote the projection heads for force and motion representations, respectively, while $\mathrm{sg}(\cdot)$ denotes the stop-gradient operation that prevents gradients from propagating to the teacher.

3.5 Training and Inference

Progressive Noise Injection. Future interaction dynamics are inherently uncertain, and limited training data can cause the student to overfit. To improve generalization, we introduce progressive noise injection by adding Gaussian noise to the student queries. Let $\mathbf{Q}$ and $\widetilde{\mathbf{Q}}$ be the original and perturbed student queries, respectively, $p \in [0, 1]$ be the normalized training progress, and $\sigma(p)$ be the corresponding noise scale. We perturb the queries with RMS-scaled Gaussian noise:

$$\widetilde{\mathbf{Q}} = \mathbf{Q} + \sigma(p) \cdot \mathrm{RMS}(\mathbf{Q}) \cdot \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{6}$$

where $\mathrm{RMS}(\mathbf{Q})$ denotes the root mean square of the query values. The noise scale $\sigma(p)$ remains zero until p_{start}, then increases linearly to $\sigma_{\max}$ at p_{end}, and remains at $\sigma_{\max}$ thereafter. This schedule first allows the student to learn meaningful representations through alignment and then improves its robustness to query perturbations.

Action Generation via Flow Matching. We train both the teacher and student policies using flow matching. Given a ground-truth action chunk $\mathbf{a}$, we sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\tau \sim \mathcal{U}(0, 1)$, construct $\mathbf{a}_\tau = (1 - \tau)\boldsymbol{\epsilon} + \tau\mathbf{a}$, and define the target velocity as $\mathbf{v}^* = \mathbf{a} - \boldsymbol{\epsilon}$. Let c_S and c_T represent the student and teacher conditioning information, respectively. Their flow matching losses are:

$$\mathcal{L}_S = \mathbb{E}\Big[\|\mathbf{v}_S(\mathbf{a}_\tau, \tau, c_S) - \mathbf{v}^*\|_2^2\Big], \qquad \mathcal{L}_T = \mathbb{E}\Big[\|\mathbf{v}_T(\mathbf{a}_\tau, \tau, c_T) - \mathbf{v}^*\|_2^2\Big]. \tag{7}$$

Both policies use the same noise sample and interpolation time for consistent velocity prediction. The final training objective is:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_S + \mathcal{L}_T + \lambda\mathcal{L}_{\mathrm{distill}}, \tag{8}$$

where $\lambda = 0.5$ controls the weight of the distillation loss.

Inference. During inference, only the student policy is retained, while the teacher and future-dynamics encoder are removed. The student generates actions using current observations without requiring future force or motion information.

4 Experiments

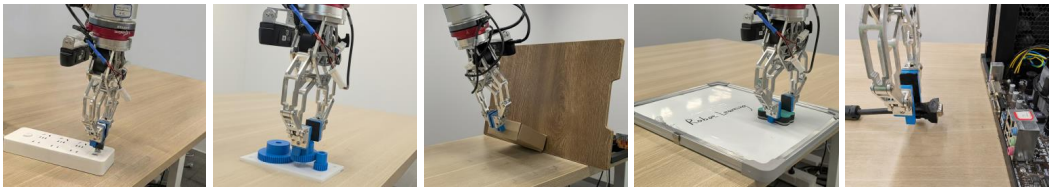
In this section, we evaluate our method on contact-rich manipulation tasks through extensive experiments. We first report results on five real-world tasks and assess out-of-distribution generalization. We then conduct ablation studies on future prediction targets, model architecture, and training strategies, followed by visualizations of the learned latent dynamics.

4.1 Experimental Setup

Hardware Platform and Data Collection Our main experiments use a 7-DoF xArm 7 manipulator equipped with wrist-mounted and fixed third-person cameras and a 6D force/torque sensor. We collect successful demonstrations through 3D SpaceMouse teleoperation at 30 Hz, recording synchronized visual observations, robot states, force/torque measurements, and actions for policy training. Detailed hardware configurations and data collection procedures are provided in Appendix A.

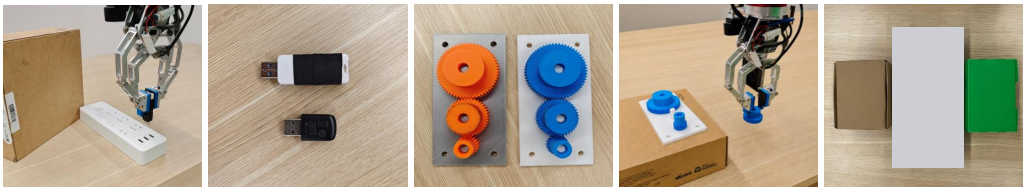
Tasks and Evaluation Metrics. We evaluate ForceLDM on five real-world contact-rich manipulation tasks: USB Insertion, Gear Assembly, Box Flipping, Board Wiping, and VGA Plugging, as shown in Figure 3. For each task, we collect 80–100 successful demonstration trajectories and evaluate each method over 20–30 real-world rollouts. We report task success rate over repeated real-world trials as the primary evaluation metric. Detailed task descriptions, success criteria, and evaluation protocols are provided in Appendix A.

Compared Baselines. We compare our method against several strong baselines. **π_0 [5] w/o Force** is a flow-matching-based Vision-Language-Action model pretrained on large-scale robot data. **π_0 w/ Force** is a force-augmented variant of π_0 that converts force measurements into tokens and concatenates them with the input sequence. **TA-VLA** [52] is a force-aware VLA model that introduces an auxiliary future-force prediction objective. **ForceVLA** [46] is a force-aware VLA model that adopts a Mixture-of-Experts architecture [16] to model visual and force modalities with specialized



(a) USB Insertion (b) Gear Assembly (c) Box Flipping (d) Board Wiping (e) VGA Plugging

Figure 3: Five real-world contact-rich manipulation tasks used in our experiments.



(a) Visual Occlusion (b) Unseen objects (c) Unseen Color (d) Unseen Height (e) Unseen Size and Material

Figure 4: Out-of-distribution generalization settings used in our real-world evaluation.

experts. **Seer [42] w/ Force** augments Seer with force input and uses future RGB reconstruction as auxiliary supervision. For a fair comparison, all methods are fully fine-tuned on the same collected dataset using the same training configuration. Detailed baseline implementations and training settings are provided in Appendix D.

Implementation Details We build ForceLDM on π_0 [5], using two camera views and a 10-step history of 6D force/torque measurements as inputs. Future optical flow is estimated with pretrained RAFT [40] and encoded using SDXL-VAE [34]. We use 16 learnable motion queries and one learnable force query, and align teacher–student representations at layers $\{8, 12, 16\}$ with a distillation loss weight of $\lambda = 0.5$. We fully fine-tune a multi-task policy for 30K steps on a single A100-80G GPU with a batch size of 16, using AdamW and a cosine learning-rate schedule that decays from 2.5×10^{-5} to 2.5×10^{-6} after 1,000 warmup steps. At inference, we use 10 flow-matching steps to generate each 32-step action chunk and execute all 32 actions before predicting the next chunk. Detailed model architecture and training hyperparameters are provided in Appendices B and C.

4.2 Main Results

Table 1 summarizes the performance on in-distribution contact-rich manipulation tasks. ForceLDM achieves the highest average success rate of 87.1%, outperforming Seer [42], the strongest baseline, by 14.9 percentage points. ForceLDM performs particularly well on insertion tasks requiring millimeter-level alignment, achieving 92.0% on USB Insertion and 85.0% on VGA Plugging. We attribute this to learning future contact dynamics, which helps the model understand physical interactions and adjust the insertion pose through contact-based exploration. In contrast, baselines that model visual dynamics or directly predict future force may overfit to irrelevant details or sensor noise, rather than learning control-relevant information. Despite these gains, ForceLDM underperforms TA-VLA on Gear Assembly due to limited spatial perception, which sometimes causes grasping failures near the workspace boundary. Further analysis is provided in Appendix G.

4.3 Out-of-Distribution Generalization

OOD Evaluation Settings. We evaluate OOD generalization on three representative contact-rich tasks, as shown in Figure 4. For USB Insertion, we consider **Visual Occlusion**, where the external camera view is partially blocked by an object, and **Unseen Object**, where the USB connector is replaced with a novel instance. For Gear Assembly, we consider **Unseen Color**, which changes the object appearance, and **Unseen Height**, which changes the target assembly height. For Box Flipping, we consider **Unseen Material**, where the original wooden box is replaced with a 3D-printed box, leading to different friction properties, and **Unseen Size**, which evaluates the policy's ability to flip boxes of different sizes. All OOD settings preserve the original task objectives and success criteria to evaluate robustness to visual and physical changes. Each OOD setting is evaluated over 15 trials.

Table 1: Comparison of success rates (%) on in-distribution (ID) contact-rich manipulation tasks.

Model	USB Insertion	Gear Assembly	Box Flipping	Board Wiping	VGA Plugging	Avg.
π_0 [5] w/o force	44.0	46.7	55.0	60.0	0.0	41.1
π_0 [5] w/ force	64.0	83.3	70.0	70.0	5.0	58.5
TA-VLA [52]	80.0	**93.3**	80.0	75.0	10.0	67.7
ForceVLA [46]	52.0	66.7	65.0	65.0	5.0	50.7
Seer [42] w/ Force	76.0	70.0	**95.0**	70.0	50.0	72.2
Ours	**92.0**	83.3	**95.0**	**80.0**	**85.0**	**87.1**

Table 2: Comparison of success rates (%) on out-of-distribution (OOD) tasks under visual occlusion and changes in object instance, color, assembly height, material, and size.

Model	USB Insertion		Gear Assembly		Box Flipping		Avg.
	Object	Occlusion	Height	Color	Material	Size	
π_0 [5] w/o force	20.0	26.7	0.0	40.0	26.7	0.0	18.9
π_0 [5] w/ force	40.0	73.3	0.0	46.7	26.7	33.3	36.7
TA-VLA [52]	20.0	66.7	13.3	**66.7**	13.3	26.7	34.5
ForceVLA [46]	26.7	20.0	0.0	40.0	0.0	13.3	16.7
Seer [42] w/ Force	46.7	20.0	**53.3**	60.0	73.3	40.0	48.9
Ours	**86.7**	**86.7**	46.7	60.0	**80.0**	**53.3**	**68.9**

Results. Table 2 reports the OOD evaluation results. ForceLDM achieves the highest average success rate of 68.9%, outperforming Seer w/ Force, the strongest baseline, by 20.0 percentage points. Notably, ForceLDM achieves a success rate of 86.7% on USB Insertion under both unseen objects and visual occlusion, demonstrating strong robustness to changes in object appearance and visual occlusion. It also achieves the best performance on Box Flipping under material and size changes, showing its ability to handle changes in object properties and contact conditions. These results suggest that learning future dynamics from optical flow and force signals helps the model focus on physically relevant interactions rather than appearance-specific details, improving generalization across different OOD conditions.

4.4 Ablation Studies

Ablation on Future Prediction Targets. We study which future modalities are most useful for dynamics distillation, as different signals capture different aspects of physical interaction. We compare four choices: optical flow alone, force alone, force with RGB observations, and force with optical flow. As shown in Table 3, different modalities provide complementary benefits. Future force is particularly useful for tasks requiring sustained contact control, achieving 90.0% success on Box Flipping, but performs poorly on VGA Plugging (30.0%), where precise motion and alignment are critical. In contrast, optical flow provides direct motion cues that benefit high-precision manipulation tasks, achieving 92.0% on USB Insertion, but performs less effectively on tasks requiring sustained force control, such as Box Flipping. Adding future RGB observations to force does not improve performance, possibly because the model focuses on irrelevant visual details rather than control-relevant interaction dynamics. Combining force and optical flow achieves the highest average success rate of 90.7%, substantially outperforming force alone at 68.0% and optical flow alone at 70.7%. These results support our choice of future force and optical flow as complementary supervision signals for dynamics distillation.

Ablation on Model Architecture. We study the importance of key components in our teacher–student architecture for learning action-relevant future dynamics. We consider three variants by removing the distillation loss $\mathcal{L}_{\text{distill}}$, teacher action loss $\mathcal{L}_T$, and stop-gradient operation, respectively. As shown in Table 4, removing $\mathcal{L}_{\text{distill}}$ reduces the average success rate from 85.1% to 42.8%, indicating that action supervision alone is insufficient to transfer future dynamics from the teacher to the student. Removing $\mathcal{L}_T$ also leads to a substantial performance drop, suggesting that the teacher must be

Table 3: Ablation of future modalities used for dynamics distillation. Visual, Force, and Motion denote future RGB observations, force/torque signals, and optical flow, respectively. We report success rates (%) on USB Insertion, Box Flipping, and VGA Plugging.

Future Target			Success Rate (%)			
Visual	Force	Motion	USB Insertion	Box Flipping	VGA Plugging	Avg.
–	–	✓	**92.0**	60.0	60.0	70.7
–	✓	–	84.0	90.0	30.0	68.0
✓	✓	–	80.0	85.0	35.0	66.7
–	✓	✓	**92.0**	**95.0**	**85.0**	**90.7**

Table 4: Ablation of teacher–student distillation components. $\mathcal{L}_{\text{distill}}$ and $\mathcal{L}_T$ denote the representation alignment loss and teacher action loss, respectively. Stop Gradient denotes gradient stopping on teacher targets. We report success rates (%) under both in-distribution (ID) and out-of-distribution (OOD) conditions.

Configuration	USB Insertion		Box Flipping		VGA Plugging	Avg.
	ID	OOD	ID	OOD		
w/o $\mathcal{L}_{\text{distill}}$	44.0	30.0	85.0	30.0	25.0	42.8
w/o $\mathcal{L}_T$	60.0	36.7	80.0	16.7	10.0	40.7
w/o Stop Gradient	68.0	50.0	60.0	16.7	20.0	42.9
Full Model	**92.0**	**86.7**	**95.0**	**66.7**	**85.0**	**85.1**

grounded in action prediction to learn useful representations for distillation. Similarly, removing stop-gradient leads to representation collapse, resulting in a substantial performance drop. These results validate the effectiveness of our teacher–student architecture and its key design choices.

Ablation on Training Strategies. Table 5 studies the importance of latent dynamics distillation and progressive noise injection in ForceLDM. First, we compare latent dynamics distillation with an alternative strategy that directly predicts future signals through joint prediction. This variant achieves an average success rate of only 62.1%, substantially below the full model. This suggests that direct prediction may encourage the model to focus on irrelevant details and sensor noise rather than learning control-relevant dynamics. Second, removing progressive noise reduces the average success rate from 85.1% to 74.1%. The performance drop is larger in OOD settings than in ID settings, suggesting that progressive noise injection helps improve the model's generalization ability.

4.5 Visualization

Visualization of Latent Dynamics. To examine whether the learnable future queries capture future motion and contact dynamics, we train lightweight probes to predict future optical flow and contact forces from the future query representations at layer 12 of the frozen policy. Figure 5 shows five stages of a USB insertion task: approach, stuck, realignment, insertion, and return. When the gripper becomes stuck, the predicted optical flow indicates an upward motion for recovery. During realignment, the predicted flow shows the direction of the next adjustment, followed by downward motion during insertion. The predicted force follows the overall trend of the measured force, capturing the main changes during two contact events. Although the predicted values differ from the measurements, the predicted force is smoother and does not reproduce high-frequency noise in the measured signal. These results suggest that the learned representations capture future motion and contact dynamics while avoiding overfitting to irrelevant details.

Additional Visualizations and Failure Case Analysis. Additional visualizations of successful task executions under ID and OOD conditions are provided in Appendix F. Representative failure cases and an analysis of common failure modes are presented in Appendix G.

Table 5: Ablation of the training strategies used in ForceLDM. Joint Prediction combines action learning with direct future-dynamics prediction, while w/o Progressive Noise removes progressive noise injection into the student's future queries.

Training Strategy	USB Insertion		Box Flipping		VGA Plugging	Avg.
	ID	OOD	ID	OOD		
Joint Prediction	84.0	76.7	65.0	20.0	65.0	62.1
w/o Progressive Noise	84.0	73.3	90.0	40.0	**85.0**	74.1
Full Model	**92.0**	**86.7**	**95.0**	**66.7**	**85.0**	**85.1**

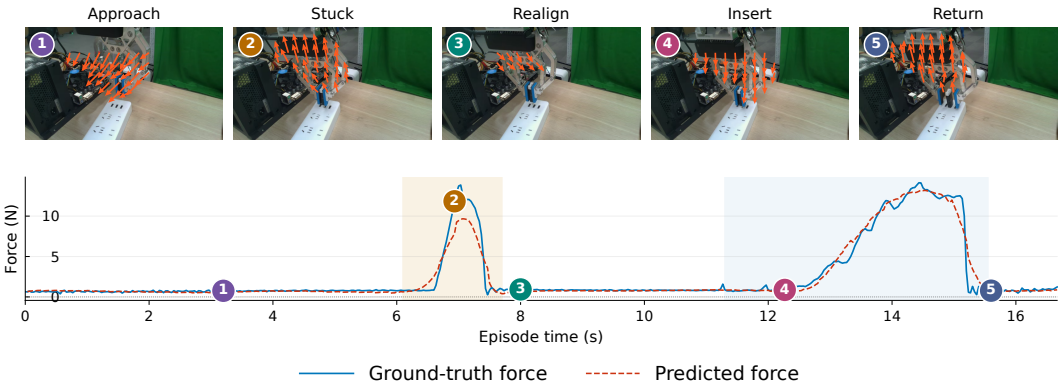


Figure 5: **Probing Latent Dynamics.** We freeze the trained model and fit lightweight probes to predict optical flow and contact forces from its intermediate hidden states. **Top:** Predicted future 2D motion vectors derived from optical flow (bright orange-red arrows) at five stages of the USB insertion task. **Bottom:** Measured contact force (blue solid) and predicted force (red dashed).

5 Conclusion

We propose **ForceLDM**, a force-aware latent dynamics model for contact-rich manipulation. At its core, ForceLDM distills future force/torque and optical-flow cues from a privileged teacher into a student policy, enabling it to capture action-relevant interaction dynamics using only current observations at inference. Experiments on five real-world contact-rich manipulation tasks demonstrate that ForceLDM outperforms force-aware VLA baselines in both task performance and generalization to unseen visual and physical variations. Ablation studies validate the complementary roles of future force and optical flow, together with the benefits of progressive noise injection. These results highlight the value of learning future physical dynamics for force-aware robotic manipulation.

Limitations. Despite the promising results of ForceLDM, several limitations remain. First, our experiments use small-scale datasets, and whether ForceLDM maintains its advantage as training data scales up remains unclear. Second, optical-flow estimation errors caused by occlusion or motion blur may affect the learned dynamics representations. Third, 2D optical flow lacks explicit depth information, which may limit the model's ability to capture spatial relationships during manipulation. Future work will explore scaling up training data to improve performance and combining 3D scene flow with force feedback to better model future interaction dynamics.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 62402367, 62232008, and 2024CSJGG01500, the Fundamental Research Funds for the Central Universities, and the Jiangsu Funding Program for Excellent Postdoctoral Talent under Grant No.2025ZB315. The authors declare no competing interests.

References

[1] Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine intelligence 15*, pages 103–129, 1995.

[2] Jianxin Bi, Kevin Yuchen Ma, Ce Hao, Mike Zheng Shou, and Harold Soh. VLA-Touch: Enhancing vision-language-action model with dual-level tactile feedback. *IEEE Robotics Autom. Lett.*, 11(7):8487–8494, 2026.

[3] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: an open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[4] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Conference on Robot Learning*, volume 305, pages 17–40, 2025.

[5] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems*, 2025.

[6] Hao Chen, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Renrui Zhang, Xiaoqi Li, Xiao He, Yandong Guo, Chi-Wing Fu, Shanghang Zhang, et al. Fast-in-slow: A dual-system vla model unifying fast manipulation within slow reasoning. In *Advances in Neural Information Processing Systems*, 2025.

[7] Haonan Chen, Jiaming Xu, Hongyu Chen, Kaiwen Hong, Binghao Huang, Chaoqi Liu, Jiayuan Mao, Yunzhu Li, Yilun Du, and Katherine Driggs-Campbell. Multi-modal manipulation via multi-modal policy consensus. In *IEEE International Conference on Robotics and Automation*, pages 408–415, 2026.

[8] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *IEEE International Conference on Robotics and Automation*, pages 2786–2793, 2017.

[9] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

[10] Peng Hao, Chaofan Zhang, Dingzhe Li, Xiaoge Cao, Xiaoshuai Hao, Shaowei Cui, and Shuo Wang. TLA: Tactile-language-action model for contact-rich manipulation. *Robot Learning*, 2026.

[11] Zihao He, Hongjie Fang, Jingjing Chen, Hao-Shu Fang, and Cewu Lu. FoAR: Force-aware reactive policy for contact-rich robotic manipulation. *IEEE Robotics and Automation Letters*, 10(6):5625–5632, 2025.

[12] Liang Heng, Haoran Geng, Kaifeng Zhang, Pieter Abbeel, and Jitendra Malik. ViTacFormer: Learning cross-modal representation for visuo-tactile dexterous manipulation. In *Robotics: Science and Systems*, 2026.

[13] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *International Conference on Machine Learning*, 2025.

[14] Binghao Huang, Yixuan Wang, Xinyi Yang, Yiyue Luo, and Yunzhu Li. 3D-ViTac: Learning fine-grained manipulation with visuo-tactile sensing. In *Conference on Robot Learning*, 2024.

[15] Jialei Huang, Shuo Wang, Fanqi Lin, Yihang Hu, Chuan Wen, and Yang Gao. Tactile-vla: unlocking vision-language-action model's physical knowledge for tactile generalization. *arXiv preprint arXiv:2507.09160*, 2025.

[16] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[17] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning*, 2025.

[18] Geonhyup Lee, Yeongjin Lee, Kangmin Kim, Seongju Lee, Sangjun Noh, Seunghyeok Back, and Kyoobin Lee. Manipforce: Force-guided policy learning with frequency-aware representation for contact-rich manipulation. In *IEEE International Conference on Robotics and Automation*, pages 22127–22134, 2026.

[19] Jiuzhou Lei, Chang Liu, Yu She, Xiao Liang, and Minghui Zheng. Learning when to see and when to feel: Adaptive vision-torque fusion for contact-aware manipulation. *arXiv preprint arXiv:2604.01414*, 2026.

[20] Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. In *International Conference on Learning Representations*, 2026.

[21] Yang Li, Hongru Jiang, Junjie Xia, Hongquan Zhang, Jinda Du, Yunsong Zhou, Jia Zeng, Ce Hao, Jieji Ren, Qiaojun Yu, et al. Forcevla2: Unleashing hybrid force-position control with force awareness for contact-rich manipulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8911–8920, 2026.

[22] Yao Li, Peiyuan Tang, Wuyang Zhang, Chengyang Zhu, Yifan Duan, Weikai Shi, Xiaodong Zhang, Zijiang Yang, Jianmin Ji, and Yanyong Zhang. FAVLA: A force-adaptive fast-slow VLA model for contact-rich robotic manipulation. *arXiv preprint arXiv:2602.23648*, 2026.

[23] Fanqi Lin, Ruiqian Nai, Yingdong Hu, Jiacheng You, Junming Zhao, and Yang Gao. OneTwoVLA: A unified vision-language-action model with adaptive reasoning. In *International Conference on Learning Representations*, 2026.

[24] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, 2025.

[25] Wenhai Liu, Junbo Wang, Yiming Wang, Weiming Wang, and Cewu Lu. Forcemimic: Force-centric imitation learning with force-motion capture system for contact-rich manipulation. In *IEEE International Conference on Robotics and Automation*, pages 1105–1112. IEEE, 2025.

[26] Zhuoyang Liu, Jiaming Liu, Hao Chen, Jiale Yu, Ziyu Guo, Chengkai Hou, Chenyang Gu, Xiangju Mi, Renrui Zhang, Kun Wu, et al. LaST_0: Latent spatio-temporal chain-of-thought for robotic vision-language-action model. In *International Conference on Machine Learning*, 2026.

[27] Yunfan Lou, Yifan Ye, Yankai Fu, Jun Cen, Xiaowei Chi, Yaoxu Lyu, Peidong Jia, Sirui Han, Zhihe Lu, and Shanghang Zhang. Dream-tac: A unified tactile world action model for contact-rich robot manipulation. *arXiv preprint arXiv:2606.08737*, 2026.

[28] Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-H0.7: A latent world-action model from egocentric videos. *arXiv preprint arXiv:2605.00078*, 2026.

[29] Henrique Morimitsu, Xiao-Bin Zhu, Roberto Marcondes Cesar, Xiangyang Ji, and Xu-Cheng Yin. DPFlow: Adaptive optical flow estimation with a dual-pyramid framework. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17810–17820, 2025.

[30] NVIDIA. GR00T N1.5. `https://research.nvidia.com/labs/gear/gr00t-n1_5/`, 2025.

[31] Steven Oh, Jason Jingzhou Liu, Tony Tao, Philip Han, Kenneth Shaw, Satoshi Funabashi, Ruslan Salakhutdinov, and Deepak Pathak. FACTR 2: Learning external force sensing for commodity robot arms improves policy learning. In *Conference on Robot Learning*, 2026.

[32] Carlota Parés-Morlans, Nils Kuhn, Isabel Liu, Alberta Longhini, and Jeannette Bohg. Demystifying when and why vlas fail in contact-rich tasks and how to fix them. In *Conference on Robot Learning*, 2026.

[33] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. In *Robotics: Science and Systems*, 2025.

[34] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, 2024.

[35] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. SpatialVLA: Exploring spatial representations for visual-language-action model. In *Robotics: Science and Systems*, 2025.

[36] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, pages 627–635, 2011.

[37] Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. MemoryVLA: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. In *International Conference on Learning Representations*, 2026.

[38] Haoming Song, Delin Qu, Yuanqi Yao, Qizhi Chen, Qi Lv, Yiwen Tang, Modi Shi, Guanghui Ren, Maoqing Yao, Bin Zhao, et al. Hume: Introducing system-2 thinking in visual-language-action model. *arXiv preprint arXiv:2505.21432*, 2025.

[39] Yue Su, Sijin Chen, Haixin Shi, Mingyu Liu, Zhengshen Zhang, Ningyuan Huang, Weiheng Zhong, Zhengbang Zhu, Yuxiao Liu, and Xihui Liu. World guidance: World modeling in condition space for action generation. In *International Conference on Machine Learning*, 2026.

[40] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419, 2020.

[41] Shuai Tian, Yupeng Zheng, Yuhang Zheng, Songen Gu, Yujie Zang, Yuxing Qin, Weize Li, Haoran Li, Wenchao Ding, and Dongbin Zhao. VT-WAM: Visual-tactile world action model for contact-rich manipulation. *arXiv preprint arXiv:2607.02503*, 2026.

[42] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *International Conference on Learning Representations*, 2025.

[43] Marcel Torne, Karl Pertsch, Homer Walke, Kyle Vedder, Suraj Nair, Brian Ichter, Allen Z Ren, Haohuan Wang, Jiaming Tang, Kyle Stachowicz, et al. MEM: Multi-scale embodied memory for vision language action models. In *Conference on Robot Learning*, 2026.

[44] Han Xue, Jieji Ren, Wendi Chen, Gu Zhang, Yuan Fang, Guoying Gu, Huazhe Xu, and Cewu Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. In *Robotics: Science and Systems*, 2025.

[45] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

[46] Jiawen Yu, Hairuo Liu, Qiaojun Yu, Jieji Ren, Ce Hao, Haitong Ding, Guangyu Huang, Guofan Huang, Yan Song, Panpan Cai, et al. ForceVLA: Enhancing vla models with a force-aware moe for contact-rich manipulation. In *Advances in Neural Information Processing Systems*, 2025.

[47] Yujie Zang, Yuhang Zheng, Xiangli Nie, Yupeng Zheng, Shuai Tian, Songen Gu, Chen Gao, Zining Wang, Shuicheng Yan, and Wenchao Ding. TacForeSight: Force-guided tactile world model for contact-rich manipulation. *arXiv preprint arXiv:2606.11184*, 2026.

[48] Jia Zeng, Qingwen Bu, Bangjun Wang, Wenke Xia, Li Chen, Hao Dong, Haoming Song, Dong Wang, Di Hu, Ping Luo, et al. Learning manipulation by predicting interaction. In *Robotics: Science and Systems*, 2024.

[49] Chaofan Zhang, Peng Hao, Xiaoge Cao, Xiaoshuai Hao, Shaowei Cui, and Shuo Wang. VTLA: Vision-tactile-language-action model with preference learning for insertion manipulation. *arXiv preprint arXiv:2505.09577*, 2025.

[50] Kaidi Zhang, Heng Zhang, Zhengtong Xu, Zhiyuan Zhang, Md Rakibul Islam Prince, Xiang Li, Xiaojing Han, Yuhao Zhou, Arash Ajoudani, and Yu She. TacVLA: Contact-aware tactile fusion for robust vision-language-action manipulation. *arXiv preprint arXiv:2603.12665*, 2026.

[51] Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, et al. DreamVLA: a vision-language-action model dreamed with comprehensive world knowledge. In *Advances in Neural Information Processing Systems*, 2025.

[52] Zongzheng Zhang, Haobo Xu, Zhuo Yang, Chenghao Yue, Zehao Lin, Huan-ang Gao, Ziwei Wang, and Hao Zhao. TA-VLA: Elucidating the design space of torque-aware vision-language-action models. In *Conference on Robot Learning*, 2025.

[53] Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. FLARE: Robot learning with implicit world modeling. In *Conference on Robot Learning*, 2025.

[54] Yuhang Zheng, Songen Gu, Weize Li, Yupeng Zheng, Yujie Zang, Shuai Tian, Xiang Li, Ce Hao, Chen Gao, Si Liu, et al. Omnivta: Visuo-tactile world modeling for contact-rich robotic manipulation. *arXiv preprint arXiv:2603.19201*, 2026.

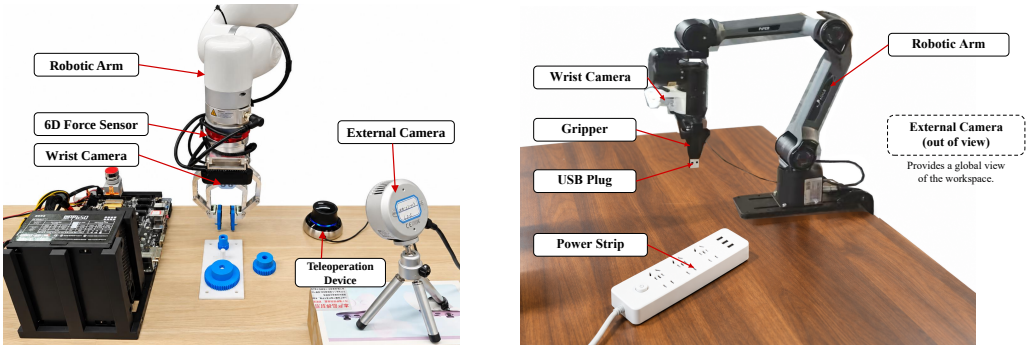
[55] Zhide Zhong, Haodong Yan, Junfeng Li, Xiangchen Liu, Xin Gong, Tianran Zhang, Wenxuan Song, Jiayi Chen, Xinhu Zheng, Hesheng Wang, and Haoang Li. Flowvla: Visual chain of thought-based motion reasoning for vision-language-action models. *arXiv preprint arXiv:2508.18269*, 2025.

[56] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Robotics: Science and Systems*, 2025.

[57] Xinyue Zhu, Binghao Huang, and Yunzhu Li. Touch in the wild: Learning fine-grained manipulation with a portable visuo-tactile gripper. In *Advances in Neural Information Processing Systems*, 2025.

[58] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183, 2023.

A Detailed Experimental Setup



(a) xArm 7 (b) Piper-X

Figure 6: Real-world experimental setups for (a) the main xArm 7 experiments and (b) the supplementary Piper-X evaluation.

A.1 Robot Platform

We conduct most experiments on xArm 7 and further evaluate our method on Piper-X with a different system configuration, as shown in Figure 6. The main setup uses a 7-DoF xArm 7 equipped with a wrist-mounted 6D force/torque sensor, a third-person RealSense L515 camera, and an Orbbec Gemini 336L wrist camera, using π_0 as the policy backbone. The supplementary setup uses a 6-DoF AgileX Piper-X with joint-torque estimates derived from motor currents, a RealSense D435 camera, and an Orbbec Gemini 335 camera, with GR00T N1.5 as the policy backbone. Table 6 compares their configurations.

Table 6: Comparison of experimental setups across robot platforms.

Component	xArm 7	Piper-X
DoF	7	6
Force input	6D F/T sensor	Joint torque
Cameras	RealSense L515 + Gemini 336L	RealSense D435 + Gemini 335
Policy backbone	π_0	GR00T N1.5

A.2 Data Collection

We collect demonstrations through 3D SpaceMouse teleoperation, recording synchronized camera observations, robot states, force/torque measurements, and actions. To improve data diversity, we vary the initial robot pose and include demonstrations that recover from failed attempts. For example, in USB insertion, the operator deliberately misaligns the plug with the port and then demonstrates corrective motions to complete the insertion, instead of always starting from an aligned pose. Approximately 20–30% of the demonstrations include such recovery behaviors. For each robot platform, we combine demonstrations from all tasks into a single multi-task dataset for policy training. Dataset statistics for both robot platforms are summarized in Table 7.

A.3 Evaluation Protocol

For each task, the robot starts from a fixed initial pose, while the object is placed at one of a predefined set of evaluation locations. Each method is evaluated on the same set of object locations with the same number of trials. For the main in-distribution evaluation on xArm 7, each method is evaluated over 25 trials on USB Insertion, 30 trials on Gear Assembly, and 20 trials each on Box Flipping, Board Wiping, and VGA Plugging. A trial is considered successful only if the task is completed within 2 minutes and the contact force remains below the safety threshold of 25 N. Otherwise, the

Table 7: Dataset statistics for the xArm 7 and Piper-X robot platforms.

Robot Platform	Task	Demos	Frames	Duration (h)
xArm 7	USB Insertion	100	55,667	0.52
	Gear Assembly	80	64,091	0.59
	Box Flipping	80	71,795	0.66
	Board Wiping	80	70,810	0.66
	VGA Plugging	80	69,874	0.65
	Total	**420**	**332,237**	**3.08**
Piper-X	USB Insertion	100	48,438	0.45
	Charger Insertion	100	41,651	0.39
	Total	**200**	**90,089**	**0.83**

trial is counted as a failure. The success rate is calculated as the number of successful trials divided by the total number of evaluation trials.

A.4 Task Descriptions

We describe each task and its key challenges below, with representative successful rollouts provided in Appendix F.

- **USB Insertion.** The robot must align a USB plug with the target port on a power strip and insert it while adjusting its position and orientation. This task requires precise alignment due to the tight clearance between the plug and the port. The task is successful when the plug is fully inserted into the port.
- **Gear Assembly.** The robot must pick up a gear, align its center hole with the target shaft, and move it downward to complete the assembly. Since the hole is occluded during alignment, the robot must rely on force feedback to adjust its position. The task is successful when the gear is fully mounted on the shaft.
- **Box Flipping.** The robot must flip a box from a flat position to an upright position using a fixed support. During flipping, the robot must continuously adjust its motion based on force feedback to prevent the box from slipping or excessive force from being applied. The task is successful when the box stands upright on its narrow face.
- **Board Wiping.** The robot must pick up an eraser and use it to wipe the marked region of a board. During wiping, the robot must maintain sufficient contact force to effectively remove the markings. The task is successful when all markings in the target region are removed.
- **VGA Plugging.** The robot must pick up a VGA connector, align it with the target socket, and insert it horizontally. The irregular shape of the connector requires precise alignment and force-guided position adjustments during insertion. The task is successful when the connector is fully inserted into the socket.

B Model Architecture Details

Backbone. We use the pretrained PaliGemma backbone from π_0 [5], comprising a SigLIP image encoder and a Gemma-2B language model with approximately 2.92B parameters in total. It takes two RGB images from the external and wrist cameras, each resized with padding to 224×224, together with the task instruction. The image and instruction tokens form a shared visual–language prefix for both action experts.

Teacher and Student Experts. Both experts use the same approximately 300M-parameter architecture with 18 Transformer layers and hidden width 1024. Their Transformer weights are initialized from the same pretrained π_0 action expert and then optimized separately within joint training. Both receive the shared prefix, robot state, and force history, and predict 32-step action chunks through flow matching. The teacher additionally receives encoded future force and motion tokens, while the student uses learnable future queries.

Force Encoding. We flatten ten historical 6D force/torque measurements and use a one-layer MLP to produce one force token for each expert. A separate one-layer MLP encodes the teacher's 32-step future force sequence into one token of the same width.

Future Motion Encoding. RAFT [40] estimates optical flow between the current and future images for both views. We convert each flow field to RGB by mapping direction to hue and using normalized, clipped magnitude to control color strength against a white background. A frozen SDXL-VAE [34] encodes these motion images. After 2×2 latent patch merging, a two-layer MLP and layer normalization project the spatial features to width 1024. A Q-Former-style compressor with 16 learnable queries aggregates the features from both views through cross-attention, producing 16 motion tokens.

Student Future Queries. The student uses 17 learnable queries of width 1024: one for future force and 16 for future motion, matching the teacher's privileged tokens. Each expert concatenates its force-history token, state token, future-related tokens, and noisy action tokens. Future queries attend to causal observations but not to noisy action tokens.

Representation Alignment. We align future-related hidden states at layers $\{8, 12, 16\}$ using separate two-layer force and motion projectors with Swish activations. The projected student states are matched to stop-gradient teacher targets using cosine distance.

C Training Hyperparameters

ForceLDM contains approximately 3.56B trainable parameters, including a 2.92B vision–language backbone and two 311M action experts. We jointly fine-tune the backbone, both action experts, and the additional modules, while keeping the RAFT and SDXL-VAE preprocessing models frozen. The training hyperparameters are summarized in Table 8.

Table 8: Training hyperparameters for ForceLDM.

Parameter	Value
Optimization	
Training hardware	1 NVIDIA A100, 80 GB
Training steps	30,000
Batch size	16
Optimizer	AdamW
Peak learning rate	2.5×10^{-5}
Final learning rate	2.5×10^{-6}
Learning-rate schedule	Cosine decay
Learning-rate warmup	1,000 steps
Adam (β_1, β_2)	$(0.9, 0.95)$
Gradient clipping (global norm)	1.0
Distillation	
Alignment layers	$\{8, 12, 16\}$
Distillation weight λ	0.5
Motion / force queries	16 / 1
Query Perturbation	
Maximum query noise scale $\sigma_{\max}$	0.3
Noise ramp start p_{start}	0.3
Noise ramp end p_{end}	0.7

D Baseline Implementation Details

We implement all baselines except ForceVLA within our π_0-based framework, following their official open-source implementations. All methods are trained on the same task datasets for 30,000 optimization steps with a batch size of 16, using the optimizer settings in Table 8. For ForceVLA, we use its official codebase to preserve the original fusion architecture.

π_0 w/o Force. The base π_0 [5] combines a PaliGemma vision–language backbone with a Transformer action expert. The backbone encodes the two camera views and language instruction, while the action expert processes robot-state and noisy-action tokens conditioned on the vision–language context. It predicts 32-step action chunks through flow matching without force input.

π_0 w/ Force. We retain the original backbone, action expert, and flow-matching objective, while introducing an MLP to encode the force history into a single token. This force token is concatenated with the state and noisy-action tokens as input to the action expert. The model thus incorporates force feedback without explicitly modeling future interaction dynamics.

TA-VLA. Following TA-VLA [52], we encode the historical force sequence using a two-layer MLP and feed the resulting force token into the action expert alongside the state and noisy-action tokens. Unlike π_0 w/ Force, this baseline extends the flow-matching objective to jointly predict future action chunks and force sequences. The additional force-prediction loss encourages the model to learn force-related information during action generation. We set the auxiliary force-prediction loss weight to 0.1.

ForceVLA. ForceVLA [46] extends π_0 with FVLMoE, a force-aware Mixture-of-Experts fusion module that dynamically integrates pretrained vision–language features with real-time 6-axis force feedback during action decoding. The module uses modality-specific experts to adaptively fuse multimodal information, enabling force-aware action prediction. We train ForceVLA on our demonstrations using its official implementation.

Seer w/ Force. Seer [42] introduces visual foresight to guide action prediction through future visual representations. We adapt this design to π_0 by introducing ten learnable foresight tokens per camera view to extract future visual features from the vision–language backbone. These features are directly provided to the action expert for flow-matching-based action prediction, while a separate masked image decoder reconstructs future RGB images under an MSE loss. We additionally incorporate historical force observations and jointly optimize action prediction and future RGB reconstruction. The force history is encoded by a two-layer MLP into a single token, which is concatenated with the state and noisy-action tokens as input to the action expert. We set the RGB reconstruction loss weight to 0.1.

E Additional Experiments

E.1 Evaluation on Another Robot Platform

To examine whether our method can be applied to different robot platforms and VLA architectures, we conduct additional real-world experiments on an alternative robotic platform. Compared with our main experiments using xArm 7 and π_0, these experiments employ a different robot, force-sensing method, camera configuration, and VLA backbone.

Experimental Setup. We use an AgileX Piper-X manipulator and adopt GR00T N1.5 [30] as the VLA policy backbone. The force-sensing and camera configurations are described in Appendix A and compared with the main xArm 7 setup in Table 6. To implement ForceLDM on this backbone, we duplicate the policy's action-decoding head to construct the teacher and student experts, which are trained using the same future-dynamics distillation framework as in the main experiments. We consider two tasks: inserting a USB plug and inserting a charger plug into their corresponding sockets on a power strip. We use the separate Piper-X dataset described in Appendix A.2, with 100 demonstrations per task (200 in total), for platform-specific policy adaptation. We evaluate each method over 20 trials per task. All compared methods use the same task initialization and evaluation protocol.

Results. Table 9 presents the success rates on two real-world manipulation tasks using Piper-X. ForceLDM achieves 80% on USB insertion and 60% on charger insertion, averaging 70% and outperforming GR00T N1.5 and TA-VLA by 50 and 7.5 percentage points, respectively. In contrast, directly adding torque observations reduces GR00T N1.5's average success rate to 17.5%, potentially due to noisy torque signals and the dominance of visual features. By modeling action-relevant future

interaction dynamics, ForceLDM better leverages force feedback for precise manipulation. These results demonstrate the effectiveness of our method on a different robot platform and VLA backbone.

Table 9: Real-world manipulation success rates (%) on the Piper-X platform. Each method is evaluated over 20 trials per task, with the average success rate reported across both tasks.

Method	**USB Insertion**	**Charger Insertion**	**Avg.**
GR00T N1.5	15.0	25.0	20.0
GR00T N1.5 + Torque Input	25.0	10.0	17.5
TA-VLA	70.0	55.0	62.5
ForceLDM (Ours)	**80.0**	**60.0**	**70.0**

E.2 Sensitivity to the Prediction Horizon

The prediction horizon determines how far into the future ForceLDM learns to model interaction dynamics. To investigate its effect, we evaluate three horizons $H \in \{16, 32, 50\}$ on Box Flipping and Gear Assembly under identical settings. Specifically, H controls both the action-chunk length and the future-prediction horizon, where the model predicts H future force steps and optical flow between I_t and I_{t+H}. As shown in Table 10, $H = 16$ and $H = 32$ achieve comparable performance, while increasing the horizon to $H = 50$ substantially reduces success rates. This suggests that excessively long horizons may introduce greater uncertainty in future interaction dynamics and make precise contact control more challenging. We therefore adopt $H = 32$, which achieves the highest average success rate.

Table 10: Effect of the prediction horizon on task success rates (%).

Horizon H	**Box Flipping**	**Gear Assembly**	**Average**
16	100.0	86.7	93.4
32	**100.0**	**90.0**	**95.0**
50	90.0	70.0	80.0

E.3 Sensitivity to Optical-Flow Quality

ForceLDM learns future interaction dynamics from estimated optical flow, whose quality may vary across different estimators. To examine whether our method relies on a specific flow estimator, we compare RAFT (default) and DPFlow [29] on USB and charger insertion using the Piper-X and GR00T N1.5 setup described in Appendix E.1, while keeping all other settings fixed. We include GR00T N1.5 and TA-VLA as baselines, using the results reported in Table 9. As shown in Table 11, replacing RAFT with DPFlow changes the success rate from 80% to 75% on USB insertion and from 60% to 70% on charger insertion. The comparable performance suggests that our method is not strongly dependent on the choice of flow estimator, as both provide useful motion cues for learning future interaction dynamics.

Table 11: Comparison of optical-flow estimators on real-world manipulation success rates (%) using Piper-X.

Configuration	**USB Insertion**	**Charger Insertion**	**Avg.**
GR00T N1.5	15.0	25.0	20.0
TA-VLA	70.0	55.0	62.5
ForceLDM (RAFT)	**80.0**	60.0	70.0
ForceLDM (DPFlow)	75.0	**70.0**	**72.5**

F Visualization

We provide representative successful executions of ForceLDM under both in-distribution (ID) and out-of-distribution (OOD) conditions.

Figure 7 illustrates successful executions of five real-world manipulation tasks. These tasks involve different forms of contact-rich interactions, including precise insertion, gear assembly, object flipping, and surface wiping. ForceLDM successfully completes these tasks by handling contact interactions that require precise alignment and force control. In particular, the insertion and assembly tasks require accurate positioning under tight clearances, while flipping and wiping require appropriate contact forces throughout the manipulation process. These examples highlight the effectiveness of our method in contact-rich manipulation.

Figure 8 further presents successful executions under unseen conditions, including changes in object appearance, material, size, and assembly height. Despite these variations, ForceLDM successfully completes the tasks, demonstrating its ability to handle different object properties and spatial configurations. These results further suggest that learning action-relevant future interaction dynamics helps the policy maintain effective manipulation performance beyond the training conditions.

G Failure Case Analysis

Figure 9 presents representative failure cases from our real-world experiments. We identify three common failure modes.

- **Repeated Attempts.** In USB Insertion and VGA Plugging (Figure 9(a,d)), the robot may repeatedly attempt insertion without successfully completing the task. In some cases, the robot releases the connector before it is fully inserted, resulting in task failure.
- **Unstable or Insufficient Contact.** In Box Flipping (Figure 9(c)), insufficient contact force may cause the box to slip during flipping. Similarly, in Board Wiping (Figure 9(e)), the robot may fail to maintain sufficient force to remove all markings from the board.
- **Grasping Failures.** In Gear Assembly (Figure 9(b)), the robot occasionally closes its gripper before reaching the gear, resulting in a failed grasp. This may arise from inaccurate estimation of the distance between the gripper and the gear due to limited depth information.

Similar failure modes are also observed in the baseline methods, indicating that precise contact control and reliable grasping remain challenging in real-world manipulation.

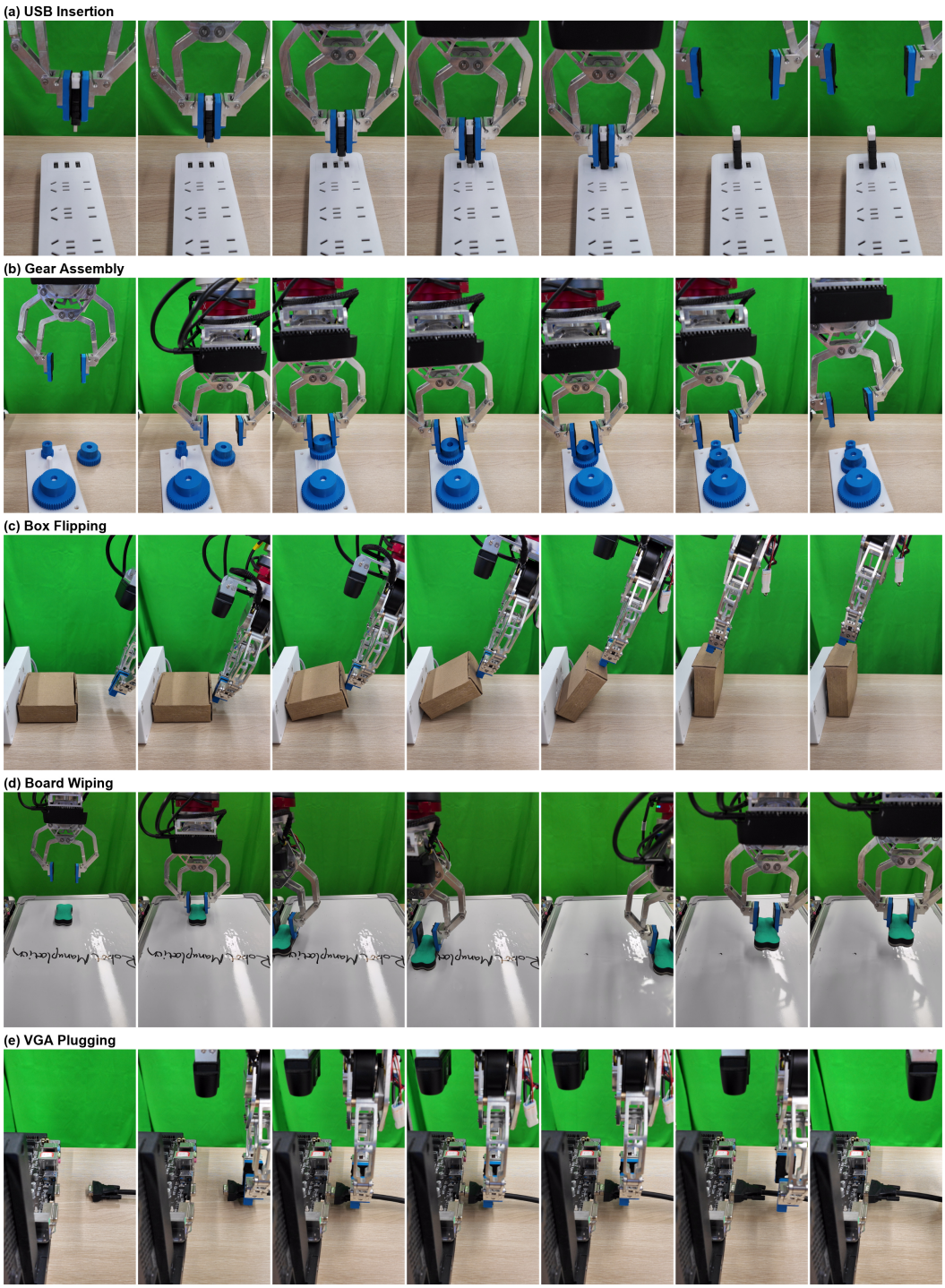


Figure 7: Successful executions of five real-world manipulation tasks: USB Insertion, Gear Assembly, Box Flipping, Board Wiping, and VGA Plugging. Each row shows seven frames from the initial state to task completion.



Figure 8: Successful executions under OOD conditions: (a) an unseen USB connector, (b) an unseen gear color, (c) an increased assembly height, (d) an unseen box material, and (e) an unseen box size. Each row shows seven frames from the initial state to task completion.

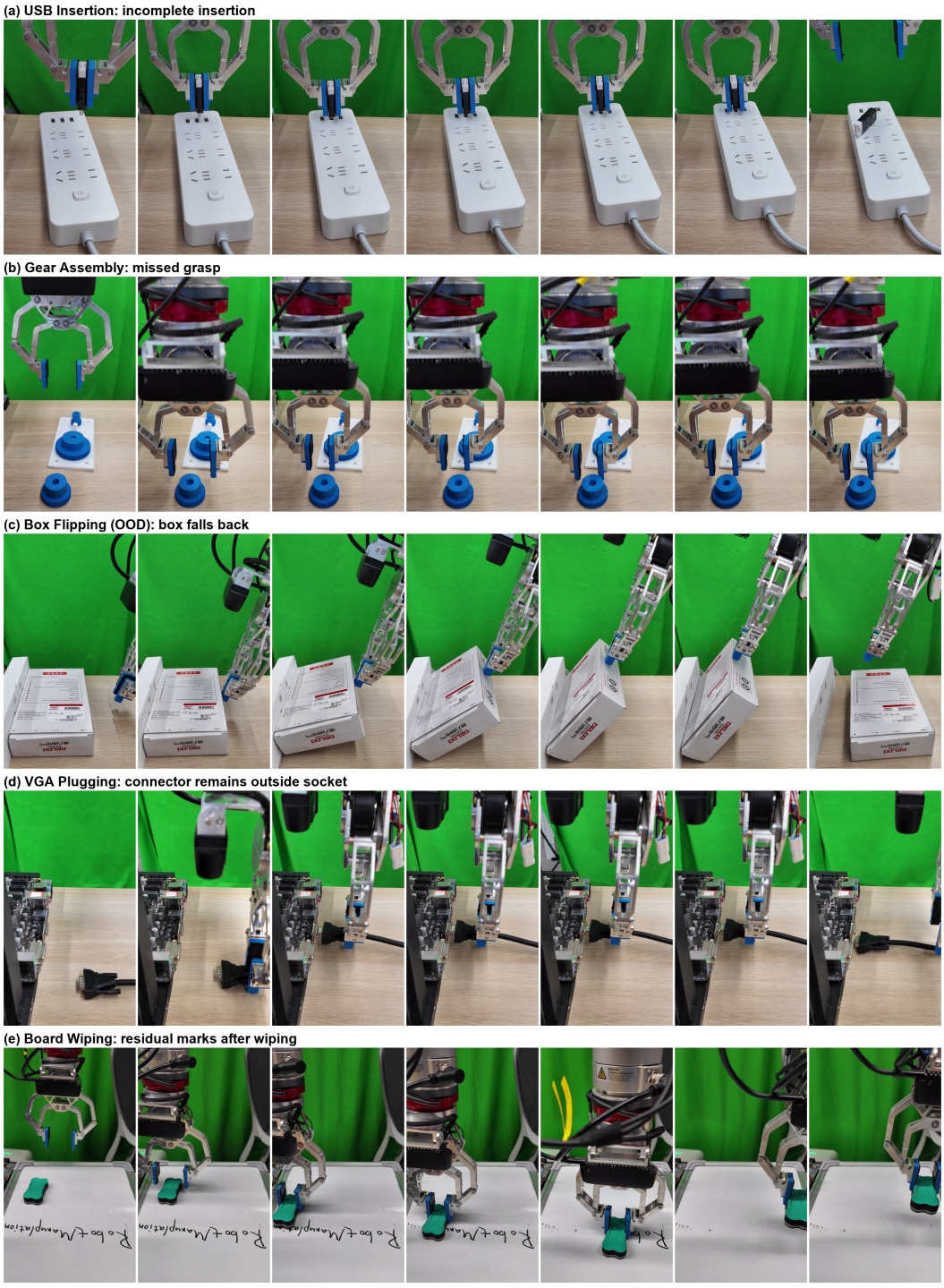


Figure 9: Representative failure cases: (a) the USB plug failing to fully insert into the target port, (b) a missed gear grasp, (c) a box falling back during flipping under OOD conditions, (d) the VGA connector released before being fully inserted, and (e) small residual marks after wiping. Each row shows seven frames from the initial state to the terminal failure state.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please see the summarized contributions in Section 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please see Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The paper should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not present theorems or formal proofs; it gives a model formulation and training objective only.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The method and training objectives are described in Section 3, and the experimental setup is described in Section 4. Detailed experimental protocols, model architecture, training hyperparameters, and baseline implementations are provided in Appendices A, B, C, and D, respectively. Code is also provided to support reproduction of the method.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code and data are publicly available through the project page at `https://tpy001.github.io/ForceLDM/`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setup is described in Section 4.1. Hardware, data collection, evaluation protocols, and task success criteria are detailed in Appendix A. Model architecture, training hyperparameters, and baseline implementations are provided in Appendices B, C, and D, respectively.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports success rates over multiple real-world rollouts, but does not include confidence intervals, error bars, or statistical significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The training hardware and main optimization settings are described in Section 4.1. Detailed training hyperparameters and baseline training settings are provided in Appendices C and D, respectively.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The manuscript describes a robotics method evaluated with teleoperated robot demonstrations and does not involve prohibited data collection, deception, or undisclosed human-subject experimentation.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

Justification: This work focuses on improving contact-rich robotic manipulation, and we do not anticipate specific negative societal impacts arising from the proposed method. We therefore do not include a separate discussion of broader impacts.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The released code and dataset do not pose a high risk of misuse requiring special safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No]

Justification: We cite the existing models and methods used in our work, but their licenses and terms of use are not explicitly documented in the manuscript.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released code and dataset are documented and made publicly available through the project page at `https://tpy001.github.io/ForceLDM/`.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not include crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work does not involve human-subject research.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as part of the core methodology. The proposed method, experiments, and conclusions do not depend on LLM-generated components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.