DreamVLA: A Vision-Language-Action Model Dreamed with Comprehensive World Knowledge

Wenyao Zhang[124*] **Hongsi Liu**[27*] **Zekun Qi**[34*] **Yunnan Wang**[12*]
Xinqiang Yu[4] **Jiazhao Zhang**[45] **Runpei Dong**[6] **Jiawei He**[4]
He Wang[45] **Zhizheng Zhang**[4] **Li Yi**[3] **Wenjun Zeng**[2] **Xin Jin**[2‡]

[1]SJTU [2]EIT [3]THU [4]Galbot [5]PKU [6]UIUC [7]USTC

Project Page Code 🤗 Hugging Face

Abstract

Recent advances in vision-language-action (VLA) models have shown promise in integrating image generation with action prediction to improve generalization and reasoning in robot manipulation. However, existing methods are limited to challenging image-based forecasting, which suffers from redundant information and lacks comprehensive and critical world knowledge, including dynamic, spatial and semantic information. To address these limitations, we propose DreamVLA, a novel VLA framework that integrates comprehensive world knowledge forecasting to enable inverse dynamics modeling, thereby establishing a perception-prediction-action loop for manipulation tasks. Specifically, DreamVLA introduces a dynamic-region-guided world knowledge prediction, integrated with the spatial and semantic cues, which provide compact yet comprehensive representations for action planning. This design aligns with how humans interact with the world by first forming abstract multimodal reasoning chains before acting. To mitigate interference among the dynamic, spatial and semantic information during training, we adopt a block-wise structured attention mechanism that masks their mutual attention, preventing information leakage and keeping each representation clean and disentangled. Moreover, to model the conditional distribution over future actions, we employ a diffusion-based transformer that disentangles action representations from shared latent features. Extensive experiments on both real-world and simulation environments demonstrate that DreamVLA achieves **76.7%** success rate on real robot tasks and **4.44** average length on the CALVIN ABC-D benchmarks.

1 Introduction

The evolution of robot learning has demonstrated impressive progress [1–13] in training policies capable of performing diverse tasks across various environments [14–27]. One promising direction is Vision-Language-Action (VLA) models, which leverage the rich understanding capabilities of pre-trained Multimodal Large Language Models (MMLMs) [28–31] to directly map natural language instructions and visual observations to robot actions [17, 1, 14]. Although these approaches [32–34, 15, 1, 35–44] have achieved impressive results, their direct mapping from observations to actions lacks the closed-loop forecasting capability that humans typically possess when understanding and reasoning about future knowledge of environments.

To incorporate future knowledge prediction into VLA, most existing methods [45, 5, 46–57] leverage a copilot generation model to generate future frames/keypoints, then predict action sequences conditioned on goal images. Several methods [58–63] integrate pixel-level image forecasting with the

[*] Equal contribution. [‡] Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

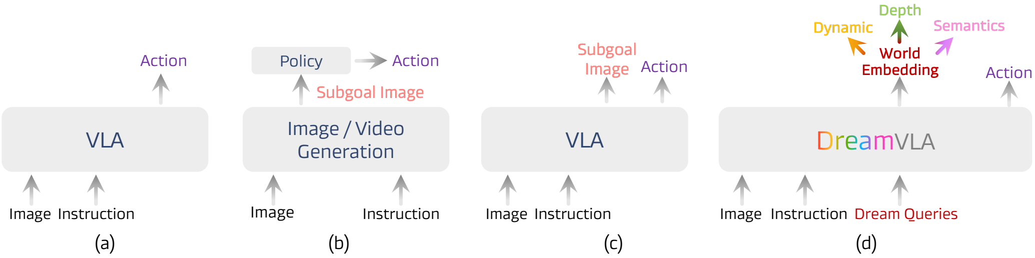

Figure 1: (a) Vanilla VLA directly maps visual observations and language instructions to actions. (b) Models leveraging separate image/video generation or copilot models to generate future frames or trajectories, subsequently guiding an action head. (c) VLA variants explicitly predict a subgoal image as an intermediate visual reasoning step prior to action generation. (d) Our proposed **DreamVLA**, which explicitly predicts dynamic regions, depth map, semantics (DINOv2 and SAM) knowledge, significantly enhances the model's action reasoning and generalization.

action prediction in a single framework, which exploits the synergy of prediction and planning and regards the prediction as an intermediate reasoning step [60] akin to those used in large language models (LLMs) [64]. Despite early success in incorporating dense visual forecasting, these methods naturally exhibit limitations: (1) *Redundant pixel information*: There exists significant overlap between forecasted images and current observations, making the prediction less efficient and effective. (2) *Lack of spatial information*: Absence of explicit 3D knowledge of environments [65–68, 24]. (3) *Lack of high-level knowledge forecasting*: Missing high-level understanding of future states, *e.g.*, semantics information. Therefore, we argue that existing methods (Figure 1 (a-c)) are insufficient to forecast future states for a more comprehensive prediction-action loop in the context of world-level future knowledge.

To address these issues, we propose DreamVLA, a novel framework that incorporates comprehensive world knowledge forecasting into the vision-language-action models, thereby establishing a perception-prediction-action loop for the manipulation task. As shown in Figure 1 (d), instead of directly generating entire future frames, our proposed method introduces *world embedding* to predict comprehensive world knowledge, which is highly relevant to robot execution, such as dynamic area, depth, and high-level semantic features. This approach aligns with the way humans interact with the world, emphasizing relevant changes and world knowledge. By dreaming/forecasting these targeted aspects of the environment, we aim to provide the model with concise and relevant intermediate representations that facilitate more effective action planning.

To obtain comprehensive world knowledge, our approach incorporates three key features: (1) *Dynamic region-based forecasting*. We leverage an off-the-shelf optical flow prediction model [69, 70] to identify dynamic regions within the scene, enabling the model to concentrate on areas of motion that are critical for task execution instead of redundant frame reconstruction. (2) *Depth-aware forecasting*. We employ depth estimation techniques [65] to generate per-frame depth maps, providing valuable spatial context that aids in understanding the three-dimensional structure of the environment. (3) *High-level foundation features*. We incorporate semantic features aligned with visual foundation models such as DINOv2 [71] and SAM [72]. In this way, DreamVLA offers a more comprehensive and effective pathway for the model to plan and execute. Furthermore, we adopt a block-wise structured attention mechanism that masks their mutual attention, preventing information leakage and keeping each representation clean and disentangled. Since the world and action embeddings occupy the same latent space and share similar statistics, a naive MLP head cannot disentangle modality-specific information or exploit their cross-modal correlations. We employ a diffusion-based transformer that disentangles action representations from shared latent features to reason actions.

Through extensive experiments on public benchmarks, we find that incorporating world knowledge prediction leads to significant performance improvements. Our method achieves state-of-the-art performance on the CALVIN benchmark (**4.44** average length), and we analyze the influence of the ingredients of our world knowledge and find that they have improvements in different aspects. Specifically, comprehensive ablation shows that predicting dynamic regions alone delivers the greatest gains, while depth and semantic cues offer smaller, roughly equal benefits. Worse, when depth or semantic prediction is used in isolation, it not only fails to help but can actually degrade performance. Extensive experiments on both simulation and real-world demonstrate the effectiveness of our method.

The key contributions of our work are summarized as follows:

- We recast the vision–language–action model as a perception–prediction–action model and make the model explicitly predict a compact set of dynamic, spatial and high-level semantic information, supplying concise yet comprehensive look-ahead cues for planning.
- We introduce a block-wise structured-attention mechanism, coupled with a diffusion-transformer decoder, to suppress representation noise from cross-type knowledge leakage and thus enable coherent multi-step action reasoning.
- DreamVLA sets a new state of the art on the CALVIN ABC-D benchmark (4.44 average task length), outperforming prior methods by up to 3.5% on the simulation platform, and boosts real-world success to 76.7%. Ablation studies confirm each component's contribution.

2 Related Works

2.1 Vision–Language–Action Models

The earliest VLA [18, 73, 2, 74–76] lay the foundation by combining pretrained vision-language representations with task-conditioned policies for manipulation and control. Inspired by the recent advances of Large Language Models [77–80] and multimodal large language models [30, 28, 81, 67, 82] and the emergence of large-scale robot datasets [14, 83–85], VLA has become a trend in robot learning. RT series [2, 86, 87] is the pioneer attempt to fine-tune the MLLM on robot demonstration datasets, resulting in strong accuracy and generalization. Building on this foundation, many advanced techniques [32, 34, 15, 1, 35, 36, 75, 37–39, 88–90, 40, 91] are developed to boost the performance. Meanwhile, considering the advantage of the diffusion model in modeling multi-peak, some researchers [92–96] employ different architectures to sample action from noise conditioned on observation, task instruction, and robot prior knowledge. Given on this manner which directly maps observation and instruction to action lacks reasoning steps like LLM [64], most existing methods [45, 5, 46–51] leverage a copilot image/video generation model to generate future frames then predict action sequences conditioned on goal images. However, the above methods still need an extra generation model, which introduces inference time and computation load. Therefore, several methods [58–63] integrate pixel-level forecasting with the action prediction in a single framework, which exploits the synergy of prediction and planning. Despite success, these methods naturally exhibit limitations in redundant reconstruction [97], and lack spatial and semantic information.

2.2 Knowledge Forecasting for Robotics

Learning future world knowledge for robot training has increasingly become popular to enable policies for achieving an action-forecasting loop. Early attempts [51, 21, 16, 45, 53, 52, 98] to implement this based on off-the-shelf video generation models [99, 55] and feed the goal images or states into policy model to conduct inverse dynamics. This two-stage training strategy is easy to implement but is limited by the performance and latency of video generation models. More advanced solutions couple forecasting with control by requiring the policy to produce, in addition to actions, explicit predictions. Concretely, these works ask the policy to output (i) high-level subtask/option sequences or language plans that decompose long-horizon goals [100–102], (ii)latent future embeddings/latent actions that compactly encode forthcoming motor intentions [90], (iii)whole sub-goal images or short visual rollouts that anticipate how the scene should evolve [58, 60], and (iv) object-centric signals (e.g., bounding boxes) that capture manipulation-relevant dynamics [85, 89]. This line of work demonstrates better performance and generalization. However, the future states are limited to redundant visual information [65, 66, 103, 71, 104, 68] or monotonous states [23, 50]. In contrast to previous work, DreamVLA proposes to predict future knowledge in an efficient (dynamic region) and effective (comprehensive knowledge) way, demonstrating strong performance and generalization.

3 Methodology

3.1 Problem Definition and Notation

We aim to improve robot execution by leveraging rich world knowledge as a guiding principle. In this context, we formulate vision–language–action reasoning as an *inverse dynamics* problem [105, 58, 51],

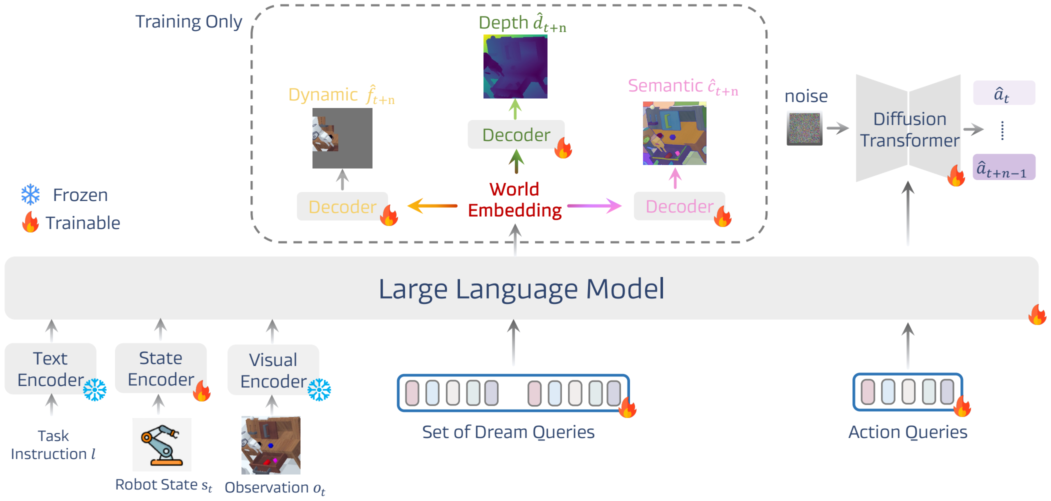

Figure 2: **Framework Overview**. Given the current robot state s_t, observation o_t, and language instruction, DreamVLA encodes multimodal inputs via frozen text, visual encoders and a tunable state encoder. These tokens, together with a learnable set of <dream> queries, are processed by a large language model to produce *world embedding*. Three lightweight decoders then project each corresponding element of this embedding into the dynamics region $\hat{f}_{t+n}$, monocular depth $\hat{d}_{t+n}$ and high-level semantics $\hat{c}_{t+n}$. A separate <action> query draws a latent action embedding, which conditions a diffusion transformer that refines Gaussian noise into an n-step action sequence $\hat{a}_{t:t+n-1}$. The dashed box highlights prediction heads that are used only during training; inference skips these heads and operates directly on the world embedding.

which regards the future world knowledge prediction as the intermediate reasoning for robot control, fully unleashing the synergy of prediction and execution. At each time step t, the robot receives three heterogeneous signals: a natural language instruction l, a raw visual frame o_t, and its proprioceptive state s_t. To inject look-ahead reasoning, we define a set of special tokens called <dream> queries [81], and concatenate all inputs into a sequence. A unified model $\mathcal{M}$ maps these inputs into a compact latent representation, which we call the *world embedding*:

$$\mathbf{w}_{t+n} = \mathcal{M}\left(l, o_t, s_t | \texttt{<dream>}\right). \tag{1}$$

Next, the world embedding predicts the comprehensive world knowledge that combines motion cues, spatial details and high-level semantics. Specifically, a set of predictor $\mathcal{P}$ extrapolates n steps ahead,

$$\hat{p}_{t+n} = \mathcal{P}\big(\mathbf{w}_{t+n}\big) = \big[\hat{f}_{t+n}, \hat{d}_{t+n}, \hat{c}_{t+n}\big], \tag{2}$$

where $\hat{f}_{t+n}$ marks dynamic regions, $\hat{d}_{t+n}$ encodes monocular depth, and $\hat{c}_{t+n}$ optionally stores high-level semantic feature (e.g. DINOv2 [71], SAM [72]).

Given *world embedding* $\mathbf{w}_{t+n}$, the <action> query is assigned to the latent action embedding by the unified model $\mathcal{M}$ to aggregate the correlated action information. A denoising-diffusion transformer $\mathcal{D}$ formulates an n-step action based on the latent feature:

$$\hat{a}_{t:t+n-1} = \mathcal{D}(\mathcal{M}\big(l, o_t, s_t, \texttt{<dream>}|\texttt{<action>}\big)), \tag{3}$$

thus completing a perception–prediction–action loop that is identical during training and inference. The remainder of this chapter details the system components—encoders, world-knowledge predictor, and diffusion-based action generator—that instantiate the above formulation.

3.2 Model Architecture

As illustrated in Figure 2, our DreamVLA framework comprises three core modules operating within a unified transformer architecture. Firstly, heterogeneous inputs—including natural language l, visual observations o_t, and proprioceptive states s_t—are individually processed by modality-specific encoders. We encode language instructions using CLIP [103] text embeddings, visual frames through a Masked Autoencoder [106] to obtain spatiotemporal patch representations, and proprioceptive signals via several convolutional and fully-connected layers. Following encoding, a set of learnable

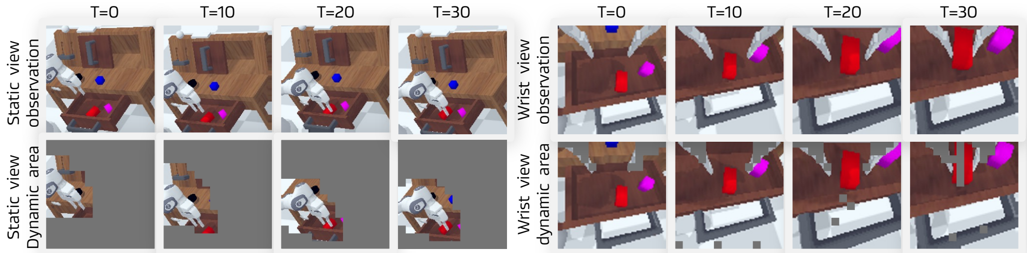

Figure 3: **Visualization of dynamic regions over time.** We show the static camera (left) and wrist-mounted camera (right) observations alongside the corresponding dynamic masks generated by our method at multiple time steps. The masks highlight dynamic regions by leveraging optical flow trajectories extracted via CoTracker [70, 69]. Compared to the original observations, our method effectively suppresses irrelevant background and focuses on interaction-relevant areas (e.g., moving objects and end-effector), enabling more structured and efficient action reasoning.

queries designated as `<dream>` and `<action>` are appended to these multimodal embeddings, where `<dream>` contains three subqueries (dynamic, depth and semantics), which could be used for the prediction of specific knowledge. Subsequently, we leverage a large language model based on GPT-2 [107] to integrate and attend across modalities and queries using carefully structured causal and non-causal attention mechanisms (Figure 4). This effectively fuses low-level perceptual signals into compact, semantically coherent representations of the world state.

Finally, specialized light-weight output heads comprising by shallow convolutional layers decode *world embedding* into explicit predictions: reconstruct anticipated dynamic region, monocular depth, and semantic features. During inference, DreamVLA skips the decoder entirely, saving substantial computation. Instead, the model outputs an *world embedding* that encapsulates predictions of future dynamics, depth, and semantics without pixel-level reconstruction, thereby retaining the accuracy gains from future-state reasoning while maintaining low latency. In parallel, we employ a denoising diffusion transformer [92] to decode latent action embedding into executable robot action sequences. Collectively, these components enable DreamVLA to perform robust, predictive vision–language–action reasoning in an end-to-end manner.

3.3 Comprehensive World Knowledge Prediction

Predicting *what will matter next* is more valuable than merely reproducing the raw future frame. DreamVLA explicitly forecasts future world knowledge that is most relevant for manipulation, including (i) motion–centric **dynamic region**, (ii) 3D **depth geometry**, and (iii) high-level **semantics**. These complementary signals provide a compact, structured surrogate for raw pixels and supply the policy with look-ahead context for inverse dynamics planning.

Motion-centric dynamic-region reconstruction. Predicting dynamic regions tells the robot *what parts of the scene are about to move*, allowing the model to capture the statistical link between the current scene, the language instruction, and the actions needed to realize the predicted motion. As shown in Figure 3, DreamVLA neither predicts dense optical flow nor synthesizes an entire future frame. Instead, we first apply CoTracker [69, 70] to extract dynamic regions, namely pixels that move with the robot end-effector or other movable objects, and then train DreamVLA to reconstruct only these regions. Furthermore, generating reconstruction targets with an asymmetrical tokenizer can further enhance performance [106]. From the perspective of discrete variational autoencoder (dVAE) [108–111], the overall optimization is to maximize the *evidence lower bound* (ELBO) [112–114, 68] of the log-likelihood $\mathrm{P}(x_i|\tilde{x}_i)$. Let x denote the original image, $\tilde{x}$ the masked motion region, and z the reconstruction target. The generative modeling can be described as:

$$\sum_{(z_i,\tilde{z}_i)\in\mathcal{D}} \log \mathrm{P}(x_i|\tilde{x}_i) \geq \sum_{(x_i,\tilde{x}_i)\in\mathcal{D}} \Big(\mathbb{E}_{z_i\sim \mathrm{Q}_\phi(\mathbf{z}|x_i)}\big[\log \mathrm{P}_\psi(x_i|z_i)\big] - D_{\mathrm{KL}}\big[z, \mathrm{P}_\theta(\mathbf{z}|\hat{z}_i)\big]\Big), \quad (4)$$

where $\mathrm{P}_\psi(x|z)$ is the tokenizer decoder to recover origin data, $\hat{z}_i = \mathrm{Q}_\phi(\mathbf{z}|\tilde{x}_i)$ denotes the masked motion region tokens from masked data and $\mathrm{P}_\theta(z|\hat{z}_i)$ reconstructs masked tokens in an autoencoding fashion. Here, the $\mathrm{P}_\theta(z|\hat{z}_i)$ is zero, and the dynamic region prediction loss can be formulated as:

$$\mathcal{L}_{\text{dyn}} = \frac{1}{|\mathcal{D}|}\sum_{x_i \in \mathcal{D}} \mathbb{E}_{z \sim Q_\phi(z|x_i)}\Big[-\log \mathrm{P}_\psi\big((x_i)_{\mathcal{M}} \mid z\big)\Big]. \tag{5}$$

Depth prediction. Predicting how the depth field will evolve tells the robot *where it should move next*, steering it toward free space and away from impending obstacles. If depth sensors are available, we supervise the DreamVLA with ground-truth maps; on low-cost platforms without depth sensing, we instead hallucinate future geometry from a single RGB stream. To do so, we treat Depth-Anything [65, 66] predictions as a self-supervised teacher and train a dedicated *depth query* to regress the aligned future map $\hat{d}_{t+n}$. The objective is a scale-normalized mean-squared error,

$$\mathcal{L}_{\text{depth}} = \tfrac{1}{HW}\sum_{i,j}\big(\hat{d}_{t+n}^{(i,j)} - \alpha\, d_{t+n}^{(i,j)}\big)^2, \tag{6}$$

$$\alpha = \frac{\sum_{i,j}\hat{d}_{t+n}^{(i,j)} d_{t+n}^{(i,j)}}{\sum_{i,j} d_{t+n}^{(i,j)\,2}}, \tag{7}$$

where α removes the global scale ambiguity that monocular methods cannot resolve. In practice, this simple loss is sufficient: the teacher provides metrically plausible depth, and the scale-normalization term encourages the model to preserve ordinal depth relationships, a property that is crucial for grasp synthesis and collision checking, while ignoring any arbitrary global scale shift.

Contrastive semantic forecasting. Predicting future semantics teaches the robot *which objects or regions will matter for the task*, providing a high-level context (for example, object identity and affordances) that guides the selection of goals and grasp choice. To learn these semantics, DreamVLA predicts future DINOv2 [71] and SAM [72] feature $\hat{c}_{t+n}$ using an InfoNCE loss [115, 68]: the ground-truth feature is the positive sample, whereas spatially shifted features act as negatives. This encourages discriminative anticipation that the model must pick the correct object semantics among plausible but wrong futures:

$$\mathcal{L}_{\text{sem}} = -\log\frac{\exp\big(\hat{c}_{t+n}^{\top} c_{t+n}/\tau\big)}{\sum_k \exp\big(\hat{c}_{t+n}^{\top} c_k/\tau\big)}, \tag{8}$$

where k represents the number of tokens in spatial, and τ denotes the temperature.

Structured attention for cross-type knowledge disentanglement. To preserve clear cross-type knowledge boundaries, `<dream>` is decomposed into three sub-queries (dynamic, depth and semantics). If these sub-queries could freely attend to one another, high-frequency flow details would contaminate depth reasoning, and semantic cues might bleed into motion features, producing noisy mixed representations. We therefore mask their mutual attention: each sub-query attends only to the shared visual, language, and state tokens, while direct links among the three are disabled, keeping their latent features disentangled and free of cross-talk. As shown in Figure 4, both `<dream>` and `<action>` queries also employ causal attention restricted to past context, which preserves temporal causality. This organized pattern mirrors the specialist routing used in Mixture-of-Experts (MoE) networks [116]. By avoiding cross-modal leakage, the structured attention supplies clean future world knowledge for action prediction, improves robustness, and maintains temporal consistency.

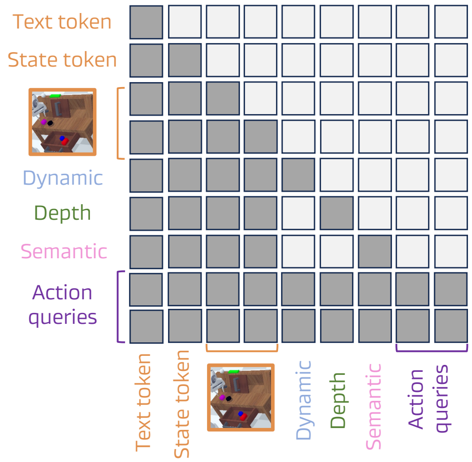

Figure 4: Block-wise structured attention.

3.4 Inverse Dynamics via Denoising Diffusion Transformer

Given two ordered observations o_t and o_{t+1}, classical inverse dynamics infers the intermediate action $\hat{a}_t$. We extend this formulation by predicting a full action sequence $\hat{a}_{t:t+n-1}$ conditioned

Table 1: **CALVIN ABC-D results.** We present the average success computed over 1000 rollouts for each task and the average number of completed tasks to solve 5 instructions consecutively (Avg. Len.). DreamVLA shows significant superiority over baselines. The best results are **bolded**.

Method	Task completed in a row					
	1	2	3	4	5	Avg. Len. ↑
Roboflamingo [32]	82.4	61.9	46.6	33.1	23.5	2.47
Susie [120]	87.0	69.0	49.0	38.0	26.0	2.69
GR-1 [16]	85.4	71.2	59.6	49.7	40.1	3.06
3D Diffusor Actor [95]	92.2	78.7	63.9	51.2	41.2	3.27
OpenVLA [1]	91.3	77.8	62.0	52.1	43.5	3.27
RoboDual [121]	94.4	82.7	72.1	62.4	54.4	3.66
UNIVLA [122]	95.5	85.8	75.4	66.9	56.5	3.80
π_0 [34]	93.8	85.0	76.7	68.1	59.9	3.84
CLOVER [123]	96.0	83.5	70.8	57.5	45.4	3.53
UP-VLA [59]	92.8	86.5	81.5	76.9	69.9	4.08
Robovlm [39]	98.0	93.6	85.4	77.8	70.4	4.25
Seer [58]	96.3	91.6	86.1	80.3	74.0	4.28
VPP [51]	95.7	91.2	86.3	81.0	75.0	4.29
DreamVLA	**98.2**	**94.6**	**89.5**	**83.4**	**78.1**	**4.44**

on the current observation o_t and future latent world embeddings $\mathbf{w}_{t+n}$. Specifically, DreamVLA first aggregates this latent embedding, already enriched with predicted future dynamics, depth, and semantics, into a compact action embedding via a dedicated action query and the model's causal attention. Since the world and action embeddings occupy the same latent space and share similar statistics, a naive MLP head cannot disentangle modality-specific information or exploit their cross-modal correlations. We therefore employ a denoising diffusion transformer (DiT) [92, 117] as the action head. Conditioned on the action embedding, DiT employs iterative self-attention and denoising to fuse perceptual forecasts with control priors and to transform Gaussian noise into an n-step trajectory $a_{t:t+n-1}$, yielding coherent, diverse, and physically grounded action sequences. The loss of action prediction can be formulated as:

$$\mathcal{L}_{\text{DiT}} = \mathbb{E}_{\tau,\varepsilon}\Big\|\varepsilon - \varepsilon_\theta\big(\sqrt{\bar{\alpha}_\tau}\,a_{t:t+n-1} + \sqrt{1-\bar{\alpha}_\tau}\,\varepsilon,\ \tau,\ \mathbf{c}\big)\Big\|_2^2, \qquad (9)$$

where ε_θ is the DiT denoiser, $\varepsilon \sim \mathcal{N}(0, I)$, $\bar{\alpha}_\tau$ follows a cosine noise schedule and $\mathbf{c}$ is the latent action embedding obtained from a large language model. Inference is performed by drawing a Gaussian sample and running the learned reverse diffusion, yielding diverse yet physically plausible trajectories that close the perception–prediction–action loop.

4 Experiments

4.1 Implementation Details

All models are implemented in PyTorch and trained on NVIDIA 8 A800 GPUs. We use an AdamW [118] optimizer with initial learning rate 10^{-3}, weight decay $1e-4$, and a cosine learning-rate schedule with 5% linear warm-up. Batch size is set to 64, we set the query length of each modality 9 and diffusion steps in DiT to 10. We weight the dynamic region, depth and segmentation prediction losses as λ_{dyn}=0.1, λ_{depth}=0.001, λ_{sem}=0.1, and the action loss as λ_{DiT}=1, respectively. We first pre-train DreamVLA on the language-free split of the CALVIN [119] and on the full DROID dataset [84]. For the LIBERO benchmark, we first pretrain DreamVLA on LIBERO-90 and then finetune on each track. The model predicts entire frames instead of comprehensive knowledge, keeping storage and computation requirements manageable. We then fine-tune DreamVLA on each target dataset using the comprehensive world knowledge forecasting objective. All models are trained for 20 epochs, and we select the checkpoint with the highest validation success rate (SR) for final evaluation.

4.2 Simulation Benchmark Experiments

Simulation setup. We evaluate DreamVLA on CALVIN [119] and LIBERO [124] benchmark. CALVIN is a simulated benchmark designed for learning long-horizon, language-conditioned robot manipulation policies. It comprises four distinct manipulation environments and over six hours

Table 2: **The extended LIBERO experiments.** DreamVLA achieves the best or competitive performance across all tracks compared to previous approaches. The best results are **bolded**.

Methods	Scores (%)				Average
	Spatial	Object	Goal	Long	
Diffusion Policy [92]	78.3	92.5	68.3	50.5	72.4
Octo [15]	78.9	85.7	84.6	51.1	75.1
OpenVLA [1]	84.7	88.4	79.2	53.7	76.5
SpatialVLA [38]	88.2	89.9	78.6	55.5	78.1
CoT-VLA [60]	87.5	91.6	87.6	69.0	81.1
DreamVLA	**97.5**	**94.0**	**89.5**	**89.5**	**92.6**

of teleoperated play data per environment, captured from multiple sensors including static and gripper-mounted RGB-D cameras, tactile images, and proprioceptive readings. We report the success rate of every track and the average length of 5 tasks. Additionally, evaluations are also conducted on LIBERO [124], a simulated benchmark spanning four suites (LIBERO-Spatial/-Object/-Goal/-Long). Each suite contains 10 tasks supported by 50 human-teleoperated demonstrations, targeting spatial reasoning, object-centric manipulation, and goal completion.

Results. As shown in Table 1, DreamVLA achieves the highest performance on ABC-D tasks, Our method surpasses Roboflamingo [32], 3D Diffusor Actor [95], OpenVLA [1], RoboDual [121], UNIVLA [122], Robovlm [39] and GR1 [16], which directly projects the RGB/depth image to action signals as shown in Figure 1(a) in the manuscripts. Compared to methods that use a copilot model to generate sub-goal images as input, like Susie [120] and CLOVER [123] as shown in Figure 1(b) in manuscripts, our model significantly achieves more accurate control. DreamVLA outperforms approaches like UP-VLA [59], Seer [58], and VPP [51] as shown in Figure 1(c), which merge whole sub-goal image foresight into one VLA to take benefits from a more integrated design and joint optimization. indicating that our method has better multi-task learning and generalization capabilities in simulation tasks. For the LIBERO benchmark [124], DreamVLA exhibits better or comparable ability across all tracks compared to previous approaches by future world knowledge prediction as shown in Table 2.

4.3 Real World Experiments

To evaluate the effectiveness of our method in the real-world, we use the Franka Panda arm to conduct real-world experiments on gripper grasping. In our setups, two RealSense D415 cameras capture RGB images. One is in a third-person view, and the other is at the end of the robotic arm, as shown in Figure 5. We collect four categories of objects for two tasks: pick and place. Additionally, we conduct experiments on drawer opening and closing tasks, as shown in the supplementary. Follow [58], we pretrain DreamVLA on the DROID [84] contains large-scale trajectories of Franka robots in varied scenes. For fair comparison, we fine-tune Diffusion Policy [92], Octo-Base [15], OpenVLA [1] and DreamVLA on collected demonstration datasets containing 100 trajectories for each task.

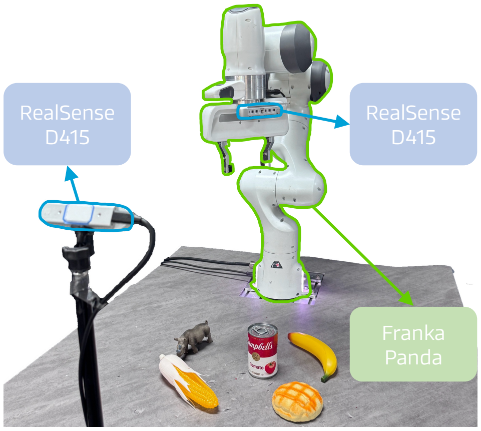

Figure 5: Real-world experiment setup.

In the experimental setup, each trial permits a maximum of 20 consecutive attempts. For the grasping experiments, objects are randomly positioned on the table surface. A trial is deemed successful if the robotic arm successfully grasps the target object within the predefined attempt limit. In the placement experiments, the robot is required to transfer the grasped object into a designated basket. Success is recorded only if both the grasping and placement operations are completed within the allowed attempts. For the drawer manipulation tasks, the drawer is placed randomly in front of the robotic arm. The experiment is considered successful if the drawer displacement exceeds 10 centimeters, indicating effective interaction. The results, presented in Table 3, demonstrate that our method performs better than other methods. More real-world experiment visualizations are shown in the supplementary section.

Table 3: **Real-world evaluation** with the Franka Robot across three tasks.

Method	Pick			Place			Drawer			Task (All)
	Bottle	Doll	Avg.	Banana	Chili	Avg.	Open	Close	Avg.	Avg.
Diffusion Policy [92]	50.0	70.0	60.0	65.0	45.0	55.0	15.0	60.0	37.5	50.8
Octo-Base [15]	50.0	60.0	55.0	40.0	50.0	45.0	20.0	50.0	35.0	45.0
OpenVLA [1]	50.0	40.0	45.0	20.0	30.0	25.0	40.0	30.0	35.0	35.0
DreamVLA	**85.0**	**80.0**	**82.5**	**80.0**	**80.0**	**80.0**	**70.0**	**65.0**	**67.5**	**76.7**

Table 4: **Performance comparison** between predicting the optical flow and the dynamic region. Notably, the * denotes that this result is from [58].

Method	Task completed in a row					
	1	2	3	4	5	Avg. Len. ↑
Vanilla VLA*	93.0	82.4	72.3	62.6	53.3	3.64
+ dynamic region	97.6	92.6	87.5	80.4	73.7	4.32
+ depth	**98.3**	94.3	88.5	82.0	77.2	4.40
+ semantics	98.2	**94.6**	**89.5**	**83.4**	**78.1**	**4.44**

4.4 Ablation Study

In this section, we design the experiments to investigate the following questions.
Q1: What is the contribution of each modal characteristic?

The core motivation of DreamVLA is to enable the model to predict comprehensive visual knowledge of the future to enhance action reasoning. However, not all types of knowledge contribute equally to subsequent execution. We consider four types of predictive knowledge: dynamic region, depth, and semantic segmentation features derived from SAM and DINO. As shown in Figure 6, we first train the model with each knowledge forecasting independently. The green dashed line denotes the performance of the Vanilla VLA baseline, which uses no knowledge prediction. Among all, predicting dynamic regions proves to be the most beneficial, because these masks explicitly flag the pixels that are about to change and therefore align almost perfectly with the policy's action semantics. By contrast, supervising the network with depth map, DINO or SAM features alone not only fails to help but often degrades performance. We analyze that this gap stems from how closely each auxiliary target matches the downstream objective: dynamic-region labels supply gradients that reinforce the action head, whereas depth regression and high-dimensional feature matching (DINO/SAM) inject large, noisy losses that dominate optimization. With the limited model attention budget, these competing gradients dilute the task-relevant features and push the backbone toward suboptimal optima, producing the observed drop below the dashed baseline.

Next, we train the model with all five knowledge heads simultaneously (All) and perform an ablation study (All-X), where we remove one knowledge signal at a time to evaluate its contribution. Removing F leads to the most significant performance drop, confirming its essential role. Interestingly, removing DINO results in similar or even better performance, suggesting that not all semantic signals are equally helpful or stable in predicting outcomes, so we only use semantic features from SAM in the subsequent ablations. Table 4 reveals a clear and decreasing return pattern in all ablations.

Q2: Auxiliary Tasks vs. Future Knowledge Prediction: which drives improvement?

Table 5 contrasts two training regimes: predicting complete world knowledge and performing auxiliary reconstructions, showing that the former is decisively superior. In our ablation, every prediction strategy is individually replaced by its reconstruction counterpart, yet each substitution consistently lowers performance: VLA trained only to redraw the current RGB, depth, semantics, or DINOv2 features can handle the first few actions but soon loses coherence, whereas a network trained to forecast the next dynamic region, depth map, and semantics preserves accuracy throughout the trajectory and carries tasks much farther before failure. The reason is that prediction provides a richer, action-oriented signal, directing learning toward the pixels that will drive the upcoming decision, while reconstruction merely revisits background detail that the control policy never actually needs.

Q3: Why do we use the optical flow as the mask instead of directly forecasting it?

To justify our choice of employing motion-centric dynamic regions over direct flow forecasting, we implement both variants under identical settings (Table 6). In the optical flow setup, the model must predict the full future flow field along with the subgoal image, which significantly increases

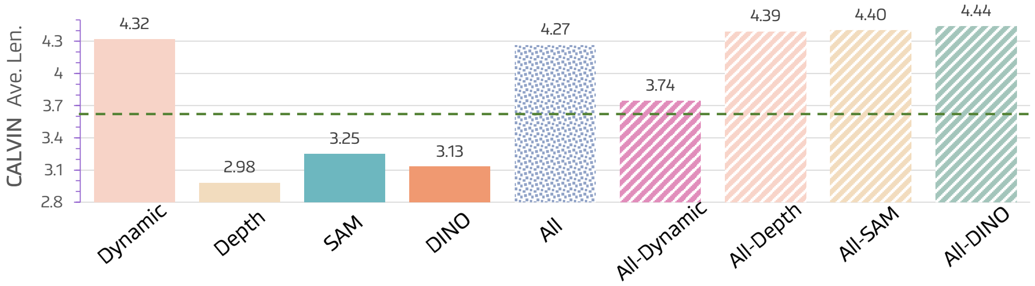

Figure 6: CALVIN ABC-D performance with respect to different combinations of knowledge prediction. All=all of five models, and All-X=taking X out of All.

Table 5: **Performance comparison** between co-training with auxiliary tasks and predicting the comprehensive world knowledge.

Method	Task completed in a row					
	1	2	3	4	5	Avg. Len.
Auxiliary	97.7	92.3	85.6	79.5	74.2	4.14
Prediction	**98.2**	**94.6**	**89.5**	**83.4**	**78.1**	**4.44**

Table 6: **Performance comparison** between predicting the optical flow and dynamic region.

Method	Task completed in a row					
	1	2	3	4	5	Avg. Len.
Optical	97.6	92.4	86.8	81.7	75.4	4.23
Dynamic	**98.2**	**94.6**	**89.5**	**83.4**	**78.1**	**4.44**

Table 7: **Performance comparison** between vanilla causal and our structured attention.

Method	Task completed in a row					
	1	2	3	4	5	Avg. Len.
Causal	94.2	86.5	78.4	71.3	62.7	3.75
Structure	**98.2**	**94.6**	**89.5**	**83.4**	**78.1**	**4.44**

Table 8: **Performance comparison** between shared and seprated queries.

Method	Task completed in a row					
	1	2	3	4	5	Avg. Len.
Shared	95.5	90.1	83.8	76.9	70.4	4.17
Separated	**98.2**	**94.6**	**89.5**	**83.4**	**78.1**	**4.44**

the training complexity. This extra burden manifests in markedly lower multi-step success rates. By contrast, our dynamic region approach merely employs the pretrained flow model to obtain a binary mask, focusing the model on "where" relevant motion occurs, bringing a significant improvement.

Q4: The effectiveness of structured attention in DreamVLA.

To demonstrate the effectiveness of our proposed structure attention mechanism in Figure 4, we swap it for a vanilla causal mask while keeping everything else fixed. In this setting, every `<dream>` query, including the one meant to capture semantics, can also read the flow and depth tokens produced in the same step; the extra cross-peek mixes unrelated signals, adds gradient noise, and quickly degrades long-horizon control. Our mask removes all query-to-query edges, so `<action>` query consults only past language, state and multimodal predictions, never their siblings. Table 7 shows the payoff: the causal variant brings a marginal improvement for Vanilla VLA, whereas the block-sparse version keeps success high throughout, confirming that blocking intra-step leakage is important.

Q5: Can we use the shared query to predict the comprehensive world knowledge?

Instead of assigning separate queries to dynamic region, depth, and semantics features, one might let a single set of shared queries predict all signals. To test this idea, we split each world-embedding vector into four equal sub-spaces, with each quarter intended to carry a different modality. Table 8 shows that the shared-query design hurts action performance: mixing modalities in the same query introduces cross-talk, so the diffusion head receives noisy features. In contrast, giving each modality its query keeps the representations disentangled and yields a clear performance gain.

5 Conclusion

We present DreamVLA, a novel Visual-Language-Action framework that enables inverse dynamics modeling through comprehensive world knowledge prediction, supporting the perception-prediction-action loop for manipulation tasks. DreamVLA leverages dynamic-region-guided knowledge forecasting, combining spatial and semantic cues to generate compact and informative representations for action planning. We introduce a block-wise structured-attention mechanism, coupled with a diffusion-transformer decoder, to suppress representation noise from cross-type knowledge leakage and thus enable coherent multi-step action reasoning. Extensive experiments in both real and simulated environments demonstrate the effectiveness of DreamVLA, achieving a 76.7% success rate on real-world robot tasks and outperforming prior methods on the CALVIN ABC-D benchmark.

Acknowledgements

This work was supported by Grants of NSFC 62302246, ZJNSFC LQ23F010008, Ningbo 2023Z237 & 2024Z284 & 2024Z289 & 2023CX050011 & 2025Z038 & 2025Z091 and supported by High Performance Computing Center at Eastern Institute of Technology and Ningbo Institute of Digital Twin.

References

[1] 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. *arXiv preprint arXiv:2406.09246*, 2024. 1, 3, 7, 8, 9, 32

[2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong T. Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: robotics transformer for real-world control at scale. In *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. 3, 32

[3] Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, et al. Video language planning. *arXiv preprint arXiv:2310.10625*, 2023.

[4] Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *Advances in Neural Information Processing Systems*, 36, 2024.

[5] Zawalski Michał, Chen William, Pertsch Karl, Mees Oier, Finn Chelsea, and Levine Sergey. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024. 1, 3

[6] Kaifeng Zhang, Zhao-Heng Yin, Weirui Ye, and Yang Gao. Learning manipulation skills through robot chain-of-thought with sparse failure guidance. *arXiv preprint arXiv:2405.13573*, 2024.

[7] Yao Mu, Tianxing Chen, Shijia Peng, Zanxin Chen, Zeyu Gao, Yude Zou, Lunkai Lin, Zhiqiang Xie, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version). In *European Conference on Computer Vision*, pages 264–273. Springer, 2025.

[8] Jiangran Lyu, Yuxing Chen, Tao Du, Feng Zhu, Huiquan Liu, Yizhou Wang, and He Wang. Scissorbot: Learning generalizable scissor skill for paper cutting via simulation, imitation, and sim2real. *arXiv preprint arXiv:2409.13966*, 2024.

[9] Wenbo Cui, Chengyang Zhao, Songlin Wei, Jiazhao Zhang, Haoran Geng, Yaran Chen, Haoran Li, and He Wang. Gapartmanip: A large-scale part-centric dataset for material-agnostic articulated object manipulation. *arXiv preprint arXiv:2411.18276*, 2024.

[10] Jinghuan Shang, Karl Schmeckpeper, Brandon B May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. *arXiv preprint arXiv:2407.20179*, 2024.

[11] Jiawei He, Danshi Li, Xinqiang Yu, Zekun Qi, Wenyao Zhang, Jiayi Chen, Zhaoxiang Zhang, Zhizheng Zhang, Li Yi, and He Wang. Dexvlg: Dexterous vision-language-grasp model at scale. *arXiv preprint arXiv:2507.02747*, 2025.

[12] Xiang Li, Varun Belagali, Jinghuan Shang, and Michael S Ryoo. Crossway diffusion: Improving diffusion-based visuomotor policy via self-supervised learning. In *IEEE International Conference on Robotics and Automation (ICRA'24)*, pages 16841–16849. IEEE, 2024.

[13] Xiang Li, Cristina Mata, Jongwoo Park, Kumara Kahatapitiya, Yoo Sung Jang, Jinghuan Shang, Kanchana Ranasinghe, Ryan Burgert, Mu Cai, Yong Jae Lee, et al. Llara: Supercharging robot learning data for vision-language policy. In *International Conference on Learning Representations (ICLR'25)*, 2025. 1

[14] Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 1, 3, 32

[15] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. 1, 3, 8, 9

[16] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *The Twelfth International Conference on Learning Representations*. 3, 7, 8, 29

[17] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 1

[18] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on robot learning*, pages 894–906. PMLR, 2022. 3

[19] Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. *arXiv preprint arXiv:2410.18647*, 2024.

[20] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

[21] Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint arXiv:2409.16283*, 2024. 3

[22] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *Conference on Robot Learning (CoRL)*, 2024.

[23] Jiangran Lyu, Ziming Li, Xuesong Shi, Chaoyi Xu, Yizhou Wang, and He Wang. Dywa: Dynamics-adaptive world action model for generalizable non-prehensile manipulation. *arXiv preprint arXiv:2503.16806*, 2025. 3

[24] Zekun Qi, Wenyao Zhang, Yufei Ding, Runpei Dong, Xinqiang Yu, Jingwen Li, Lingyun Xu, Baoyu Li, Xialin He, Guofan Fan, et al. Sofar: Language-grounded orientation bridges spatial reasoning and object manipulation. *arXiv preprint arXiv:2502.13143*, 2025. 2, 31, 32

[25] Xialin He, Runpei Dong, Zixuan Chen, and Saurabh Gupta. Learning getting-up policies for real-world humanoid robots. *arXiv preprint arXiv:2502.12152*, 2025.

[26] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025.

[27] Jiazhao Zhang, Nandiraju Gireesh, Jilong Wang, Xiaomeng Fang, Chaoyi Xu, Weiguang Chen, Liu Dai, and He Wang. Gamma: Graspability-aware mobile manipulation policy learning based on online grasping pose fusion. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1399–1405. IEEE, 2024. 1

[28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 1, 3

[29] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. *arXiv preprint arXiv:2402.07865*, 2024.

[30] OpenAI. Gpt-4v(ision) system card, 2023. URL `https://openai.com/research/gpt-4v-system-card`. 3

[31] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024. 1

[32] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. In *The Twelfth International Conference on Learning Representations*. 1, 3, 7, 8, 32

[33] Dantong Niu, Yuvan Sharma, Giscard Biamby, Jerome Quenum, Yutong Bai, Baifeng Shi, Trevor Darrell, and Roei Herzig. Llarva: Vision-action instruction tuning enhances robot learning. In *8th Annual Conference on Robot Learning*, 2024.

[34] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 1, 3, 7

[35] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024. 1, 3

[36] Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Weixian Lei, Lijuan Wang, and Mike Zheng Shou. Showui: One vision-language-action model for gui visual agent. *arXiv preprint arXiv:2411.17465*, 2024. 3

[37] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025. 3

[38] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model, 2025. URL `https://arxiv.org/abs/2501.15830`. 8

[39] Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2024. 3, 7, 8, 33

[40] Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. Flower: Democratizing generalist robot policies with efficient vision-language-action flow policies. In *7th Robot Learning Workshop: Towards Robots with Human-Level Abilities*. 3

[41] 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.

[42] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

[43] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024.

[44] 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. *arXiv preprint arXiv:2501.09747*, 2025. 1

[45] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3

[46] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024. 1, 3, 32

[47] Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. In *International Conference on Machine Learning*, pages 37321–37341. PMLR, 2024.

[48] Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. In *The Twelfth International Conference on Learning Representations*.

[49] Kaidong Zhang, Pengzhen Ren, Bingqian Lin, Junfan Lin, Shikui Ma, Hang Xu, and Xiaodan Liang. Pivot-r: Primitive-driven waypoint-aware world model for robotic manipulation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[50] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023. 3

[51] 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. *arXiv preprint arXiv:2412.14803*, 2024. 3, 7, 8

[52] Dongxiu Liu, Haoyi Niu, Zhihao Wang, Jinliang Zheng, Yinan Zheng, Zhonghong Ou, Jianming Hu, Jianxiong Li, and Xianyuan Zhan. Efficient robotic policy learning via latent space backward planning. *arXiv preprint arXiv:2505.06861*, 2025. 3

[53] Kanchana Ranasinghe, Xiang Li, Cristina Mata, Jongwoo Park, and Michael S Ryoo. Pixel motion as universal representation for robot control. *arXiv preprint arXiv:2505.07817*, 2025. 3

[54] Wenyan Yang, Ahmet Tikna, Yi Zhao, Yuying Zhang, Luigi Palopoli, Marco Roveri, and Joni Pajarinen. Symbolically-guided visual plan inference from uncurated video data. *arXiv preprint arXiv:2505.08444*, 2025.

[55] Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through neural trajectories. *arXiv preprint arXiv:2505.12705*, 2025. 3

[56] Yuhang Huang, JIazhao Zhang, Shilong Zou, XInwang Liu, Ruizhen Hu, and Kai Xu. Ladi-wm: A latent diffusion-based world model for predictive manipulation. *arXiv preprint arXiv:2505.11528*, 2025.

[57] Jiange Yang, Haoyi Zhu, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. Tra-moe: Learning trajectory prediction model from multiple domains for adaptive policy conditioning. *ArXiv*, abs/2411.14519, 2024. 1

[58] 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. *Int. Conf. Learn. Represent. (ICLR)*, 2024. 1, 3, 7, 8, 9, 28, 29

[59] Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. Up-vla: A unified understanding and prediction model for embodied agent. *arXiv preprint arXiv:2501.18867*, 2025. 7, 8, 29

[60] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. *arXiv preprint arXiv:2503.22020*, 2025. 2, 3, 8

[61] Yuyin Yang, Zetao Cai, Yang Tian, Jia Zeng, and Jiangmiao Pang. Gripper keypose and object pointflow as interfaces for bimanual robotic manipulation. *arXiv preprint arXiv:2504.17784*, 2025.

[62] 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. *arXiv preprint arXiv:2504.02792*, 2025.

[63] Hongyin Zhang, Zifeng Zhuang, Han Zhao, Pengxiang Ding, Hongchao Lu, and Donglin Wang. Reinbot: Amplifying robot visual-language manipulation with reinforcement learning. *arXiv preprint arXiv:2505.07395*, 2025. 1, 3

[64] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 2022. 2, 3

[65] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024. 2, 3, 6

[66] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024. 3, 6, 28

[67] Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. Shapellm: Universal 3d object understanding for embodied interaction. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XLIII*, volume 15101 of *Lecture Notes in Computer Science*, pages 214–238. Springer, 2024. 3, 32

[68] Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *Int. Conf. Mach. Learn. (ICML)*, 2023. 2, 3, 5, 6, 32

[69] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision*, pages 18–35. Springer, 2024. 2, 5, 27

[70] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024. 2, 5

[71] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024. 2, 3, 4, 6, 26, 28

[72] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 3992–4003. IEEE, 2023. 2, 4, 6, 26, 28

[73] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022. 3

[74] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. 3, 26, 32

[75] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. *Robotics: Science and Systems*, 2024. 3

[76] Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*, 2024. 3

[77] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 3

[78] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[79] Konstantinos I. Roumeliotis and Nikolaos D. Tselikas. Chatgpt and open-ai models: A preliminary review. *Future Internet*, 15(6):192, 2023.

[80] OpenAI. Openai o3 and o4-mini system card, 2025. URL `https://openai.com/research/o3-o4-mini-system-card`. 3

[81] Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, Xiangwen Kong, Xiangyu Zhang, Kaisheng Ma, and Li Yi. DreamLLM: Synergistic multimodal comprehension and creation. In *Int. Conf. Learn. Represent. (ICLR)*, 2024. 3, 4

[82] Yuang Peng, Yuxin Cui, Haomiao Tang, Zekun Qi, Runpei Dong, Jing Bai, Chunrui Han, Zheng Ge, Xiangyu Zhang, and Shu-Tao Xia. Dreambench++: A human-aligned benchmark for personalized image generation. *CoRR*, abs/2406.16855, 2024. 3

[83] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021. 3

[84] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. 7, 8, 30, 32

[85] Shengliang Deng, Mi Yan, Songlin Wei, Haixin Ma, Yuxin Yang, Jiayi Chen, Zhiqi Zhang, Taoyu Yang, Xuheng Zhang, Heming Cui, et al. Graspvla: a grasping foundation model pre-trained on billion-scale synthetic action data. *arXiv preprint arXiv:2505.03233*, 2025. 3

[86] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong T. Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183. PMLR, 2023. 3, 32

[87] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. RT-H: action hierarchies using language. *CoRR*, abs/2403.01823, 2024. 3, 32

[88] Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025. 3

[89] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. pi0.5: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. 3

[90] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, Shu Jiang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025. 3

[91] 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. 3

[92] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023. 3, 5, 7, 8, 9

[93] Zhi Hou, Tianyi Zhang, Yuwen Xiong, Haonan Duan, Hengjun Pu, Ronglei Tong, Chengyang Zhao, Xizhou Zhu, Yu Qiao, Jifeng Dai, et al. Dita: Scaling diffusion transformer for generalist vision-language-action policy. *arXiv preprint arXiv:2503.19757*, 2025.

[94] 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. *arXiv preprint arXiv:2410.07864*, 2024.

[95] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024. 7, 8

[96] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy. *arXiv e-prints*, pages arXiv–2403, 2024. 3

[97] 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. *arXiv preprint arXiv:2505.15659*, 2025. 3

[98] Jiaxu Wang, Qiang Zhang, Jingkai Sun, Jiahang Cao, Yecheng Shao, and Renjing Xu. Reinforcement learning with generalizable gaussian splatting. *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 435–441, 2024. URL https://api.semanticscholar.org/CorpusID:269042854. 3

[99] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024. 3

[100] Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Ran Cheng, Yaxin Peng, Chaomin Shen, et al. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. *arXiv preprint arXiv:2502.14420*, 2025. 3

[101] Fanqi Lin, Ruiqian Nai, Yingdong Hu, Jiacheng You, Junming Zhao, and Yang Gao. Onetwovla: A unified vision-language-action model with adaptive reasoning. *arXiv preprint arXiv:2505.11917*, 2025.

[102] Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, et al. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. *arXiv preprint arXiv:2502.19417*, 2025. 3

[103] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3, 4, 26

[104] Runpei Dong, Zekun Qi, Linfeng Zhang, Junbo Zhang, Jianjian Sun, Zheng Ge, Li Yi, and Kaisheng Ma. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? In *Int. Conf. Learn. Represent. (ICLR)*, 2023. 3, 32

[105] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2455–2467, 2018. 3

[106] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 4, 5, 26, 29

[107] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 5, 26

[108] Alex Graves. Practical variational inference for neural networks. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Adv. Neural Inform. Process. Syst. (NIPS)*, 2011. 5

[109] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Int. Conf. Learn. Represent. (ICLR)*, 2014.

[110] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Int. Conf. Mach. Learn. (ICML)*, 2021.

[111] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: BERT pre-training of image transformers. In *Int. Conf. Learn. Represent. (ICLR)*. OpenReview.net, 2022. 5

[112] Jorma Rissanen. Modeling by shortest data description. *Autom.*, 14(5):465–471, 1978. 5

[113] Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *ACM Conf. Comput. Learn. Theory (COLT)*, 1993.

[114] Runpei Dong, Zhanhong Tan, Mengdi Wu, Linfeng Zhang, and Kaisheng Ma. Finding the task-optimal low-bit sub-distribution in deep neural networks. In *Int. Conf. Mach. Learn. (ICML)*, 2022. 5

[115] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. 6

[116] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 6

[117] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 7, 26

[118] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Int. Conf. Learn. Represent. (ICLR)*, 2019. 7

[119] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022. 7, 28, 32

[120] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023. 7, 8, 29

[121] Qingwen Bu, Hongyang Li, Li Chen, Jisong Cai, Jia Zeng, Heming Cui, Maoqing Yao, and Yu Qiao. Towards synergistic, generalized, and efficient dual-system for robotic manipulation. *arXiv preprint arXiv:2410.08001*, 2024. 7, 8

[122] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025. 7, 8

[123] Qingwen Bu, Jia Zeng, Li Chen, Yanchao Yang, Guyue Zhou, Junchi Yan, Ping Luo, Heming Cui, Yi Ma, and Hongyang Li. Closed-loop visuomotor control with generative expectation for robotic manipulation. *arXiv preprint arXiv:2409.09016*, 2024. 7, 8, 29

[124] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: benchmarking knowledge transfer for lifelong robot learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 7, 8

[125] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 26

[126] Hao Liu, Lisa Lee, Kimin Lee, and Pieter Abbeel. Instruction-following agents with jointly pre-trained vision-language models. *CoRR*, abs/2210.13431, 2022. 32

[127] Markus Grotz, Mohit Shridhar, Tamim Asfour, and Dieter Fox. Peract2: Benchmarking and learning for robotic bimanual manipulation tasks. *CoRR*, abs/2407.00278, 2024.

[128] Atharva Mete, Haotian Xue, Albert Wilcox, Yongxin Chen, and Animesh Garg. Quest: Self-supervised skill abstractions for learning continuous control. *CoRR*, abs/2407.15840, 2024. 32

[129] Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Manon Devin, Alex X. Lee, Maria Bauzá Villalonga, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, Antoine Laurens, Claudio Fantacci, Valentin Dalibard, Martina Zambelli, Murilo Fernandes Martins, Rugile Pevceviciute, Michiel Blokzijl, Misha Denil, Nathan Batchelor, Thomas Lampe, Emilio Parisotto, Konrad Zolna, Scott E. Reed, Sergio Gómez Colmenarejo, Jon Scholz, Abbas Abdolmaleki, Oliver Groth, Jean-Baptiste Regli, Oleg Sushkov, Thomas Rothörl, José Enrique Chen, Yusuf Aytar, Dave Barker, Joy Ortiz, Martin A. Riedmiller, Jost Tobias Springenberg, Raia Hadsell, Francesco Nori, and Nicolas Heess. Robocat: A self-improving generalist agent for robotic manipulation. *Trans. Mach. Learn. Res.*, 2024, 2024. 32

[130] Shizhe Chen, Ricardo Garcia Pinel, Cordelia Schmid, and Ivan Laptev. Polarnet: 3d point clouds for language-guided robotic manipulation. In *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pages 1761–1781. PMLR, 2023. 32

[131] Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *CoRR*, abs/2406.10721, 2024. 32

[132] Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Nikhil J. Joshi, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do as I can, not as I say: Grounding language in robotic affordances. In *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pages 287–318. PMLR, 2022. 32

[133] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning, CoRL 2022, 14-18 December 2022,*

Auckland, New Zealand, volume 205 of *Proceedings of Machine Learning Research*, pages 1769–1782. PMLR, 2022.

[134] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29 - June 2, 2023*, pages 9493–9500. IEEE, 2023.

[135] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. In *Annu. Conf. Robot. Learn. (CoRL)*, 2023.

[136] Wenlong Huang, Fei Xia, Dhruv Shah, Danny Driess, Andy Zeng, Yao Lu, Pete Florence, Igor Mordatch, Sergey Levine, Karol Hausman, and Brian Ichter. Grounded decoding: Guiding text generation with grounded models for embodied agents. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[137] Haoxu Huang, Fanqi Lin, Yingdong Hu, Shengjie Wang, and Yang Gao. Copa: General robotic manipulation through spatial constraints of parts with foundation models. *CoRR*, abs/2403.08248, 2024.

[138] Kuan Fang, Fangchen Liu, Pieter Abbeel, and Sergey Levine. MOKA: Open-World Robotic Manipulation through Mark-Based Visual Prompting. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. 32

[139] Siyuan Huang, Haonan Chang, Yuhan Liu, Yimeng Zhu, Hao Dong, Abdeslam Boularias, Peng Gao, and Hongsheng Li. A3VLM: actionable articulation-aware vision language model. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, volume 270 of *Proceedings of Machine Learning Research*, pages 1675–1690. PMLR, 2024. 32

[140] Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 18061–18070. IEEE, 2024. 32

[141] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, Tengyu Liu, Li Yi, and He Wang. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2023. 32

[142] Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. In *Int. Conf. Comput. Vis. (ICCV)*, 2023. 32

[143] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 77–85, 2017. 32

[144] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Adv. Neural Inform. Process. Syst. (NIPS)*, pages 5099–5108, 2017. 32

[145] Zekun Qi, Muzhou Yu, Runpei Dong, and Kaisheng Ma. VPP: efficient conditional 3d generation via voxel-point progressive representation. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2023. 32

[146] Shaochen Zhang, Zekun Qi, Runpei Dong, Xiuxiu Bai, and Xing Wei. Positional prompt tuning for efficient 3d representation learning. *CoRR*, abs/2408.11567, 2024. 32

[147] Guofan Fan, Zekun Qi, Wenkai Shi, and Kaisheng Ma. Point-gcc: Universal self-supervised 3d scene pre-training via geometry-color contrast. In Jianfei Cai, Mohan S. Kankanhalli, Balakrishnan Prabhakaran, Susanne Boll, Ramanathan Subramanian, Liang Zheng, Vivek K. Singh, Pablo César, Lexing Xie, and Dong Xu, editors, *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, pages 4709–4718. ACM, 2024. 32

[148] Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, Xinqiang Yu, Jiawei He, He Wang, and Li Yi. Omnispatial: Towards comprehensive spatial reasoning benchmark for vision language models. *arXiv preprint arXiv:2506.03135*, 2025. 32

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: We claim in the abstract and introduction that this paper studies advancing vision-language action model via world knowledge prediction.

   Guidelines:

   - The answer NA 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 NA 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: The limitations are addressed in Section D.

   Guidelines:

   - The answer NA 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 authors 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: [NA]

   Justification: This work does not include theoretical results or proofs, similar to previous VLA studies.

   Guidelines:

   - The answer NA 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: We provide the implementation details for reproducibility in Section 4.1, and the project will be open-sourced.

   Guidelines:

   - The answer NA 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: We attach the core script and data preparation to the supplemental material. The complete code will be released in the camera-ready version, accompanied by detailed instructions for reproducibility.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the training and test details in Section 4.

   Guidelines:

   - The answer NA 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: [NA]

   Justification: We set a fixed time seed to achieve controlled generation while ensuring the reproduction of the results.

   Guidelines:

   - The answer NA 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: We provide sufficient information on the computer resources in the supplementary.

   Guidelines:

   - The answer NA 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: Our work conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA 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: [Yes]

    Justification: We discuss the broader impacts of this work in the supplementary material.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: To enhance safety, we suggest incorporating fail-safe mechanisms to halt actions under uncertainty, and using semantic filters to block ambiguous or harmful commands. Human oversight and deployment in controlled environments are also recommended to reduce potential misuse or unintended behaviors.

    Guidelines:

    - The answer NA 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: [Yes]

    Justification: We properly provide these information in Section 4.

    Guidelines:

    - The answer NA 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 assets introduced in the paper are well documented.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA 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: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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.

Appendix

A Implementation Details

A.1 DreamVLA Architecture

Text Encoder. We use the CLIP ViT-B/32 text encoder [103] to process natural language task instructions. The encoder transforms each instruction into a fixed-length embedding that captures semantic intent. These embeddings are then projected into the shared latent space and used to condition the subsequent modules, enabling effective grounding of language into perception and action.

Visual Encoder. We employ an MAE-pretrained ViT-B [106] as the vision encoder. At each timestep, images are captured from two views: eye-on-hand and eye-on-base. Each image is processed by the vision encoder to produce 196 latent vectors, which represent local patch information, along with a [CLS] token that encodes the global representation of the image. Directly inputting all 197 tokens into the transformer backbone would create a significant computational burden, particularly when processing long histories. Moreover, many image details are redundant for accomplishing manipulation tasks. To address this, we utilize the Perceiver Resampler [125] to condense the image representations and extract task-relevant features. The Perceiver Resampler employs learnable latent vectors with a shape of (num latents, dim), where num latents is significantly smaller than the number of image tokens. Through Perceiver Attention, these latent vectors condense the input image features, along with the [CLS] token, to form the final image tokens.

Robot State. The robot state consists of the arm and gripper state. The arm state includes the end-effector position and its rotation in Euler angles, resulting in a six-dimensional representation. The gripper state is a binary value indicating whether the gripper is open or closed. We tokenize the robot state using an MLP. Specifically, the gripper state is first converted into a one-hot encoding. The one-hot encoding of the gripper state and the arm state are then each passed through separate linear layers. The outputs are concatenated and passed through a final linear layer to produce the state token.

Learnable Queries. We introduce two sets of learnable query tokens, denoted as `<dream>` and `<action>`, to extract and integrate information from multimodal inputs for joint prediction.

The `<dream>` queries provide structured supervision through comprehensive knowledge prediction tasks and consist of 64 tokens in total, organized as 9 queries for each of the three modalities: dynamic motion, depth estimation and semantic features. These queries guide the model in reconstructing rich visual representations, enhancing the quality of the learned latent space.

The `<action>` query is dedicated to action sequence prediction. Their length is determined by the temporal prediction horizon, as defined in the action chunking strategy from [74].

Large Language Model. We adopt GPT-2 Medium [107] as our language backbone. GPT-2 Medium is a 24-layer, 16-head Transformer decoder with a hidden size of 1,024 and a total of approximately 345 million parameters. It was pretrained on the WebText corpus (∼8 million documents, 40 GB of text) using autoregressive language modeling to predict the next token with a byte-pair encoding vocabulary of 50,257 tokens.

Output Heads. To decode the *world embedding* into comprehensive world knowledge, we incorporate multiple task-specific output heads that predict dynamic motion regions, depth maps, and high-level semantics, including DINOv2 [71] and SAM-style segmentation features [72].

Each prediction head is implemented using a lightweight Vision Transformer (ViT) decoder, which operates on two types of tokens produced by the multimodal backbone: the latent embeddings associated with a specific modality and a set of learnable mask tokens used for reconstruction.

To retain spatial correspondence, we inject fixed sine–cosine positional encodings into the token embeddings. These tokens are then processed through several Transformer encoder layers, followed by a modality-specific linear projection head that maps each patch token to its output space, such as per-pixel depth values or semantic logits—thereby reconstructing the expected visual signals of future observations. Concrete details of each module are shown in Table 9.

Action Prediction with Diffusion Transformer To generate future actions conditioned on latent action embeddigns, we adopt a diffusion-based Transformer architecture, DiT-B [117], as our action decoder. DiT enables flexible modeling of complex action distributions by progressively denoising a sequence of latent action tokens through a series of Transformer layers, allowing the model to capture multimodal uncertainty in robot control.

Table 9: The parameters of the each module in DreamVLA.

	Hidden size	Number of layers	Number of heads
image encoder	768	12	12
perceiver resampler	768	3	8
LLM	1024	24	16
image decoder	1024	2	16
depth decoder	1024	2	16
semantic decoder	1024	2	16

We configure the DiT model with the base variant (DiT-B), using an action token embedding size equal to the hidden dimension of the fusion Transformer. The model predicts K future actions, where each action is a 7-dimensional vector that encodes the displacement of the pose and gripper state of the end effector. In our experiments, we set $K = 2$, corresponding to a 3-frame prediction window (current + 2 future steps). The model does not utilize past action context during generation (i.e., past window size is 0), focusing solely on predictive synthesis.

During training, Gaussian noise is added to the future action trajectories, and the model learns to reverse this corruption process step by step. This module operates on top of the aggregated representation via `<action>` query, enabling temporally coherent and semantically grounded action generation. The concrete detail of DiT is shown in Table 10.

Table 10: Configuration of the DiT-B model used for action prediction.

Parameter	**Value**
Model type	DiT-B
Token size	1024
Action prediction window	2 future steps (3-frame chunk)
Past context steps	0
Number of Transformer layers	12
Number of attention heads	12
Positional encoding	Learned (1D for time)
Diffusion timesteps (Train)	8
Diffusion timesteps (Inference)	10
Noise schedule	Linear
Loss function	Denoising Score Matching (L2 loss)
Precision	float32

A.2 Feature Extraction

To facilitate dynamic region prediction, we adopt a motion-based heuristic to generate coarse binary masks that highlight regions of interest. Given a sequence of consecutive RGB frames of resolution $H \times W$, we uniformly sample one keypoint every 8 pixels in both spatial dimensions, resulting in $N = \lfloor H/8 \rfloor \times \lfloor W/8 \rfloor$ sampled locations per frame. For each sampled location, we compute inter-frame displacements $(\Delta x, \Delta y)$ by tracking its position across adjacent frames using CoTracker [69]. The magnitude of displacement is converted into a scalar speed value:

$$s_{ij} = \sqrt{(\Delta x_{ij})^2 + (\Delta y_{ij})^2},$$

where (i, j) denotes the spatial coordinates of each sampled patch. We then apply a speed threshold τ (e.g., $\tau = 1$ pixel/frame) to obtain a binary motion mask. To account for small motions and ensure spatial connectivity, we perform a single-pixel morphological dilation, expanding each positive location to its eight-connected neighbors.

The resulting mask is flattened and reshaped into the form $(B, 1, L)$, where $L = \lfloor H/8 \rfloor \cdot \lfloor W/8 \rfloor$ and B is the batch size. We apply this binary mask element-wise to both predicted patch embeddings $\{\hat{p}_i\}$ and their corresponding ground-truth embeddings $\{p_i\}$ during loss computation, encouraging accurate representation in dynamic regions.

For depth supervision, we use the ground-truth depth maps provided by datasets when available. In cases where depth annotations are not provided—such as in certain real-world robot datasets—we use monocular depth estimators, specifically Depth-Anything v2 [66], to generate pseudo-ground-truth depth labels.

In addition to depth and dynamic signals, we include high-level feature supervision. For DINOv2 [71], we extract features from the final transformer layer, capturing global semantic and structural representations. For SAM [72], we utilize the output of its image encoder as dense segmentation-aware features. These diverse modalities collectively provide comprehensive supervision signals to improve the quality and generalizability of our learned visual representations.

A.3 Training Detail

The total loss can be formulated as:

$$\mathcal{L} = \lambda_{\text{dyn}}\mathcal{L}_{\text{dyn}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{sem}}\mathcal{L}_{\text{sem}} + \lambda_{\text{DiT}}\mathcal{L}_{\text{DiT}} \tag{10}$$

where $\lambda_{\text{dyn}} = 0.1, \lambda_{\text{depth}} = 0.001, \lambda_{\text{sem}} = 0.1, \lambda_{\text{DiT}} = 1.$

We train DreamVLA on 8 NVIDIA A800 GPUs. The main bottleneck is the memory bandwidth to load large spatial feature tensors, for example, of 256×64×64 for SAM. We pre-compute the features from off-the-shelf models instead of conducting inference on the fly. This approach requires extra storage space to save all the features extracted from the above foundation models, but significantly saves on training time and avoids loading models with high GPU memory usage during training. All training configurations are listed in Table 11.

Table 11: DreamVLA Training Configuration

Hyperparameters	**Value**
# GPUs	8
Batch size	8 / GPU (64 effective)
Learning rate (LR)	1e-3
LR Schedule	Constant
Weight decay	0.01
Optimizer	AdamW
Betas	[0.9, 0.999]
Epochs	20
Warm-up epochs	1
Warm-up LR schedule	Linear (1e-2 * LR)

B Experiments

B.1 Simulation Benchmark and Settings

We evaluate DreamVLA on the CALVIN benchmark [119], a simulated robotic manipulation suite designed for studying long-horizon, language-conditioned tasks. CALVIN aims to facilitate the development of agents that operate solely based on onboard sensor inputs and free-form human instructions, without access to privileged information or external supervision. The tasks in CALVIN require agents to execute long sequences of low-level control commands in response to complex language goals, reflecting realistic robotic interaction scenarios.

The benchmark includes four structurally similar but visually distinct environments, referred to as Env A, B, C, and D. Each environment features a Franka Emika Panda arm with a parallel gripper and a tabletop workspace containing manipulable elements such as a sliding door, a drawer, and a light button. The textures, object placements, and scene layouts vary across environments to encourage generalization and robustness.

Observations consist of RGB images from both fixed and gripper-mounted cameras (resized to 224×224), as well as low-dimensional robot state inputs that include the end-effector's position, orientation, and gripper status. The agent outputs a 7-dimensional continuous action vector: 6 dimensions control the spatial displacement of the gripper, and the final dimension governs the open/close state of the gripper.

The dataset contains approximately 2.4 million interaction steps and 40 million short-horizon action windows. Environments A, B, and C provide language-free demonstrations for large-scale pretraining, while annotated instructions are available in a subset of the data for downstream policy learning. We hold out Env D for evaluation to assess zero-shot generalization to unseen combinations of instructions and environment variations.

Following standard protocol [119, 58], we evaluate performance on a set of 34 diverse tasks that include object pushing, placing, rotating, and other dexterous operations. In contrast to prior work, DreamVLA not only predicts actions conditioned on visual-language observations but also simultaneously learns to infer comprehensive future world knowledge, including depth maps, dynamic saliency regions, DINOv2 features, and SAM-based segmentation maps. This multi-task supervision enables richer scene understanding and improves policy generalization. We report success rate (**SR**) as our primary evaluation metric, measuring whether the instructed task was completed correctly based on the final state of the environment.

B.2 Simulation Results

We evaluate our approach on the CALVIN ABC-D benchmark, where training is conducted on environments A, B, and C, and testing is performed exclusively in Environment D. This evaluation setting poses a strong challenge for generalization, as Environment D features novel textures, object arrangements, and visual configurations not seen during training. As reported in Table 1 in the main manuscript, DreamVLA achieves superior performance across all tasks, substantially outperforming previous state-of-the-art methods.

In particular, our model significantly outperforms two-stage inverse dynamics approaches such as Susie [120], demonstrating the effectiveness of our end-to-end architecture that unifies multimodal prediction and action generation. Compared to CLOVER [123], UP-VLA [59], Seer [58], which also incorporates visual foresight, DreamVLA benefits from a more integrated design and joint optimization, resulting in consistently stronger execution accuracy. Furthermore, our method surpasses video generation-based pretraining approaches like GR-1 [16], highlighting the advantage of coupling vision prediction with action planning in a single framework.

Notably, DreamVLA, achieves an average episode length of **4.44** on the ABC-D split, establishing a new state-of-the-art on the CALVIN benchmark and validating the benefits of predicting future knowledge. The qualitative results as shown in Figure 7.

B.3 Visualization

As shown in Figure 8 and Figure 9, we visualize the model's predictions of dynamic regions and depth maps. Although supervision is applied only to dynamic regions, DreamVLA is able to reconstruct semantically meaningful representations of the entire scene. This surprising generalization ability can be attributed to two factors. First, in long-horizon manipulation sequences, the robot arm is in constant motion and frequently interacts with various objects, causing most task-relevant regions to become dynamic at some point in time. This ensures that a large portion of the scene is eventually observed under dynamic supervision. Second, although static regions are not explicitly supervised, the input frames inherently contain global visual context—including background structures, object appearances, and spatial layout—which the model can leverage to hallucinate and complete missing details. As a result, DreamVLA implicitly learns to integrate temporal dynamics with static priors, leading to coherent and accurate predictions beyond the explicitly labeled regions.

Although the predicted depth maps are relatively coarse due to the patch-level reconstruction inherent in MAE-style decoders [106], they still provide valuable guidance for downstream tasks. In particular, the model benefits from anticipating future depth, which helps refine action decisions and improves spatial awareness.

B.4 Additional Ablation Study

Q6: Effect of the query count per modality inside <dream> queries.

Each <dream> query contains three groups of elements: dynamic, depth, and semantics, each assigned K queries. We vary $K \in \{4,\ 9,\ 16\}$ to examine its influence. When $K = 4$, the limited capacity prevents the model from encoding fine-grained motion, geometry, and semantics, so accuracy drops even though memory usage is lowest. With $K = 9$, each modality has sufficient bandwidth without overloading the backbone, yielding the best success rate and the longest uninterrupted task execution. Increasing to $K = 16$ introduces redundant tokens that compete for attention and raise GPU memory, bringing no extra gain and slightly lower generalization.

Table 12: **Performance comparison** between different numbers of <dream> queries.

Number	Task completed in a row					
	1	2	3	4	5	Avg. Len.
4	97.2	92.6	86.4	80.7	75.1	4.32
9	**98.2**	**94.6**	**89.5**	**83.4**	**78.1**	**4.44**
16	98.1	93.0	86.9	81.0	73.9	4.33

B.5 Real-world Settings

In our real-world training setup, we use a history length of 7, with the model jointly predicting the next 3 future visual representations and action steps. The visual backbone is initialized with a ViT-B model pre-trained using MAE [106], and inference is accelerated using bfloat16 mixed-precision without any observed degradation in

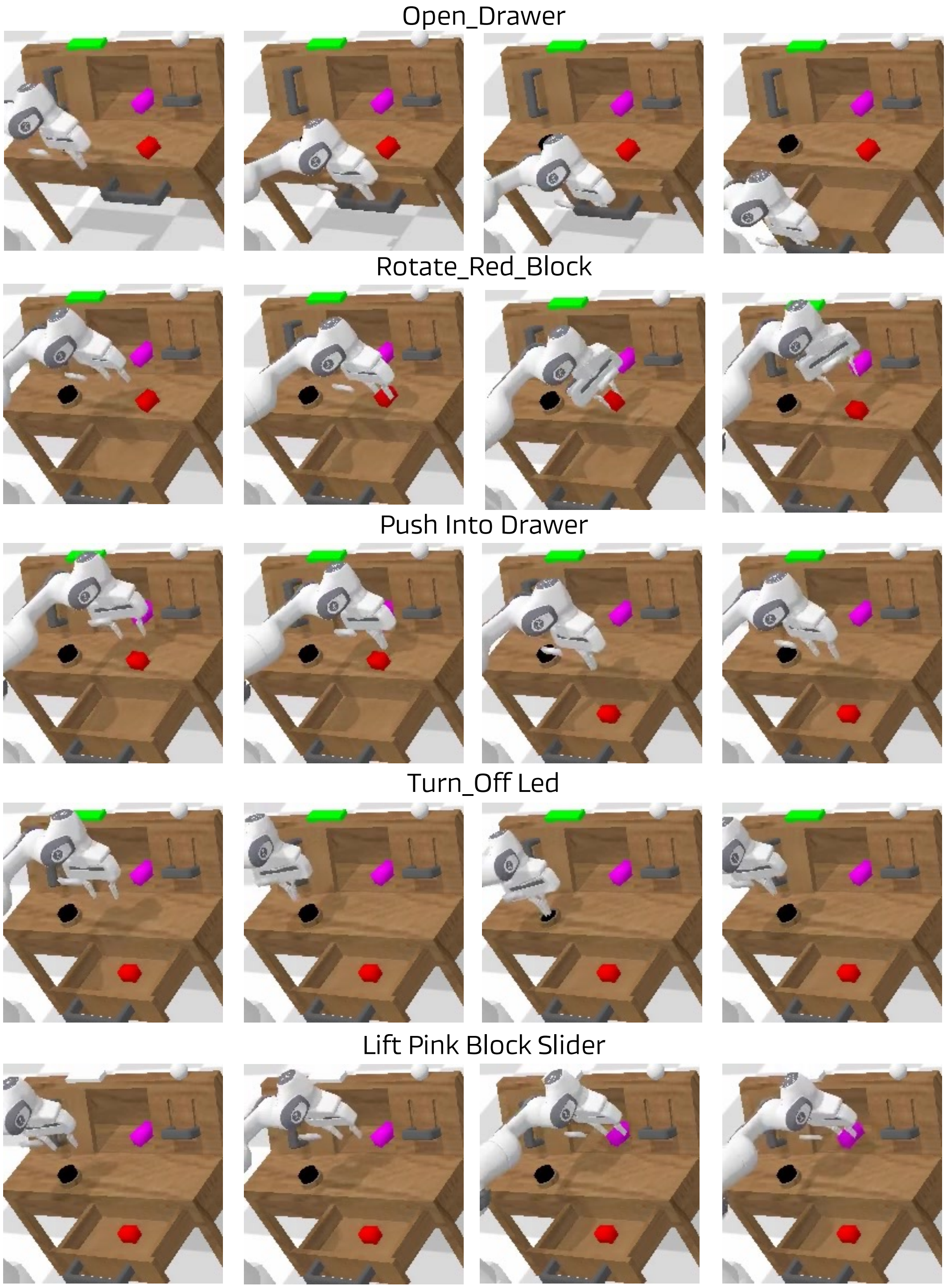

Figure 7: **Qualitative results** of the CALVIN long horizon task.

task performance. This configuration strikes a balance between computational efficiency and policy stability in manipulation tasks.

For pretraining, we leverage a large-scale dataset such as DROID [84], which contains approximately 76,000 successful robot trajectories collected in diverse settings. For downstream adaptation, we fine-tune the model

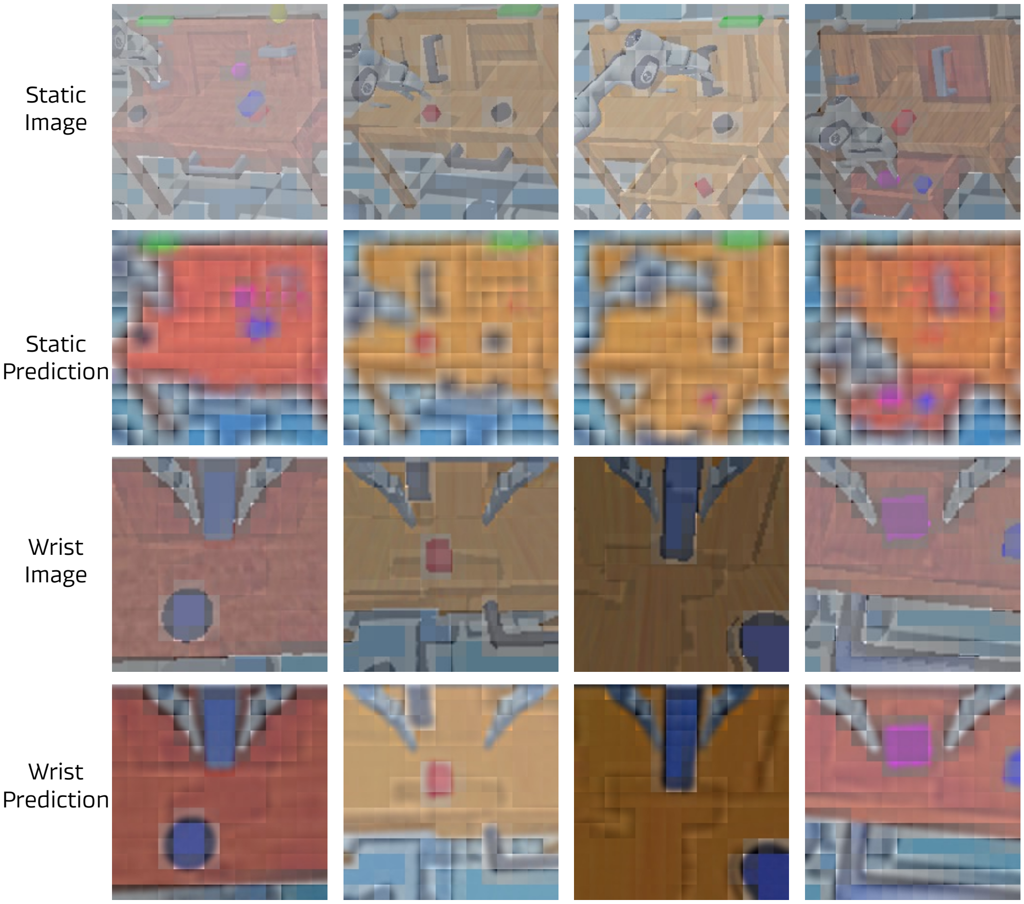

Figure 8: **Visualization results** of the dynamic region predictions.

using 100 task-specific demonstrations for each task collected with SoFar [24]. As shown in Figure 10, we present the qualitative results of real-world experiments.

B.6 Inference latency

model part	inference time
image, text and state encoders	12 ms
observation forward pass w/dream query	19 ms
w/o dream query	16 ms
action forward pass (10 step)	60 ms
total	91 ms
w/o dream query	88 ms

Table 13: Inference time of our model on a NVIDIA GeForce RTX 4090 GPU, we test 5 times and take average time.

Table 13 reports end-to-end latency for processing two camera images on an NVIDIA GeForce RTX 4090. At inference time, no explicit image decoding is required, and the system runs at 11 Hz. The results show: (i) Auxiliary cues incur minimal overhead. Our "dream queries" are token-level predictions (no explicit image decoding and no external models). The incremental cost is 3 ms (3.4%), i.e., 91 ms vs. 88 ms without dream queries. (ii) Latency is dominated by the action head rather than the auxiliary cues. A 10-step action head contributes about 60 ms. This cost scales with the number of sampling steps and model size; it can be reduced by using fewer steps, a smaller DiT variant, faster samplers. (iii) For latency-critical applications, we can prune

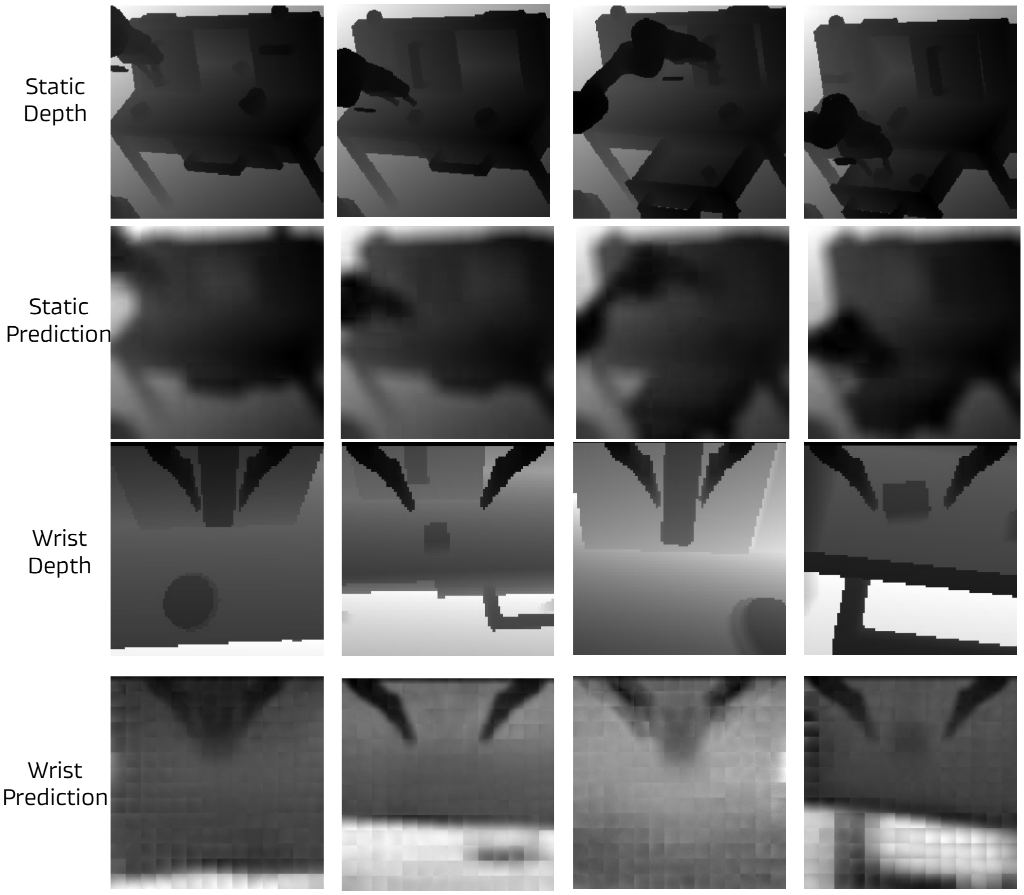

Figure 9: **Visualization results** of the depth maps.

dream queries (e.g., keep only dynamic regions, or dynamic+depth) and/or increase action chunking or run the action head asynchronously to amortize computation, without changing the observation pathway.

C Additional Related Works

C.1 Language-Grounded Robot Manipulation

Language-grounded robot Manipulation adopts the human language as a general instruction interface. Existing works can be categorized into two groups: **i)** *End-to-end* models like RT-series [2, 86, 87] built upon unified cross-modal Transformers with tokenized actions [74, 126–128, 32, 129], large vision-language-action (VLA) models built from VLMs [1], or 3D representations [130, 46, 131]. Training on robot data such as Open X-Embodiment [14] and DROID [84], a remarkable process has been made. However, the data scale is still limited compared to in-the-wild data for training VLMs. **ii)** *Decoupled* high-level reasoning and low-level actions in large vision-language models and small off-the-shelf policy models, primitives [132–138, 24], or articulated priors [139, 140].

D Limitation & Future Works

While DreamVLA demonstrates solid vision-language-action and achieves state-of-the-art performance on CALVIN [119], its current scope is still narrow: it practises mainly parallel-gripper manipulation, relies on RGB-centric data, and is trained on scenes with limited geometric and material diversity. We therefore plan to (i) add dexterous-hand demonstrations with rich contact annotations [141, 142], (ii) introduce 3D point clouds [143, 144, 104, 68, 145, 146, 67, 147] and spatial information [24, 148], tactile—and fuse them into volumetric world states, and (iii) extend data collection and on-policy fine-tuning to bolster generalization and long-horizon robustness.

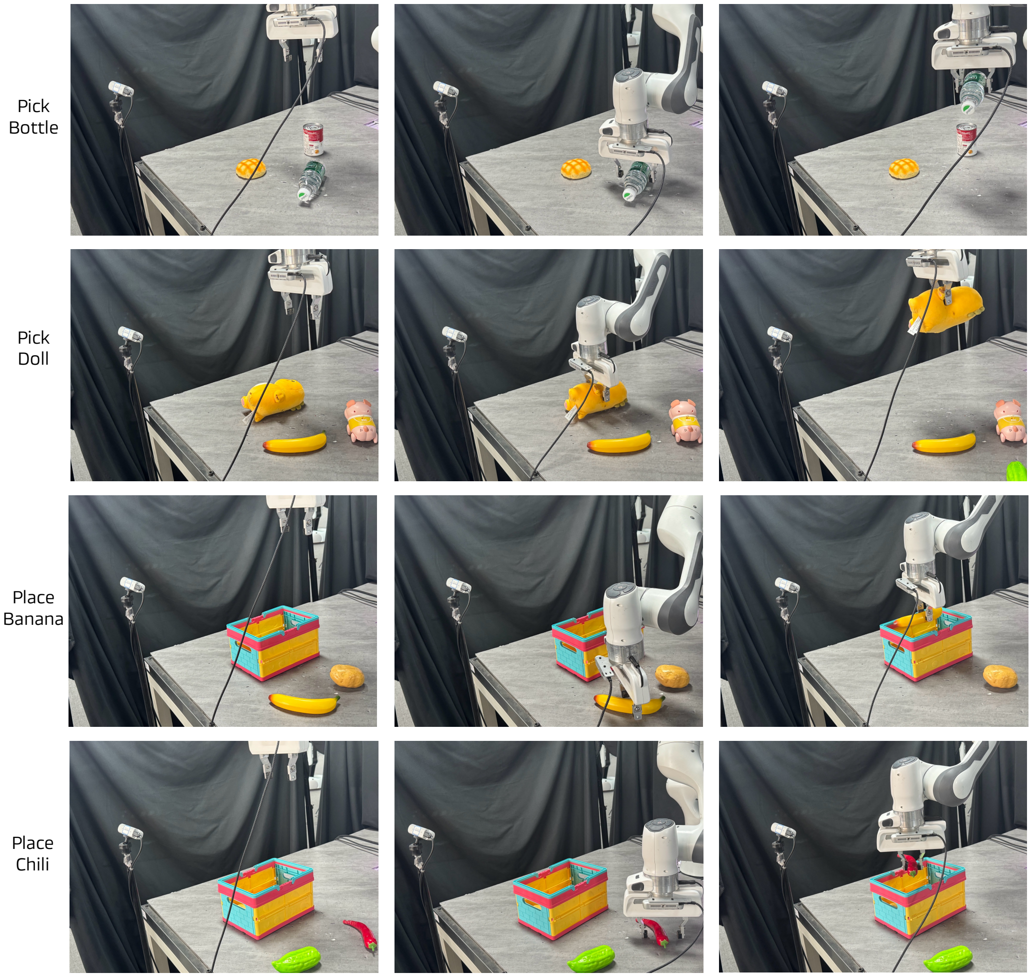

Figure 10: **Qualitative results** of real world language-grounded manipulation.

E Additional Discussions and Future Work

i. Scaling Laws. A promising direction for future exploration involves investigating scaling behavior in DreamVLA. In particular, we plan to study how increasing the capacity of key components—such as the backbone visual encoder or the size of the language model—affects model performance. This includes replacing the current text encoder with larger-scale language models (e.g., LLaMA-2 or GPT variants) to assess the impact of richer linguistic understanding on multimodal reasoning and action generation.

ii. Integration with Additional Baselines. We also aim to evaluate DreamVLA in conjunction with more recent and diverse baselines. For example, RoboVLMs [39] incorporate a wide range of vision-language backbones and offer a unified framework for robotic policy learning. Combining DreamVLA with these baselines can help standardize performance comparisons and reveal architectural synergies between pretrained vision-language models and action-centric transformers.

iii. Contribution of Multi-View Observations. Our current framework leverages both fixed and egocentric camera views. In future work, we plan to conduct a detailed ablation study to quantify the contribution of each view modality to task performance. This analysis will provide insights into how multi-view information improves spatial reasoning and robustness, especially in occluded or ambiguous scenarios.

iv. Extension to More Complex and Long-Horizon Tasks. While DreamVLA demonstrates strong performance on the CALVIN benchmark, we are interested in extending the framework to more complex, long-horizon tasks that involve extended temporal dependencies, delayed rewards, and multi-stage subgoals. This includes evaluating on benchmarks that require sustained interaction, sequential tool use, or high-level planning. Addressing these challenges will require not only more powerful temporal modeling but also better integration of memory, goal abstraction, and hierarchical reasoning mechanisms.

v. Application to Robotic Navigation and Humanoid. Beyond tabletop manipulation, DreamVLA could be adapted to robot navigation tasks in indoor or semi-structured environments. By learning to predict dynamic regions, obstacles, and semantic scene components, the model could support instruction-driven navigation and path planning under multimodal supervision, especially in settings where map-based planning is infeasible.

Furthermore, another compelling extension is applying DreamVLA to humanoid robots, which require reasoning over whole-body motion, balance, and physically grounded interactions. The modularity of our framework allows for integration with additional proprioceptive inputs and more complex action spaces. This line of work would explore how multimodal predictive learning can scale to full-body motor control and human-like task execution.

F Broader Impacts

DreamVLA proposes a new training paradigm for vision-language-action (VLA) modeling, going beyond the conventional mapping from visual observations and language to actions. Instead of directly predicting actions from high-dimensional input, our framework first encourages the model to predict comprehensive world knowledge, including depth, dynamic motion, segmentation, and semantic features, before generating actions. This intermediate representation improves action grounding and generalization.

A key strength of DreamVLA lies in its simplicity and efficiency: by adding only a lightweight decoder and a set of learnable queries, we significantly enhance the performance of existing VLA backbones with minimal parameter overhead. This makes the method both scalable and compatible with current VLM-based architectures, paving the way for more robust and transferable policies.

Practically, this design can benefit the development of assistive robots' navigation and humanoid robots, where it is essential for agents to generalize across novel environments and language goals. Furthermore, since our method leverages unlabeled perceptual signals during training, it reduces reliance on curated language-instruction datasets, which are often expensive and domain-specific.

Overall, DreamVLA offers a practical, extensible, and training-efficient framework for improving VLA systems, and we hope it inspires further research into multimodal abstraction and low-cost robot learning.