DUAL-STREAM DIFFUSION FOR WORLD-MODEL AUGMENTED VISION-LANGUAGE-ACTION MODEL

Anonymous authors
Paper under double-blind review

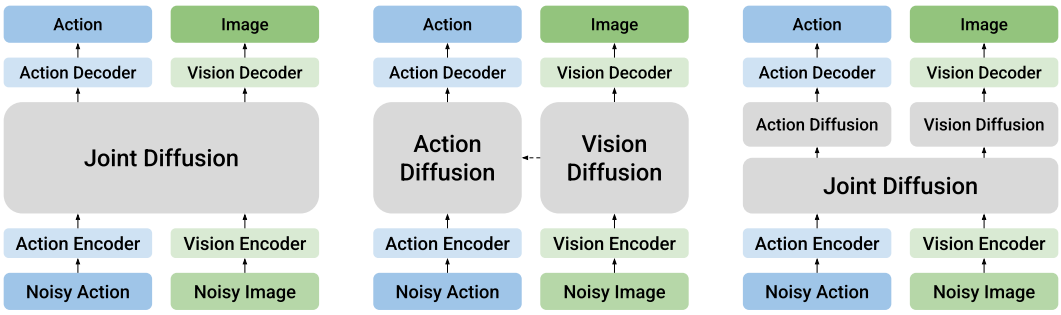
ABSTRACT

Recently, augmenting Vision-Language-Action models (VLA) with world modeling has shown promise in improving robotic policy learning. However, it remains challenging to jointly predict next-state observations and action sequences because of the inherent difference between the two modalities. To address this, we propose DUal-STream diffusion (DUST), a world-model augmented VLA framework that handles the modality conflict and enhances the performance of VLA models across diverse tasks. Specifically, we propose a multimodal diffusion transformer architecture that explicitly maintains separate modality streams while still enabling cross-modal knowledge sharing. In addition, we introduce independent noise perturbations for each modality and a decoupled flow-matching loss. This design enables the model to learn the joint distribution in a bidirectional manner while avoiding the need for a unified latent space. Based on the decoupling of modalities during training, we also introduce a joint sampling method that supports test-time scaling, where action and vision tokens evolve asynchronously at different rates. Through experiments on simulated benchmarks such as RoboCasa and GR-1, DUST achieves up to 6% gains over baseline methods, while our test-time scaling approach provides an additional 2–5% boost. On real-world tasks with the Franka Research 3, DUST improves success rates by 13%, confirming its effectiveness beyond simulation. Furthermore, pre-training on action-free videos from BridgeV2 yields significant transfer gains on RoboCasa, underscoring DUST's potential for large-scale VLA pretraining.

1 INTRODUCTION

Vision-Language-Action models (VLAs) have recently emerged as powerful candidates for general robotic policies (Black et al., 2025; NVIDIA et al., 2025). These models build on the representational power of Vision-Language Models (VLMs) pretrained on internet-scale multimodal datasets by finetuning on robotics datasets, enabling them to generate actions that transfer to novel objects, scenes, and instructions (Zawalski et al., 2024). Despite strong perception and instruction grounding, they fail to model the dynamics of how actions affect the environment, and thus lack an explicit understanding of underlying physical processes. To address this, recent works have augmented VLAs with world modeling objectives, which teach the model to additionally predict future visual observations (Guo et al., 2024; Zheng et al., 2025; Liang et al., 2025). This joint prediction enables the model to more effectively capture the dynamics that govern both actions and their visual consequences, resulting in improved performance and generalization.

Existing works that augment VLAs with world modeling such as PAD (Guo et al., 2024) and EnerVerse (Huang et al., 2025) utilize unified joint diffusion model structures (See Figure 1a), where the two modalities are concatenated together and modeled with a single unified model. However, this implicitly assumes that the two modalities share a common latent space, which forces the model to reconcile two fundamentally different objectives of predicting low-dimensional temporally smooth action trajectories and reconstructing high-dimensional spatially structured visual observations. Because of this heterogeneity, a joint latent space requires high capacity encoder–decoder networks to preserve reconstruction accuracy. In contrast, approaches such as Video Policy (Liang et al., 2025) and Video Prediction Policy (Hu et al., 2025) adopt a causal diffusion design (See Figure 1b) that separates the two modalities into distinct models with unidirectional conditioning. While this respects the structural differences between modalities, it restricts information flow to a single di-



(a) Unified Joint Diffusion Model (b) Causal Diffusion Model (c) Dual-Stream Diffusion Model

Figure 1: **Architectures for world modeling augmentations to VLAs.** (a) Unified Joint Diffusion concatenates action and vision tokens and generates both with a single model. (b) Causal Diffusion uses separate models with one-way conditioning. (c) Dual-Stream Diffusion maintains separate streams for each modality while enabling cross-modal knowledge transfer through shared attention.

rection and prevents bidirectional knowledge transfer. These two design choices highlight a trade-off between cross-modal integration and modality-specific fidelity. To bridge these contrasting approaches, we propose DUal-STream Diffusion (**DUST**), a model for VLAs that preserves distinct modality streams yet facilitates information exchange across them (See Figure 1c).

DUST introduces a dual-stream multimodal diffusion transformer that maintains distinct pathways for action and vision tokens, while enabling interaction through shared attention layers. On top of this architecture, we develop a decoupled diffusion training algorithm that independently noises each modality and optimizes over a decoupled flow-matching loss, so that actions and observations can be optimized according to their different statistical structures, and allow the model to capture cross-modal causal relationships. Building upon the decoupled noising, we also propose a joint sampling method that allows test-time scaling via asynchronous denoising of the two modalities.

We validate DUST across simulated, real-world, and transfer learning settings. In simulation, DUST outperforms baselines such as GR00T-N1 and FLARE on the RoboCasa and GR-1 benchmarks, with consistent gains across all task categories and data scales, with up to 5% improvement on RoboCasa and 6% on GR-1. On a real Franka Research 3 arm, DUST achieves the highest success rates across diverse pick-and-place tasks, obtaining performance boosts of over 12% compared to baselines. In addition, by pretraining on action-free video from BridgeV2, DUST substantially improves performance while finetuning on downstream RoboCasa tasks, showing that it can effectively leverage large-scale passive video data for data-efficient policy learning. Finally, we experiment with our asynchronous joint diffusion sampling approach to test-time scaling, showing 2-6% advantage over naive sampling methods. Together, these results highlight DUST's scalability, real-world effectiveness, and ability to transfer knowledge from video pretraining.

2 Related Works

Vision-Language-Action models (VLAs). Vision–Language–Action models (VLAs) have recently emerged as a promising paradigm for general robot policy learning, leveraging Vision–Language models (VLMs) trained on internet-scale datasets. Building on the strong representational capacity of VLMs (Dai et al., 2023; Team, 2024; Xiao et al., 2024), VLA architectures adapt them for robotics tasks by either autoregressive action token generation (Kim et al., 2024; Brohan et al., 2023; Wu et al., 2024; Cheang et al., 2024) or diffusion model-based action experts (Black et al., 2025; NVIDIA et al., 2025). We choose diffusion modeling for the action generation setup. Beyond these designs, extensions include cross-embodiment latent action modeling (Ye et al., 2025; Bu et al., 2025) and reasoning-oriented approaches for complex task execution (Zawalski et al., 2024). Despite these advances, most methods focus only on action distribution approximation methods that learn expert demonstrations without explicitly capturing the underlying physical dynamics. In contrast, our approach integrates a world-modeling component that captures future physical dynamics, which enables more effective action generation.

World-modeling for robotic policy learning. Prior work has augmented VLAs by incorporating world modeling, by representing future states alongside action generation. One line of research, including PAD (Guo et al., 2024) and EnerVerse (Huang et al., 2025), employs unified architectures that jointly model future images and actions through diffusion, as shown in Figure 1a. UWM (Zhu et al., 2025) extends upon this by introducing modality-specific time schedules, while FLARE (Zheng et al., 2025) introduces implicit world-modeling, where by aligning mid-level features to future image embeddings instead of directly diffusing them. UVA (Li et al., 2025a) first embeds both modalities into a joint latent space before using modality-specific decoders to project back to their native state spaces. A complementary direction, pursued by Video Policy (Liang et al., 2025), Video Prediction Policy (Hu et al., 2025), adopts architectures with disjoint components that permit only unidirectional conditioning, as shown in Figure 1b.

Another key design choice concerns how future states are represented. A common approach, as in PAD (Guo et al., 2024), PIDM (Tian et al., 2025), and This&That (Wang et al., 2024) is to directly reconstruct the RGB image observation after executing the generated action chunk. In contrast, works such as DINO-WM (Zhou et al., 2024) and FLARE (Zheng et al., 2025) replace raw image prediction with the generation of future observation embeddings, extracted from pretrained encoders such as DINO-V2 (Oquab et al., 2023) and Q-Former (Li et al., 2023). We adopt the latter strategy, as embedding-level targets emphasize the semantic structure of future states while avoiding the need to reproduce pixel-level details, which are often irrelevant for downstream control but costly to model.

3 PRELIMINARIES

Problem setup. Let $\mathcal{D} = \{T_1, T_2, ...\}$ be the expert demonstration trajectories, where each trajectory $T_i = \{I, \{(O_t, A_t)\}_{t=0}^{L}\}$ consists of task instruction I and observations O_t and action sequences A_t. In specific, we denote the observations at timestep t as $O_t = (o_t^v, o_t^s)$, where o_t^v is the visual observation and o_t^s is the robot proprioceptive state. Actions are grouped in chunks (Zhao et al., 2023; Chi et al., 2023) such that $A_t = (a_t, a_{t+1}, ..., a_{t+k-1})$ where k is the size of chunk. Our goal is to train a model that predicts A_t given the observations O_t and instruction I.

Vision-Language-Action model (VLA). To achieve this, we follow the common practice introduced in recent diffusion-based VLA models (Black et al., 2025; NVIDIA et al., 2025). Specifically, we use a pretrained Vision-Language model (VLM; Li et al. 2025b) to extract high-level semantic information from the image observations and text instruction. Then, the extracted representations are used as conditions for the action expert, *e.g.*, through cross-attention layers in diffusion transformers (DiT; Peebles & Xie 2022) for action prediction.

The action expert is trained by using Flow Matching objective (Lipman et al., 2023). Formally, given the action sequence A_t, we sample a random timestep $\tau \in [0, 1]$ and Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to create noisy action $A_t^\tau = \tau A_t + (1-\tau)\epsilon$. Let Φ_t be the features extracted from VLM module using the current image observation o_t^v and instruction I. Then, we train a velocity model $V_\theta(\phi_t, A_t^\tau, o_t^s)$ that aims to predict the ground truth velocity field $A_t - \epsilon$ by following Flow Matching loss:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{A_t^\tau, \tau}\Big[\big\|V_\theta(A_t^\tau, \Phi_t, o_t^s) - (A_t - \epsilon)\big\|^2\Big], \tag{1}$$

where we sample $\tau \sim \text{Beta}(\frac{s-\tau}{s}; 1.5, 1.0)$ with $s = 0.999$ following common practice (Black et al., 2025; NVIDIA et al., 2025). During inference, we sample noise $A_t^0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, then use Euler's method to generate action chunks over N_A denoising steps as follows:

$$A_t^{\tau+\Delta\tau} = A_t^\tau + V_\theta(\Phi_t, A_t^\tau, o_t^s)\Delta\tau, \quad \text{where } \Delta\tau = 1/N_A. \tag{2}$$

World-modeling. The goal of world-modeling is to learn through the prediction of future states. We consider predicting future image observation o_{t+k}^v, which is obtained by executing the action chunk A_t with chunk size k. However, direct pixel-level prediction may lead to focusing on learning of high-frequency details that are irrelevant for policy learning. To this end, we aim to reconstruct the representation of a future image observation instead, which we obtain by re-using the vision encoder in our VLM. We denote $\tilde{o}_{t+k}$ to be the future image embedding, and our world-modeling goal is to predict this, conditioned on VLM features Φ_t and proprioceptive state o_t^s.

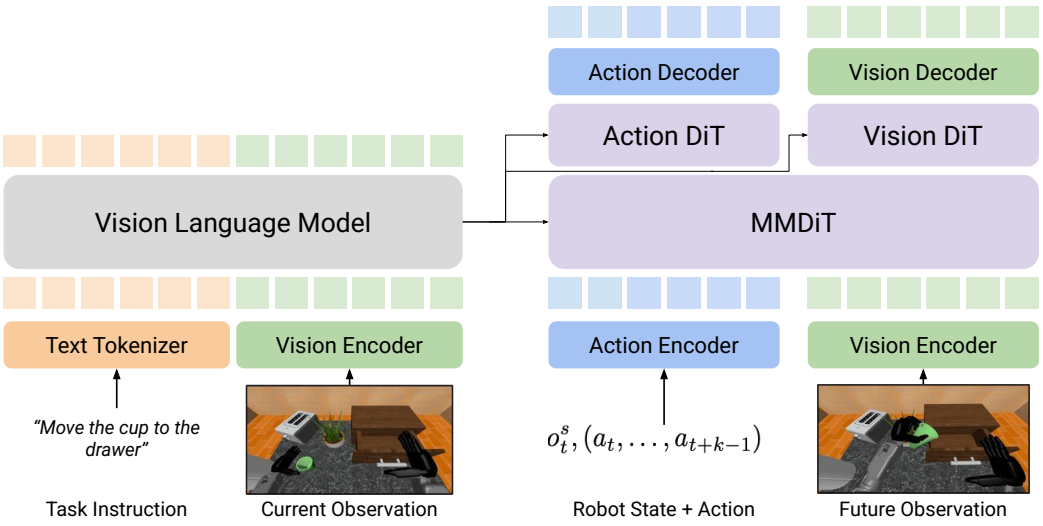


Figure 2: **Dual-Stream diffusion (DUST) architecture.** Our architecture has **(1)** VLM model VLM$\phi(\cdot)$ that processes current observation and task instruction to produce semantic representations, and **(2)** diffusion model π_θ which conditions on these representations to generate actions and future observation embeddings. π_θ comprises a MM-DiT stack that preserves distinct action and vision streams, followed by modality-specific DiT decoders.

4 METHOD

In this section, we present the *DUal-STream diffusion (DUST)* model for joint world modeling and action prediction. We first introduce the design of our method, which builds on a Multi-modal diffusion transformer (MMDiT; Esser et al. 2024) in Section 4.1. Then we propose a decoupled training algorithm that uses independent noise scheduling in Section 4.2. Lastly, we introduce an inference-time scaling method that jointly samples the action and future visual states in Section 4.3.

4.1 DUST ARCHITECTURE

In addition to the action prediction model V_θ, we aim to predict future visual state $\tilde{o}_{t+k}$ using diffusion modeling. To this end, we modify the original DiT action expert to jointly predict them, where we demonstrate our architecture in Figure 2. Given VLM feature Φ_t, DUST takes the triplet $(o_t^s, A_t^\tau, \tilde{o}_{t+k}^\tau)$ as input, which is composed of the robot proprioceptive state, noised action sequences, and noised future observation embedding. The multimodal sequence input is first passed through a stack of MMDiT blocks. Within each MMDiT block, the two modality streams are propagated separately, concatenated only during the shared attention layer to enable cross-modal exchange, and then split back into their respective pathways. To further decouple their dynamics, each stream receives its own timestep embedding via Adaptive LayerNorm (AdaLN) (Peebles & Xie, 2022), which allows action and vision tokens to be trained with distinct noise levels. After traversing the MMDiT layers, the streams are routed into modality-specific DiT layers for fine-grained denoising. This enables the vision pathway to reconstruct semantically consistent future embeddings and the action pathway to refine low-level trajectories, thereby improving joint modeling of control and world dynamics.

4.2 JOINT TRAINING ALGORITHM

We now introduce a joint training algorithm based on a decoupled diffusion framework. Our design is inspired by Diffusion Forcing (Chen et al., 2025), which trains diffusion models to denoise sequences with independent per-token noise levels. Our setting replaces the per-token noising with *per-modality* noising. Specifically, actions and future image embeddings are noised independently, which leads to valid diffusion objectives for each modality while enabling the model to capture causal relationships between them.

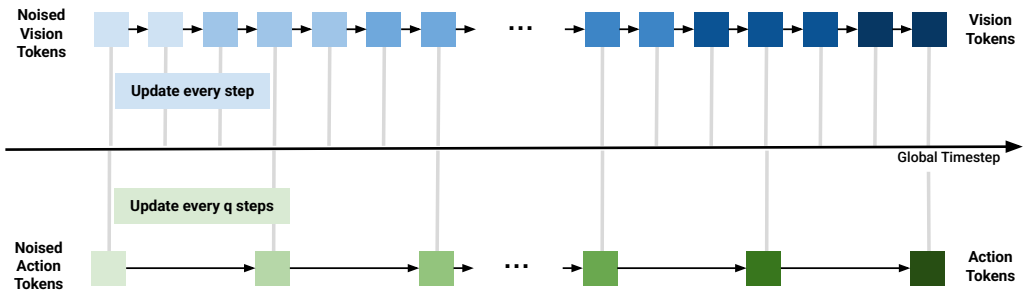


Figure 3: **Overview of vision-action joint sampling.** During inference, we sample over N_A steps for action tokens and $N_o = q \times N_A$ steps for vision tokens. The global timestep advances by $\Delta\tau_o = 1/N_o$, and action tokens are updated only every q steps in $\Delta\tau_A = 1/N_A$ strides. Default q value is 1, and increasing it allows test-time scaling.

Decoupled noise scheduling. Let the two modalities be the action chunk $A_t \in \mathbb{R}^{k \times d_A}$ and the future observation embedding $\tilde{o}_{t+k} \in \mathbb{R}^{d_o}$, where d_A and d_o are the dimensions of the action space and future image embedding, respectively. During training, we sample timestep independently, with $\tau_A \in [0, 1]$ for actions and $\tau_o \in [0, 1]$ for future observations. Let $\epsilon_A, \epsilon_o \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be sampled Gaussian noise, with which we noise A_t and $\tilde{o}_{t+k}$, giving the noisy action sequences and noisy future observations as $A_t^{\tau_A} = \tau_A A_t + (1 - \tau_A)\epsilon_A$ and $\tilde{o}_{t+k}^{\tau_o} = \tau_o \tilde{o}_{t+k} + (1 - \tau_o)\epsilon_o$, respectively. The diffusion model V_θ predicts the velocity field of each modality, conditioned on the VLM feature Φ_t. Let us denote $V_\theta(A_t^{\tau_A}, \tilde{o}_{t+k}^{\tau_o}, \Phi_t, o_t^s) = [V_\theta^A, V_\theta^o]$ be the outputs of diffusion model. Then, the training objective for each action and image observations (*i.e.*, world-modeling) are given as follows:

$$\begin{aligned} \mathcal{L}_{\text{A}}(\theta) &= \mathbb{E}_{A_t^{\tau_A}, \tilde{o}_{t+k}^{\tau_o}} \left[\left\| V_\theta^A - (A_t - \epsilon_A) \right\|^2 \right] \\ \mathcal{L}_{\text{WM}}(\theta) &= \mathbb{E}_{A_t^{\tau_A}, \tilde{o}_{t+k}^{\tau_o}} \left[\left\| V_\theta^o - (\tilde{o}_{t+k} - \epsilon_o) \right\|^2 \right] \end{aligned} \tag{3}$$

To effectively train the model over this joint objective, we adopt the results of Rojas et al. (2025), where we can decompose the joint objective of diffusing two modalities into the sum of unimodal diffusion losses, given that we utilize independent noise injection for each modality. Concretely, we can utilize the following sum of flow matching losses:

$$\mathcal{L}_{\text{Joint}}(\theta) = \mathcal{L}_{\text{A}}(\theta) + \lambda_{\text{WM}} \mathcal{L}_{\text{WM}}(\theta), \tag{4}$$

where $\lambda_{\text{WM}} > 0$ is a weighting hyperparameter for world modeling loss.

4.3 Vision-action joint sampling and inference-time scaling

During inference, we jointly sample actions and vision in parallel, by enjoying bidirectional dependencies in which generated actions constrain plausible future states, and predicted states guide action generation. Yet, the two modalities differ in their requirements, with image diffusion models operating in a high-dimensional space that typically demands many denoising steps, whereas action diffusion converges with far fewer. To address this, we build on DUST's decoupled noising scheme and introduce a test-time scaling strategy, where vision tokens are evolved on a finer timescale than actions through asynchronous forward Euler sampling.

During inference, we first sample noise $A_t^0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_A)$ and $\tilde{o}_{t+k}^0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_v)$. We denote number of diffusion steps for actions tokens N_A and vision observation tokens $N_o = q \times N_A$ $(q \in \mathbb{N})$. We utilize asynchronous forward Euler integration (Figure 3) for the sampling process, with global timestep progressing by $\Delta\tau_o = 1/N_o$, for which vision tokens are updated every time, but action tokens update only every $\Delta\tau_A = 1/N_A = q\Delta\tau_o$.

$$\tilde{o}_{t+k}^{\tau_o + \Delta\tau_o} = \tilde{o}_{t+k}^{\tau_o} + V_\theta^o \Delta\tau_o, \quad A_t^{\tau_A + \Delta\tau_A} = \begin{cases} A_t^{\tau_A} + V_\theta^A \Delta\tau_A & \text{if } (\tau_A N_o \bmod q = 0) \\ A_t^{\tau_A} & \text{otherwise} \end{cases} \tag{5}$$

Since increasing diffusion step count comes as the cost of higher inference time, for our main experiments we set $N_o = N_A = 4$ following (NVIDIA et al., 2025; Zheng et al., 2025). We experiment with test-time scaling via increasing N_o in Section 5.3.

Table 1: **Evaluation on RoboCasa.** Success rates (%) on RoboCasa benchmark for 8 pick-and-place (PnP), 6 contraption open/close (OP/CL), and 10 other miscellaneous tasks. 100, 300, and 1,000 demos are used for training. †: reproduced results.

	100 Demos				300 Demos				1,000 Demos			
Method	PnP	OP/CL	Other	Avg.	PnP	OP/CL	Other	Avg.	PnP	OP/CL	Other	Avg.
GR00T-N1	0.215	0.603	0.468	0.417	0.272	0.660	0.466	0.450	0.323	0.757	0.508	0.508
+ FLARE†	0.230	0.648	0.498	0.446	0.380	0.767	0.562	0.553	0.459	0.837	0.682	0.646
+ DUST	**0.295**	**0.760**	**0.510**	**0.501**	**0.423**	**0.807**	**0.581**	**0.585**	**0.483**	**0.863**	**0.686**	**0.663**

Table 2: **Evaluation on GR-1.** Success rates (%) on GR-1 benchmark for 16 pick-and-place (PnP) and 8 articulated (Art.). 300 and 1,000 demos are used. †: reproduced results.

	300 Demos			1,000 Demos		
Method	PnP	Art.	Avg.	PnP	Art.	Avg.
GR00T-N1	0.176	0.283	0.203	0.307	0.310	0.308
+ FLARE†	0.340	0.330	0.337	0.393	0.324	0.363
+ DUST	**0.358**	**0.367**	**0.360**	**0.422**	**0.413**	**0.420**

Table 3: **Evaluation on real-world tasks.** Success rates (%) of 4 pick-and-place (PnP) tasks for real-world Franka robot experiments. See Fig. 4 for the task instructions. †: reproduced results.

Method	Task 1	Task 2	Task 3	Task 4	Avg.
GR00T-N1	0.583	0.750	0.500	0.354	0.547
+FLARE†	0.625	0.729	0.500	0.375	0.557
+DUST	**0.833**	**0.792**	**0.625**	**0.458**	**0.677**

5 EXPERIMENTS

In this section, we empirically assess the effectiveness of DUST. Section 5.1 presents results from simulated environments (RoboCasa, GR-1) and real-world (Franka Research 3) tasks. In Section 5.2, we investigate transferability by pretraining on action-free video data from BridgeV2, followed by finetuning on the RoboCasa benchmark to assess whether learned world modeling capabilities can be transferred from video data to robot tasks. In Section 5.4, we analyze the various components of our methodology through ablation studies.

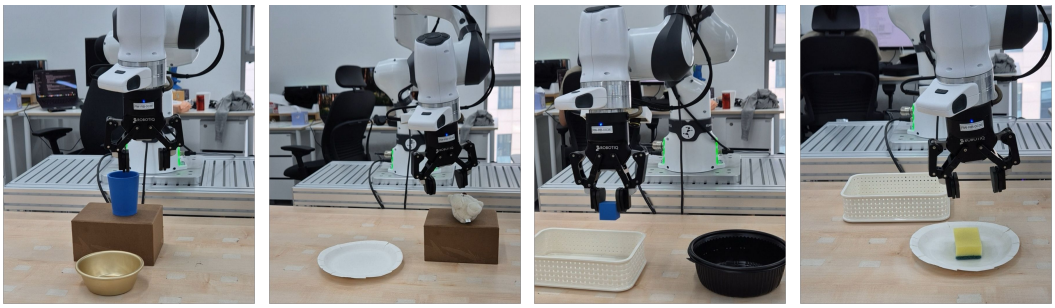
Implementation. For the vision–language model (VLM), we adopt the Eagle-2 model (Li et al., 2025b), which processes both image observations and task instructions. Semantic features are extracted from the 12th layer of the VLM and used as conditioning signals for the diffusion module. The diffusion backbone consists of 12 MM-DiT blocks, followed by 4 DiT blocks per modality, dedicated respectively to future image embedding and action generation. Conditioning with VLM features is applied in an interleaved manner, with alternating self-attention and cross-attention layers. Future image embeddings are derived from SIGLIP-2 (Tschannen et al., 2025) representations of future image observations produced by the Eagle 2 model. Each image yields 256 tokens, which are reduced to 64 tokens via 2×2 average pooling. In total, the diffusion module operates on 1 state token, 16 action tokens, and 64 future image tokens. We utilize 1.0 as the λ value for weighing the loss function, giving equal weight to the world-modeling and action-modeling components.

Baselines. Our primary baselines are the vanilla GR00T-N1 model (NVIDIA et al., 2025), which is currently the state-of-the-art VLA model, and a variant trained with FLARE loss (Zheng et al., 2025). Due to a lack of code release, our FLARE baseline was carefully reimplemented to use the same VLM backbone and world-modeling target as DUST (see Section A.2 for details). For fair comparison, all models are trained with a frozen pretrained VLM module and with the diffusion action expert module randomly initialized.

5.1 MAIN RESULTS

First, we verify the efficacy of DUST across 2 simulated environments and 1 real-world setting. For the simulated setting, we utilize RoboCasa (Nasiriany et al., 2024) and GR-1 (NVIDIA et al., 2025) as our benchmarks, each representing single robot arm manipulation and humanoid manipulation. For the real-world setting, we propose 4 pick-and-place tasks with the Franka Research 3 robot arm.

RoboCasa kitchen. RoboCasa is a single arm manipulation benchmark with a focus on kitchen environment interaction tasks. We utilize a suite of 24 tasks, including turning sink faucets, closing drawer doors, and moving objects. The training dataset is drawn from the publicly available dataset from RoboCasa. We experiment over 100, 300, and 1000 training episodes per tasks.



(a) PnP Task 1 (b) PnP Task 2 (c) PnP Task 3 (d) PnP Task 4

Figure 4: **Real-world task instructions.** For the real-world experiments, we utilize 4 pick-and-place tasks with the Franka Research 3 robot. The tasks are categorized by their distinct source-target pairs (box, bowl, plate, etc.) and each contains 4 different objects (cup, doll, cube, sponge).

GR-1 tabletop tasks. GR-1 is a humanoid robot benchmark with a focus on dexterous tabletop manipulation of everyday objects. We utilize a total of 24 tasks, mostly comprised of pick-and-place tasks, with some tasks having additional articulated requirements, such as closing a drawer or microwave. The training dataset is taken from GR00T-N1 (NVIDIA et al., 2025). We experiment over 300 and 1000 training episodes per task.

Real-world setup. We conduct real-world experiments using a 7-DoF Franka Research 3 robotic arm, where both state and action spaces are parameterized by the arm's joint positions together with a binary gripper state. Evaluation is performed on a suite of four pick-and-place tasks in a tabletop setting, where each task is defined by the distinct source–target configuration. Within every task we evaluate across four different object categories (doll, cup, box, sponge) to capture variations in geometry, size, and physical properties. The training corpus consists of 60 expert demonstrations per task, gathered via teleoperation on the same Franka platform.

Simulation results. Tables 1 and 2 show that DUST consistently outperforms GR00T-N1 and GR00T-N1+FLARE across both RoboCasa and GR-1 benchmarks, covering all task categories and demonstration scales. On RoboCasa with 100 demonstrations per task, DUST improves the average success rate by 18% over GR00T-N1 and 5% over FLARE, and this advantage remains as the number of demonstrations increases, confirming both data efficiency and scalability. On GR-1, a more challenging benchmark, DUST again surpasses both baselines at 300 and 1000 demonstrations, yielding improvements in both task categories.

Real-world results. Table 3 presents results on Franka robot pick-and-place tasks. DUST consistently outperforms prior methods, achieving the highest success rate on every task with average improvements of 13These gains, observed across diverse object types and source–target configurations, demonstrate DUST's robustness in physical environments and its promise for practical deployment. As illustrated in Figure 5, the incorporation of world modeling enables the policy to anticipate the future end-effector pose and accurately align with the target object.

5.2 Transfer Learning

Collecting high-quality teleoperated robot demonstrations is expensive and labor-intensive, while vast amounts of action-free video can be gathered at minimal cost through human recordings or internet-scale crawling (Ye et al., 2025; Dass et al., 2023; Wang et al., 2025). Leveraging such large-scale video datasets allows models to acquire generalizable representations of object dynamics and scene evolution without relying on low-level action annotations. DUST's dual-stream architecture is naturally suited for this

Table 4: **Evaluation for transfer learning.** Success rates (%) on RoboCasa with or without BridgeV2 video data pretraining.

Method	Video Pretrain	PnP	OP/CL	Other	Avg.
GR00T-N1	✗	0.215	0.603	0.468	0.417
+ **DUST**	✗	0.295	0.760	0.510	0.501
+ **DUST**	✓	**0.423**	**0.807**	**0.581**	**0.585**

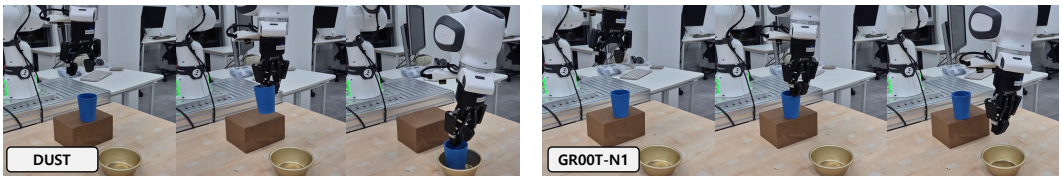


Figure 5: **Qualitative comparison on a real-world pick-and-place task** (Instruction : "*Pick up the blue cup on the brown box and place it in the golden bowl.*") The sequence on the right shows GR00T-N1, which directly generates action sequences, and on the left is DUST, which incorporates explicit world modeling. While GR00T-N1 produces actions that bring the gripper near the cup, it fails to align precisely with the rim and is unsuccessful in grasping. By contrast, DUST leverages its internal prediction of future states by estimating where generated actions will position the gripper, allowing it to consistently adjust and achieve alignment with the desired position for grasping.

Table 5: **Results of test-time scaling with asynchronous joint sampling.** Success rates (%) on RoboCasa and GR-1 with our test-time scaling approach using asynchronous joint sampling. For scaling, we increase N_o, the number of diffusion steps for vision tokens.

	RoboCasa 100 demos				RoboCasa 1000 demos				GR-1 1000 demos		
N_o	PnP	OP/CL	Other	Avg.	PnP	OP/CL	Other	Avg.	PnP	Art.	Avg.
4	0.295	0.760	0.510	0.501	0.483	0.863	0.686	0.663	0.422	0.413	0.420
16	0.308	0.733	0.524	0.504	0.498	0.856	0.690	0.668	0.447	0.463	0.451
32	0.248	0.753	0.568	0.508	0.501	0.868	0.724	0.686	0.471	0.472	**0.471**
64	0.290	0.770	0.548	**0.518**	0.509	0.881	0.736	**0.697**	0.430	0.511	0.450

setting, as it enables pretraining on action-free video to accumulate world-modeling knowledge prior to finetuning as a policy, thereby bridging the gap between inexpensive large-scale video data and costly teleoperated robot data.

During the pretraining stage, the model is trained exclusively on the video component of the BridgeV2 dataset (Walke et al., 2023), optimizing only the world-modeling term of the flow-matching loss while randomly initializing the action tokens. After pretraining, we finetune the model on the RoboCasa dataset using 100 demonstrations per task. Table 4 shows that incorporating video pretraining yields substantial gains, with DUST achieving an average success rate of 0.585 compared to 0.501 without pretraining. These results highlight that large-scale passive video data can effectively transfer to downstream policy learning, improving data efficiency and generalization while reducing dependence on expensive robot demonstrations.

5.3 Test-time scaling for joint sampling

While our main experiments adopt the same number of diffusion steps for both actions and vision, this symmetry may not be optimal. The higher dimensionality and structural complexity of image embeddings typically requires more denoising iterations than the lower-dimensional and temporally smooth action tokens. To account for this, we introduce a test-time scaling strategy in which vision tokens are allocated additional diffusion steps while action tokens steps are fixed, thereby enabling finer-grained refinement of visual representations. Specifically, we follow the asynchronous joint sampling procedure outlined in Section 4.3. We increase the number of vision denoising steps N_o from its default value of 4 to 16, 32, and 64, while keeping the number of action token steps fixed at $N_A = 4$. Experiments are conducted using DUST checkpoints finetuned on RoboCasa with 100 and 1000 demonstrations per task, as well as GR-1 with 1000 demonstrations per task.

As shown in Table 5, increasing the number of vision denoising steps leads to mostly steady performance gains up to 64 steps. On RoboCasa, we observe improvements of roughly 2–3% at 64 steps, while on GR-1 the best results occur at 32 steps, yielding a 5% gain. These findings indicate that allocating additional diffusion steps to vision tokens can substantially enhance VLA performance by allowing more precise refinement of visual representations. However, the improvements come at the expense of higher inference time, highlighting a tunable trade-off between efficiency and accuracy. Further ablations on the role of modality decoupling in this process are provided in Section A.1.

Table 6: **Ablation study.** Success rates (%) on RoboCasa benchmark with 100 demos/task (a) ablating over architecture and training algorithm, (b) depth of MMDiT, and (c) the loss weigh λ_{WM} for world-modeling loss.

(a) Architectural and training

Arch.	Noise	PnP	OP/CL	Other	Avg.
DiT	Joint	0.240	0.633	0.340	0.380
DiT	Decoupled	0.248	0.613	0.454	0.425
MMDiT	Joint	0.160	0.677	0.382	0.382
MMDiT	Decoupled	0.295	0.760	0.510	0.501

(b) MMDiT depth

Layers	Avg.
6	0.474
10	0.483
12	0.501
14	0.493

(c) Effect of λ_{WM}

λ_{WM}	Avg.
0.2	0.343
0.5	0.489
1.0	0.501
2.0	0.496

5.4 Ablation study

DUST components analysis. We next conduct an ablation study to disentangle the contributions of DUST's two core design elements: the dual-stream MMDiT architecture and decoupled training algorithm. To this end, we evaluate three alternative configurations: (1) a baseline DiT model trained with a uniform noise schedule applied jointly to both action and vision tokens, serving as a standard single-stream reference, (2) a DiT model with decoupled noising, where AdaLN conditioning is applied independently to each modality, but the token streams still share a single feed-forward pathway, and (3) an MMDiT model with uniform noise levels, corresponding to the unmodified multi-stream MMDiT architecture with separate actions and vision streams. This design allows us to isolate the relative benefits of modality-specific noise schedules and of the dual-stream transformer structure itself. Results on RoboCasa with 100 demonstrations per task (Figure 6a) show that both components are indispensable. Removing the dual-stream MMDiT structure results in a performance drop of approximately 8%, while removing decoupled noise leads to an even larger 12% reduction. These findings confirm that the two design choices contribute complementary gains, with MMDiT enabling structured cross-modal representation learning, while decoupled noising allows each modality to evolve under dynamics appropriate to its scale and complexity.

Loss weight hyperparameter λ_{WM} and MMDiT layer count. Next, we analyze the effect of the loss weighting coefficient λ_{WM}, which balances the two flow-matching terms in our objective. Larger λ_{WM} values emphasize world modeling, while smaller values emphasize action modeling. As shown in Figure 6c, experiments on RoboCasa with 100 demonstrations per task indicate that performance remains stable in the range $\lambda_{\text{WM}} \in [0.5, 2.0]$, but degrades when moving outside this interval. This suggests that effective learning requires weighting the two objectives relatively evenly. Next, we study the ratio of MMDiT to DiT layers. Fixing the total number of layers in π_θ to 16, we vary the number of MMDiT layers to adjust the trade-off between cross-modal knowledge transfer and per-modality specialization. Results (Figure 6b) show that while performance is generally stable across configurations, the best outcome is obtained with 12 MMDiT layers and 4 DiT layers, highlighting the benefit of heavily leveraging cross-modal processing.

6 Conclusion

In this work, we introduced DUal-STream Diffusion (DUST), a world-model augmented VLA framework that decouples the diffusion of actions and future observations while still enabling cross-modal knowledge transfer. By maintaining separate modality streams linked through shared attention, DUST avoids the limitations of a unified latent space and captures causal dependencies between modalities. Extensive experiments show that DUST consistently outperforms baselines on both simulated benchmarks (RoboCasa, GR-1) and real-world Franka Research 3 tasks, underscoring its scalability and robustness. Beyond architecture and training, we also proposed a test-time scaling strategy with asynchronous joint sampling, which further improves performance by allocating finer-grained diffusion to high-dimensional vision tokens. Finally, pretraining on action-free video (BridgeV2) demonstrates that DUST can exploit large-scale passive data for efficient transfer to downstream robotics. Together, these contributions establish DUST as a versatile and extensible framework for bridging world modeling, video pretraining, and scalable inference in VLA models.

Reproducibility statement

We provide detailed descriptions and diagrams of our architecture and training algorithms in Section 4.1, 4.2, and A.2. We utilize publicly released datasets for simulation setting experiments, and our real-world experiments are easy to reproduce and clearly explained. We also attach pseudocode for our training algorithm and test-time scaling strategy in Section A.6.

References

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

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. In *Robotics: Science and Systems*, 2025.

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

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems*, 2025.

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*, 2023.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

Shivin Dass, Karl Pertsch, Hejia Zhang, Youngwoon Lee, Joseph J. Lim, and Stefanos Nikolaidis. Pato: Policy assisted teleoperation for scalable robot data collection. In *Robotics: Science and Systems*, 2023.

Patrick Esser, Sumith Kulal, A. Blattmann, Rahim Entezari, Jonas Muller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

Yanjiang Guo, Yucheng Hu, Jianke Zhang, Yen-Jen Wang, Xiaoyu Chen, Chaochao Lu, and Jianyu Chen. Prediction with action: Visual policy learning via joint denoising process. In *Advances in Neural Information Processing Systems*, 2024.

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

Siyuan Huang, Liliang Chen, Pengfei Zhou, Shengcong Chen, Zhengkai Jiang, Yue Hu, Yue Liao, Peng Gao, Hongsheng Li, Maoqing Yao, and Guanghui Ren. Enerverse: Envisioning embodied future space for robotics manipulation. *arXiv preprint arXiv:2501.01895*, 2025.

Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. In *IEEE International Conference on Robotics and Automation*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, 2023.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *Robotics: Science and Systems*, 2025a.

Zhiqi Li, Guo Chen, Shilong Liu, Shihao Wang, Vibashan VS, Yishen Ji, Shiyi Lan, Hao Zhang, Yilin Zhao, Subhashree Radhakrishnan, Nadine Chang, Karan Sapra, Amala Sanjay Deshmukh, Tuomas Rintamaki, Matthieu Le, Ilia Karmanov, Lukas Voegtle, Philipp Fischer, De-An Huang, Timo Roman, Tong Lu, Jose M. Alvarez, Bryan Catanzaro, Jan Kautz, Andrew Tao, Guilin Liu, and Zhiding Yu. Eagle 2: Building post-training data strategies from scratch for frontier vision-language models. *arXiv preprint arXiv:2501.14818*, 2025b.

Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning*, 2023.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems*, 2024.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE International Conference on Computer Vision*, 2022.

Kevin Rojas, Yuchen Zhu, Sichen Zhu, Felix X. F. Ye, and Molei Tao. Diffuse everything: Multimodal diffusion models on arbitrary state spaces. In *International Conference on Machine Learning*, 2025.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

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

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, 2023.

Boyang Wang, Nikhil Sridhar, Chao Feng, Mark Van der Merwe, Adam Fishman, Nima Fazeli, and Jeong Joon Park. This&that: Language-gesture controlled video generation for robot planning. *arXiv preprint arXiv:2407.05530*, 2024.

Wenhao Wang, Jianheng Song, Chiming Liu, Jiayao Ma, Siyuan Feng, Jingyuan Wang, Yuxin Jiang, Kylin Chen, Sikang Zhan, Yi Wang, et al. Genie centurion: Accelerating scalable real-world robot training with human rewind-and-refine guidance. *arXiv preprint arXiv:2505.18793*, 2025.

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 *International Conference on Learning Representations*, 2024.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *International Conference on Learning Representations*, 2025.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

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

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.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, Avnish Narayan, You Liang Tan, Guanzhi Wang, Qi Wang, Jiannan Xiang, Yinzhen Xu, Seonghyeon Ye, Jan Kautz, Furong Huang, Yuke Zhu, and Linxi Fan. Flare: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

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

Table 7: **Results of test-time scaling with synchronous joint sampling.** Success rates (%) on RoboCasa and GR-1 with our test-time scaling approach using synchronous joint sampling. For scaling, we increase both N_o, N_A, the number of diffusion steps for vision tokens and action tokens, respectively.

	RoboCasa 100 demos				RoboCasa 1,000 demos				GR-1 1000 demos		
N_o	PnP	OP/CL	Other	Avg.	PnP	OP/CL	Other	Avg.	PnP	Art.	Avg.
4	0.295	0.760	0.510	0.501	0.483	0.863	0.686	0.663	0.422	0.413	0.420
16	0.197	0.685	0.450	0.425	0.472	0.854	0.621	0.630	0.422	0.413	0.420
32	0.210	0.710	0.424	0.424	0.450	0.807	0.630	0.614	0.406	0.438	0.406
64	0.181	0.654	0.416	0.397	0.460	0.817	0.601	0.608	0.399	0.405	0.401

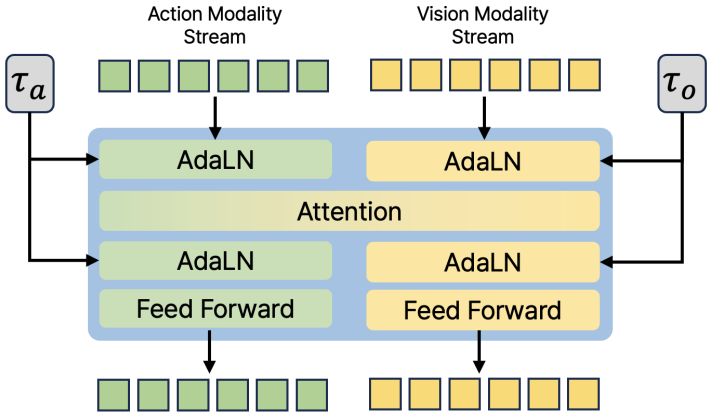


Figure 6: **Modified MMDiT.** MMDiT block is used with separate timestep embeddings being used as conditions for each modality.

A Appendix

A.1 Test-time scaling of naive joint sampling

In Section 5.3, we explored test-time scaling DUST by increasing N_o, the number of vision token diffusion steps, while keeping N_A, the action diffusion step count, fixed at 4. While we have seen great performance gains through the asynchronous joint sampling, it is natural to ask whether simply increasing diffusion steps for both modalities could be enough.

In Table 7, we give results to an ablation study, where both $N_A = N_o$ are increased together, instead of fixing N_A and increasing N_o. We can see that without the decoupling of number of steps between modalities, simply increasing diffusion steps actually leads to deterioration in performance. This lends credibility to our initial hypothesis of only vision tokens needing more diffusion steps, and shows that the asynchronous component of our test-time scaling method is crucial to its success.

A.2 Implementation and Training Details

Additional Implementation Details. We base our architecture on the GR00T-N1 (NVIDIA et al., 2025) codebase, from which we get the pretrained Eagle-2 VLM model. For vision tokens, they pass through an encoder made up of a 3-layer MLP with 2D sinusoidal positional encoding with SiLU activation. The vision decoder is a 2-layer MLP with ReLU activation. Action tokens utilize the linear encoder-decoder pair given in the original codebase, alongside 1D sinusoidal positional encoding.

The MMDiT blocks used in our model are a slight modification of the original in that the AdaLN layers for each modality stream take the conditioning timestep embeddings from independent sources instead of utilizing a global timestep embedding. We show this in more detail in Figure 6.

Baselines. The GR00T-N1 baseline is trained on the original released code, while the FLARE baseline does not release official code or checkpoints. Hence, for FLARE, we do not utilize the Q-Former architecture of the original paper, but reimplement the FLARE loss to utilize the same world modeling target as ours, which is the SIGLIP embeddings from the model VLM. This allows fair comparison between dual-stream diffusion world modeling of DUST and the implicit world modeling of FLARE. For the alignment module of FLARE we use a small MLP, with similar architecture to that of REPA (Yu et al., 2025), which inspired FLARE.

Batch Size and Iteration Count. We vary batch size and training time per dataset.

- For the RoboCasa (Nasiriany et al., 2024) dataset, we train using global batch size 32, with 2 A100 GPUs. For each training dataset scale, the time until convergence varies, with 100 demos requiring 60k steps, 300 demos requiring 420k steps, and 1000 demos requiring 600k steps. The long convergence time is mostly due to the small global batch size.
- For the GR-1 (NVIDIA et al., 2025) dataset, we train using global batch size of 960, with 8 H200 GPUs over 60k steps. We noted training on GR-1 was very sensitive to batch size and required large scale training for meaningful training results.
- For the real-world dataset, we train using global batch size of 32, with 2 A100 GPUs over 60k steps.
- For the transfer learning setup, we first train with BridgeV2 (Walke et al., 2023) video data using global batch size of 32, with 2 A100 GPUs for 120k steps. Then, we finetune using the RoboCasa 100 demo dataset with the same GPU setup for 60k steps.

Common Training Details. Excluding batch size and iteration count, all experiments are done with the same training hyperparameters. We optimize with AdamW (Loshchilov & Hutter, 2019) using a base learning rate of 1e-4, with $\beta_1 = 0.95$, $\beta_2 = 0.999$, and $\epsilon =$ 1e-8. Weight decay of 1e-5 is applied with the exemption of bias and LayerNorm weights. The learning rate follows a cosine decay schedule with a 5% warmup period.

A.3 Simulation Benchmarks

RoboCasa Kitchen. RoboCasa is a single arm manipulation benchmark with a focus on kitchen environment interaction tasks. We utilize a suite of 24 tasks that span a wide range of common household manipulations, including turning sink faucets, closing drawer doors, and moving objects. Tasks are categorized into 8 pick-and-place tasks, 6 contraption open/close tasks, and 10 other miscellaneous tasks. Training data is drawn from the publicly available dataset from RoboCasa which was generated with MimicGen (Mandlekar et al., 2023) within the MuJoCo simulation environment (Todorov et al., 2012), with a Franka Emika Panda robot arm serving as the manipulator. Image observations include 3 viewpoints from the left, right, and wrist. The robot state/action space is parameterized with 7 Degrees of Freedom (DoF), consisting of end-effector position and rotation together with a binary gripper pose. We experiment over 100, 300, and 1000 training episodes per task, testing data efficiency and scaling properties.

GR-1 Tabletop Tasks. GR-1 is a humanoid robot benchmark with a focus on dexterous tabletop manipulation of everyday objects. We utilize a total of 24 tasks consisting of 16 pick-and-place tasks, and 8 articulated tasks, the latter adding the requirement of closing containers such as microwaves and cabinets after pick-and-place. Training data utilizes data from GR00T-N1 (NVIDIA et al., 2025), where the dataset was generated with DexMimicGen (Jiang et al., 2025) in the MuJoCo simulation environment (Todorov et al., 2012). The simulated robot is a GR-1 humanoid robot with Fourier dexterous hands, enabling fine-grained grasping and manipulation. Image observations are taken from a single egocentric view from the robot's head. The state/action space consists of 29 DoF in total, 17 DoF corresponding to the GR-1 robot's arms and waist, and 6 DoF for each of the Fourier hands. We experiment over 300 and 1000 training episodes per task.

A.4 Real-World Experiment Details

Our tasks consist of 4 tasks, which have the following task instruction templates:

- Pick up the {*Object*} on the brown box and place it in the golden bowl.

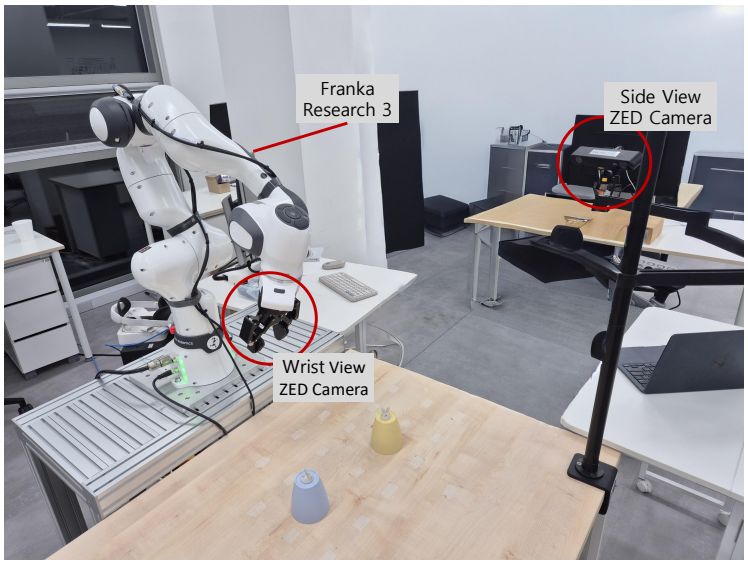


Figure 7: **Real-world experimental setting.** We utilize the Franka Research 3 robot with two ZED cameras, one on the wrist and one to the side.

- Pick up the {*Object*} on the brown box and place it on the white plate.
- Pick up the {*Object*} in the white basket and place it in the black bowl.
- Pick up the {*Object*} on the white plate and place it in the white basket.

Each task contains the four object categories - Teddy Bear, Blue Cube, Blue Cup, and Sponge. During evaluation each object-task configuration gets 6 evaluations, meaning 24 trials per task. We predetermine a set of varied configuration of where to place the source-target locations, on where the source location the object is placed, and the direction it is facing. This allows for more fair comparison in real-world experiments that typically have high stochasticity. When an object has been partially placed in the target destination but the center of gravity is outside of said target, we denote that as a half success and count is as 0.5 successes. We note there were very few cases of this happening.

A.5 LLM Usage Disclosure

We acknowledge that large language models (LLMs) were used in the preparation of this manuscript to assist with writing quality. LLMs were used to find grammatical errors, suggest alternative vocabulary, and detect potential typographical issues. All substantive ideas, analyses, and conclusions presented in this paper are the work of the authors.

A.6 DUST PSEUDOCODE

Algorithm 1: DUST Training

Input: Dataset D, weight λ_{WM}, steps, batch size, optimizer hyperparams.
Models and encoders/decoders as described in text below.
Output: Trained parameters θ.

Initialize θ, optimizer (AdamW);
for *step* $\leftarrow 1$ **to** *steps* **do**
  `// 1) Minibatch`
  Sample a minibatch $B \subset D$ of size bs;
  `// 2) Conditioning`
  $\Phi \leftarrow \text{VLM}_\phi(o_t^v, \ell)$; `// VLM semantic representations`
  `// 3) Modality-decoupled noising`
  Sample $\tau_A, \tau_o \sim \mathcal{U}(0,1)$ and $\epsilon_A, \epsilon_o \sim \mathcal{N}(0, I)$;
  $A_t^{\tau_A} \leftarrow \tau_A A_t + (1-\tau_A)\epsilon_A$;
  $\tilde{o}_{t+k} \leftarrow \text{VLM}_{\text{img}}(o_{t+k}^v)$; `// future obs embedding`
  $\tilde{o}_{t+k}^{\tau_o} \leftarrow \tau_o \tilde{o}_{t+k} + (1-\tau_o)\epsilon_o$;
  `// 4) Per-modality encoders`
  $X_A \leftarrow \text{Enc}_A([o_t^s, A_t^{\tau_A}])$; $X_o \leftarrow \text{Enc}_o([\tilde{o}_{t+k}^{\tau_o}])$;
  `// 5) Dual-stream MMDiT stack (AdaLN per modality)`
  for $i \leftarrow 1$ **to** N_{MMDiT} **do**
    $(X_A, X_o) \leftarrow \text{MMDiT}_i(X_A, X_o, \Phi, \tau_A, \tau_o)$;
  `// 6) Modality-specific DiT stack`
  for $i \leftarrow 1$ **to** N_{DiT} **do**
    $X_A \leftarrow \text{DiT}_i^A(X_A, \Phi, \tau_A)$;
    $X_o \leftarrow \text{DiT}_i^o(X_o, \Phi, \tau_o)$;
  `// 7) Per-modality Decoders`
  $V_\theta^A \leftarrow \text{Dec}_A(X_A)$; $V_\theta^o \leftarrow \text{Dec}_o(X_o)$;
  `// 8) Flow-matching losses (linear path)`
  `// For the linear path,` $u_A = \frac{d}{d\tau_A}(\tau_A A_t + (1-\tau_A)\epsilon_A) = A_t - \epsilon_A$
  $u_A \leftarrow A_t - \epsilon_A$; $u_o \leftarrow \tilde{o}_{t+k} - \epsilon_o$;
  $L_A \leftarrow \text{MSE}(V_\theta^A, u_A)$; $L_{\text{WM}} \leftarrow \text{MSE}(V_\theta^o, u_o)$;
  $L_{\text{joint}} \leftarrow L_A + \lambda_{\text{WM}} L_{\text{WM}}$;
  `// 9) Update`
  zero_grad(); **backward**(L_{joint}); **clip_grad_norm**(θ); **step**();
return θ;

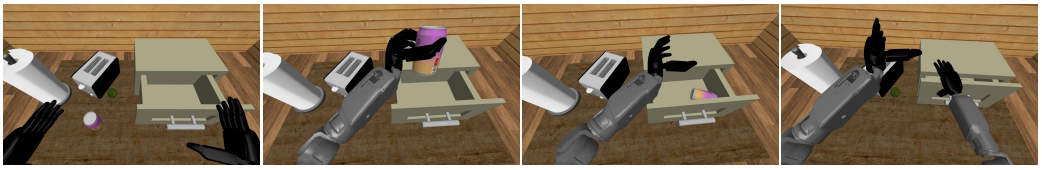
Algorithm 2: DUST Test-Time Scaling - Asynchronous Joint Sampling

Input: Trained model π_θ; horizon T; diffusion step counts N_A, N_o with $N_o > N_A$; ratio $q = N_o/N_A$;
Output: Predicted action sequence A_t and future observation embedding $\tilde{o}_{t+k}$;

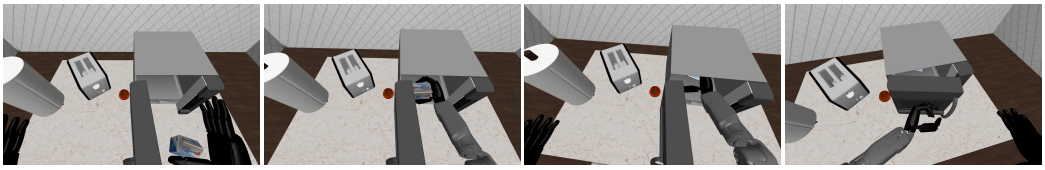
Initialize $\tau_A, \tau_o = 0$;
Initialize noisy tokens $A_t^{\tau_A} \sim \mathcal{N}(0, I)$, $\tilde{o}_{t+k}^{\tau_o} \sim \mathcal{N}(0, I)$;
Set $\Delta\tau_o = 1/N_o$, $\Delta\tau_A = 1/N_A = q\Delta\tau_o$
for $n_A \leftarrow 1$ **to** N_A **do**
  `// outer loop: action updates`
  for $j \leftarrow 1$ **to** q **do**
    `// inner loop:` q `vision updates`
    $\tau_o \leftarrow \tau_o + \Delta\tau_o$;
    $\tau_A \leftarrow \tau_A + \Delta\tau_o$;
    $\tilde{o}_{t+k}^{\tau_o} \leftarrow \tilde{o}_{t+k}^{\tau_o} + V_\theta^o \Delta\tau_o$;
  $A_t^{\tau_A} \leftarrow A_t^{\tau_A} + V_\theta^A \Delta\tau_A$;
return *Final denoised* A_t^1, $\tilde{o}_{t+k}^1$;

A.7 Example GR-1 Rollouts

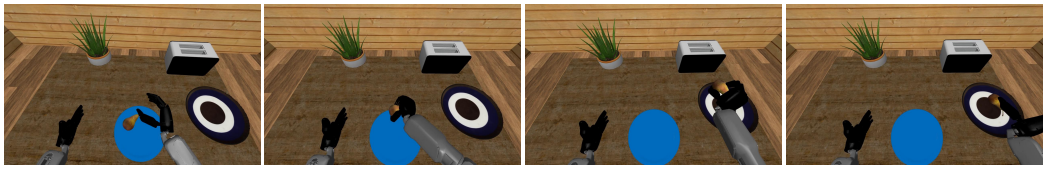
We showcase example rollouts of DUST trained on GR-1 with 1000 demos per task.



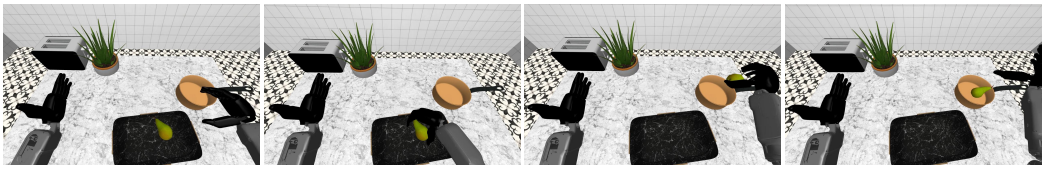
(a) **(GR-1)** *Pick up the can, place it into the drawer and close the drawer.*



(b) **(GR-1)** *Pick up the milk, place it into the microwave and close the microwave*



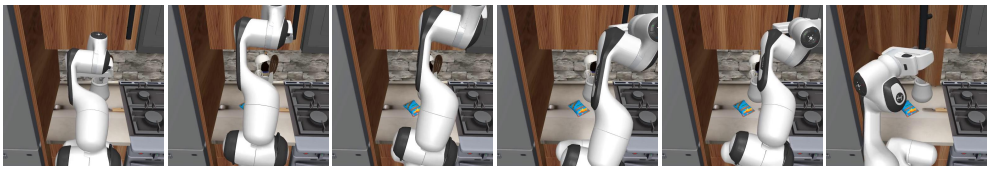
(c) **(GR-1)** *Pick the pear from the plate and place it in the plate*



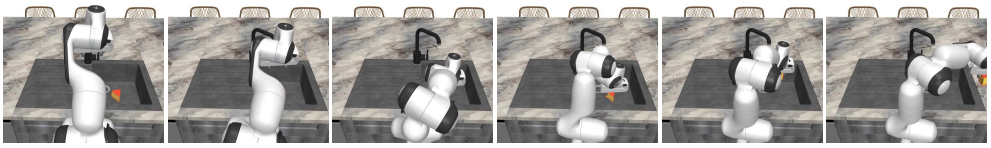
(d) **(GR-1)** ***Pick the pear from the tray and place it in the pot***

A.8 Example RoboCasa Rollouts

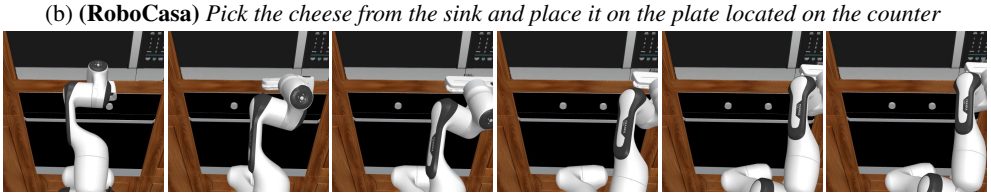
We showcase example rollouts of DUST trained on RoboCasa with 1000 demos per task.



(a) **(RoboCasa)** *Open the cabinet door*



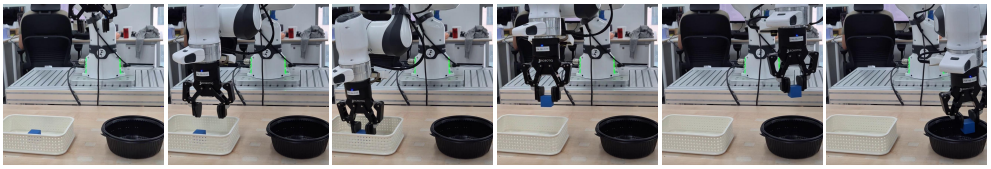
(b) **(RoboCasa)** *Pick the cheese from the sink and place it on the plate located on the counter*



(c) **(RoboCasa)** *Turn on the microwave*

A.9 Example Real-world Rollouts

We showcase example rollouts of DUST trained on our real-world Franka Research 3 dataset with 60 demos per task.



(a) **(Franka)** *Pick up the blue cube in the white basket and place it in the black bowl*

(b) **(Franka)** *Pick up the sponge on the brown box and place it on the white plate*