VIDAR: EMBODIED VIDEO DIFFUSION MODEL FOR GENERALIST MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Scaling general-purpose manipulation to new robot embodiments remains challenging: each platform typically needs large, homogeneous demonstrations, and end-to-end pixel-to-action pipelines may degenerate under background and viewpoint shifts. Based on previous advances in video-based robot control, we present Vidar, consisting of an embodied video diffusion model as the generalizable prior and a masked inverse dynamics model (MIDM) as the adapter. We leverage a video diffusion model pre-trained at Internet scale, and further continuously pre-train it for the embodied domain using 750K multi-view trajectories collected from three real-world robot platforms. For this embodied pre-training, we introduce a unified observation space that jointly encodes robot, camera, task, and scene contexts. The MIDM module learns action-relevant pixel masks without dense labels, grounding the prior into the target embodiment's action space while suppressing distractors. With only ∼20 minutes of human demonstrations on an unseen robot (∼1% of typical data), Vidar outperforms state-of-the-art baselines and generalizes to unseen tasks, backgrounds, and camera layouts. Our results suggest a scalable recipe for "one prior, many embodiments": strong, inexpensive video priors together with minimal on-robot alignment.

1 INTRODUCTION

Robotic manipulation spans skills such as stable object handling, in-hand reorientation, and multi-point contact control—capabilities underlying cloth folding, liquid pouring, and tool-assisted assembly. Vision-language-action (VLA) models (O'Neill et al., 2024; Kim et al., 2024; Liu et al., 2024a; Zhang & Yan, 2023) have made encouraging progress via large-scale multimodal pre-training, yet extending them to general-purpose or bimanual settings remains difficult. The core obstacle is control complexity: the action space grows combinatorially with added joints, while success hinges on tight temporal coordination, accurate contact dynamics, and long-horizon reasoning. These factors amplify data requirements and heighten sensitivity to platform-specific details. In practice, progress is constrained by **data scarcity**: human demonstration corpora typically contain only tens to hundreds of hours (Intelligence et al., 2025; Liu et al., 2024a), orders of magnitude smaller than Internet-scale video collections with hundreds of thousands of hours (Wang et al., 2024). Collecting demonstrations is labor-intensive, expensive, and coupled to hardware, leaving a key question: *how can a new robot embodiment achieve precise, generalizable control with limited domain-specific data?*

A natural answer lies in *video*. Unlike text or static images, video is both abundant and intrinsically suited to capture the temporal dynamics and interaction cues—affordances, contacts, motion continuity—that demonstrations aim to convey. Leveraging such signals enables robots to acquire embodiment-agnostic interaction knowledge at scale, and later specialize to new morphologies with far fewer platform-specific samples. To turn this rich modality into a transferable prior, we adopt video generative models, which learn distributions of *plausible, temporally coherent rollouts* rather than task-specific labels. This generative formulation enforces physical consistency, supports counterfactual reasoning about "what could happen next", and shifts dependence from costly demonstrations to abundant raw video. Recent advances in video diffusion models trained on web-scale corpora already show strong semantic grounding and temporal fidelity (Liu et al., 2024b; Wang et al., 2025; Kong et al., 2024; Bao et al., 2024), making them well-suited to serve as general interaction priors for low-shot embodiment alignment. Meanwhile, for robot control, our objective departs from vanilla

video diffusion. Rather than producing photorealistic clips, we require *actionable* rollouts that are consistent with robot actuation, accurate in contact dynamics, and robust across embodiments.

Based on previous advances in using video generative models for robot control (Du et al., 2023), we propose **Video Diffusion for Action Reasoning (Vidar)**, to better explore the transferable prior from videos for efficient bimanual manipulation with decoupled video generation and action prediction. To build a good video prior for embodied control, we propose a three-stage training pipeline: Internet-scale videos are used for general pre-training (where off-the-shelf checkpoints can be adopted), large cross-embodiment robotic datasets are used for embodied domain pre-training, and a small number of robot-specific demonstrations are used for target domain fine-tuning. Specifically for embodied domain pre-training, we align 750K multi-view bimanual clips spanning three robot platforms into a unified observation space. Such pre-training on a unified space ensures control feasibility, promotes physically credible contacts, and mitigates viewpoint and morphology gaps. During inference, we further enhance rollout quality (e.g., physical plausibility, instruction relevance) by applying test-time scaling (Jaech et al., 2024).

On the action side, the main challenge is to decode videos into reliable controls despite background clutter, distractors, and partial observability of hands and tools. We address this with a **Masked Inverse Dynamics Model (MIDM)**, which learns to attend selectively to action-relevant regions without pixel-level supervision. By filtering out irrelevant content, MIDM provides robust action decoding and facilitates the transfer of video priors across domains. Moreover, such a lightweight model can be trained using only a small number of demonstrations. Together, these components transform raw Internet video into a transferable and controllable prior, enabling precise manipulation with only a small number of demonstrations.

Empirically, Vidar achieves state-of-the-art performance on the RoboTwin (Chen et al., 2025) benchmark under the challenging multi-task setting. In the real world, it attains strong performance with only **20 minutes of human demonstrations** (roughly 3 per task) on a previously unseen robotic platform. Despite this minimal supervision, it outperforms leading baselines by large margins—**58%** over VPP (Hu et al., 2024) and **40%** over UniPi (Du et al., 2023). Moreover, it generalizes robustly to novel scenarios, such as environments with reflective surfaces, indicating that video-pretrained priors can support both data-efficient adaptation and semantically grounded control.

2 METHOD

In this section, we formally define our problem and present Vidar in detail. The overview of our method is shown in Figure 1.

2.1 PROBLEM FORMULATION

This work investigates the challenges of bimanual manipulation for everyday activities, tasks that are inherently resistant to standardization. Our experiments are conducted using the common Aloha robot platform (Liu et al., 2024a; Fu et al., 2024), with detailed hardware specifications given in Appendix F. The problem is formulated as follows.

Let $\mathcal{L}$ denote the language instruction space, $\mathcal{O}$ the observation space, and $\mathcal{A}$ the action space. Our goal is to learn a conditional manipulation policy

$$\pi : \mathcal{L} \times \mathcal{O} \to \mathbb{P}(\mathcal{A}),$$

where $\mathbb{P}(\cdot)$ denotes probability measures over the corresponding space. Learning this policy directly is highly challenging. It requires large-scale demonstrations that jointly cover language, observation, and action, which are expensive and hardware-specific. Moreover, robots differ in sensing, morphology, and viewpoints, making policies learned on one platform difficult to transfer. Finally, successful manipulation depends on fine-grained contact events and long-horizon temporal coherence; photorealistic video generation alone does not guarantee *actionability*.

We address these challenges by elevating the action space to the video domain, where richer semantic information is preserved and an abundance of large-scale data is available for learning a strong, transferable prior. We factorize the policy through the video space $\mathcal{V}$:

$$\pi = I \circ G, \qquad G : \mathcal{L} \times \mathcal{O} \to \mathbb{P}(\mathcal{V}), \quad I : \mathcal{V} \to \mathcal{A}.$$

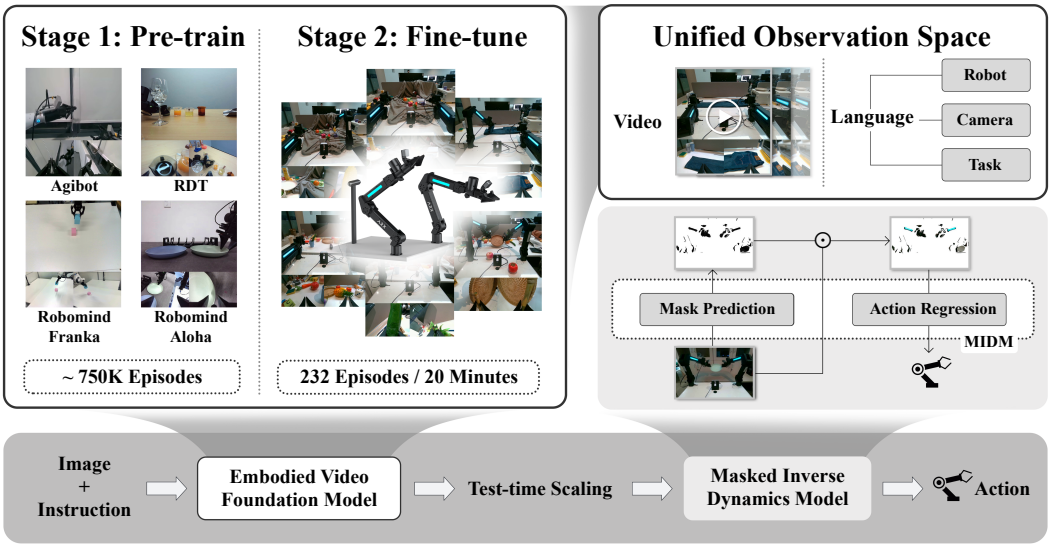


Figure 1: The overall pipeline of Vidar, where various video sources are leveraged for transferring to a new platform with limited demonstrations. A unified observation space handles heterogeneous, multi-view robotic videos and language instructions, enabling the pre-training of an embodied foundation video model on about 750,000 multi-view bimanual robotic episodes. After fine-tuning it with only 20 minutes of human demonstrations on an unseen robot platform, we adopt test-time scaling to select the best video during inference. Meanwhile, the masked inverse dynamics model (MIDM) converts videos to actions, where masks are learned to attend to action-relevant regions for background-robust action regression.

Here, G is a video generation model that produces temporally coherent, physically plausible rollouts conditioned on task and observations, while I is an inverse dynamics model that maps short video windows into robot-specific controls. This two-stage design shifts most of the representation burden to G, which can be pretrained on abundant Internet and robotic video, and leaves only a lightweight I to be trained with limited demonstrations on the target platform.

Concretely, G is conditioned on proprioceptive traces and embodiment tokens, and trained on 750K multi-view bimanual clips from three robot platforms, aligned into a unified observation space. This ensures that generated rollouts remain feasible under different morphologies and camera setups, and that contact and motion continuity are preserved. At inference, test-time scaling (Jaech et al., 2024) with physics-aware reranking further improves temporal coherence and physical plausibility. The inverse dynamics component I is instantiated as a **Masked Inverse Dynamics Model (MIDM)**, which learns to attend to action-relevant regions such as hands, tools, and contact patches without pixel-level labels. By filtering out background clutter and distractors, MIDM provides robust action decoding and enables effective transfer of video priors across domains.

This formulation directly addresses the identified challenges: large-scale video pretraining reduces the need for triply-labeled demonstrations; conditioning and unified observation mitigate embodiment shifts; contact- and flow-consistent objectives make rollouts actionable; and MIDM grounds them to the target robot with few demonstrations. Together, these design choices turn raw video into transferable interaction priors that enable efficient and precise adaptation to new robotic embodiments.

2.2 Video Generation Model

We adopt rectified flow (Lipman et al., 2023; Liu et al., 2023) models, which generate high-quality videos by modeling pixel flow over time. Specifically, the model parameterizes a flow function $v : \mathcal{V} \times \mathbb{R} \times \mathcal{L} \to \mathcal{V}$, which models the velocity of pixels as they transition from noisy video frames to target frames under certain conditions and time, leading to the following ODE over a video x_t

(with x_0 sampled from a Gaussian and x_1 as the output video):

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = v(x_t, t, c), \quad t \in [0, 1]. \tag{1}$$

To learn the non-trivial mapping between the Gaussian distribution and the video distribution, we train the "velocity" v to approximate the constant flow from x_0 to x_1 during training. The flow matching loss is:

$$L_G = \mathbb{E}_{c,t,x_0,x_1}\left[\|(x_1 - x_0) - v(tx_1 + (1-t)x_0, t, c)\|^2\right]. \tag{2}$$

Unified Observation Space. To mitigate viewpoint and morphology gaps between heterogeneous embodiments, we design a unified observation space for multi-view embodied data. The space does not include actions: the video diffusion model only learns world evolution, allowing it to generalize efficiently across robots with different morphologies. Denoting the maximum number of cameras as V, we can define a unified observation space $\mathcal{U} \subseteq \mathcal{L} \times \mathcal{O}$ (also see Figure 1):

$$\mathcal{U} = \{\langle \mathbf{o}, \mathbf{l} \rangle | \mathbf{o} = \mathrm{aggregate}(\mathbf{I}^{(1)}, \mathbf{I}^{(2)}, \cdots, \mathbf{I}^{(V)}), \mathbf{l} = \mathrm{concatenate}(l_r, l_c, l_t)\}, \tag{3}$$

where $\mathrm{aggregate}(\mathbf{I}^{(1)}, \mathbf{I}^{(2)}, \cdots, \mathbf{I}^{(V)})$ is the aggregation of image views (some views may be missing), and l_r, l_c, and l_t are instructions related to the robotic platform, camera, and task, respectively. Specifically, each observation $\mathbf{o}$ is constructed as $\mathbf{o} = \bigoplus_{k=1}^{V} \phi_{r_k}(\mathbf{I}^{(k)})$, where $\mathbf{I}^{(k)}$ is the RGB image from camera k, ϕ_{r_k} is a spatial resizing function. This operation produces a consistent tensor shape across platforms and preserves both semantic and kinematic context. Instead of relying solely on task information and a single view, we condition the video generation model on the robot, camera, task information, and multiple views, thereby providing rich context for video prediction with unified representations across heterogeneous embodiments.

Embodied Pre-training and Fine-tuning on Unified Observation Space. We pre-train our video generation model on a large-scale corpus of approximately 750K open-sourced episodes, all projected into the unified observation space (Equation (3)). This dataset includes three camera views, a configuration commonly adopted in bimanual setups (Fu et al., 2024; AgiBot-World-Contributors et al., 2025) to provide abundant context information for precise control. At fine-tuning time, we apply supervised fine-tuning (SFT) on all model parameters using a small number of human demonstration episodes collected from the target platform. To ensure precise adaptation without overfitting, we augment the dataset by clipping variable-length videos from random starting points. In this way, the limited domain-specific data is fully used, and the model learns to predict videos at various states.

Test-time Scaling. Test-Time Scaling (TTS) refers to using additional test-time compute to improve performance without retraining the model, which is widely explored in large language models (Muennighoff et al., 2025). Diffusion-based video generation is inherently stochastic, resulting in significant variance in the quality, physical plausibility, and task relevance of sampled rollouts. Naïvely sampling a single trajectory may result in incoherent or suboptimal generations, especially in cluttered or ambiguous scenes. To address this, we propose a test-time scaling strategy: given an observation prefix $\mathbf{o}_1$, we generate K candidate video trajectories $\{\tilde{\mathbf{v}}_{1:T}^{(i)}\}_{i=1}^{K}$ using different random seeds. We then rank these trajectories using a pretrained evaluator (e.g., CLIP or a vision-language model) q_η and select the highest-scoring one, i.e., $\arg\max_i \; q_\eta(\tilde{\mathbf{v}}_{1:T}^{(i)})$. This approach aligns the predicted video distribution with task-relevant, actionable videos through "rejection sampling", reducing sampling variance and consistently improving the quality of generated demonstrations.

2.3 Masked Inverse Dynamics Model

Inverse dynamics models often suffer from poor generalization due to the presence of background noise, texture biases, and visual distractions in high-dimensional observations (Tan et al., 2025). Explicitly localizing action-relevant regions is challenging without dense annotations, and existing segmentation methods (Yuan et al., 2025) often fail to capture both arms in the bimanual settings, let alone produce temporally consistent segmentations (see Figure 6 in Appendix C). While prior work has explored information-theoretic formulations of state-space models (Nguyen et al., 2021), scaling to high dimensions remains challenging.

To solve the problem, we introduce a masked inverse dynamics model (MIDM) that learns to focus on task-relevant regions via implicit mask prediction. The model consists of two components: (1) a mask prediction network U that outputs a spatial mask $m \in [0,1]^{H \times W}$ from an input frame x, and (2) an action regression network R that predicts the action from the masked frame. Formally:

$$m = U(x), \quad \hat{a} = R(\text{Round}(m) \odot x),$$

where $\odot$ denotes element-wise multiplication, and Round(.) means rounding to the nearest integer. The model is trained by minimizing the following loss:

$$L_I = \mathbb{E}_{x,a}\left[l(\hat{a} - a) + \lambda \|m\|_1\right],$$

where $l(\cdot)$ is the Huber loss. The second ℓ_1-norm regularization term promotes spatial sparsity, encouraging the model to focus on minimal, task-critical regions without any segmentation supervision. We train it using straight-through estimators. Notably, this framework is not restricted to predicting embodiment-specific actions; embodiment-agnostic actions can also be predicted.

By utilizing reliable action signals as supervision, rather than noisy annotations, this lightweight module demonstrates robust generalization to unseen environments and backgrounds with limited demonstrations, as evidenced by our experimental results (Section 3).

3 Experiments

We now present experimental studies, with the goal to verify the following hypotheses:

H1: Vidar achieves superior success rates with only 20 minutes of target domain demonstrations;

H2: Vidar generalizes effectively to unseen tasks and backgrounds;

H3: Pre-training with a unified observation space benefits embodied video generation;

H4: Masked inverse dynamics models exhibit greater generalization ability than the baseline.

3.1 Experimental Setup

3.1.1 Datasets

Our pre-training data integrates episodes sourced from Agibot-World (AgiBot-World-Contributors et al., 2025), RoboMind (Wu et al., 2024b), and RDT (Liu et al., 2024a). The resulting dataset for real-world experiments comprises 746,533 episodes. For the simulation experiments, we additionally add episodes from Egodex (Hoque et al., 2025).

For adaptation to *unseen* platforms, we collect target robot data in both the simulation and real-world domains. For the simulation domain, we employ two configurations: a low data setting and a standard one. In the low data setting, we collect 20 episodes per task using the Aloha (agilex) robot with an adjusted camera to fully capture the robot arms, under clean scenarios of the RoboTwin platform (Chen et al., 2025). In the standard data setting, we follow the leaderboard of RoboTwin by collecting 50 episodes per task with original camera viewpoints, where partial arm occlusion occurs more frequently. For the real-world domain, we collect 20 minutes of human demonstration videos, covering 81 tasks across 232 episodes.

All datasets collected consist of descriptions of the robot type and camera placements, as well as task instructions, as is required by the unified observation space. The collected embodiment-specific dataset is used both for fine-tuning the video diffusion model and training the masked inverse dynamics model. Notably, all these target domains are unseen during pre-training. Further details of our dataset can be found in Appendix A.

3.1.2 Training and Inference

We evaluate our method using two representative video generation models (with off-the-shelf checkpoints pre-trained on Internet-scale videos): the open-source Wan2.2 (Wang et al., 2025) for simulation experiments, and Vidu 2.0 (Bao et al., 2024) for the more diverse and challenging real-world tests. We use their pre-trained checkpoints and continue pre-training the models with a batch size of 128. The first 10,000 steps involve pre-training on the diverse robotics dataset, followed by 12,000

fine-tuning steps for Wan2.2 and 13,000 fine-tuning steps for Vidu 2.0. To reduce inference costs, we uniformly downsample the training videos to 8 frames per second (fps). For the masked inverse dynamics model, we adopt the U-Net (Ronneberger et al., 2015) structure as the mask prediction network and the ResNet (Xu et al., 2024) structure as the action regression network, with $\lambda = 3 \times 10^{-3}$ (effects of λ are shown in Appendix C). We train it exclusively on the fine-tuning dataset. We use AdamW (Loshchilov & Hutter, 2019) for all our training. Here are some detailed settings of real-world experiments.

We use open-loop control for Vidar; the videos are generated in a single batch, without subsequent generation after the initial run. Using 8 NVIDIA Ampere-series 80GB GPUs, generating one video with 60 frames (7.5 seconds duration at 8fps) costs about 25 seconds. The time cost can be reduced using distillation or quantization, which are beyond the scope of this paper. For test-time scaling, we choose $K = 3$, and three videos with different random seeds are generated in parallel, evaluated by GPT-4o (Hurst et al., 2024). Additionally, we disable test-time scaling for simulation experiments for better reproducibility. More details of training and inference can be found in Appendix B.

3.1.3 BASELINES

For simulation experiments, we choose Pi0.5 (Intelligence et al., 2025) as our baseline. For real-world experiments, we perform preliminary experiments over multiple baselines and find that adaptation with only 20 minutes of videos and about 3 demonstrations per task is too challenging for vision-language-action models. Thus, we choose two baselines that also incorporate video-level prior knowledge: UniPi and VPP. To ensure fair comparison, we reproduce these methods over the advanced Vidu 2.0 model. We also compare Vidar (built on Wan 2.2) with Pi0.5 using a larger real-world dataset for completeness; the results are provided in Appendix D.

Pi0.5. We use the official checkpoint of Pi0.5, which has already been pre-trained on more than 10k hours of robot data, and follow the official framework to finetune the base model on multi-task demonstration data.

UniPi. We follow the official framework by fine-tuning the Vidu 2.0 model directly on our demonstration data and training an inverse dynamics model using a ResNet-based architecture.

VPP. We use the same checkpoint for the video generation model as Vidar, because VPP mainly focuses on how to use a given video generation model. Additionally, we train a diffusion model to generate short action sequences based on the latent features during one-step video generation and CLIP (Radford et al., 2021) embeddings of the task instructions. Notably, they use closed-loop control, which means new action sequences are generated and executed after previous executions.

3.2 EXPERIMENTAL RESULTS

H1: Success Rates. For the simulation experiments, we evaluate our method on the RoboTwin 2.0 (Chen et al., 2025) benchmark using the more challenging multi-task setting (in contrast, the official leaderboard trains a separate policy for each task), and the results for two data settings are summarized in Table 1. Compared to the strong baseline Pi0.5 (Intelligence et al., 2025), our method achieves a state-of-the-art average success rate across all scenarios. More detailed results can be found in Appendix C.

For real-world experiments, we test two baseline methods and our method under three scenarios: seen task and background (six tasks), unseen task (five tasks), and unseen background (six tasks). Detailed explanations are as follows:

- **Seen Tasks & Backgrounds**: three pick-and-place tasks (e.g., grasp the tomato using the left arm), two daily-life tasks (e.g., flip a dice using the right arm), and one bimanual task (lift the basket). The background we use is a cluttered office desk setup, featuring several computers situated behind the workspace.
- **Unseen Tasks**: three daily-life tasks (e.g., stack the bowl on the steamer using the left arm) and two semantic tasks (e.g., grasp the shortest bread using the left arm).

Table 1: Average success rates across 50 tasks, 100 episodes for the RoboTwin benchmark. Vidar consistently surpasses the strong Pi0.5 baseline across all settings and scenarios. The Pi0 results are taken directly from the official leaderboard, where each task is trained and evaluated independently under standard data settings, making them easier and not directly comparable.

Data Regime	Low		Standard	
Scenario	Clean	Randomized	Clean	Randomized
Pi0*	-	-	46.42%	16.34%
Pi0.5	25.0%	9.2%	44.8%	14.2%
Vidar (Ours)	**60.0%**	**15.7%**	**65.8%**	**17.5%**

- **Unseen Backgrounds**: two pick-and-place tasks (e.g., grasp a tomato using the left arm) and four daily-life tasks (e.g., flip a dice using the left arm). For unseen backgrounds, we include a studio setup with a typical green screen and a daily workspace setting with two cupboards containing robot supplies, which exhibit reflective surfaces.

The success rates are shown in Table 2. We find that our method achieves superior success rates over all three scenarios, demonstrating the effectiveness of our method. UniPi does not utilize heterogeneous robotic data, which restricts its generalization under limited demonstrations; VPP uses predicted features from a single denoising forward pass for action prediction, which leads to noise and instability—particularly in unseen environments. More details can be found in Appendix C.

Table 2: Success rates of different methods and configurations over robot manipulation tasks. Vidar achieves high success rates across all three scenarios, with great generalization ability to unseen tasks and backgrounds.

Method	Seen Tasks & Backgrounds	Unseen Tasks	Unseen Backgrounds
VPP	4.5%	13.3%	0.0%
UniPi	36.4%	6.7%	22.2%
Vidar (Ours)	**68.2%**	**66.7%**	**55.6%**

To further demonstrate the generality of our approach, we also perform additional real-world experiments using the open-sourced Wan2.2 and HunyuanVideo models (Kong et al., 2024). Notably, Vidar surpasses Pi0.5, achieving a 35% higher average success rate on the 7 seen tasks and a 54% higher success rate on the 7 unseen tasks. Related results are provided in Appendix D.

H2: Generalization Ability. For the simulation benchmark, we show that our model generalizes effectively to randomized scenarios, despite being trained only on clean scenarios (see Appendix C). For real-world experiments, quantitative results of our superior generation ability are shown in unseen scenarios of Table 2. We also provide some demonstrations in Figure 2, where Vidar demonstrates robust generalization to unseen tasks and backgrounds with strong semantic understanding. In Appendix E, we present more visualizations, including failure cases.

H3: Effectiveness of Pre-training. We evaluate the video generation quality of the original Vidu 2.0 model and our pre-trained version over the unseen target domain using VBench (Huang et al., 2024). As is shown in Table 3, we find that pre-training using large-scale robotic videos under our unified observation space enhances both the consistency and quality of generated frames, which are important for robot control tasks.

H4: Effectiveness of MIDM. Based on empirical observations of the Aloha robot's error tolerances, we define a successful prediction as having a maximum infinity norm error of less than 0.06 for joint positions and less than 0.6 for gripper positions, for both arms. Using this criterion, we evaluate the success rates of our masked inverse dynamics model (MIDM) and a ResNet baseline on both training and testing sets. The results, presented in Table 4, show that our MIDM demonstrates superior generalization compared to the baseline. Additionally, examples of the learned masks are provided in

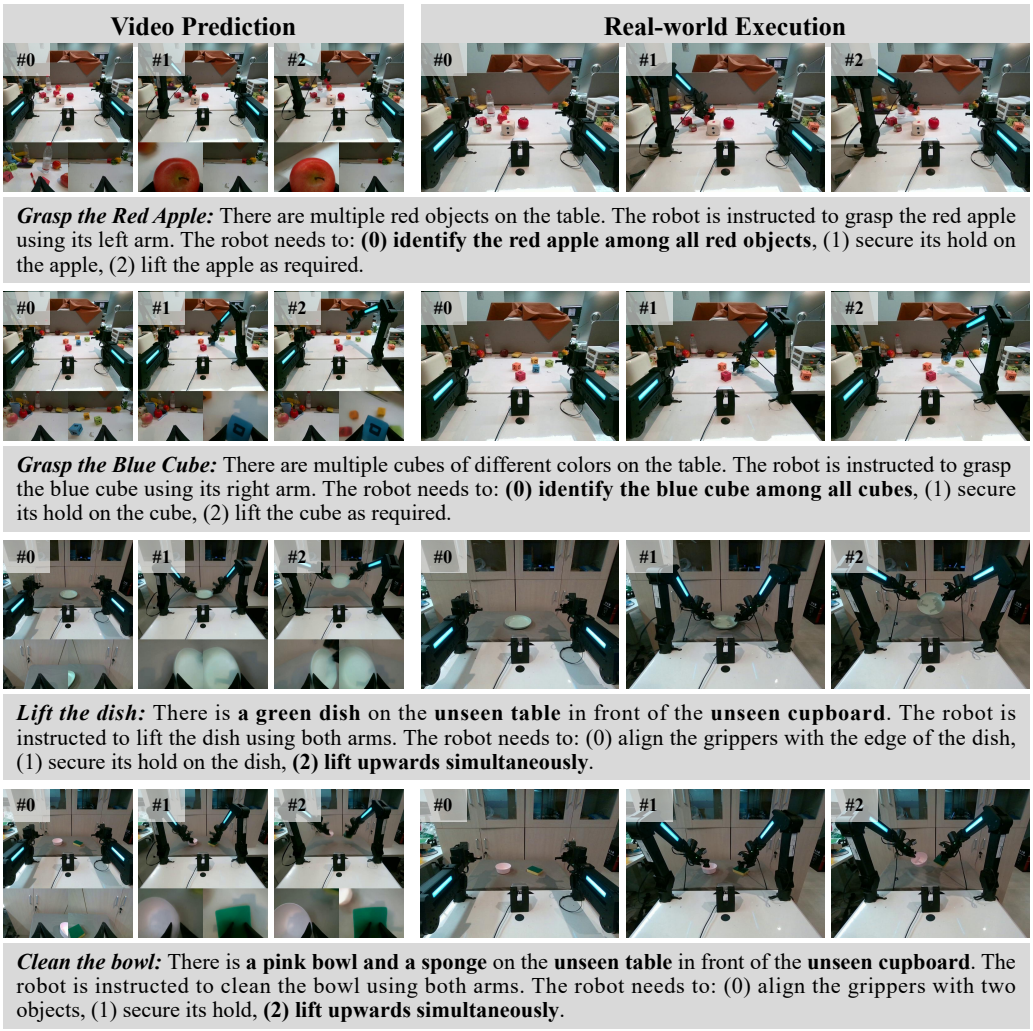


Figure 2: Videos of predictions (left) and corresponding executions (right) of Vidar for challenging tasks. It can handle unseen tasks and unseen backgrounds with strong semantic understanding.

Table 3: VBench video quality measurements for different video model configurations in the unseen target domain. Embodied pre-training over the unified observation space benefits video generation.

Configuration	Subject Consistency	Background Consistency	Imaging Quality
Vidu 2.0	0.565	0.800	0.345
+ Embodied Pre-training	**0.855**	**0.909**	**0.667**

Figure 3. Without any additional supervision, MIDM effectively captures action-relevant features and generalizes well to unseen backgrounds.

3.3 Ablation Study

We conduct an ablation study of our method by evaluating success rates on the same tasks presented in Table 2. The results are summarized in Table 5, where "w/o MIDM" means using the ResNet baseline. We find that both masked inverse dynamics models and test-time scaling are beneficial to the success rates.

Table 4: Training and testing success rates and testing l_1 errors of different inverse dynamics models. Both the baseline and MIDM achieve high performances during training, but MIDM generalizes better during test time.

Inverse Dynamics Model	Training Accuracy	Testing Accuracy	Testing l_1 Error
ResNet	**99.9%**	24.3%	0.0430
MIDM (Ours)	**99.9%**	**49.0%**	**0.0308**

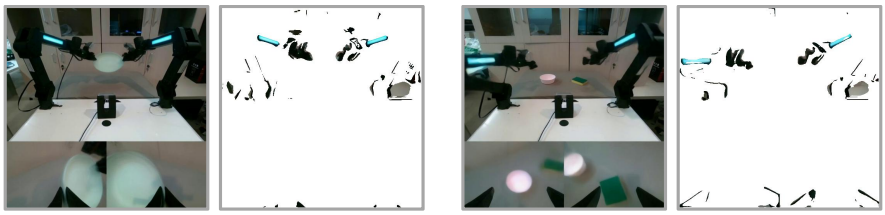


Figure 3: Input images and corresponding masked images learned by the masked inverse dynamic model (MIDM). The two cases are from an unseen background with complex reflective surfaces, while the predicted mask images still focus on the essential parts of robotic arms.

4 Related Work

Vision-Language-Action Models (VLAs). Vision-Language-Action (VLA) models integrate perception, language understanding, and action generation to enable general-purpose embodied intelligence. However, existing VLA systems rely heavily on large-scale, task-specific datasets—often requiring hundreds of thousands of trajectories—to achieve robust performance. For instance, RT-1 (Brohan et al., 2023) uses 130K real-world episodes, while RT-2 (Zitkovich et al., 2023) scales to 1B image-text pairs. Recent works like OpenVLA (Kim et al., 2024), Octo (Ghosh et al., 2024), Pi0 (Black et al., 2024), and RDT-1B (Liu et al., 2024a) further expand to millions of demonstrations across diverse embodiments. Despite these efforts, such data-intensive pipelines present a scalability bottleneck and limit generalization to novel tasks and domains, motivating more data-efficient and transferable alternatives. However, actions are typically coupled to their VLA models, which constrains efficient transfer to heterogeneous embodiments.

Coupled Video Generation and Action Synthesis. Recent efforts couple video generation with action synthesis to enhance physical consistency and interoperability. For instance, Video-Prediction-Policy (VPP) (Hu et al., 2024) learns implicit inverse dynamics conditioned on future representations predicted by a video diffusion model, while UVA (Li et al., 2025) unifies video-action latent spaces with lightweight diffusion heads. Other works, such as VidMan (Wen et al., 2024), integrate low-level action representations with video prediction to predict actions. These approaches bridge spatiotemporal dynamics between vision and action, offering novel solutions for complex manipulation tasks. However, these approaches require end-to-end joint training of video generation and action prediction models, which limits their flexibility and adaptability.

Table 5: Ablation study of Vidar, where "w/o MIDM" means using the ResNet baseline. In three scenarios, both masked inverse dynamics models and test-time scaling are beneficial to the success rates.

Configuration	Seen Tasks & Backgrounds	Unseen Tasks	Unseen Backgrounds
Vidar w/o TTS	45.5%	33.3%	44.4%
Vidar w/o MIDM	59.1%	26.7%	22.2%
Vidar (Ours)	**68.2%**	**66.7%**	**55.6%**

Video Generation Models for Embodied AI. Motivated by one implementation of world models (Ha & Schmidhuber, 2018), video generation models for embodied AI also predict future scene dynamics to assist robotic planning and policy learning. Current works such as UniPi (Du et al., 2023), RoboDreamer (Zhou et al., 2024), Gen2Act (Bharadhwaj et al., 2024), CLOVER (Bu et al., 2024), SuSIE (Black et al., 2023), and GR-1 (Wu et al., 2024a) mainly adopt text-conditioned video generation or frame prediction with a fixed single-camera view, demonstrating how a single-arm robot physically compliantly executes a series of actions. Another orthogonal line of research, such as Dreamitate (Liang et al., 2024), focuses on tool use video predictions, where tools serve as mediators to simplify manipulation tasks. Additionally, Genie 2 (Parker-Holder et al., 2024) generates interactive 3D environments via autoregressive latent diffusion, enabling scalable training for embodied agents. These models implicitly encode physical laws through video synthesis, reducing reliance on real-world robotics data. However, current works either remain limited to single-arm robots and rely on a single camera view, or rely on the availability and functionality of specific tools, which restricts their applicability in more complex real-world scenarios; meanwhile, the success of Instant3D (Li et al., 2024) provides a compelling demonstration that the diffusion models can effectively adapt to multi-view formats. Moreover, most existing methods do not utilize heterogeneous embodied videos for pre-training.

5 Conclusion

We presented Vidar, a generalizable framework for bimanual robotic manipulation that addresses the core challenges of data scarcity and embodiment heterogeneity. By combining large-scale, diffusion-based video pretraining over a unified observation space with a masked inverse dynamics model, Vidar enables accurate action prediction from multi-view visual observations and language instructions, requiring only minimal demonstrations in new environments. Our experiments demonstrate that Vidar consistently outperforms existing methods and exhibits strong generalization to unseen tasks and backgrounds, highlighting its capacity for semantic understanding and transfer.

6 Ethics Statement

Vidar has the potential to accelerate the deployment of capable bimanual robots in real-world environments. However, the rise of generalist robotic systems also introduces important considerations regarding privacy, safety, and accountability, especially as these systems are deployed in sensitive domains involving close human-robot interaction.

7 Reproducibility Statement

We submit our code, including the HunyuanVideo diffusion model and the masked inverse dynamics model, in the supplemental materials. Meanwhile, the Wan2.2 model that we use is open-sourced and widely available. Appendix A describes our dataset in detail, and Appendix B provides comprehensive information on training and inference. All datasets used for pre-training are publicly available.

REFERENCES

AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, Shu Jiang, Yuxin Jiang, Cheng Jing, Hongyang Li, Jialu Li, Chiming Liu, Yi Liu, Yuxiang Lu, Jianlan Luo, Ping Luo, Yao Mu, Yuehan Niu, Yixuan Pan, Jiangmiao Pang, Yu Qiao, Guanghui Ren, Cheng Ruan, Jiaqi Shan, Yongjian Shen, Chengshi Shi, Mingkang Shi, Modi Shi, Chonghao Sima, Jianheng Song, Huijie Wang, Wenhao Wang, Dafeng Wei, Chengen Xie, Guo Xu, Junchi Yan, Cunbiao Yang, Lei Yang, Shukai Yang, Maoqing Yao, Jia Zeng, Chi Zhang, Qinglin Zhang, Bin Zhao, Chengyue Zhao, Jiaqi Zhao, and Jianchao Zhu. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *CoRR*, abs/2503.06669, 2025. doi: 10.48550/ARXIV.2503.06669. URL https://doi.org/10.48550/arXiv.2503.06669.

Fan Bao, Chendong Xiang, Gang Yue, Guande He, Hongzhou Zhu, Kaiwen Zheng, Min Zhao, Shilong Liu, Yaole Wang, and Jun Zhu. Vidu: a highly consistent, dynamic and skilled text-to-video generator with diffusion models. *CoRR*, abs/2405.04233, 2024. doi: 10.48550/ARXIV.2405.04233. URL https://doi.org/10.48550/arXiv.2405.04233.

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. *CoRR*, abs/2409.16283, 2024. doi: 10.48550/ARXIV.2409.16283. URL https://doi.org/10.48550/arXiv.2409.16283.

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. *CoRR*, abs/2310.10639, 2023. doi: 10.48550/ARXIV.2310.10639. URL https://doi.org/10.48550/arXiv.2310.10639.

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. *CoRR*, abs/2410.24164, 2024. doi: 10.48550/ARXIV.2410.24164. URL https://doi.org/10.48550/arXiv.2410.24164.

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 Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu (eds.), *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. doi: 10.15607/RSS.2023.XIX.025. URL https://doi.org/10.15607/RSS.2023.XIX.025.

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. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/fad8962279154544ed69bb63eb14d677-Abstract-Conference.html.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Qiwei Liang, Zixuan Li, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. Robotwin 2.0: A scalable

data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *CoRR*, abs/2506.18088, 2025. doi: 10.48550/ARXIV.2506.18088. URL https://doi.org/10.48550/arXiv.2506.18088.

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. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *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. URL http://papers.nips.cc/paper_files/paper/2023/hash/1d5b9233ad716a43be5c0d3023cb82d0-Abstract-Conference.html.

Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile ALOHA: learning bimanual mobile manipulation with low-cost whole-body teleoperation. *CoRR*, abs/2401.02117, 2024. doi: 10.48550/ARXIV.2401.02117. URL https://doi.org/10.48550/arXiv.2401.02117.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In Dana Kulic, Gentiane Venture, Kostas E. Bekris, and Enrique Coronado (eds.), *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024. doi: 10.15607/RSS.2024.XX.090. URL https://doi.org/10.15607/RSS.2024.XX.090.

David Ha and Jürgen Schmidhuber. World models. *CoRR*, abs/1803.10122, 2018. URL http://arxiv.org/abs/1803.10122.

Ryan Hoque, Peide Huang, David J. Yoon, Mouli Sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. *CoRR*, abs/2505.11709, 2025. doi: 10.48550/ARXIV.2505.11709. URL https://doi.org/10.48550/arXiv.2505.11709.

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. *CoRR*, abs/2412.14803, 2024. doi: 10.48550/ARXIV.2412.14803. URL https://doi.org/10.48550/arXiv.2412.14803.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench: Comprehensive benchmark suite for video generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 21807–21818. IEEE, 2024. doi: 10.1109/CVPR52733.2024.02060. URL https://doi.org/10.1109/CVPR52733.2024.02060.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll L. Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, and Dane Sherburn. Gpt-4o system card. *CoRR*, abs/2410.21276, 2024. doi: 10.48550/ARXIV.2410.21276. URL https://doi.org/10.48550/arXiv.2410.21276.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura M. Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *CoRR*, abs/2504.16054, 2025. doi: 10.48550/ARXIV.2504.16054. URL https://doi.org/10.48550/arXiv.2504.16054.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, and Ilge Akkaya. Openai o1 system card. *CoRR*, abs/2412.16720, 2024. doi: 10.48550/ARXIV.2412.16720. URL https://doi.org/10.48550/arXiv.2412.16720.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Paul Foster, Pannag R. 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. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2024. URL https://proceedings.mlr.press/v270/kim25c.html.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Daquan Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models. *CoRR*, abs/2412.03603, 2024. doi: 10.48550/ARXIV.2412.03603. URL https://doi.org/10.48550/arXiv.2412.03603.

Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=2lDQLiH1W4.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *CoRR*, abs/2503.00200, 2025. doi: 10.48550/ARXIV.2503.00200. URL https://doi.org/10.48550/arXiv.2503.00200.

Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Conference on Robot Learning,*

6-9 November 2024, Munich, Germany, volume 270 of *Proceedings of Machine Learning Research*, pp. 3943–3960. PMLR, 2024. URL https://proceedings.mlr.press/v270/liang25b.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

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. *CoRR*, abs/2410.07864, 2024a. doi: 10.48550/ARXIV.2410.07864. URL https://doi.org/10.48550/arXiv.2410.07864.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, Lifang He, and Lichao Sun. Sora: A review on background, technology, limitations, and opportunities of large vision models. *CoRR*, abs/2402.17177, 2024b. doi: 10.48550/ARXIV.2402.17177. URL https://doi.org/10.48550/arXiv.2402.17177.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel J. Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *CoRR*, abs/2501.19393, 2025. doi: 10.48550/ARXIV.2501.19393. URL https://doi.org/10.48550/arXiv.2501.19393.

Tung D. Nguyen, Rui Shu, Tuan Pham, Hung Bui, and Stefano Ermon. Temporal predictive coding for model-based planning in latent space. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8130–8139. PMLR, 2021. URL http://proceedings.mlr.press/v139/nguyen21h.html.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alexander Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew E. Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Paul Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Haoshu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I. Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng,

Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi Jim Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J. Joshi, Niko Sünderhauf, Ning Liu, Norman Di Palo, Nur Muhammad (Mahi) Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R. Sanketi, Patrick Tree Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham D. Sonawani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Liangwei Xu, Xuanlin Li, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and RT-X models : Open x-embodiment collaboration. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pp. 6892–6903. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611477. URL https://doi.org/10.1109/ICRA57147.2024.10611477.

Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer, Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Hadsell, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale foundation world model. 2024. URL https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL http://proceedings.mlr.press/v139/radford21a.html.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells III, and Alejandro F. Frangi (eds.), *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, volume 9351 of *Lecture Notes in Computer Science*, pp. 234–241. Springer, 2015. doi: 10.1007/978-3-319-24574-4_28. URL https://doi.org/10.1007/978-3-319-24574-4_28.

Hengkai Tan, Yao Feng, Xinyi Mao, Shuhe Huang, Guodong Liu, Zhongkai Hao, Hang Su, and Jun Zhu. Anypos: Automated task-agnostic actions for bimanual manipulation. *CoRR*, abs/2507.12768, 2025. doi: 10.48550/ARXIV.2507.12768. URL https://doi.org/10.48550/arXiv.2507.12768.

Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan

Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Xiaofeng Meng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *CoRR*, abs/2503.20314, 2025. doi: 10.48550/ARXIV.2503.20314. URL https://doi.org/10.48550/arXiv.2503.20314.

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, Ping Luo, Ziwei Liu, Yali Wang, Limin Wang, and Yu Qiao. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=MLBdiWu4Fw.

Youpeng Wen, Junfan Lin, Yi Zhu, Jianhua Han, Hang Xu, Shen Zhao, and Xiaodan Liang. Vidman: Exploiting implicit dynamics from video diffusion model for effective robot manipulation. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/481c70828a4ff20d31a646cc6cc95f3d-Abstract-Conference.html.

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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024a. URL https://openreview.net/forum?id=NxoFmGgWC9.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, Zhen Zhao, Guangyu Li, Zhao Jin, Lecheng Wang, Jilei Mao, Xinhua Wang, Shichao Fan, Ning Liu, Pei Ren, Qiang Zhang, Yaoxu Lyu, Mengzhen Liu, Jingyang He, Yulin Luo, Zeyu Gao, Chenxuan Li, Chenyang Gu, Yankai Fu, Di Wu, Xingyu Wang, Sixiang Chen, Zhenyu Wang, Pengju An, Siyuan Qian, Shanghang Zhang, and Jian Tang. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *CoRR*, abs/2412.13877, 2024b. doi: 10.48550/ARXIV.2412.13877. URL https://doi.org/10.48550/arXiv.2412.13877.

Huazhi Xu, Xiaoyan Luo, and Wencong Xiao. Multi-residual unit fusion and wasserstein distance-based deep transfer learning for mill load recognition. *Signal Image Video Process.*, 18(4): 3187–3196, 2024. doi: 10.1007/S11760-023-02981-6. URL https://doi.org/10.1007/s11760-023-02981-6.

Chengbo Yuan, Suraj Joshi, Shaoting Zhu, Hang Su, Hang Zhao, and Yang Gao. Roboengine: Plug-and-play robot data augmentation with semantic robot segmentation and background generation. *CoRR*, abs/2503.18738, 2025. doi: 10.48550/ARXIV.2503.18738. URL https://doi.org/10.48550/arXiv.2503.18738.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=vSVLM2j9eie.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=kHjOmAUfVe.

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 Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

Table 6: Detailed information about datasets. Dataset instructions correspond to the robot and camera components of the unified observation space (see Figure 1). The platforms in the RoboTwin and the Vidar datasets are unseen during pre-training.

Dataset	Dataset Instruction
Agibot	The whole scene is in a realistic, industrial art style with three views: **a fixed high camera, a movable left arm camera, and a movable right arm camera**. The **genie-1 robot** is currently performing the following task:
RDT	The whole scene is in a realistic, industrial art style with three views: **a fixed front camera, a movable left arm camera, and a movable right arm camera**. The **aloha robot** is currently performing the following task:
RoboMind Franka	The whole scene is in a realistic, industrial art style with three views: **a fixed camera on the opposite side, a fixed left camera, and a fixed right camera**. The **franka robot** is currently performing the following task:
RoboMind Aloha	The whole scene is in a realistic, industrial art style with three views: **a fixed front camera, a movable left arm camera, and a movable right arm camera**. The **aloha robot** is currently performing the following task:
Egodex	The whole scene is in a realistic, industrial art style with one view: **a movable front camera**. The **person** is currently performing the following task:
RoboTwin (Low Data)	The whole scene is in a realistic, industrial art style with three views: **a fixed rear camera, a movable left arm camera, and a movable right arm camera**. The **aloha robot** is currently performing the following task:
RoboTwin (Standard Data)	The whole scene is in a realistic, industrial art style with three views: **a fixed front camera, a movable left arm camera, and a movable right arm camera**. The **aloha robot** is currently performing the following task:
Vidar	The whole scene is in a realistic, industrial art style with three views: **a fixed rear camera, a movable left arm camera, and a movable right arm camera**. The **aloha robot** is currently performing the following task:

A Dataset Details

The details of our datasets are presented in Table 6. For the RoboTwin dataset, we adjust the camera positions to capture both arms in full, rather than only the end-effectors, to improve training of the masked inverse dynamics model. For the Agibot-World dataset, episodes are segmented into shorter clips using their frame-level annotations. We also filter out episodes with fewer than three views or shorter than four seconds. For each dataset, we provide information about the associated robot and camera setups, which form part of the unified observation space. We also leverage GPT-4o to augment its task annotations for the Vidar dataset. During pre-training, sampling ratios for each dataset are set proportionally to their sizes.

It is worth noting that the RoboTwin dataset and our Vidar dataset differ from all pre-training datasets in these aspects: for example, beyond the use of different robot arms, the central camera is positioned far behind and high above the scene in the Vidar dataset—significantly different from the views in the pre-training datasets.

You are a skilled robot video ranker. Your task is to identify the index of the video with the highest quality based on the provided image clips and video caption. When evaluating the images, consider both their physical accuracy and how well they align with the video caption. Each image contains three views, and you must assess their consistency, ensuring there are no abrupt appearances or disappearances of objects or color blocks between frames. When determining the index, if there is a tie, output the smallest video index. For example: if video 1 has the highest quality, output 1; if video 2 has the highest quality, output 2; if all the videos have the same quality, output 1. We have {n_videos} videos, each containing {n_imgs_per_video} images, for you to evaluate. The caption for video_reference is '{caption}'. The images are arranged in the following sequence:
{img_seq}
Please assess the quality of the videos and provide the index of the one with the highest quality, without any explanations.

Figure 4: Prompt for GPT-4o evaluation. Variables in the curly braces should be replaced by corresponding values. Specifically, one line of "img_seq" describes one video, and is formatted as "-**video_1**: image_1, image_2, ..., image_{n_imgs_per_video}".

Table 7: Hyperparameters of MIDM.

Hyperparameter	Value
Number of Parameters	92 Million
U-Net Down-sampling/Up-sampling Layers	5
ResNet Structure	ResNet-50
Action Prediction Loss	Huber Loss
Learning Rate	5×10^{-4}
Warm-up	6000 Steps
AdamW β	$(0.9, 0.999)$
AdamW ϵ	10^{-8}
AdamW Weight Decay	10^{-2}

B Training and Inference Details

Using 64 NVIDIA Ampere-series 80GB GPUs, we train Vidu 2.0 for 23,000 iterations (10,000 for pre-training and the remainder for fine-tuning), which takes about 64 hours. Note that for all fine-tuning procedures, we employ full-parameter fine-tuning. For MIDM, we use 8 NVIDIA Hopper-series 80GB GPUs for 60,000 training iterations, taking about 5 hours. Additional MIDM hyperparameters are provided in Table 7. We trained Pi0.5 model as a baseline for 55,000 iterations for real-world tasks, with action horizon chosen to be 16. Additional Pi0.5 hyperparameters are provided in Table 8

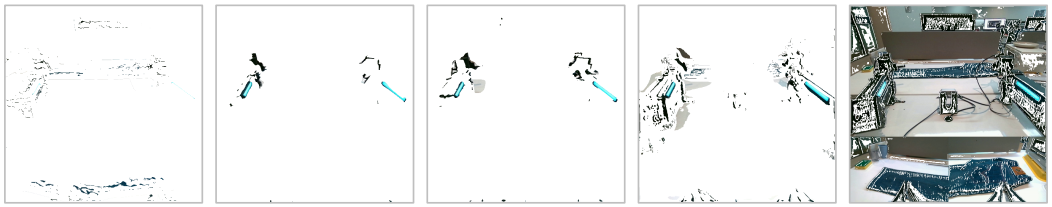
During inference, the video diffusion models are deployed in the cloud, while only the lightweight MIDM is executed locally. For test-time scaling, we uniformly sample 5 – 7 frames from the generated videos and use GPT-4o to select the best result. The prompt we use is shown in Figure 4, focusing on physical plausibility and alignment with the textual instruction. Each pairwise comparison costs about \$0.003, and it accounts for about 25% of the total latency when $K = 3$.

C Experimental Results

For the simulation experiments, we set $K = 1$ (i.e., no test-time scaling) for better reproducibility. During testing, we limit the maximum steps to 180, which means there are three model inferences with 60 steps each. We use Pi0.5 as our baseline and trained both Vidar and Pi0.5 under two data regimes. Specifically, under the standard-data setting, models are trained under the clean scenario with 50 episodes for each task; under the low-data setting, models are trained under clean scenario with 20 episodes with adjusted camera views to test their adaptivity to different views. Notably,

Table 8: Hyperparameters of Pi0.5.

Hyperparameter	Value
Number of Parameters	2 Billion
Learning Rate	2.5×10^{-5}
Batch Size	32
Warm-up	1000 Steps
Optimizer	AdamW
AdamW β	$(0.9, 0.95)$
AdamW ϵ	10^{-8}
AdamW Weight Decay	10^{-10}



(a) $\lambda = 10^{-1}$ (b) $\lambda = 10^{-2}$ (c) $\lambda = 3 \times 10^{-3}$ (d) $\lambda = 10^{-3}$ (e) $\lambda = 10^{-4}$

Figure 5: Masked images learned by the masked inverse dynamic model (MIDM) with different values of λ.

we adopt a multi-task setting instead of training the model separately for each task, which is more challenging. The testing success rates averaged over 100 episodes are shown in Table 9 and Table 10.

For the real-world experiments, a more detailed version of Table 2 is provided in Table 11. We also investigate the impact of the hyperparameter λ in MIDM, with results summarized in Table 12 and illustrated in Figure 5. Notably, $\lambda = 3 \times 10^{-3}$ yields the best performance.

We also test the performance of an existing segmentation model, RoboEngine (Yuan et al., 2025). As is shown in Figure 6, it often identifies only one arm per frame, fails to recognize grippers in wrist camera views, or lacks temporal consistency across frames.

D Additional Real-World Experiments

We also run additional real-world experiments using the Wan 2.2 model and the HunyuanVideo model as our backbone. The training hyperparameters are detailed in Table 13. Utilizing 64 NVIDIA Hopper-series 80GB GPUs, the training of the HunyuanVideo Model required approximately 54 hours.

Figure 6: Segmentation results. We split the concatenated video frames into three views and apply RoboEngine (Yuan et al., 2025) to each view.

Table 9: Success rates of Vidar and Pi0.5 on the RoboTwin 2.0 benchmark under clean scenario.

Data Regime	Low		Standard	
Task	Pi0.5	Vidar	Pi0.5	Vidar
Adjust Bottle	95.0%	100.0%	98.0%	63.0%
Beat Block Hammer	40.0%	85.0%	54.0%	93.0%
Blocks Ranking RGB	11.0%	55.0%	25.0%	52.0%
Blocks Ranking Size	1.0%	35.0%	10.0%	21.0%
Click Alarmclock	59.0%	100.0%	93.0%	95.0%
Click Bell	63.0%	95.0%	92.0%	100.0%
Dump Bin Bigbin	35.0%	50.0%	49.0%	72.0%
Grab Roller	81.0%	100.0%	96.0%	96.0%
Handover Block	2.0%	5.0%	4.0%	2.0%
Handover Mic	17.0%	0.0%	31.0%	24.0%
Hanging Mug	0.0%	0.0%	8.0%	1.0%
Lift Pot	14.0%	90.0%	3.0%	93.0%
Move Can Pot	18.0%	60.0%	23.0%	48.0%
Move Pillbottle Pad	11.0%	70.0%	31.0%	72.0%
Move Playingcard Away	58.0%	100.0%	74.0%	97.0%
Move Stapler Pad	1.0%	35.0%	19.0%	28.0%
Open Laptop	42.0%	50.0%	46.0%	73.0%
Open Microwave	11.0%	20.0%	21.0%	43.0%
Pick Diverse Bottles	20.0%	55.0%	24.0%	67.0%
Pick Dual Bottles	23.0%	85.0%	54.0%	87.0%
Place A2B Left	5.0%	45.0%	53.0%	86.0%
Place A2B Right	3.0%	55.0%	43.0%	91.0%
Place Bread Basket	33.0%	75.0%	46.0%	82.0%
Place Bread Skillet	1.0%	85.0%	41.0%	79.0%
Place Burger Fries	28.0%	80.0%	78.0%	93.0%
Place Can Basket	19.0%	50.0%	25.0%	38.0%
Place Cans Plasticbox	4.0%	0.0%	17.0%	69.0%
Place Container Plate	62.0%	100.0%	80.0%	98.0%
Place Dual Shoes	1.0%	0.0%	4.0%	9.0%
Place Empty Cup	20.0%	100.0%	95.0%	92.0%
Place Fan	8.0%	45.0%	21.0%	55.0%
Place Mouse Pad	3.0%	60.0%	19.0%	74.0%
Place Object Basket	31.0%	35.0%	66.0%	55.0%
Place Object Scale	14.0%	85.0%	40.0%	75.0%
Place Object Stand	36.0%	95.0%	64.0%	90.0%
Place Phone Stand	19.0%	75.0%	30.0%	82.0%
Place Shoe	13.0%	80.0%	61.0%	89.0%
Press Stapler	58.0%	90.0%	80.0%	98.0%
Put Bottles Dustbin	0.0%	0.0%	11.0%	3.0%
Put Object Cabinet	0.0%	0.0%	33.0%	22.0%
Rotate QRcode	27.0%	65.0%	51.0%	65.0%
Scan Object	4.0%	45.0%	17.0%	47.0%
Shake Bottle	86.0%	100.0%	93.0%	99.0%
Shake Bottle Horizontally	72.0%	100.0%	100.0%	99.0%
Stack Blocks Three	4.0%	15.0%	9.0%	25.0%
Stack Blocks Two	37.0%	80.0%	57.0%	90.0%
Stack Bowls Three	3.0%	45.0%	37.0%	39.0%
Stack Bowls Two	22.0%	95.0%	75.0%	92.0%
Stamp Seal	7.0%	50.0%	28.0%	68.0%
Turn Switch	27.0%	60.0%	10.0%	60.0%
Average	25.0%	60.0%	44.8%	65.8%

Table 10: Success rates of Vidar and Pi0.5 on the RoboTwin 2.0 benchmark under randomized scenario.

Data Regime	Low		Standard	
Task	Pi0.5	Vidar	Pi0.5	Vidar
Adjust Bottle	16.0%	65.0%	35.0%	39.0%
Beat Block Hammer	5.0%	10.0%	5.0%	10.0%
Blocks Ranking RGB	0.0%	0.0%	0.0%	4.0%
Blocks Ranking Size	0.0%	0.0%	0.0%	0.0%
Click Alarmclock	30.0%	35.0%	67.0%	74.0%
Click Bell	16.0%	25.0%	42.0%	68.0%
Dump Bin Bigbin	16.0%	10.0%	25.0%	10.0%
Grab Roller	28.0%	30.0%	34.0%	37.0%
Handover Block	2.0%	0.0%	0.0%	0.0%
Handover Mic	0.0%	0.0%	4.0%	8.0%
Hanging Mug	0.0%	0.0%	2.0%	0.0%
Lift Pot	0.0%	10.0%	0.0%	4.0%
Move Can Pot	4.0%	0.0%	4.0%	0.0%
Move Pillbottle Pad	6.0%	20.0%	2.0%	4.0%
Move Playingcard Away	19.0%	40.0%	13.0%	22.0%
Move Stapler Pad	0.0%	0.0%	6.0%	7.0%
Open Laptop	26.0%	30.0%	2.0%	24.0%
Open Microwave	4.0%	0.0%	8.0%	5.0%
Pick Diverse Bottles	10.0%	0.0%	6.0%	9.0%
Pick Dual Bottles	17.0%	15.0%	6.0%	30.0%
Place A2B Left	1.0%	10.0%	7.0%	7.0%
Place A2B Right	0.0%	15.0%	7.0%	17.0%
Place Bread Basket	10.0%	15.0%	15.0%	7.0%
Place Bread Skillet	0.0%	10.0%	12.0%	8.0%
Place Burger Fries	14.0%	5.0%	47.0%	11.0%
Place Can Basket	4.0%	0.0%	2.0%	4.0%
Place Cans Plasticbox	0.0%	0.0%	11.0%	13.0%
Place Container Plate	38.0%	55.0%	31.0%	23.0%
Place Dual Shoes	0.0%	0.0%	0.0%	3.0%
Place Empty Cup	4.0%	20.0%	32.0%	33.0%
Place Fan	0.0%	0.0%	2.0%	10.0%
Place Mouse Pad	0.0%	10.0%	1.0%	14.0%
Place Object Basket	5.0%	10.0%	6.0%	4.0%
Place Object Scale	1.0%	0.0%	9.0%	13.0%
Place Object Stand	22.0%	35.0%	10.0%	10.0%
Place Phone Stand	7.0%	25.0%	3.0%	18.0%
Place Shoe	6.0%	40.0%	13.0%	24.0%
Press Stapler	13.0%	40.0%	58.0%	48.0%
Put Bottles Dustbin	0.0%	0.0%	3.0%	0.0%
Put Object Cabinet	0.0%	0.0%	1.0%	3.0%
Rotate QRcode	4.0%	10.0%	0.0%	1.0%
Scan Object	0.0%	5.0%	1.0%	6.0%
Shake Bottle	56.0%	65.0%	67.0%	75.0%
Shake Bottle Horizontally	46.0%	60.0%	59.0%	73.0%
Stack Blocks Three	0.0%	0.0%	0.0%	3.0%
Stack Blocks Two	8.0%	5.0%	9.0%	16.0%
Stack Bowls Three	0.0%	15.0%	10.0%	4.0%
Stack Bowls Two	6.0%	35.0%	31.0%	30.0%
Stamp Seal	1.0%	0.0%	4.0%	7.0%
Turn Switch	17.0%	10.0%	0.0%	33.0%
Average	9.2%	15.7%	14.2%	17.5%

Table 11: Detailed success rates of various methods and configurations on robot manipulation tasks. "L", "R", and "B" indicate the use of the left arm, right arm, or both arms, respectively. "w/o TTS" and "w/o MIDM" correspond to the ablation settings described in Table 5.

Scenario/Task	Success Rate				
Seen Tasks & Backgrounds	UniPi	VPP	**Vidar**	w/o TTS	w/o MIDM
Grasp the Tomato (L)	60.0%	0.0%	60.0%	60.0%	80.0%
Grasp the Tomato (R)	20.0%	0.0%	80.0%	60.0%	60.0%
Lift the Basket (B)	66.7%	0.0%	66.7%	33.3%	33.3%
Flip the Dice to Point One on the Top (L)	0.0%	33.3%	33.3%	0.0%	33.3%
Grab the Bottle (L)	0.0%	0.0%	66.7%	33.3%	33.3%
Get the Toast from the Toaster (L)	66.7%	0.0%	100.0%	66.7%	100.0%
Average	36.4%	4.5%	**68.2%**	45.5%	59.1%
Unseen Tasks	UniPi	VPP	**Vidar**	w/o TTS	w/o MIDM
Place the Bowl on the Steamer (L)	0.0%	0.0%	66.7%	66.7%	33.3%
Grasp the Shortest Bread (L)	0.0%	0.0%	66.7%	33.3%	0.0%
Grasp the Shortest Bread (R)	0.0%	0.0%	66.7%	33.3%	33.3%
Wipe the Table with Rag (L)	33.3%	33.3%	66.7%	33.3%	66.7%
Wipe the Table with Rag (R)	0.0%	33.3%	66.7%	0.0%	0.0%
Average	6.7%	13.3%	**66.7%**	33.3%	26.7%
Unseen Backgrounds	UniPi	VPP	**Vidar**	w/o TTS	w/o MIDM
Grasp the Tomato (L)	0.0%	0.0%	66.7%	33.3%	0.0%
Flip the Die to Point One on the Top (L)	0.0%	0.0%	33.3%	33.3%	0.0%
Flip the Die to Point One on the Top (R)	0.0%	0.0%	33.3%	0.0%	0.0%
Pick the Carrot on the Green Plate (L)	33.3%	0.0%	33.3%	66.7%	0.0%
Place the Bowl on the Plate (L)	33.3%	0.0%	66.7%	66.7%	33.3%
Place the Bowl on the Plate (R)	66.7%	0.0%	100.0%	66.7%	100.0%
Average	22.2%	0.0%	**55.6%**	44.4%	22.2%

Table 12: Analysis of hyperparameter λ in the masked inverse dynamics model. The success rates are robust over a large range, and $\lambda = 3 \times 10^{-3}$ achieves the best performance.

λ	Training Accuracy	Testing Accuracy	Testing l_1 Error
10^{-1}	99.1%	7.1%	0.0670
10^{-2}	99.8%	39.9%	0.0331
3×10^{-3}	**99.9%**	**49.0%**	**0.0308**
10^{-3}	99.9%	40.7%	0.0338
10^{-4}	99.8%	24.4%	0.0461

For the Wan 2.2 model, we evaluate it alongside Pi0.5 (also trained for 55,000 iterations). Since Pi0.5 is typically fine-tuned on a larger dataset, we expand the fine-tuning set to 2,307 episodes. We then evaluate both models on 14 tasks (7 seen and 7 unseen), with results reported in Table 14. As shown, Vidar consistently outperforms Pi0.5 on average.

We evaluate the HunyuanVideo model version on six tasks. The results are presented in Table 15 and Figure 7. We plan to open-source our implementations related to both the HunyuanVideo Model and our MIDM.

E Additional Visualizations

More demonstrations, including two failed cases, are shown in Figure 8.

Table 13: Hyperparameters of the Wan 2.2 model and the HunyuanVideo Model.

Hyperparameter	Wan2.2	HunyuanVideo
Number of Parameters	5 Billion	13 Billion
Learning Rate (Pre-train)	2×10^{-5}	1×10^{-4}
Learning Rate (Fine-tune)	2×10^{-5}	5×10^{-5}
Warm-up	200 Steps	500 steps
Optimizer	AdamW	AdamW
AdamW β	$(0.9, 0.999)$	$(0.9, 0.999)$
AdamW ϵ	10^{-8}	10^{-8}
AdamW Weight Decay	0.1	0
Pre-training Steps	10,000	10,000
Fine-tuning Steps	12,000	2,000

Table 14: Success rates of Vidar reproduced using the Wan2.2 Model. "L", "R", and "B" indicate the use of the left arm, right arm, or both arms, respectively.

Seen Cases	Vidar	Pi0.5
Flip the Die to Show One on Its Top (L)	40.0%	20.0%
Grasp the Fruit from the Basket (R)	60.0%	0.0%
Click the Mouse's Left Button (L)	80.0%	0.0%
Lift the Basket (B)	80.0%	20.0%
Pour Water into the Cup (L)	100.0%	25.0%
Wipe the Table with a Blue Rag (L)	100.0%	100.0%
Grab the Blue Cup (R)	25.0%	75.0%
Average	69.3%	34.3%
Unseen Cases	**Vidar**	**Pi0.5**
Close the Laptop (L)	80.0%	0.0%
Close the Laptop (R)	60.0%	0.0%
Throw the Paper Ball (L)	80.0%	40.0%
Shake the Water Bottle (R)	60.0%	0.0%
Grasp and Place Two Breads (B)	90.0%	20.0%
Wipe the Desk with a Blue Towel (B)	20.0%	0.0%
Manipulate the Controller Handle (L)	80.0%	30.0%
Average	67.1%	12.9%

Table 15: Success rates of Vidar reproduced using the HunyuanVideo model.

Task	Success Rate
Grasp the Apple (Seen Task)	75.0%
Grab the Bottle (Seen Task)	100.0%
Grasp the Cup by its Handle (Seen Task)	25.0%
Lift the Steamer (Unseen Task)	50.0%
Grasp the Apple and Put it into the Steamer (Unseen Task)	100.0%
Stack One Cube on Top of the Other Cube (Unseen Task)	20.0%
Average	58.3%

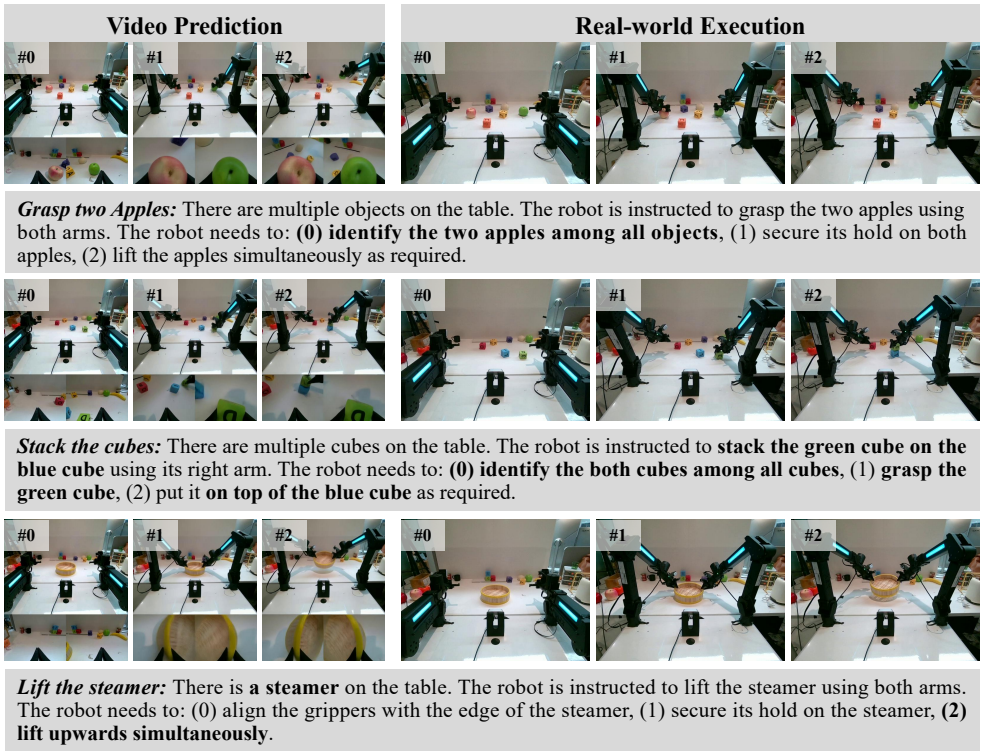


Figure 7: Videos of predictions (left) and corresponding executions (right) of Vidar reproduced using the HunyuanVideo model for challenging tasks.

F Hardware Details

Hardware details are shown in Figure 9 and Table 16. One important assumption underlying this approach is that the intermediate video modality contains all the information for action prediction. However, this assumption fails to hold for many robotic systems due to the specific platform setting, including the Aloha platform. In the pre-training part of Figure 1, we can see that the arm joints frequently fall outside the camera's field of view, even with three different views available. In our target domain, we adjust the center camera to a new position where two arms can be fully captured, shown in the fine-tuning part of Figure 1. In this way, our robotic platform differs from any platform we encountered during pre-training, serving as an ideal testbed for showing adaptation capability with scarce data.

G Large Language Model Usage

We utilize large language models to refine the writing. For example, we write a draft and let the model check the grammar errors.

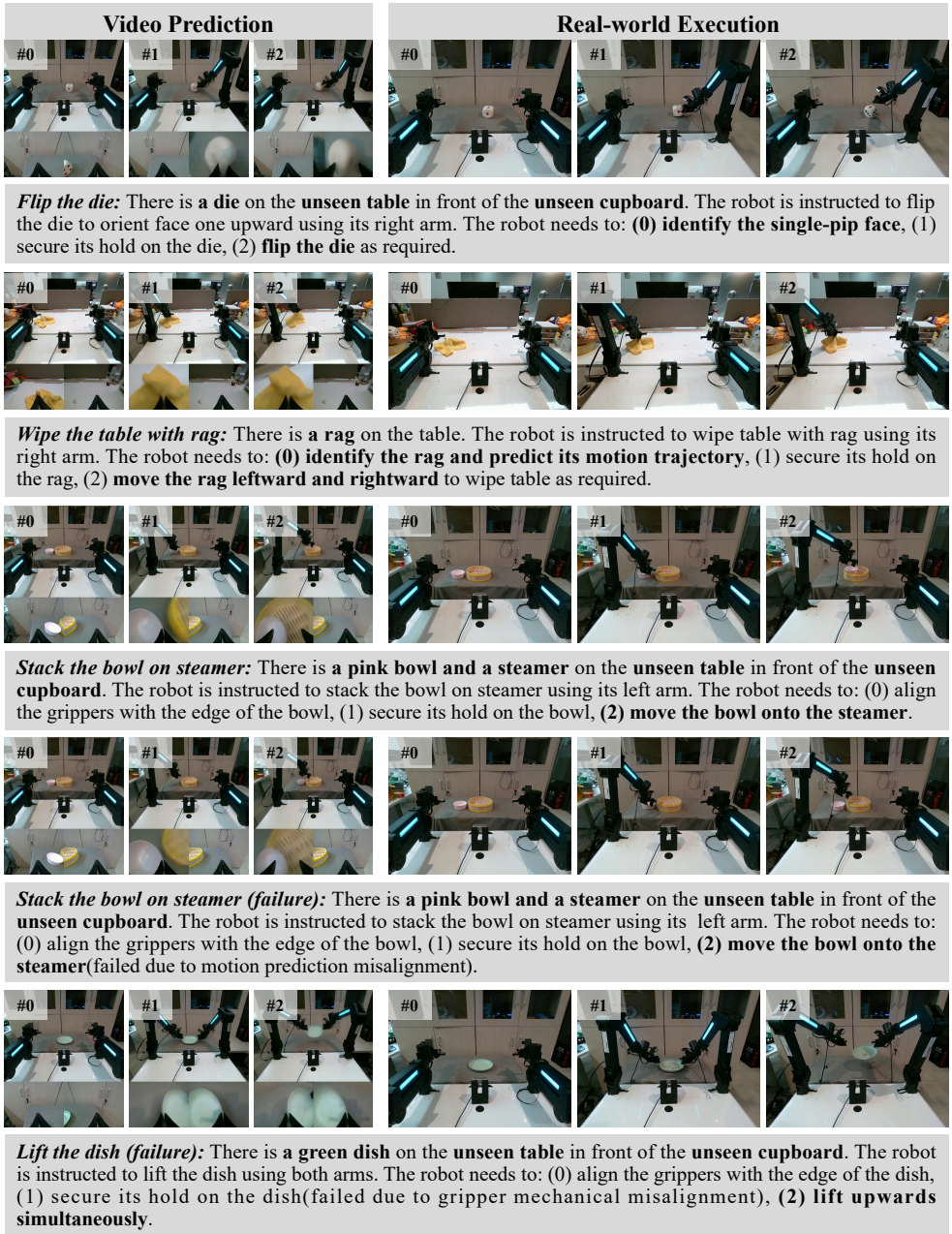


Figure 8: Videos of predictions (left) and corresponding executions (right) of Vidar for more challenging tasks. Both successful and failed cases are presented.

Figure 9: Our robotic platform.

Table 16: Hardware Information.

Parameter	Value
Degree of Freedom	$2 \times (6 + 1) = 14$
Cameras	3 RGB Cameras
Arm weight	$3.9\,\mathrm{kg}$
Arm Valid Payload	$1.0\,\mathrm{kg}$
Arm Reach	$0.6\,\mathrm{m}$
Arm Repeatability	$1\,\mathrm{mm}$
Gripper Range	0 - $80\,\mathrm{mm}$
Gripper Max Force	$10\,\mathrm{N}$