ANT GROUP

Scaling Mixture-of-Experts Video Pretraining for Embodied Intelligence

Shuailei Ma*, Jiaqi Liao*, Xinyang Wang*, Jingjing Wang*, Chaoran Feng, Zijing Hu, Chong Bao, Zichen Xi, Yuqi Gan, Weisen Wang, Yanhong Zeng, Qin Zhao, Zifan Shi, Wei Wu, Hao Ouyang, Qiuyu Wang, Shangzhan Zhang, Jiahao Shao, Yipengjing Sun, Liangxiao Hu, Lunke Pan, Nan Xue, Kecheng Zheng, Yinghao Xu, Xing Zhu, Yujun Shen, Ka Leong Cheng†

*Equal Contribution †Project Lead

Despite the recent promise in robot control, video generative models suffer from a domain mismatch due to their primary focus on content creation. For example, their design inherently prioritizes visual fidelity and creativity over computational efficiency and physical realism. In this work, we present `LingBot-Video`, a DiT-based video pretraining paradigm specifically tailored for embodied intelligence. From the ***architecture*** perspective, we adopt the Mixture-of-Experts (MoE), instead of dense, framework to achieve a better trade-off between modeling capacity and inference efficiency, and manage to scale it up from scratch. From the ***data*** perspective, we construct a data profiling engine that augments standard internet videos with extensive robot-oriented footage, encompassing manipulation, navigation, and egocentric perspectives, to equip the base model with an intrinsic understanding of actions and world dynamics. From the ***training*** perspective, we develop a multi-dimensional reward system to enforce the alignment regarding physical rationality and task completion, going beyond standard criteria such as aesthetics, prompt-following, and motion consistency. Comprehensive evaluations validate its performance and efficiency as a video foundation model. We contribute `LingBot-Video` as the inaugural large-scale, open-source MoE video foundation model to the community, in a pioneering effort to bridge digital creativity and physical actuation.

Website: https://technology.robbyant.com/lingbot-video
Github: https://github.com/robbyant/lingbot-video
Checkpoints: https://huggingface.co/robbyant/lingbot-video

Robbyant

1 Introduction

Beyond their success in content creation, diffusion-based [7, 9, 36, 37, 81, 89, 105] and autoregressive [32, 101, 119] video models have demonstrated remarkable ability to synthesize temporally coherent and photorealistic sequences conditioned on text, images, and other control signals [10, 23, 31, 92, 96]. This capability has motivated a growing body of work that interprets video models as implicit simulators of the physical world, enabling their use in robotics [2, 52, 55], autonomous driving [77, 78], and interactive environments [10]. In this paradigm, video models serve not only as generative systems but also as predictive world models [4] that support planning, policy learning, and imagination-based control. However, translating these models from passive video generation to active embodied reasoning and intelligence remains an open challenge.

Despite their promise, a fundamental gap persists between video generation models and embodied intelligence requirements. Most video foundation models are optimized for perceptual quality—such as realism, aesthetics, and text alignment—rather than physical correctness or controllability. While these objectives yield visually compelling results, they do not explicitly enforce consistency with physical interaction constraints, such as contact stability, rigid-body dynamics, or long-horizon state consistency under intervention. This highlights a key tension: while internet-scale video provides rich visual diversity, it does not guarantee fidelity to the constraints of embodied interaction.

Figure 1. Samples of *Text-to-Image* and *Text-to-Video* tasks generated by `LingBot-Video`. `LingBot-Video` can produce images and videos with high visual fidelity, rich details, and strong text-prompt alignment across diverse scenes and subjects.

Existing efforts to bridge video generation and embodied intelligence typically fall short across three tightly coupled dimensions: architecture, data, and training objectives. Architecturally, most diffusion-based video transformers rely on dense computation, where all parameters are activated uniformly across tokens and timesteps, resulting in prohibitive inference costs and limited scalability. While recent sparse Mixture-of-Experts (MoE) formulations [50, 57] demonstrate promising efficiency gains in large language models (LLM) [1, 5, 8, 94, 99, 121], their adoption in video generation remains limited. Data-wise, training corpora are dominated by internet videos lacking robot embodiment priors or precise interaction dynamics, leading to weak grounding in physical simulation. From a training perspective, current alignment strategies primarily optimize aesthetic quality and text-video correspondence, without incorporating explicit physical feasibility, task completion, or long-horizon reward signals. Consequently, existing systems struggle to simultaneously achieve scalability, physical consistency, and embodiment grounding.

In this work, we propose `LingBot-Video`, a DiT-based video pretraining paradigm specifically designed for embodied intelligence. Our approach addresses the aforementioned limitations through three integrated components. First, we introduce a **Mixture-of-Experts (MoE) video frameword**, which enables sparse conditional computation, improving inference efficiency while scaling model capacity for complex spatiotemporal dynamics. Second, we construct **a robot-augmented pretraining corpus** that unifies internet-scale videos with robot manipulation, navigation, and egocentric datasets, thereby injecting explicit embodiment and interaction priors into the model. Third, we develop **a multi-dimensional reward system** that extends beyond aesthetic objectives to incorporate physical rationality and task-oriented success signals, encouraging the model to learn dynamics and interactions consistent with embodied environments. Together, these components enable a more physically grounded and computationally efficient video foundation model.

Overall, we present LingBot-Video, to the best of our knowledge the **first** large-scale open-source Mixture-of-Experts video foundation model for embodied intelligence, bridging the gap between digital video generation and physical actuation. Our contributions are threefold:

- We introduce a sparse Mixture-of-Experts (MoE) video diffusion framework with a scalable training paradigm, enabling scalability–efficiency trade-off for spatiotemporal modeling.
- We develop a dedicated data profiling engine that systematically analyzes, filters, and rebalances heterogeneous video sources. This enables effective integration of large-scale internet videos with embodied datasets, resulting in improved grounding in physical interactions, action semantics, and embodiment-specific dynamics.
- We propose a multi-dimensional reward system that incorporates physical plausibility and task-level success signals, extending beyond conventional perceptual and text-alignment objectives.

2 Method

2.1 Task-Unified Single-Stream Diffusion Transformer

Our architecture adopts a cascaded design consisting of a task-unified base generator and a refiner. The base generator employs a Single-Stream Diffusion Transformer to process compact visual latents and multimodal conditions. We use Qwen3-VL-4B [6] to extract condition from multimodal instructions. Wan2.1-VAE [105] is employed for efficient visual latent compression. This section is organized as follows: first, we introduce the **Unified Input Formulation** and **Single-Stream Diffusion Transformer**; then, we detail our scaling strategy via **Sparse Mixture-of-Experts**; finally, we present the **Cascaded Refiner**.

Unified Input Formulation. We represent each training sample as a single token sequence consisting of visual latent patch and condition tokens. Specifically, after projecting visual patches and condition features into the same hidden dimension, we concatenate them along the sequence dimension to form a unified input. Under this formulation, we handle Text-to-Image (T2I), Text-to-Video (T2V), and Image-to-Video (TI2V) tasks within a single framework, representing image targets as a special single-frame ($T = 1$) video generation case. To resolve the structural discrepancy between condition and visual tokens, we employ a 3D MM-RoPE [11, 71, 113] mechanism to place them in non-overlapping temporal coordinate ranges, eliminating the need for task-specific architectures or encoders.

Single-Stream Diffusion Transformer. Efficiency and scalability guide the design of LingBot-Video. Inspired by the scalability of LLMs' decoder-only architecture [1, 5, 94, 99] and the parameter efficiency of single-stream designs, we adopt a streamlined single-stream diffusion transformer backbone. After lightweight modality-specific input projections, all visual and condition tokens share the same transformer blocks. This single-stream architecture maximizes parameter reuse and facilitates dense cross-modal interactions at every layer. Compared to dual-stream architectures [24] that process modalities via separate pathways, our unified backbone handles all tokens as a single sequence. This design groups multi-modal features into larger, unified GEMM computations, aiming to improve Model FLOPs Utilization (MFU) [17] and reduce kernel launch overhead. Furthermore, while dual-stream counterparts require frequent concatenation and splitting of conditioning and latent tokens before and after attention within every block, our model processes the unified sequence continuously. This reduces memory-bandwidth-bound tensor reorganization and layout conversion overhead [19], which is particularly beneficial under sequence-parallel distributed layouts [43, 48]. Consequently, in distributed scaling scenarios, this design simplifies communication patterns and helps avoid extra load imbalance and synchronization overhead from dual-stream partitioning.

Multi-Modal 3D RoPE. To separate condition and visual tokens while preserving video geometry within a single stream, we map all tokens to a Multi-Modal 3D RoPE [11, 71, 113] coordinate system. Specifically, given L condition tokens and an $F \times H \times W$ visual latent grid, conditioning tokens use temporal-only coordinates $(i, 0, 0)$ for $i = 1, \ldots, L$, while visual patch tokens use $(L + 1 + f, h, w)$. Query and key head dimensions are split across the temporal, vertical, and horizontal axes, applying the corresponding rotary frequencies independently. This maintains spatial locality and temporal order while keeping attention fully single-stream.

QK-Norm. To stabilize attention in deep transformer backbones, we normalize queries and keys with per-head RMSNorm before computing attention, following prior large-scale vision and diffusion transformer practice [21, 24]. This stabilization mechanism helps control attention-logit growth and feature scales during high-resolution training,

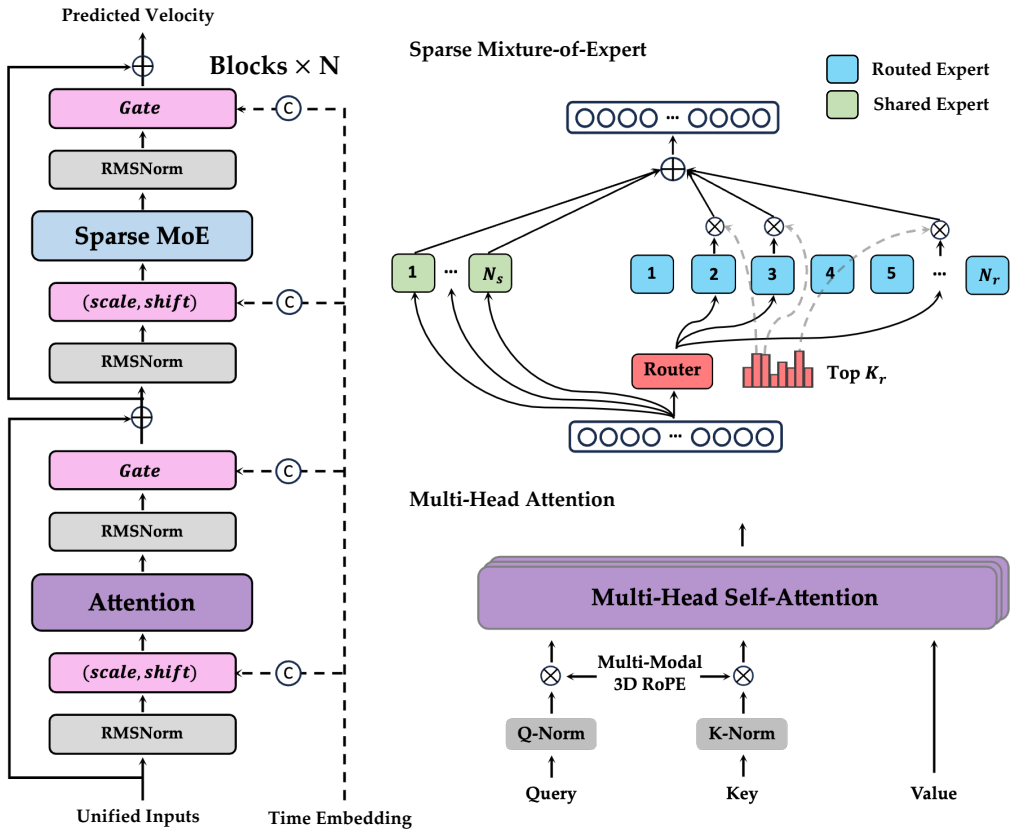


Figure 2. Overview of the task-unified single-stream diffusion transformer. Unified inputs are processed by stacked transformer blocks, where timestep modulation controls the attention and Sparse MoE branches; the attention branch applies QK-Norm and multi-modal 3D RoPE, while the MoE branch combines always-on shared experts with top-K_r routed experts before predicting velocity.

particularly under mixed precision. Additionally, each block utilizes RMSNorm around the attention and feed-forward residual branches to keep activations bounded without introducing modality-specific pathways.

AdaLN-Single Modulation. To reduce modulation overhead, we adopt the adaLN-single design [13, 105, 106] by computing a shared timestep modulation once before the transformer stack. Each block adds a layer-specific trainable modulation table to this shared signal and uses the resulting shift, scale, and gate parameters to modulate its attention and feed-forward branches. The shared projection is zero-initialized following DiT-style adaptive normalization [70], while the per-layer modulation tables are initialized with small random values, providing stable residual learning without per-block timestep MLPs.

2.2 Scaling with Sparse Mixture-of-Experts

The Mixture-of-Experts (MoE) paradigm is fundamentally motivated by the need to model highly diverse and complex data distributions, utilizing specialized subnetworks (experts) to capture distinct patterns or domains [42, 45]. Video generation, which aims to simulate the continuous physical world, demands the model to capture the visual representations of complex physical processes (such as fluid motion and three-dimensional spatial consistency) along with varying motion trajectories and rich material textures [9, 37, 105]. Consequently, MoE is a natural candidate for scaling up video diffusion models to capture the vast complexity of physical laws [69, 82]. In a conventional dense model, the Feed-Forward Network (FFN) forces all tokens to share the same parameter pathway. In unified video pretraining, this leads to severe subtask interference [42, 45, 67] since a single set of parameters must simultaneously model asymmetric domains, such as spatial textures versus temporal motion, and accommodate diverse task formats spanning T2I, T2V, and TI2V conditions. Particularly for diffusion models, the training dynamics inherently exhibit a shift

between distinct noise regimes, requiring the model to resolve the discrepancy between low-noise detail reconstruction and high-noise global layout formulation.

Sparse Mixture-of-Experts (MoE) has been robustly proven in large language models as a powerful paradigm to scale parameter capacity under a constant computational budget [18, 25, 50, 57, 86]. Driven by this success, we incorporate the sparse MoE framework into our base generator to achieve capacity-compute decoupling. This allows us to scale the total parameter capacity—creating a vast repository for diverse physical priors—while strictly controlling per-token active FLOPs. This decoupling is especially critical for high-resolution video generation, which easily scales to million-token (1M+) spatio-temporal sequences where dense scaling would incur prohibitive computational overhead. Diffusion-based video generators repeatedly process the full visual latent sequence across denoising steps [37, 70], making per-token transformer computation a dominant scaling pressure for long videos. Sparse routing reduces the active feed-forward computation at each denoising step while preserving total parameter capacity, making long-video capacity scaling more practical.

Sparse MoE Architecture. In each transformer block, we preserve the single-stream FFN residual branch structure and replace only its dense feed-forward computation with a token-choice sparse MoE layer [25, 50, 86], as shown in Fig. 2. Our MoE design incorporates key architectural principles of DeepSeekMoE [18, 57], specifically fine-grained expert segmentation and shared expert isolation, to encourage expert specialization while maintaining shared common priors. Given the modulated FFN input $\mathbf{u}_t$ of token t, the Sparse MoE layer computes a branch output from shared experts and routed experts:

$$m(\mathbf{u}_t) = \sum_{i=1}^{N_\mathrm{s}} E_i^{(\mathrm{s})}(\mathbf{u}_t) + \sum_{j \in \mathcal{R}_\mathrm{b}(\mathbf{u}_t)} g_{t,j} E_j^{(\mathrm{r})}(\mathbf{u}_t), \tag{1}$$

where $\mathcal{R}_\mathrm{b}(\mathbf{u}_t)$ is the selected routed-expert set, and $E_i^{(\mathrm{s})}(\cdot)$ and $E_j^{(\mathrm{r})}(\cdot)$ denote the i-th shared expert function and j-th routed expert function, respectively. Each expert is implemented as a SwiGLU MLP [84]. We denote N_s as the number of shared experts and N_r as the total number of routed experts. The gate $g_{t,j}$ is defined by the routing equations below. The resulting MoE branch output $m(\mathbf{u}_t)$ is injected back into the transformer block through the gated residual branch of the single-stream block. To maximize expert specialization and reduce knowledge hybridity [18], we adopt a fine-grained expert segmentation strategy, setting the intermediate dimension of both the shared and routed experts to a smaller width than a standard dense FFN counterpart. Under this design, the shared experts provide a common pathway to capture general physical principles and spatial consistency across all tokens, while the routed experts capture specialized features.

For token t, we compute routed-expert affinities with a sigmoid router [57]:

$$\alpha_{t,j} = \mathrm{Sigmoid}\left(\mathbf{u}_t^\top \mathbf{r}_j\right), \tag{2}$$

Here, $\alpha_{t,j}$ is the router affinity between token t and routed expert j, and $\mathbf{r}_j \in \mathbb{R}^d$ is the learnable router embedding of the j-th routed expert. To control communication cost in distributed training, we adopt a DeepSeek-style group-limited routing strategy [57]. We divide the N_r routed experts into N_g groups. After adding the online correction bias, $\tilde{\alpha}_{t,j} = \alpha_{t,j} + b_j$, we select the top K_g groups, where each group is scored by the sum of its top-2 bias-corrected affinity scores. Let $\mathcal{G}_\mathrm{b}(\mathbf{u}_t)$ denote the set of experts belonging to these selected groups. Within $\mathcal{G}_\mathrm{b}(\mathbf{u}_t)$, we select the top K_r experts as $\mathcal{R}_\mathrm{b}(\mathbf{u}_t)$. The overall MoE architecture is depicted in Fig. 2.

Online Bias Correction for Load Balancing. To maintain load balance while preserving representation capacity, we use an auxiliary-loss-free load-balancing strategy [57, 108]. We introduce a dynamic correction bias b_j for each expert, which is adjusted during training and used only for selecting the top experts:

$$g'_{t,j} = \begin{cases} \alpha_{t,j}, & \text{if } \tilde{\alpha}_{t,j} \in \mathrm{TopK}_{K_\mathrm{r}}(\{\tilde{\alpha}_{t,k} \mid k \in \mathcal{G}_\mathrm{b}(\mathbf{u}_t)\}), \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Here, $g'_{t,j}$ is the sparse unnormalized gate: it keeps the original affinity $\alpha_{t,j}$ for selected experts and sets all unselected experts to zero. The final gating values are obtained by normalizing the selected original affinity scores and applying a route scaling factor γ:

$$g_{t,j} = \gamma \frac{g'_{t,j}}{\sum_{k=1}^{N_\mathrm{r}} g'_{t,k}}. \tag{4}$$

Here, $g_{t,j}$ is the final routing weight applied to expert j for token t, γ is the route scaling factor, and k indexes routed experts in the normalization term. During training, the correction bias for expert j is updated online at each optimizer step using the sign of its load deviation:

$$b_j \leftarrow b_j - \eta \, \mathrm{sign}(n_j - \bar{n}), \tag{5}$$

where b_j is the correction bias of expert j, n_j is the number of valid token assignments selecting expert j (accumulated globally across ranks), $\bar{n}$ is the average load per expert, and η is the learning rate for bias updates. When bias centering is enabled, we mean-center the bias after each update:

$$b_j \leftarrow b_j - \frac{1}{N_\mathrm{r}} \sum_{k=1}^{N_\mathrm{r}} b_k. \tag{6}$$

This centering preserves relative load-balancing signals while keeping the router input-dependent and stable.

Sequence-Wise Auxiliary Loss. Video generation naturally involves long spatio-temporal token sequences, where batch-level expert-balance statistics can hide routing imbalance within individual videos. We therefore adopt the sequence-wise auxiliary balance loss from DeepSeek-V3 [57], which encourages balanced expert usage within each packed video sequence rather than only at the global batch level. For a packed batch containing S sequences, the sequence-wise balance loss is defined as:

$$\mathcal{L}_\mathrm{seq} = \frac{1}{S} \sum_{s=1}^{S} \sum_{j=1}^{N_\mathrm{r}} f_j^{(s)} P_j^{(s)}, \tag{7}$$

Here, S is the number of packed sequences, s indexes a sequence, and j indexes a routed expert. We compute the average normalized routing probability for expert j in sequence s as:

$$P_j^{(s)} = \frac{1}{T_s} \sum_{t=1}^{T_s} p_{t,j}, \qquad p_{t,j} = \frac{\alpha_{t,j}}{\sum_{k=1}^{N_\mathrm{r}} \alpha_{t,k}}. \tag{8}$$

Here, T_s is the token length of the s-th packed unified sequence, $p_{t,j}$ is the normalized routing probability of token t choosing expert j, and $P_j^{(s)}$ is its average over the sequence. We compute the normalized assignment frequency as:

$$f_j^{(s)} = \frac{N_\mathrm{r}}{K_\mathrm{r} T_s} c_j^{(s)}, \qquad c_j^{(s)} = \sum_{t=1}^{T_s} \mathbf{1}\left[\alpha_{t,j} \in \mathrm{TopK}_{K_\mathrm{r}}(\{\alpha_{t,k} \mid 1 \le k \le N_\mathrm{r}\})\right]. \tag{9}$$

Here, $c_j^{(s)}$ is the number of tokens in sequence s for which expert j belongs to the unbiased top-K_r set computed from the raw affinity $\alpha_{t,j}$ before online bias correction and group-limited routing. The indicator $\mathbf{1}[\cdot]$ returns one when this selection condition is true and zero otherwise. The normalized frequency $f_j^{(s)}$ is detached from the computation graph so that gradients only flow through $P_j^{(s)}$. The auxiliary balance loss is applied to all transformer blocks and added to the diffusion loss:

$$\mathcal{L}_\mathrm{aux} = \lambda_\mathrm{aux} \mathcal{L}_\mathrm{seq}, \tag{10}$$

$$\mathcal{L} = \mathcal{L}_\mathrm{diff} + \mathcal{L}_\mathrm{aux}, \tag{11}$$

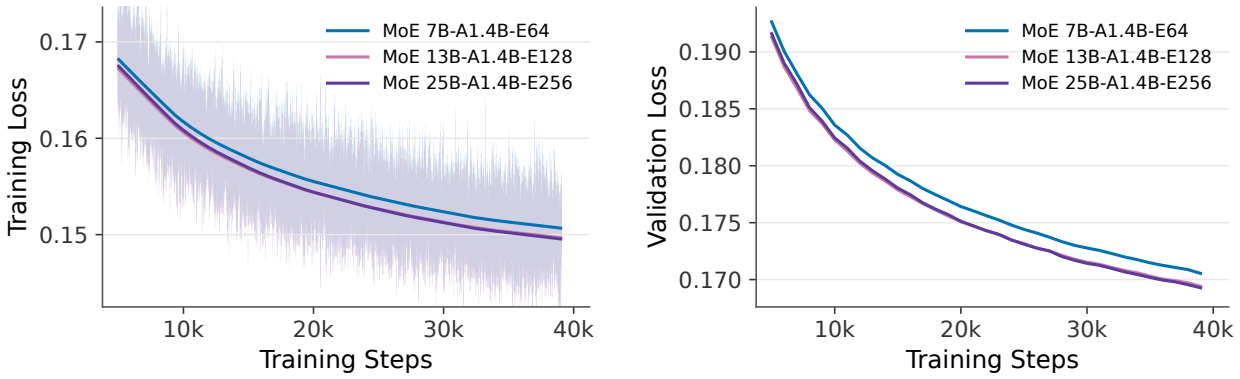
where λ_aux is the auxiliary loss weight, $\mathcal{L}_\mathrm{aux}$ is the weighted sequence-wise balance loss, $\mathcal{L}_\mathrm{diff}$ is the diffusion training loss, and $\mathcal{L}$ is the final training objective. By routing each token to K_r active routed experts, we keep the active computational cost per token comparable to a standard dense counterpart, while the total parameter capacity scales with the size of the expert pool N_r. This decoupling ensures that the generator scales its representation capacity to capture complex physical priors without incurring a linear increase in per-token FLOPs.

2.3 Sparse Mixture-of-Experts Recipe Exploration

To identify the optimal sparse routing configuration for unified video diffusion pre-training, we systematically explore the MoE design space. We structure our exploration along two primary dimensions: first, the scaling of the expert pool (total parameter capacity) under a fixed active compute budget; second, the granularity of routing (fine-grained

routing with many small experts versus coarse routing with fewer large experts) under a fixed total parameter budget. All configurations are evaluated on a unified pre-training mixture.

Expert Scale & Capacity-Compute Decoupling. We first evaluate the impact of expanding the total parameter capacity by scaling the number of available routed experts $E \in \{64, 128, 256\}$ while holding the active parameter scale strictly constant at 1.4 B per token. As shown in Fig. 3, scaling the expert count yields consistent improvements in both training and validation losses, demonstrating the efficacy of capacity-compute decoupling. However, the performance gain from $E = 128$ to $E = 256$ is marginal compared to the significant improvement from $E = 64$ to $E = 128$. Considering this trade-off among performance, latency, and memory footprint, we choose $E = 128$ as the default expert scale, which captures most of the observed capacity benefit while avoiding the additional communication and storage overhead of a larger expert pool.



(a). Training loss. **(b).** Validation loss.

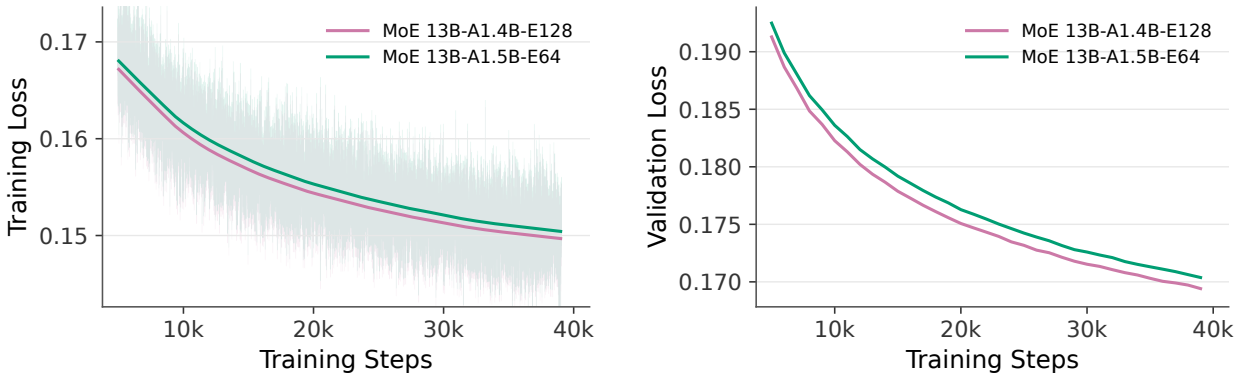
Figure 3. Expert-count recipe comparison using training and validation loss. Training curves show raw logged losses in the background and smoothed curves for visualization; validation curves are unsmoothed and aligned to the training-loss step range.

Fine-Grained Specialization vs. Coarse Routing. Next, we ablate the partitioning of FFN parameters under a fixed total parameter budget of 13 B. We compare a fine-grained routing recipe (*MoE 13B-A1.4B-E128*, which routes each token to $K_\mathrm{r} = 8$ out of 128 experts) against a coarse routing recipe (*MoE 13B-A1.5B-E64*, which routes each token to $K_\mathrm{r} = 4$ out of 64 experts). As illustrated in Fig. 4, despite the coarse routing model having a higher active parameter count (and thus a larger FLOP footprint per token), it performs consistently worse than the fine-grained counterpart throughout training. This performance gap highlights the advantages of fine-grained expert specialization [18, 57]. By dividing the FFN parameters into more, smaller experts, the model gains access to a dramatically larger combinatorial routing space ($\binom{128}{8}$ vs $\binom{64}{4}$), following the sparse top-K routing principle of MoE models [25, 50, 86]. This enables visual tokens representing heterogeneous modalities and noise levels to form more customized execution pathways, whereas routing to fewer, larger experts leads to parameter-level gradient conflicts [42, 45, 67] and knowledge hybridity [18].

2.4 Scaling Experiments

To verify the scalability and efficiency of our sparse single-stream diffusion transformer, we conduct scaling experiments across models of varying sizes, following the established practice of analyzing model scaling behavior and sparse MoE efficiency [25, 38, 46, 50, 86]. We first benchmark the model under a compute-comparable regime, evaluating the benefits of sparse parameters against standard dense baselines. We then investigate the scaling trajectories of our architecture up to 120 B total parameters. Finally, we analyze the inference efficiency of our sparse architecture under varying sequence lengths.

Active-Parameter Comparable Scaling. As illustrated in Fig. 5, we compare *MoE 13B-A1.4B* (13B total parameters, 1.4B active parameters) against a standard *Dense 1.3B* baseline under a similar active compute budget. The sparse model exhibits a substantial performance advantage in both training and validation losses throughout the training process. In video pre-training, modeling complex spatio-temporal features and continuous physical interactions requires a high representation capacity [37, 70]. Under equivalent computational constraints, the tenfold increase in total parameter capacity of the sparse MoE model provides a significantly larger repository for physical-world priors, consistent with the



(a). Training loss. **(b).** Validation loss.

Figure 4. Active-capacity recipe comparison using training and validation loss. Training curves show raw logged losses in the background and smoothed curves for visualization; validation curves are unsmoothed and aligned to the training-loss step range.

capacity-compute decoupling enabled by sparse expert routing [25, 50, 86]. This effectively resolves the feature-capacity bottleneck of the dense baseline, enabling the sparse model to achieve superior performance without inflating the compute budget.

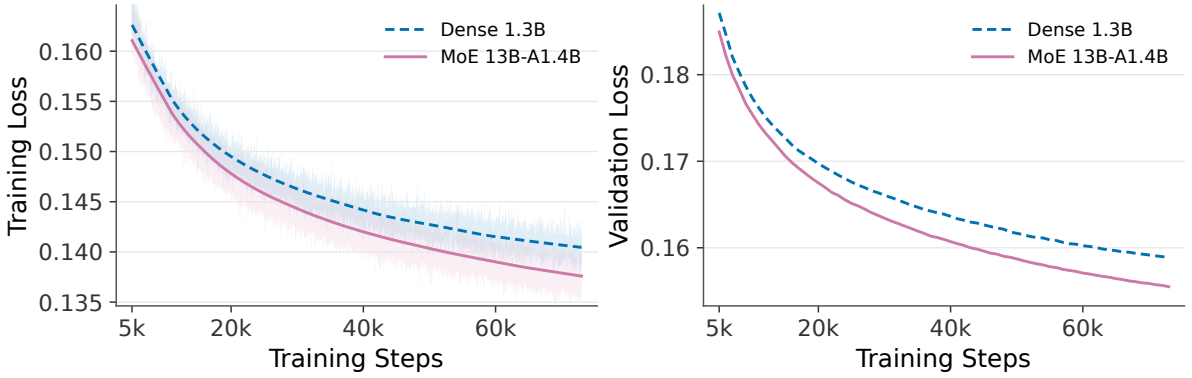
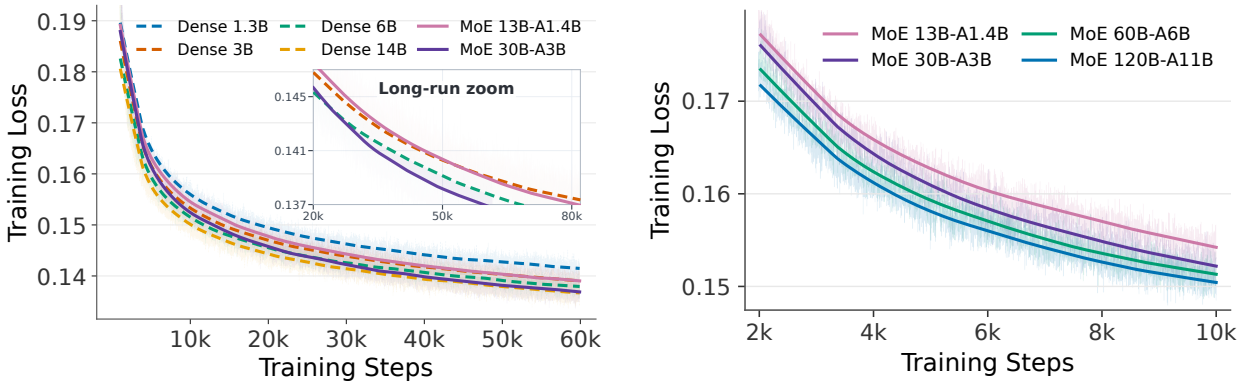


Figure 5. Comparable active-parameter scaling comparison between Dense 1.3B and MoE 13B-A1.4B using training and validation loss. Training curves show raw logged losses in the background and smoothed curves for visualization; validation curves are unsmoothed and aligned to the training-loss step range.

Cross-Compute Dominance. Furthermore, our sparse architecture demonstrates striking cross-compute dominance over dense baselines. As shown in Fig. 6a, both *MoE 13B-A1.4B* and *MoE 30B-A3B* consistently outperform dense models with roughly twice their active parameter scale. Specifically, *MoE 30B-A3B* closely approaches the performance of Dense 14B. This proves that capacity-compute decoupling fundamentally accelerates the scaling efficiency of video diffusion models, unlocking superior representation power.

Predictable Scaling Laws up to 120B. To probe the scaling potential of our architecture under practical resource constraints, we conduct an early-stage scaling study by increasing the total parameter count from 13 B to 120 B, with active parameters scaled proportionally from 1.4 B to 11 B (*MoE 13B-A1.4B*, *MoE 30B-A3B*, *MoE 60B-A6B*, and *MoE 120B-A11B*). As shown in Fig. 6b, when the models are compared at aligned training steps, larger sparse models consistently achieve lower training loss, exhibiting a predictable scale-dependent improvement consistent with neural scaling-law observations [38, 46]. Although these runs are not trained to full convergence due to resource constraints, the early training trajectories provide a practical initial validation of the scalability of our sparse single-stream design with auxiliary-loss-free load balancing and sequence-wise auxiliary loss.

Inference Efficiency. To verify that sparse capacity does not translate into prohibitive inference costs, we benchmark depth-matched dense and MoE DiT variants across sequence lengths ranging from 16K to 1M (1,048,576) tokens,



(a). Dense and sparse scaling.

(b). MoE scaling at aligned steps.

Figure 6. Training-loss comparison for compute-comparable scaling experiments. The faint background traces show the raw logged training losses; the bold curves are smoothed only for visualization and are not used to alter the underlying measurements.

following the sparse MoE literature's emphasis on conditional computation and routing overhead [25, 50, 57, 86].

As illustrated in Fig. 7, we analyze the inference speed ratio, defined as $r = T_{\text{dense}}/T_{\text{MoE}}$, where $r > 1$ indicates that *MoE 30B-A3B* is faster than the corresponding dense baseline. We first compare the MoE model against its active-parameter equivalent, *Dense 3B*, to assess routing overhead. At a sequence length of 1M tokens, where attention and FFN computations dominate execution [111], the sparse model achieves near-parity latency with a ratio of $0.97\times$. Next, we evaluate the efficiency gains of sparsity against larger dense models. As the parameter scale increases, the MoE architecture demonstrates substantial latency advantages: at 1M tokens, the speed ratios against *Dense 6B*, *Dense 14B*, and *Dense 30B* reach $1.50\times$, $2.59\times$, and $3.18\times$, respectively. These results demonstrate that our sparse framework successfully scales model capacity while preserving the inference efficiency of a 3B-scale model, offering a highly practical architecture for long-context video generation.

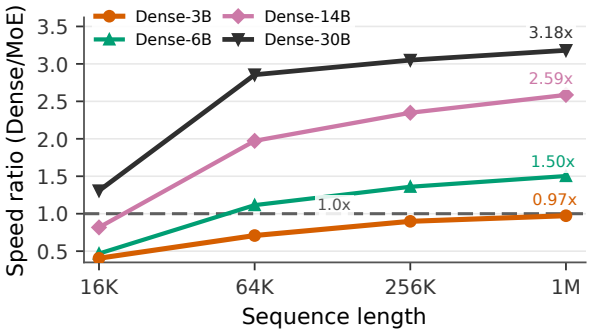


Figure 7. MoE-to-dense speed ratio, computed as dense latency divided by *MoE 30B-A3B* latency.

2.5 Cascaded Refiner

Directly generating high-resolution videos is computationally expensive due to the massive number of spatio-temporal tokens. To balance the trade-off between computational complexity and generation quality, we adopt a cascaded design. This follows the common practice of decomposing high-resolution diffusion generation into a base stage and one or more refinement or super-resolution stages [9, 36, 79]. A high-capacity base generator first models the overall motion and scene layout at a lower resolution. Subsequently, a dedicated second-stage refiner increases the resolution. In practice, we upsample the video from 480p to 1080p. During training, we simulate the base generator's outputs by downsampling the target video. We then apply synthetic degradations, including Gaussian blur and compression, to construct a degraded low-resolution input. Such synthetic degradation pipelines are widely used in blind restoration and real-world super-resolution to improve robustness to imperfect low-resolution inputs [109, 122]. This degraded video is spatially upsampled to the target resolution in pixel space. We then encode it using the VAE encoder to obtain the condition latent $\boldsymbol{x}_{\text{lr}}$. Processing the upsampled conditioning directly in the latent space avoids intermediate pixel-space decoding. This reduces computational overhead while preserving visual quality [12, 112, 124, 125].

Instead of denoising from pure Gaussian noise, the refiner learns a conditional rectified flow starting from the

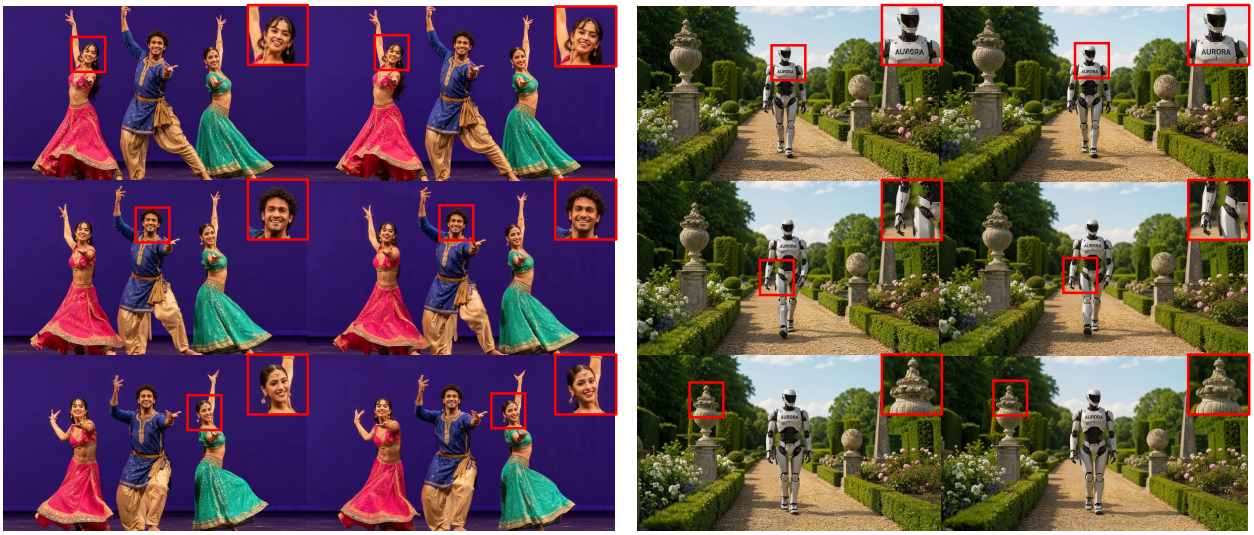


Figure 8. Refiner Generation. We compare the base video generation against the refined video generation. The left example's comprehensive prompt is "*Three dancers, a man in the center and two women on either side, are performing a traditional Indian dance on a stage...*". The right example's comprehensive prompt is "*A humanoid robot named 'AURORA' walks steadily down a gravel path in a meticulously manicured formal garden...*".

degraded condition $\boldsymbol{x}_{\text{lr}}$ toward the clean target latent $\boldsymbol{x}_0$ encoded from the target video. This formulation is built on rectified flow and flow-matching objectives, which learn continuous transport dynamics through vector-field regression [56, 60, 98]. During training, the refiner threshold $\tau \sim \text{Uniform}(0.85, 0.95)$ defines the maximum noise level for the conditional trajectory. We perturb $\boldsymbol{x}_{\text{lr}}$ with Gaussian noise ϵ to form the noisy starting condition $\boldsymbol{x}_\tau = (1-\tau)\boldsymbol{x}_{\text{lr}} + \tau\epsilon$. For a sampled training timestep $t \in [0, \tau]$, the perturbed latent $\boldsymbol{x}_t$ and the target velocity $v^\star_{\text{ref}}$ are formulated as:

$$\boldsymbol{x}_t = \left(1 - \frac{t}{\tau}\right)\boldsymbol{x}_0 + \frac{t}{\tau}\boldsymbol{x}_\tau, \quad v^\star_{\text{ref}} = \frac{\boldsymbol{x}_\tau - \boldsymbol{x}_0}{\tau}. \tag{12}$$

The model is optimized using the same flow-matching loss formulation as the base generator, restricted to timesteps $t \le \tau$. This thresholded trajectory limits denoising to noise regimes near the degraded condition. Consequently, the refiner preserves the base model's global semantics and motion. Its capacity is dedicated to restoring high-frequency details, sharpening textures, and correcting local artifacts.

During inference, the low-resolution video generated by the base stage is upsampled to the target resolution in pixel space. We then encode it to obtain the condition latent $\boldsymbol{x}_{\text{lr}}$. We perturb $\boldsymbol{x}_{\text{lr}}$ with Gaussian noise ϵ at a starting timestep $t = \tau_{\text{inf}}$ (e.g., $\tau_{\text{inf}} = 0.85$). This yields the noisy starting latent $\boldsymbol{x}_{\tau_{\text{inf}}} = (1-\tau_{\text{inf}})\boldsymbol{x}_{\text{lr}} + \tau_{\text{inf}}\epsilon$. Denoising is performed by integrating the rectified flow ODE trajectory backwards from $t = \tau_{\text{inf}}$ to $t = 0$, producing the clean video latent. As shown in Fig. 8, the refiner significantly enhances face appearance (left), while successfully restoring high-frequency details and producing sharp, legible OCR text (right).

3 Data

The performance of video generation models is inextricably linked to the scale, quality, and diversity of their training data. However, naively scaling up data collection yields diminishing returns and is often bottlenecked by practical factors such as acquisition costs and computational overhead. To address these challenges, we have developed an integrated and scalable data infrastructure built through several synergistic modules. The **Data Profiling Engine** extracts multi-dimensional attributes for each sample—covering structural, semantic, motion, camera, and quality aspects—providing a unified foundation for downstream processing. The **World-Knowledge Topological Graph** organizes visual concepts into a hierarchical semantic structure, enabling distribution-aware sampling and enhancing long-tail coverage. The **Dense Structured Captioning** module generates hierarchical textual descriptions to provide

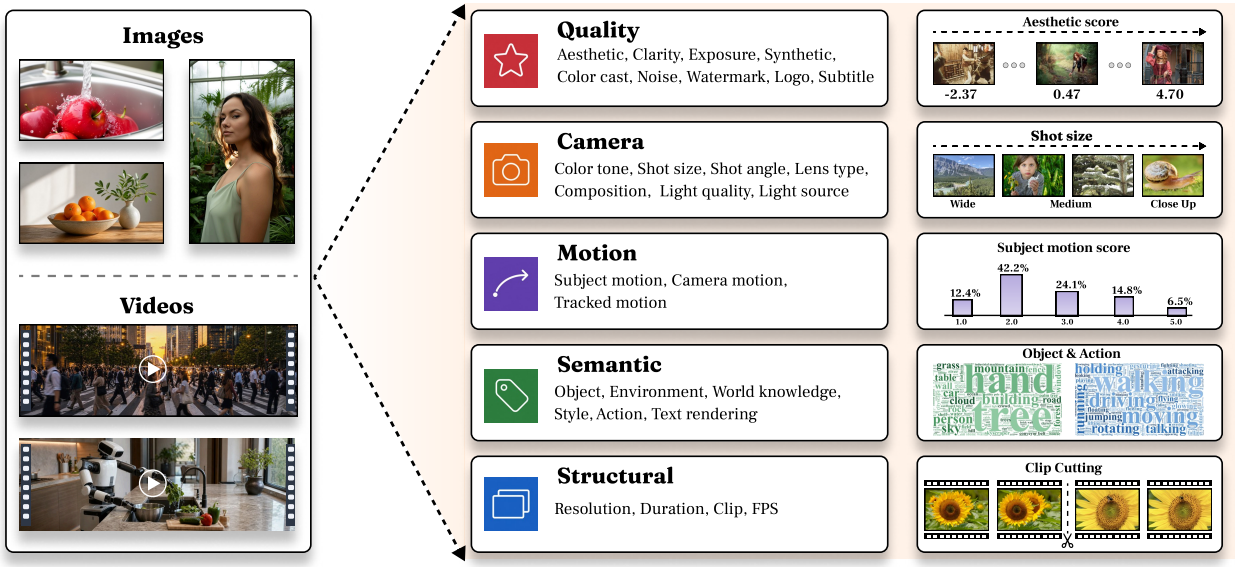


Figure 9. Overview of the Data Profiling Engine. Each image or video sample is annotated across five complementary dimensions (structural, semantic, motion, camera, and quality) into a structured profile record, which then drives downstream filtering, balanced sampling, and captioning.

fine-grained semantic supervision, while a complementary **Caption Rewriter** maps brief user prompts into this same structured format during inference, thereby closing the prompt distribution gap between training and deployment. Building upon this infrastructure, we organize the curated corpus into a stage-wise **Data Curriculum** (Sec. 3.5), which governs the volume and composition of data consumed during each phase of progressive pre-training (Sec. 5).

3.1 Data Profiling Engine

The Data Profiling Engine converts raw multimodal samples into structured, multi-dimensional records that capture structural, semantic, motion, camera, and quality attributes. Rather than relying on free-form annotations, we project every sample onto a fixed schema, providing a unified and queryable representation for heterogeneous image and video data. These standardized records drive all subsequent processing stages, from filtering and sampling to captioning. The core annotations are generated by powerful vision–language models (VLMs), further augmented by a suite of specialized scoring and detection models detailed below. Fig. 9 provides an overview of this pipeline.

Structural Metadata. This dimension records fundamental media attributes, including spatial resolution, native frame rate, and duration. For video content, we employ TransNetV2 [91] to detect shot boundaries and segment the video stream into clip-level units, ensuring that each training clip represents a single, temporally coherent shot.

Semantic Labels. Each sample is assigned semantic annotations along with category-level confidence scores, which serve as the foundation for the World-Knowledge Topological Graph used during distribution-aware sampling. Concurrently, the VLM annotator [6, 72] extracts structured tags to decompose the scene into its constituent elements: foreground and background objects, environmental settings, world-knowledge entities, visual styles, actions, and rendered text. This provides both a high-level holistic understanding of the scene and explicit entity anchors for precise data curation.

Motion and Temporal Dynamics. For video data, the engine characterizes motion along three complementary signals: camera motion, subject motion, and tracked motion. The VLM annotator decouples camera motion from subject motion, rating the intensity of each. Complementing these VLM-based estimates, a geometry-grounded *tracked-motion* signal derived from LocoTrack [16] point tracking screens out near-static or degenerate clips that would otherwise receive spurious motion scores. Together, these cues enable the downstream curation stage to effectively balance static and highly dynamic content.

Camera and Cinematic Attributes. Each sample is further tagged with seven distinct cinematic attributes, generated by

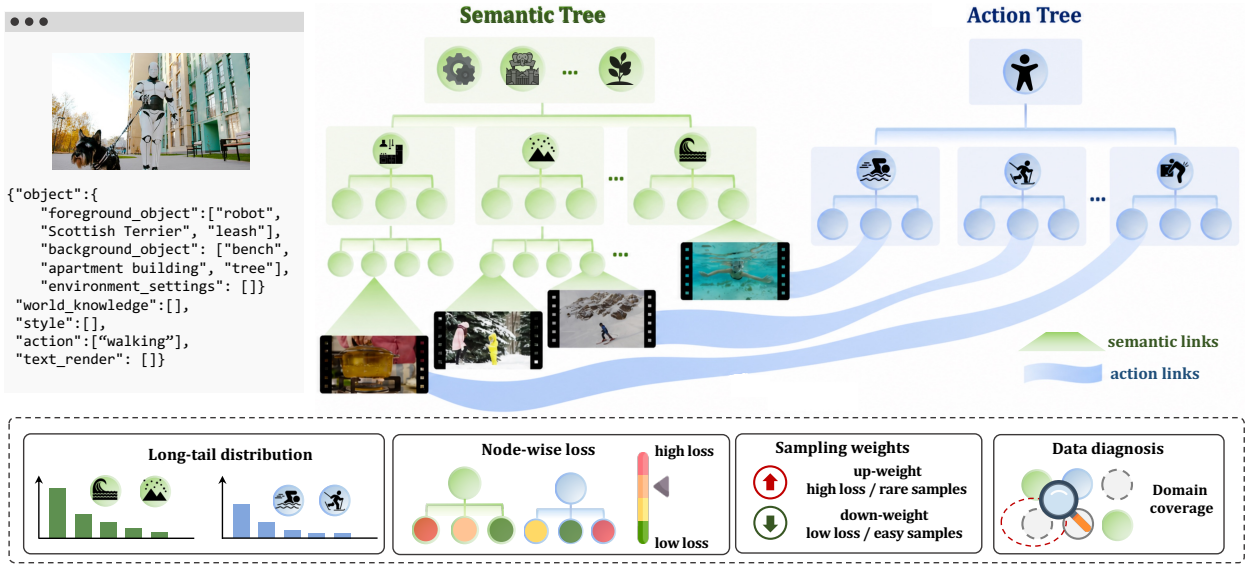


Figure 10. World-Knowledge Topological Graph. Structured tags from the Data Profiling Engine link each sample to a semantic tree of visual concepts and, for videos with actions, to an action tree of dynamic behaviors. The graph serves as a control surface for data curation: node statistics and training-feedback signals are used to up-weight rare or difficult concepts, down-weight saturated easy modes, and identify under-covered domains for targeted data expansion.

the VLM annotator distilled on cinematography data. These attributes include color tone, shot size, shot angle, lens type, composition, light quality, and light source (e.g., a warm-toned, medium-wide, high-angle shot lit by hard daylight). Making these attributes explicit facilitates fine-grained control over cinematic styles during generation and supports style-aware data balancing during curation.

Quality and Aesthetic Signals. Visual quality is evaluated using both dedicated learned scorers and VLM judgments. We compute an aesthetic score using HPSv3 [62]. Additionally, we estimate the likelihood that a sample is synthetic (AI-generated) using OmniAID [30] to detect and down-weight generated imagery. Concurrently, the base pass of the VLM annotator provides ordinal ratings for clarity and exposure, along with binary artifact indicators that flag watermarks, logos, subtitles, color casts, and noise. Collectively, these signals inform the automated filtering and reweighting stage, ensuring that low-quality and artifact-laden samples are removed or down-weighted prior to training.

3.2 World-Knowledge Topological Graph

To make data curation distribution-aware rather than quality-aware only, we construct a World-Knowledge Topological Graph [11] that organizes samples along two complementary axes: a semantic concept tree and an action tree. The semantic tree is shared by images and videos, providing a common vocabulary for objects, scenes, visual styles, and world-knowledge entities. The action tree is attached only to video samples and captures the dynamic behaviors that are especially important for video pre-training, such as object manipulation, sports, daily activities, and human gestures. Together, the two views allow us to query not only what appears in a sample, but also what happens in it.

Semantic Concept Tree. We first extract semantic tags from the Data Profiling Engine, including foreground and background objects, environment settings, world-knowledge entities, styles, and rendered text (Fig. 10). Directly embedding these raw tags is unreliable because many tags are sparse, ambiguous, or dominated by proper nouns and domain-specific terms. We therefore expand each tag with a short textual introduction that describes its visually grounded meaning, and then embed the concatenation of the tag and its introduction. The resulting representation is clustered together with frequently used Wikipedia concepts [68, 110] to form a large hierarchical inventory. This hierarchy contains 50,000 fine-grained leaf concepts and 1,000 intermediate visual categories. At lower levels, embedding-based clustering [102] provides high coverage and scalability. At the top level, however, purely automatic clustering tends to disagree with human perceptual granularity; for example, visually similar categories may be separated by ontology names, while visually distinct categories may be merged because their text descriptions are semantically close. We

therefore use an LLM-assisted **Discover–Classify–Consolidate** procedure to merge the $1{,}000$ intermediate categories into 25 visually coherent top-level groups. The discovery stage proposes an initial flat taxonomy from representative categories, the classification stage assigns all intermediate categories while allowing bounded new-bucket proposals, and the consolidation stage merges overlapping buckets, renames unclear buckets, and absorbs under-populated ones. This produces a human-aligned tree in which each sample is assigned to a fine-to-coarse semantic path.

Action Tree. For videos, we additionally construct an action tree from the action tags produced by the profiler. Because raw action tags are short, noisy, and redundant, we first normalize their surface forms and then expand each tag with a short VLM-generated description grounded in representative video snippets. We embed the concatenation of the action tag and its description, and cluster the resulting representations into canonical action nodes. This description-augmented clustering reduces ambiguity in short action phrases while grouping visually similar motion patterns. After clustering, we apply lightweight review and correction to high-frequency or ambiguous nodes, and discard unreliable long-tail actions. This produces several hundred canonical action nodes covering manipulation, human gestures, sports, and daily activities. All samples are indexed by their semantic paths, while videos with reliable action annotations are additionally linked to one or more canonical action nodes, giving us a joint semantic-action profile.

Distribution-Aware Sampling and Data Diagnosis. The graph is used as a control surface for the late-stage data curriculum. During challenge-focused continual training, we up-weight rare or difficult semantic and action nodes, especially those related to manipulation, physical contact, and long-tail human activities, while down-weighting over-represented generic video nodes. Beyond static graph statistics, we further close the loop with training feedback from the earlier pre-training stages. After the first two stages, we aggregate the denoising loss by semantic nodes and, for video samples, by action nodes, obtaining a node-aware estimate of data difficulty that informs the sampling weights in subsequent stages. Because every node maintains sample counts and representative examples, the graph also exposes coverage holes: missing actions and under-represented object categories. We use these signals to guide targeted data acquisition and to rebalance the training mixture, improving long-tail diversity without blindly increasing corpus size.

3.3 Dense Structured Captioning

Following FIBO [33], which shows that training on long structured captions substantially improves prompt adherence and controllability, we annotate all training data with dense structured JSON captions. Our captions cover four types of data—images, videos, VLA videos, and egocentric videos. All four types share the same schema: video, VLA, and egocentric captions simply add a few extra fields on top of the image schema.

- **Image captions.** Each image caption has four parts: (1) a comprehensive description of the whole image; (2) a set of camera tags (color tone, shot size, shot angle, lens type, composition, light quality, and light source), each chosen from a fixed set of values; (3) an optional world-knowledge list for named entities that appear in the image; and (4) a list of prominent elements. For every element, the caption describes its location, relative size, shape and color, texture, relationship to other elements, and orientation. If the element is a person, the caption also describes pose, gender, skin tone, expression, and clothing; if it is a group of repeated objects, the caption marks it as a cluster and gives a rough count.
- **Video captions.** Video captions add temporal information on top of the image schema. The global description is written in two parts: what happens in the scene, and how the camera moves. In addition, every prominent element carries a list of timestamped actions, e.g., a hand enters the frame during $[2.67s, 3.67s]$ to place a food item. The caption therefore tells the model not only what is in the video, but also who does what, and when.
- **VLA captions.** Robot-manipulation videos use the same video schema. Robotic arms and grippers are described as prominent elements like any other object, and their timestamped actions split each episode into clear phases, e.g., moving down to grasp, opening the gripper to release, and retracting. Gender and skin-tone fields are dropped for this data.
- **Egocentric captions.** First-person videos also use the video schema. The camera-movement field describes the wearer's head and body motion, and the wearer's hands are annotated as prominent elements, with timestamped actions describing how they interact with objects. Gender and skin-tone fields are likewise dropped.

One example caption for each data type is provided in Sec. A.

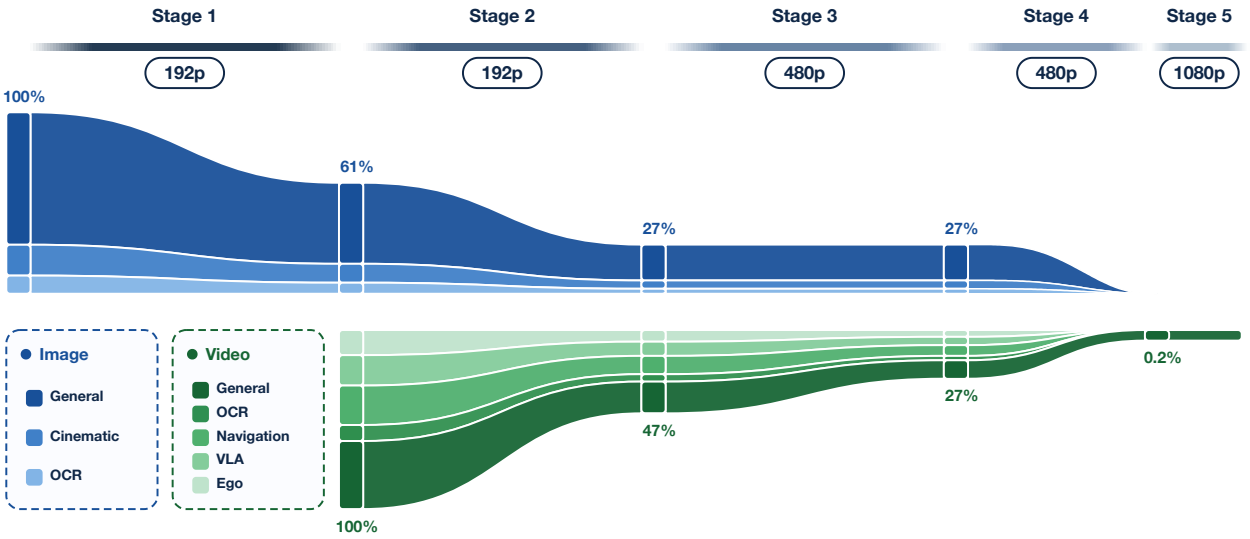


Figure 11. Data curriculum across the five progressive pre-training stages. The image stream (blue) and video stream (green) are decomposed into their constituent sources, with band width indicating each source's relative proportion; percentages denote the fraction of data retained at each stage relative to the modality's initial pool.

3.4 Caption Rewriter

The generator is trained conditionally on the dense structured captions described in Sec. 3.3—which are lengthy, attribute-rich, JSON-formatted descriptions. During inference, however, users typically provide brief, free-form prompts. To bridge this train–inference distribution gap, we introduce a *Caption Rewriter* that decomposes prompt expansion and formatting into a two-stage pipeline, avoiding the unreliability of a single-step mapping model.

In the first stage, **Expand**, a zero-shot large language model (e.g., Qwen3.6-27B [72]) processes the user prompt into a compact, action-centric natural-language description (under a thousand characters). This stage intentionally generates a concise, non-enumerative description that omits overly specific fine attributes—such as texture, exact colors, relative sizes, and precise camera parameters—which are highly prone to hallucination. The second stage, **Map**, utilizes a Qwen3.6-27B fine-tuned with LoRA [39] to transform this intermediate prose into the final structured JSON caption. Although the Map stage populates the complete schema, it operates under a strictly *bounded completion* constraint: it must preserve all explicit information from the prose verbatim (including scene layout, subjects, timestamped actions, and named entities) and is only permitted to plausibly impute low-risk missing fields. This strategy confines potential hallucinations and allows the two-stage rewriter to produce significantly more reliable and better-structured captions than a single model tasked with expanding and structuring a prompt simultaneously.

Both stages seamlessly support text-to-video (T2V) and text-image-to-video (TI2V) generation. For TI2V, a conditioning first frame is provided as the $t{=}0$ ground truth, ensuring the visual appearance adheres strictly to the image while motion dynamics follow the text. Additionally, a target duration is appended to every prompt to constrain all predicted action timestamps to fall within the video's duration.

3.5 Data Curriculum

Building upon the curated corpus, we organize the training data into a five-stage curriculum aligned with the progressive pre-training schedule (Sec. 5); here we focus on how the data itself evolves across these stages. The modality mix and per-source composition shift systematically—from low-resolution (192p) image-only data, through joint image–video data with embodiment-rich footage at increasing resolution, to a small high-quality video refinement set. The cascaded refiner (Sec. 2) is trained separately on a high-quality 1080p subset of this corpus. Fig. 11 summarizes how the modality mix and per-source proportions evolve across the five stages.

Stage 1: Curated low-resolution image pool. The initial stage trains exclusively on 192p images, retaining the samples that pass aesthetic-quality and minimum-resolution filters while rigorously discarding the lowest-tier content.

Stage 2: Video introduction and source expansion. Stage 2 introduces video content alongside images at a matching 192p resolution: a large corpus of video clips joins a quality-tightened, smaller pool of images. Video clips are admitted through a combination of resolution, aesthetic, and motion filters; notably, the motion criterion integrates a geometry-grounded tracking signal with a VLM-based motion assessment to robustly filter out near-static clips. Crucially, this stage marks the injection of over 70,000 hours of embodiment-oriented footage into the corpus—robot manipulation (VLA) spanning real-robot, simulated, open-source, and third-person perspectives across both humanoid and quadruped platforms, together with navigation and egocentric video—alongside text-rich video. Concurrently, the image filters are tightened to enforce higher aesthetic standards, a strict minimum resolution, and freedom from color cast, noise, and blur.

Stage 3: Higher-resolution re-filtering. Stage 3 scales both the video and image streams to 480p, tightening the corresponding aesthetic and motion criteria proportionally; the higher-resolution bar substantially narrows the admitted data in both modalities. We purposefully retain high-motion video to strengthen coverage of dynamic content, while holding the image stream to the rigorous quality gates established in Stage 2.

Stage 4: Challenge-focused rebalancing. At the same 480p resolution, this stage employs source-aware curation to align the corpus with the demands of embodied intelligence. Abundant general and curated videos are subjected to stringent quality standards covering resolution, aesthetics, clarity, motion, and freedom from artifacts such as watermarks, whereas scarce yet high-value embodiment sources, including manipulation, navigation, egocentric footage, and benchmark datasets, undergo only minimal filtering to maximize their coverage. This deliberate asymmetry trades redundant general footage for the long-tail, action-centric data that is paramount for embodied intelligence, yielding a mixture far more embodiment-heavy than the raw source counts would suggest.

Stage 5: High-quality refinement subset. The final stage is a small, high-quality video refinement set at 1080p, used to train the cascaded refiner (Sec. 2): a curated subset drawn from the general source, with samples held to the strictest aesthetic, resolution, and technical-quality bars, and only a tiny fraction (well under 1% of the initial video pool) retained.

4 Infrastructure

4.1 Pre-Training Infrastructure

Large-scale video pre-training imposes distinct system bottlenecks compared to image-only or language-only workloads [9, 17, 36, 105]. The training pipeline must ingest heterogeneous image and video streams, manage variable sequence lengths dictated by resolution and duration. To achieve high training throughput and memory efficiency, our pre-training infrastructure is co-designed to address six core challenges: token-budgeted data loading, composable multi-dimensional parallelism, activation-memory management, compile-first graph capture, explicit communication prefetching, and non-blocking runtime I/O [54, 61, 66, 76, 88, 128].

4.1.1 Heterogeneous Data Pipeline and Token-Budgeted Packing

Rather than batching by a fixed number of samples, the data pipeline constructs each mini-batch based on a target token budget. The dataset estimates both visual and conditioning token lengths under the sampled configuration, allowing the sampler to dynamically decide whether to append the sample based on the remaining token capacity of the batch. This online length-aware scheduling is critical because both sides of the packed input are highly variable: visual tokens come from images and videos with diverse native resolutions, aspect ratios, and durations, while conditioning tokens vary across multimodal conditions; together, these factors produce orders-of-magnitude differences in token counts.

Packed one-dimensional batch. Following VAE and condition encoding, each sample yields visual tokens x_i and conditioning tokens y_i. Rather than stacking heterogeneous samples into a rectangular tensor, we concatenate all valid segments into a single one-dimensional sequence: $[x_1, y_1, x_2, y_2, \ldots, x_N, y_N]$. The data loader constructs packed sequence metadata, such as cumulative sequence lengths and attention masks, enabling variable-length attention kernels (e.g., FlashAttention [19]) to process the packed batch in a single forward pass while preventing cross-sample attention leakage. This unified representation allows images, videos, and different conditioning modalities to be processed jointly without partitioning dataloaders by modality or resolution. To maximize token utilization and eliminate padding waste,

a smart-fill sampler dynamically searches rank-local candidate pools to fill any residual token budget of each packed batch when the next video clip exceeds the leftover limit [49].

4.1.2 Composable Parallel Training

We organize the distributed training stack into four composable parallel dimensions: data parallelism (DP) [53, 88], fully sharded data parallelism (FSDP) [76, 126], sequence parallelism (SP) [43], and expert parallelism (EP) [41, 50, 75, 86]. The training system represents these dimensions via named multi-dimensional device meshes, allowing each training run to configure and compositionally select the required parallel modes without altering the core training loop. This unified abstraction integrates throughput scaling, model-state sharding, long-sequence context partition, and MoE token routing under a single parallelization plan [54, 61, 66, 85, 115, 128].

Data parallelism (DP). DP serves as the primary outer dimension for throughput scaling. Each DP process group consumes a distinct shard of the token-budgeted data stream, while gradients, loss metrics, and optimizer states are synchronized globally. For hybrid data sharding, the data-parallel mesh is decomposed into replication and sharding dimensions, allowing cross-node replication and node-local parameter sharding to be configured independently [53, 76, 88].

Fully sharded data parallelism (FSDP). We utilize FSDP to shard parameters, gradients, and optimizer states across the sharding dimension, extending to Hybrid Sharded Data Parallelism (HSDP) for multi-node scalability. FSDP serves as the primary model-state scaling mechanism for dense transformer blocks and non-expert modules [76, 126]. FSDP sharding is applied downstream of activation checkpointing and compilation decisions, ensuring that sharding hooks, mixed-precision policies, and high-precision ignored parameters are resolved in a deterministic sequence. This ordering is critical to keeping numerically sensitive parameters (e.g., normalizations, routing scales) replicated in FP32 while compiling the surrounding tensor computations in lower precision [21].

Sequence parallelism (SP). To support long-context video sequences, we integrate Ulysses sequence parallelism (SP) [43]. Before the transformer blocks, the packed sequence is padded to align with the Ulysses group size and sliced along the token dimension. Within the attention block, all-to-all collectives transpose token shards into head shards for multi-head attention computation, and a subsequent all-to-all collective restores the sequence-sharded layout. Finally, the sequence is gathered and unpadded to reconstruct the original one-dimensional packed layout. This makes sequence length a distributed resource across the cluster rather than a single-accelerator memory limit.

Expert parallelism (EP). EP scales sparse MoE layers by distributing the expert pool across ranks and routing each token to the devices that host its selected experts [25, 41, 50, 57, 75, 86]. During the MoE forward pass, routed tokens are dispatched through all-to-all communication, processed by local experts, and then combined back to their original sequence positions, preserving global routing semantics without requiring every GPU to host every expert. In our implementation, the expert dispatch and combine path is accelerated with DeepEP [20].

4.1.3 Activation Checkpointing

Activation checkpointing manages the memory footprint of long packed video sequences by recomputing selected forward activations during the backward pass [14]. The system supports multiple checkpointing granularities: full block-level recomputation, layer-level selective checkpointing, operation-level selective checkpointing, and memory-budget-driven checkpointing [48]. This granularity is configurable, enabling the training recipe to dynamically balance recomputation overhead and memory constraints across different model scales and parallel configurations. Crucially, the checkpointing path is implemented as non-reentrant to prevent lifetime conflicts between activation recomputation and expert communication collectives.

4.1.4 Compile-First Graph Capture

Rather than serving as an optional compatibility feature, graph compilation is a key throughput optimization. The training pipeline applies `torch.compile` with the Inductor backend before FSDP2 sharding, allowing local block computations to be fused before sharding hooks are attached [3]. In our internal training benchmark, combining full-block activation checkpointing with compilation improves Model Flops Utilization (MFU) by approximately $1.9\times$, indicating that compile-first graph capture recovers a substantial fraction of accelerator utilization otherwise lost to unfused block-level execution.

4.1.5 Asynchronous Monitoring and Distributed Checkpoint I/O

Runtime I/O operations are co-designed with the training loop to prevent metric logging and state serialization from stalling the accelerators. The monitoring pathway is fully asynchronous: training ranks enqueue logging events (e.g., scalars, histograms, plots) onto a bounded queue, which are processed and written to disk by a background thread. GPU tensors are transferred to host memory before queueing, and the bounded queue prevents logging backpressure from stalling computation. This non-blocking design allows the system to monitor comprehensive diagnostic metrics—such as step-time breakdowns, MFU, MoE load balance, and routing statistics—without impacting training throughput [64].

4.2 Post-Training Infrastructure

Existing reinforcement-learning training frameworks are largely designed around language or vision-language models. In these settings, trajectories are naturally represented as token sequences, and policy log-likelihoods can usually be obtained from per-token probabilities. Around this assumption, prior work has built mature RLHF/RLVR objectives and systems, including PPO, DPO, and GRPO, as well as infrastructure that decouples large-model training, generation, and reward computation [40, 63, 74, 80, 83, 87, 97, 123]. Video diffusion model post-training violates these token-centric assumptions: its optimization target is tightly coupled with the denoising process or latent trajectory, its intermediate states are high-dimensional video latents rather than lightweight text tokens, and log-probability computation, credit assignment, and reward evaluation differ substantially from language modeling.

Recent work such as Flow-GRPO, DanceGRPO, DiffusionNFT, and AWM has explored reinforcement-learning post-training for diffusion models [58, 116, 118, 127]. Similar RL-based post-training ideas have also started to appear in recent frontier generative-model technical reports [12, 112], suggesting that RL is becoming an increasingly important component of diffusion model post-training.

4.2.1 Diffusion-Native RL System Design

RL post-training for large MoE video generation models places exceptionally heavy demands on infrastructure. A single training sample corresponds to a sequence on the order of 100K tokens; the intermediate states of video RL—latent trajectories and per-step sampling statistics—reach the gigabyte scale, far larger than text tokens; and the sheer parameter count of MoE models puts enormous pressure on both weight synchronization and memory management.

We therefore design a diffusion-native RL infrastructure for video diffusion models. The system organizes conditioning, latent trajectories, rewards, and transition-level training data under a unified data abstraction; supports GRPO-style reverse-process optimization and forward-process objectives; and is compatible with both LoRA-style parameter-efficient finetuning and full-model finetuning [39]. To handle the memory and communication pressure introduced by large intermediate video states, the infrastructure decouples rollout, reward evaluation, and training into separate execution roles. It uses a communication-aware data abstraction for large latent objects and a service-oriented reward layer for heterogeneous reward models, server-side decoding, and request batching. This design keeps the RL stack compatible with existing diffusion pipelines while making large-scale video diffusion model post-training more efficient [64, 103].

4.2.2 RL System Performance

The infrastructure is also heavily optimized for speed: full-parameter weight synchronization of the 30B model completes in 20 seconds per step; gigabyte-scale intermediate states are exchanged between rollout and training within 50 milliseconds across multiple GPU nodes; and the system sustains an end-to-end MFU of 43.9% over the full RL step.

4.3 Serving Infrastructure

`LingBot-Video` is served through a Diffusers-compatible model package and an SGLang Diffusion runtime [103, 129]. The serving stack is designed to satisfy three practical requirements: a portable model package for open deployment, a numerically aligned reference path for tolerance-based regression testing, and an optimized multi-GPU path for long-video generation. The same runtime supports text-to-image (T2I), base text-to-video (T2V), and text-image-to-video (TI2V) generation and can optionally invoke a second-stage refiner after base video generation.

4.3.1 Diffusers-Compatible Model Package

To maximize accessibility and lower the deployment barrier for the open-source community, we organize our release artifact as a Diffusers-compatible model package [103]. Prioritizing compatibility with standard Diffusers APIs ensures that developers and researchers can deploy the model out of the box using generic PyTorch environments, without requiring complex software configurations or custom binary compilation. Under this packaging scheme, the model is organized as a unified root directory where the base DiT and the refiner DiT are stored as independent components. Both stages share the same conditioning and autoencoder weights. A lightweight overlay builder constructs the runtime view expected by Diffusers and SGLang by dynamically exposing the selected DiT component as the active `transformer` via configuration files. This design allows the base and refiner serving to reuse the package without duplicating shared weights or copying weight tensors, keeping the release artifact compact and clean.

4.3.2 SGLang-Native Serving Backend

To optimize serving efficiency and maximize inference throughput, we build our deployment runtime on top of SGLang [129], a widely adopted acceleration engine renowned for its performance in serving diffusion-based pipelines. Leveraging SGLang allows our deployment stack to tap into highly optimized CUDA kernels and distributed scheduling policies. Within this framework, the serving runtime provides three distinct execution paths: a direct Diffusers path, a generic SGLang Diffusion backend, and a specialized `LingBot-Video`-native adapter. The direct Diffusers path serves as a readable, eager-mode reference and debugging baseline for checking scheduler modifications and model behavior under a controlled numerical setting. The `LingBot-Video`-native adapter keeps the Diffusers-style denoising loop closely aligned with this reference path while registering our custom pipeline directly into SGLang's serving surface and distributed execution hooks. This architectural separation allows the core denoising logic to remain stable and easy to audit, while enabling SGLang's optimized compute kernels and distributed execution policies to be applied transparently behind the same model interface.

4.3.3 Distributed Video Serving

To accommodate diverse production requirements, our SGLang deployment framework provides two recommended inference configurations: sharding long-video token sequences across multiple GPUs via context parallelism and utilizing batched classifier-free guidance (CFG) to evaluate conditional and unconditional branches efficiently. Depending on the deployment objective, users can select between the following two specialized execution modes:

- **Fidelity-First Version:** Designed for regression testing and numerical-consistency checks, this baseline configuration follows the master training codebase as closely as possible in scheduler logic, routed-expert execution, and precision policy. It preserves standard grouped matrix multiplication (grouped GEMM) for routed experts, employs vectorized token padding and restoration, and maintains a hybrid precision layout. Specifically, it executes standard transformer layers and text encoders in BF16, while keeping numerically sensitive parameters—such as normalization layers, routing gates, and scale-shift modulation tables—in FP32. In practice, this mode serves as a conservative reference gate by keeping implementation-induced deviations within expected numerical tolerance, rather than prioritizing maximum throughput.
- **Speed-First Version:** Tailored for high-throughput serving, rapid visual screening, and low-latency interactive applications, this profile optimizes compute-heavy bottlenecks while maintaining the same scheduler and model interfaces. It replaces the standard routed expert execution path with highly optimized FP8 SGLang Triton kernels, significantly reducing memory footprint and kernel execution times. Combined with sequence sharding and parallelized guidance evaluation, this accelerated version provides substantial throughput improvements with limited observed impact on video generation quality.

5 Training

5.1 Progressive Pre-Training

Large-scale video pre-training is highly challenging to optimize when high-resolution videos, multiple conditioning modalities, and heterogeneous data sources are introduced simultaneously from the beginning of training [9, 36, 105]. For a sparse Mixture-of-Experts (MoE) diffusion model, this challenge is compounded by the need for the router to

establish specialized expert pathways across different tasks and modalities [18, 25, 86]. Exposing the model to full-scale training complexity from the outset may therefore increase the risk of optimization instability, routing collapse, and suboptimal sample quality [25, 69].

To address these challenges, we design a progressive pre-training curriculum that introduces learning objectives in an optimization-friendly order [9, 36]. The curriculum first builds stable frame-level visual priors, then adds temporal modeling, expands task conditioning, harmonizes the data distribution, and finally refines high-resolution details. This staged design separates static appearance learning from dynamic evolution and allows the sparse router to specialize gradually rather than being exposed to all sources of heterogeneity at once [31, 89]. We organize the curriculum into five stages, each adding one source of training complexity to the previous stage.

Stage 1: Image-Only Prior Acquisition. The first stage establishes fundamental text-visual semantic alignment and robust single-frame visual priors [13, 24]. By formulating images as single-frame ($T = 1$) video sequences within our unified framework, we enable the model to learn object shapes, textures, and scene aesthetics without the added complexity of temporal dynamics [31, 89]. Crucially, this image-first warmup provides a stable optimization environment for the sparse router, allowing it to initialize effectively before encountering heterogeneous video tasks [18, 25].

Stage 2: Low-Resolution Temporal Learning. Stage 2 introduces video data into the image-video training mixture while deliberately maintaining a simplified task formulation. Image samples continue to reinforce frame-level visual priors, while video samples are trained with text-to-video (T2V) generation so that the model can focus on temporal dynamics, camera motion, and frame-to-frame consistency before addressing multi-condition formatting [7, 37]. This phase adapts static visual priors into coherent temporal representations without abruptly escalating the optimization difficulty [31, 89].

Stage 3: Multi-Task Conditioning. Stage 3 keeps image samples in the training mixture and broadens the video-side task definition from pure T2V to a joint mixture of T2V and text-image-to-video (TI2V) generation [9, 106]. The primary objective is to teach the model to effectively leverage visual conditions: in the TI2V task, the model must faithfully preserve the provided initial frame while predicting temporally coherent future frames [23, 106]. By introducing this requirement only after foundational temporal learning has stabilized, we avoid conflating early motion acquisition with the more challenging constraint of visual condition preservation.

Stage 4: Weighted Distribution Harmonization. Stage 4 focuses on optimizing the data distribution. Large-scale web data exhibits severe quality variance and source imbalance, which can degrade training stability and expert specialization in late-stage pre-training. We therefore switch to a weighted sampler, which samples from high-value and high-quality data sources according to predefined sampling weights via alias tables. This stage lets the model stabilize under a cleaner and more balanced mixture before entering the final refinement phase.

Stage 5: High-Resolution Refinement. The final stage functions as a high-resolution refiner, trained on video data. Its primary purpose is to dedicate training capacity to the generation of high-frequency spatial details and the correction of local artifacts in high-resolution samples, all while maintaining temporal sharpness in videos (e.g., minimizing texture flickering) [36, 79, 124]. Following the cascaded refiner configuration described in Sec. 2, this stage optimizes a conditional refinement trajectory. It maps upsampled low-resolution base-stage outputs to clean high-resolution targets, ensuring the model retains the base-stage semantics and motion dynamics while successfully restoring fine-grained spatial details [9, 124].

5.2 Post-Training

5.2.1 Reinforcement Learning with Multi-Aspect Rewards

Reward Modeling. Existing reinforcement learning (RL) methods in post-training for visual generation typically rely on holistic reward models that output a single scalar score to represent overall human preference or general text-alignment [47, 93, 114]. However, video generation is inherently multi-dimensional and prone to complex failure modes, such as static mode collapse, temporal hallucinations, and physical implausibility, which global, coarse-grained rewards fail to penalize effectively. To address this and provide fine-grained optimization signals, we decouple the evaluation into a comprehensive suite of six specialized reward models to optimize visual quality and physical dynamics during the training process: vision quality, text-video alignment, dynamic degree, motion coherence, human motion consistency, and physical plausibility.

- **Vision Quality.** Following LongCat-Video [12], we employ HPSv3 [62] as the foundation for VQ evaluation to jointly assess visual quality and text-video alignment. We leverage different statistics over all frames in the video to design complementary reward signals that separately capture overall visual fidelity and robust caption alignment. The resulting composite reward penalizes blurriness, artifacts, and low-resolution outputs while remaining sensitive to caption-level alignment.
- **Text-Video Alignment.** We propose a fine-grained, action-centric text-video alignment reward based on temporal Visual Question Answering (VQA) to evaluate text-video alignment. We first parse the training caption into structured entities, associating each with specific actions and precise temporal windows. To ensure evaluation accuracy while reducing computational overhead, we introduce an adaptive temporal slicing strategy with a bifurcated windowing mechanism: dynamic actions receive padded windows, whereas static-temporal criteria enforce strict boundaries. We then employ Qwen3.6-27B [72] as a zero-shot evaluator to answer batched verification queries on the sliced frames. The text-video alignment reward is computed as a weighted satisfaction rate normalized to $[0, 1]$, where weights are dynamically assigned based on action complexity. Unlike traditional global semantic matching metrics, this fine-grained approach ensures that accurately generating multi-stage actions yields a significantly higher reward, effectively penalizing temporal hallucinations and missing actions.
- **Dynamic Degree.** We introduce a Dynamic Degree reward evaluated by Qwen3.6-27B [72] to measure and encourage appropriate motion intensity. Specifically, the generated video is spatially downsampled to a maximum height of 480p for efficiency and evaluated using a specialized motion-assessment prompt. The VLM outputs a structured information containing a discrete motion score from 1 to 5, which we linearly map to a continuous reward scalar in $[0, 1]$. By relying on a VLM rather than traditional optical flow, this reward captures semantically meaningful subject dynamics. This counteracts the tendency of text-to-video models to generate static, image-like sequences, effectively breaking the static mode collapse without compromising temporal consistency.
- **Motion Coherence.** Text-to-video models generate videos at a fixed playback frame rate (e.g., 24 fps), yet the generated motion often suffers from a slow-motion effect: it looks slower than it would in the real world, as if captured by a high-speed camera but played back at 24 fps. Following Pulse-of-Motion [27], we estimate from the generated frames alone how fast the motion actually unfolds. The reward mechanism is designed to guide the model toward generating motion that appears natural at a standard 24 fps playback rate. It fully rewards content exhibiting realistic physical speed while penalizing outputs that display an artificial slow-motion effect, thereby steering the model to produce videos with naturally paced dynamics.
- **Human-Motion Consistency.** We train a generative human-motion consistency model using a comprehensive distillation approach to evaluate human motion. We build a high-quality dataset of real and synthesized videos to generate five-dimensional artifact scores, specifically targeting impossible topology, facial distortion, hand deformity, limb count errors, and semi-transparent bodies, along with step-by-step reasoning to assess both spatial correctness and temporal fidelity. We subsequently fine-tune a vision-language mixture-of-experts model [72] on this curated corpus. By enforcing supervision over the explicit reasoning traces, the model transcends superficial scalar regression, thereby internalizing the underlying structural and semantic constraints. During the reinforcement learning stage, it outputs structured rationales and discrete scores, which are mapped to a continuous reward scalar in $[0, 1]$. This mechanism explicitly addresses the spatial distortions and temporal-semantic misalignments, guiding the policy toward physically plausible and text-aligned human motion.
- **Physical Plausibility.** We introduce a Physical Plausibility reward to assess whether generated video trajectories unfold within a coherent physical scene, where task-relevant entities remain present, spatially grounded, and consistent with the intended task evolution. Rather than measuring only perceptual quality, this evaluator is caption-conditioned and frame-evidence-based: it establishes the expected actors, target objects, and final-state cues from the caption, but assigns scores solely based on visible phenomena in the sampled frames. It first assesses foundational physical constraints along three complementary axes: motion causality, which verifies whether objects remain stationary or continue moving unless affected by a plausible external force; object permanence and non-penetration, which checks for boundary loss, penetration, impossible overlap, and unsupported appearance or disappearance; and material-kinematic realism, which examines whether entities follow plausible material behavior and rigid-body motion. Building upon these physical foundations, the evaluator also incorporates an assessment of task completion. It verifies that the intended physical state changes actually materialize by checking for correct action occurrence, accurate object manipulation, and the successful achievement of the specified final spatial state. The resulting scores are aggregated into a unified physical reward. By jointly evaluating adherence

to fundamental physical laws and the realization of task-driven state changes, this design ensures that generated videos maintain a coherent, physics-based world where dynamic manipulations and state changes are both visually grounded and physically plausible.

Reward Aggregation. The six reward signals are aggregated into a single advantage through decoupled per-reward normalization, described in Sec. 5.2.2.

5.2.2 On-Policy GRPO Training

We post-train `LingBot-Video` with Group Relative Policy Optimization (GRPO) [58, 83, 118] to maximize the multi-aspect rewards. Our setup follows the single-step exploration paradigm of Flash-GRPO [34]: each rollout is stochastic at exactly one denoising step shared within a group, the noise is injected via Coefficients-Preserving Sampling (CPS) [107], the policy gradient is reweighted to balance timesteps, and training is strictly on-policy without KL regularization. We detail each component below.

Group-Shared Single-Step Exploration. GRPO estimates advantages by comparing a group of G rollouts of the same prompt, which requires stochastic sampling. In Flow-GRPO and DanceGRPO [58, 118], every denoising step is stochastic and all steps share the same trajectory-level advantage, which causes a severe credit-assignment problem [12, 35, 51]. Following Flash-GRPO [34], we instead make exactly one denoising step stochastic. For each group, a step index k is drawn uniformly from the first half of the inference schedule and shared by all G rollouts. The transition $t_k \to t_{k+1}$ is sampled stochastically, while all other steps remain deterministic ODE steps. Each trajectory then contains exactly one stochastic transition, advantages within a group are compared at the same timestep, and the policy update is applied exactly to the transition that produced the variation, giving precise credit assignment.

Coefficients-Preserving Sampling. For the stochastic step we adopt CPS [107], a DDIM-style transition [90], instead of the common SDE conversion [58]. The Euler–Maruyama step of the converted SDE injects more noise than the schedule prescribes [107], leaving visible artifacts that corrupt reward scores, and its diffusion coefficient explodes at high noise levels, requiring an extra clipping patch [12]; CPS avoids both by construction. Let $1 = t_0 > t_1 > \cdots > t_N = 0$ be the inference schedule. At step t_i, the velocity prediction $\hat{v}_\theta$ gives the clean and noise estimates:

$$\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_{t_i} - t_i \hat{v}_\theta, \qquad \hat{\epsilon} = \boldsymbol{x}_{t_i} + (1 - t_i)\hat{v}_\theta. \tag{13}$$

The CPS transition to t_{i+1} is:

$$\boldsymbol{x}_{t_{i+1}} = \mu_\theta + s_i \epsilon, \qquad \mu_\theta = (1 - t_{i+1})\hat{\boldsymbol{x}}_0 + \sqrt{t_{i+1}^2 - s_i^2}\,\hat{\epsilon}, \tag{14}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is fresh Gaussian noise, $s_i = t_{i+1}\sin(\eta\pi/2)$ is the injected noise scale, and $\eta \in [0, 1]$ controls the exploration strength ($\eta = 0.7$ in our experiments). Since $(t_{i+1}^2 - s_i^2) + s_i^2 = t_{i+1}^2$, the total noise after the transition matches the schedule coefficient exactly: samples stay on the marginal path of the deterministic sampler, and rewards are always evaluated on clean videos. Following [107], the transition log-likelihood takes the simplified form $\log \pi_\theta(\boldsymbol{x}_{t_{i+1}} \mid \boldsymbol{x}_{t_i}) \propto -\|\boldsymbol{x}_{t_{i+1}} - \mu_\theta\|^2$.

Timestep-Balanced Gradient Reweighting. The strength of the policy gradient at step k scales with the transition gain:

$$\kappa_k = 2 s_k \left| \frac{\partial \mu_\theta}{\partial \hat{v}_\theta} \right|, \qquad \frac{\partial \mu_\theta}{\partial \hat{v}_\theta} = (1 - t_k)\, t_{k+1} \cos\!\left(\frac{\eta\pi}{2}\right) - t_k(1 - t_{k+1}), \tag{15}$$

which varies strongly across the schedule, so a uniformly sampled critical step would let a few timesteps dominate training [12, 34, 35]. We reweight each transition by the inverse gain, normalized to unit mean over the schedule:

$$\lambda_k = \frac{\kappa_k^{-1}}{\frac{1}{N}\sum_{j=0}^{N-1} \kappa_j^{-1}}. \tag{16}$$

Here, N is the number of inference steps, and the normalizer is computed analytically over the schedule grid rather than from batch statistics, since all samples in an update batch share one timestep. This equalizes update magnitudes across timesteps without changing the effective learning rate.

Multi-Reward Advantage Normalization. The rewards of Sec. 5.2.1 have different scales and variances, so we normalize each reward independently before fusing them with weights [59]:

$$\hat{A}^{(i)} = \sum_r w_r \frac{R_r\big(\boldsymbol{x}_0^{(i)}, c\big) - \mu_r}{\sigma_r + \delta}, \tag{17}$$

where $R_r(\boldsymbol{x}_0^{(i)}, c)$ is the r-th reward of sample i under prompt c, w_r is its weight, μ_r and σ_r are the mean and standard deviation of R_r within the group, and δ is a small constant.

Objective. The final loss is the reweighted policy gradient on the sampled critical transitions:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\mathbb{E}\left[\lambda_k \, \hat{A}^{(i)} \log \pi_\theta\big(\boldsymbol{x}_{t_{k+1}}^{(i)} \mid \boldsymbol{x}_{t_k}^{(i)}\big)\right]. \tag{18}$$

Each rollout batch is consumed by a single gradient update, so training is strictly on-policy and importance ratios stay at one. Following [34, 117, 118], we use no KL penalty and no reference model, and we fine-tune all model parameters.

5.2.3 Negative-Aware Finetuning with Real-World Videos

Most existing RL post-training methods for diffusion models rely on reward models to provide optimization signals [58, 118], which introduces the risk of reward hacking in the video domain. To mitigate this, we leverage real-world videos as direct preference signals: a real video clip serves as the positive (chosen) sample, while the model-generated video under the same prompt serves as the negative (rejected) sample. This data configuration shares a similar motivation with RealDPO [15], which also pairs real videos with generated ones. For the optimization we adopt the forward-process optimization framework of DiffusionNFT [127], bypassing the need to backpropagate through denoising trajectories.

Preference Pair Construction. At each training step, we sample a batch of real video clips from a curated dataset and encode them through the pretrained VAE into clean latents $\boldsymbol{x}_0^w$ (chosen). The active policy generates N videos per prompt via the inference pipeline, and their VAE latents form the rejected pool $\{\boldsymbol{x}_{0,i}^l\}_{i=1}^N$. Each rejected latent $\boldsymbol{x}_0^l$ is paired with the corresponding chosen latent $\boldsymbol{x}_0^w$ to form a preference pair. Both latents are perturbed to a shared random timestep $t \sim \mathcal{U}(0,1)$ with a shared noise vector $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$\boldsymbol{x}_t^w = (1-t)\boldsymbol{x}_0^w + t\boldsymbol{\varepsilon}, \qquad \boldsymbol{x}_t^l = (1-t)\boldsymbol{x}_0^l + t\boldsymbol{\varepsilon}. \tag{19}$$

The clean latent is recovered from the noised state via $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_t - t \cdot \hat{v}$, where $\hat{v}$ denotes the predicted velocity. Operating on single-step noised states avoids trajectory-level log-probability computation, keeping each update computationally lightweight.

Negative-Aware Optimization. We maintain two prediction pathways sharing the same base model: (i) the *active policy* θ, which receives gradients; and (ii) the *old policy*, an exponential moving average (EMA) copy of θ that stabilizes the optimization by preventing the active policy from changing too rapidly. For any noised state $\boldsymbol{x}_t^s$ ($s \in \{w, l\}$), the two pathways produce velocity predictions $\hat{v}_\theta^s$ and $\hat{v}_{\text{old}}^s$.

Following DiffusionNFT [127], we construct an implicit positive policy that blends the active and old velocity predictions to imitate the target, and an implicit negative policy that extrapolates past the old policy to suppress it:

$$\hat{v}_{\text{pos}} = \beta\hat{v}_\theta + (1-\beta)\hat{v}_{\text{old}}, \qquad \hat{v}_{\text{neg}} = (1+\beta)\hat{v}_{\text{old}} - \beta\hat{v}_\theta, \tag{20}$$

where $\beta \in (0, 1]$ controls how strongly the active policy is allowed to deviate from the old policy. DiffusionNFT defines the per-sample loss as a reward-weighted mixture of positive and negative branches:

$$\mathcal{L}_{\text{NFT}}(r) = r\big\|\hat{v}_{\text{pos}} - v\big\|^2 + (1-r)\big\|\hat{v}_{\text{neg}} - v\big\|^2, \tag{21}$$

where $v = \boldsymbol{\varepsilon} - \boldsymbol{x}_0$ is the ground-truth flow matching velocity and $r \in [0, 1]$ is a normalized reward.

Pairwise Preference Adaptation. We adapt this formulation to our pairwise setting by treating the real video as the optimal sample ($r = 1$) and the generated video as the negative sample. This yields:

$$\mathcal{L}_{\text{chosen}} = \mathcal{L}_{\text{NFT}}(1)^w = \big\|\hat{v}_{\text{pos}}^w - v^w\big\|^2, \qquad \mathcal{L}_{\text{reject}} = \mathcal{L}_{\text{NFT}}(r)^l = r\big\|\hat{v}_{\text{pos}}^l - v^l\big\|^2 + (1-r)\big\|\hat{v}_{\text{neg}}^l - v^l\big\|^2. \tag{22}$$

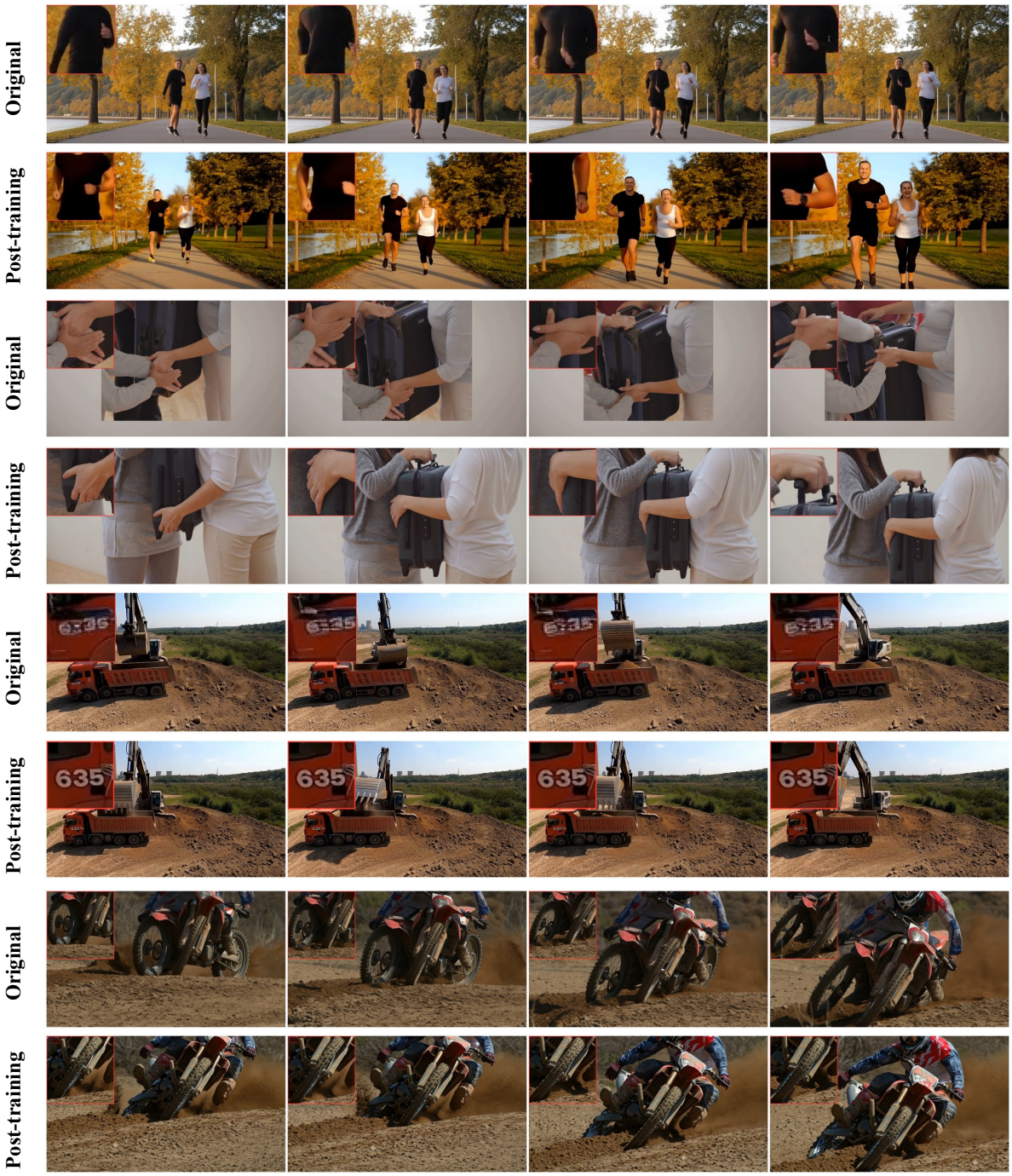


Figure 12. Qualitative comparison on general video quality before and after post-training, demonstrating marked improvements in several fundamental video generation domains. Post-training effectively resolves critical artifacts including inconsistent hand and limb synthesis, blurred or incorrect text rendering, and structural object deformation.

In principle, r for the rejected sample could be assigned by a reward model according to generation quality. For simplicity and to avoid introducing reward model overhead, we set $r = 0$. We also regularize the active policy against a frozen copy of the base model to prevent the policy from drifting too far during fine-tuning:

$$\mathcal{L}_{\text{KL}} = \tfrac{1}{2}\big(\|\hat{v}_\theta^w - \hat{v}_{\text{ref}}^w\|^2 + \|\hat{v}_\theta^l - \hat{v}_{\text{ref}}^l\|^2\big). \tag{23}$$

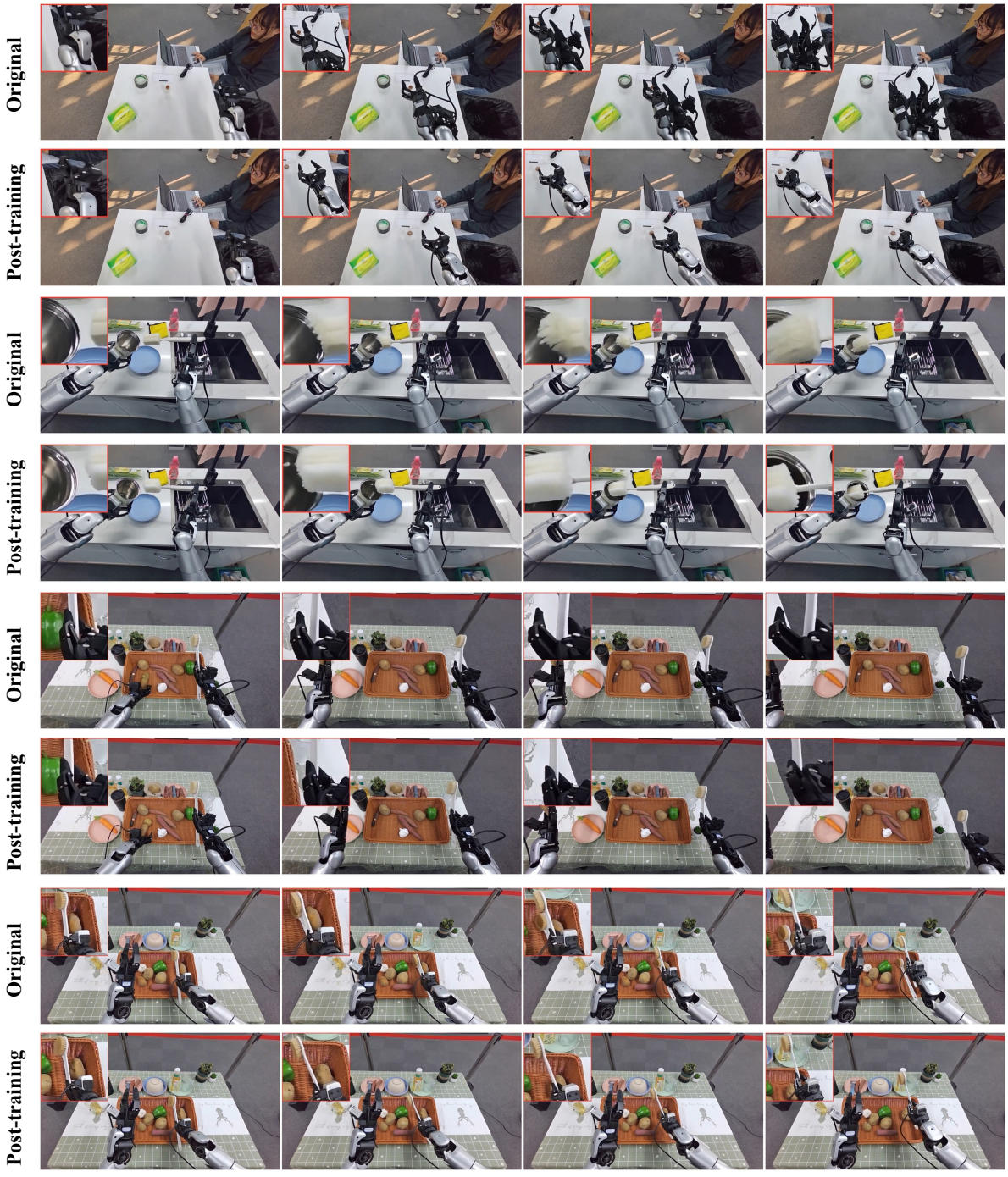


Figure 13. Qualitative comparison on embodied scenarios before and after post-training. The post-training phase significantly enhances physical plausibility by resolving baseline artifacts such as structural distortion of the arm and grasped objects, non-physical penetration, premature object release, and object duplication.

The total training objective is:

$$\mathcal{L}_{\text{RealNFT}} = \mathcal{L}_{\text{chosen}} + \mathcal{L}_{\text{reject}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}. \tag{24}$$

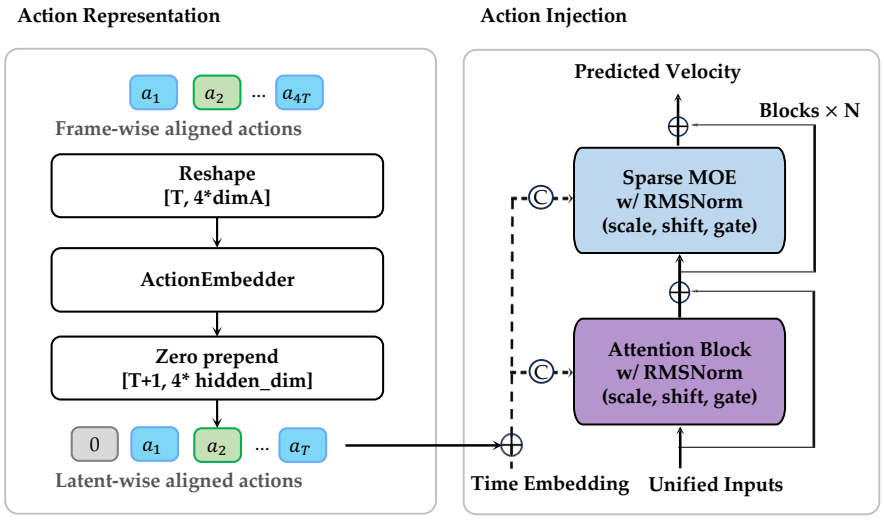


Figure 14. Architecture of `LingBot-Video-A2V`. Given frame-wise actions for the future $4T$ frames, `LingBot-Video-A2V` first converts the raw commands into relative actions, flattens the action sequence, and maps it with a learnable `ActionEmbedder` to action latents. A zero action is prepended for the initial state to temporally align action sequence with the $T + 1$ visual latents. The action embeddings are injected into the pre-trained transformer blocks with time embeddings.

5.3 Action-to-Video Post-Training

Embodied world simulation requires a model to roll out plausible future visual trajectories from the current state and a planned action sequence [95, 130]. Beyond visual generation, such simulated futures are central to embodied AI and robot planning, supporting policy evaluation [95, 100], test-time planning through imagined rollouts [104], robot-data scaling via synthetic trajectories [29], and reinforcement-learning environments for interactive policy improvement [28, 44]. To examine whether `LingBot-Video` can transfer its pre-trained understanding of physical rationality, spatial relations, and temporal evolution to this downstream setting, we further post-train it as an action-conditioned world model, denoted `LingBot-Video-A2V`. Conditioned only on the initial world state, represented by the first image and a textual state description, and on the robot action parameters, `LingBot-Video-A2V` generates action-conditioned visual rollouts of future world states. This setting poses a challenging test of physical-law modeling and action following. We organize the post-training design around three components:

- *Data recaptioning.* Pre-trained LingBot-Video excels at prompt following, where detailed prompts describe the evolution of the whole video, including physical dynamics and action descriptions. To adapt this simulator into a predictor, we rewrite the data captions so that each prompt describes only the initial state. This encourages the model to drive video evolution using only the given robot actions. We further apply a strict future-leakage check to ensure that the prompts do not reveal future observations or dynamics.
- *Action representation and injection.* Given an initial frame and future robot action chunks, we first convert the raw actions into relative actions, so that each step encodes the incremental change from the preceding state. `LingBot-Video-A2V` then flattens the full action chunk into a single sequence and projects it through the `ActionEmbedder`. As shown in Fig. 14, we prepend a zero action for the initial observation and keep the action latents temporally aligned with the visual latents. The encoded latent-wise aligned actions are injected as residual signals to modulate each transformer block, together with the time embedding. To stabilize post-training, we zero-initialize the last layer of the `ActionEmbedder`, allowing the newly introduced action branch to be integrated gradually into the pre-trained backbone.
- *Training setup.* We adopt the Fourier GR-1 post-training datasets for experiments [26]. Each training sample follows the same formulation: the model observes the initial world state, receives a sequence of robot actions, and is supervised to generate the corresponding future visual rollout. Starting from the unified pre-trained LingBot-Video, we optimize the `ActionEmbedder` together with the full transformer backbone, rather than training the action branch in isolation, to adapt the original video simulator to predict future frames conditioned on actions. We run post-training for $8k$ steps with a global batch size of 64 and a learning rate of $1e^{-5}$.

This adaptation benefits from the synergy between strong pre-trained world priors and targeted action-rollout data. LingBot-Video provides action-aware priors over physical-world causality, spatial relations, and temporal evolution, while our recaptioned, leakage-filtered GR-1 trajectories provide clean supervision for action-conditioned rollouts. As a result, `LingBot-Video-A2V` achieves high-quality action following after post-training and can be readily adapted to embodied world simulation. We include additional experimental results in Sec. 6.4.

5.4 Distillation

To improve inference efficiency, we distill `LingBot-Video` into a few-step generator following the improved Distribution Matching Distillation (DMD2) framework [120]. Let G_θ denote the student generator conditioned on the unified condition c, and let $x_0 = G_\theta(z, c)$, where $z \sim \mathcal{N}(0, I)$. For a sampled timestep t, we perturb the generated latent video as $x_t = \alpha_t x_0 + \sigma_t \epsilon$. The method matches the student distribution to the teacher distribution by minimizing a reverse-KL-style distribution-matching objective over diffusion noise levels:

$$\nabla_\theta \mathcal{L}_{\mathrm{DMD}} = \mathbb{E}_{z,t,\epsilon}\left[-w_t \left(s_{\mathrm{real}}(x_t, t, c) - s_{\mathrm{fake}}(x_t, t, c)\right) \frac{\partial G_\theta(z, c)}{\partial \theta}\right], \tag{25}$$

where s_{real} is provided by the teacher video diffusion model, and s_{fake} is estimated by an auxiliary score model trained online on samples from the current student. We also retain a lightweight GAN objective to provide real-data supervision and improve visual quality.

6 Evaluation

6.1 Internal Benchmark

To verify the capability of `LingBot-Video` as a physical world model, we conduct a comprehensive evaluation on our internal benchmark across two distinct dimensions: *General Quality*, which assesses fundamental generative capabilities, and *Embodied Domain*, which probes specialized, high-difficulty scenarios relevant to embodied AI and real-world interactions. To comprehensively evaluate the generation capabilities, the internal benchmark cover two core generation settings: *Text-to-Video (T2V)* and *Text-and-Image-to-Video (TI2V)*.

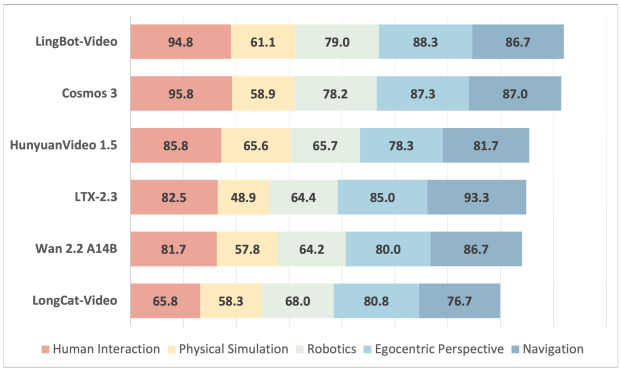
General Quality. The general domain focuses on the foundational video generation quality, ensuring that the generated videos are visually pleasing, temporally consistent, and semantically accurate.

- **Motion Quality** measures the naturalness, continuity, and physical plausibility of movements. It evaluates whether the video exhibits smooth motion trajectories and remains free from severe temporal artifacts such as flickering, structural deformation, or unnatural human actions (e.g., identity drift or clothing deformation).
- **Prompt Following** assesses how faithfully the model adheres to the input text instructions. It measures semantic alignment across multiple entities, complex counting scenarios, sequential action order, specified camera movements, *etc*.
- **Visual Consistency** quantifies the model's ability to maintain identity and scene context over time. This includes preserving the consistency of background layouts, main subject details, and specific instances across frames. For the TI2V setting, it additionally incorporates first-frame image preservation to ensure the generated video faithfully inherits the appearance and style of the input reference image.
- **Aesthetic Quality** evaluates the general visual appeal, artistic value, and stylistic execution of the generated video, emphasizing overall cinematic composition, lighting, and texture.

Embodied Domain. Beyond general quality, `LingBot-Video` is targeted at complex, interactive, and embodied scenarios. For a real-world robot equipped with cameras, the core objective is not to generate a scene from scratch, but to predict future physical interactions conditioned on the current observation (the initial frame I_0) and a control command (T). The embodied domain evaluation specifically probes the model's performance on highly specialized tasks that demand physical understanding, spatial reasoning, and interaction. It covers the following categories:

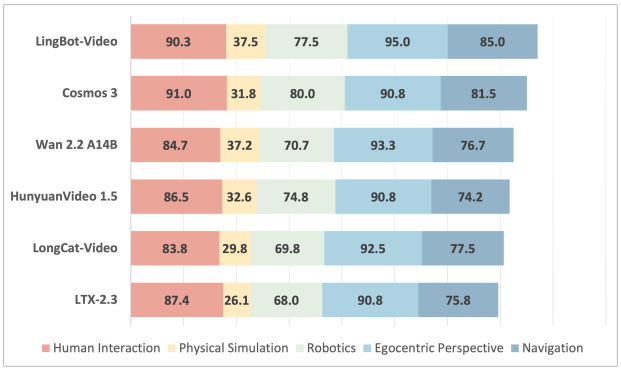
- **Human Interaction** evaluates the model's capacity to synthesize intricate physical interactions and expressive behaviors. This covers fine-grained categories such as human interaction, object manipulation, animal interaction

(a). T2V quality score.



(b). T2V domain score.

(c). TI2V quality score.



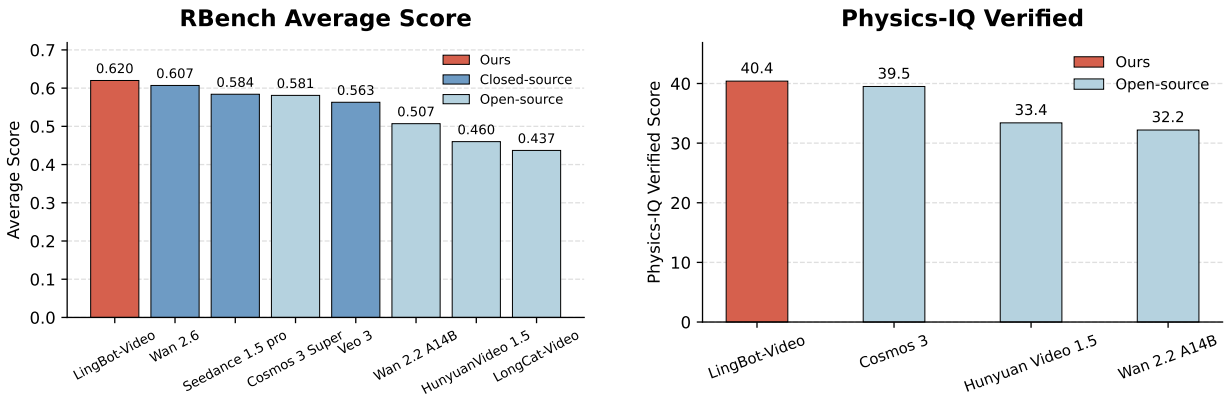
(d). TI2V domain score.

Figure 15. Quantitative evaluation on our internal benchmark. We evaluate the performance of LingBot-Video and other state-of-the-art open-source competitors across two dimensions: *general quality* (for overall visual appeal and coherence) and *embodied domain* (for specific category distributions). Top row shows results under the Text-to-Video (T2V) setting, while the bottom row illustrates results under the Text-and-Image-to-Video (TI2V) setting.

and sports dynamics. It also extends to human-related evaluation, such as detailed hand and finger motions, and subtle facial emotions and expressions.

- **Physical Simulation** focuses on how well the model adheres to intuitive physical laws, acting as a "world simulator" for embodied generation. It covers scenarios governed by classical mechanics, optics (e.g., reflections, shadows), thermodynamics, fluid dynamics, material properties, and magnetism.
- **Robotics** targets embodied agent scenarios, testing the model's capability to generate coherent movements for diverse robotic platforms, including humanoid robots, robotic arms, and quadrupedal robots.
- **Egocentric Perspective** assesses the model's proficiency in rendering first-person perspective videos, which are crucial for ego-agent.
- **Navigation** tests the model's understanding of spatial layouts and motion planning across diverse environments, including outdoor streets, autonomous driving scenes, interactive game environments, and complex indoor layouts.

Results. We compare LingBot-Video with five open-source models, including NVIDIA Cosmos 3 Super-Image-to-Video, Wan 2.2 A14B, LongCat-Video, Hunyuan Video 1.5, and LTX-2.3. As shown in Fig. 15c and Fig. 15d, LingBot-Video achieves state-of-the-art performance among all open-source competitors on the TI2V task, securing the top spot in both general quality and embodied domain scores. This demonstrates our model's superior capability in simulating precise physical trajectories, such as robotic arm manipulation and obstacle avoidance. For the T2V task (Fig. 15a and Fig. 15b), while our model ranks second in general quality, we still consistently outperform competitive baselines such as Cosmos on the embodied domain score. This edge, even in the absence of initial image conditioning, highlights that LingBot-Video possesses robust and intrinsic physical priors crucial for embodied AI applications.



(a). RBench score.

(b). Physics-IQ Verified I2V score.

Figure 16. Public benchmark score comparison. We visualize the average scores from Tab. 1 and the Physics-IQ verified scores, with `LingBot-Video` highlighted against open-source and closed-source baselines.

Models	Type	Avg.	Tasks					Embodiments			
			Manip.	Spatial	Multi-entity	Long-hor.	Reasoning	Single arm	Dual arm	Quadruped	Humanoid
`LingBot-Video`	open-source	**0.620**	**0.578**	0.643	0.444	**0.634**	0.505	0.636	0.639	**0.758**	0.689
Cosmos3 Super	open-source	0.581	0.487	0.642	0.444	0.591	0.395	0.615	0.623	0.739	0.691
LongCat-Video	open-source	0.437	0.372	0.310	0.220	0.384	0.186	0.586	0.576	0.681	0.621
Wan 2.2 A14B	open-source	0.507	0.381	0.454	0.373	0.501	0.330	0.608	0.582	0.690	0.648
HunyuanVideo 1.5	open-source	0.460	0.442	0.316	0.312	0.438	0.364	0.513	0.526	0.634	0.595
Wan 2.6	closed-source	0.607	0.546	**0.656**	0.479	0.514	**0.531**	**0.666**	**0.681**	0.723	0.667
Seedance 1.5 pro	closed-source	0.584	0.577	0.495	**0.484**	0.570	0.470	0.648	0.641	0.680	**0.692**
Veo 3	closed-source	0.563	0.521	0.508	0.430	0.530	0.504	0.634	0.610	0.689	0.637

Table 1. RBench evaluation results. We report the average score and sub-dimension scores across five task-oriented and four embodiment-specific categories for open-source and closed-source models. Scores of some models are sourced from RBench [22].

6.2 Public Benchmark

To complement our internal evaluations, we further evaluate `LingBot-Video` on public automated evaluation benchmarks on RBench [22] and Physics-IQ Verified [73] designed for video generation in embodied and physical domains.

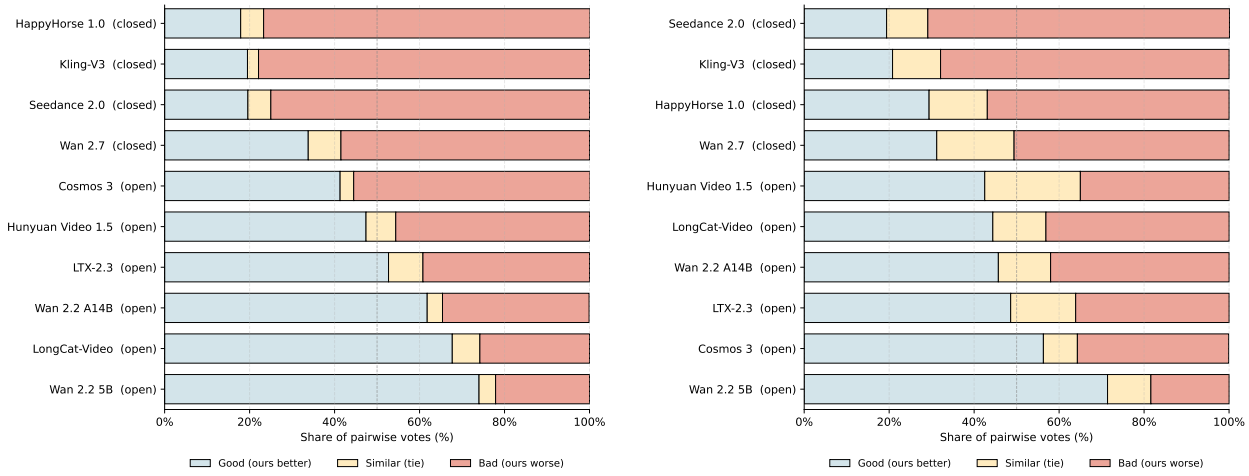
RBench. RBench [22] specifically targets the correctness of robot-centric interactions, making it a focused complement for the robotics category in our evaluation. The benchmark includes 650 text-image prompts partitioned into two primary tracks: 250 task-oriented scenarios covering five interaction types (Manipulation, Spatial Relationship, Multi-entity Collaboration, Long-horizon Planning, and Visual Reasoning) and 400 embodiment-specific scenarios spanning four robot morphologies (Single-arm, Dual-arm, Humanoid, and Quadruped). The compelling performance on RBench further validates `LingBot-Video`'s embodied generation capability, particularly in robotics-centric scenarios that demand physical coherence and precise instruction following. This aligns with the robotics-domain advantage observed in our internal benchmark (Fig. 15d) and demonstrates that `LingBot-Video`'s physical world modeling generalizes beyond our internal evaluation suite.

Physics-IQ Verified. Physics-IQ Verified [73] is a refined version of the Physics-IQ [65] benchmark for evaluating whether video generation models can predict real-world physical phenomena rather than merely producing visually plausible motion. The benchmark is built from 66 controlled physical experiments spanning solid dynamics, fluid dynamics, thermodynamics, optics, and magnetism. Each experiment is recorded from three viewpoints and two takes, yielding 396 real-world videos in total. Two evaluation modes are supported. In image-to-video (I2V), the model is conditioned on a single switch frame plus an optional text prompt and predicts the subsequent motion. In video-to-video (V2V) continuation, the model is conditioned on a 3-second conditioning video plus an optional text prompt and predicts the next 5 seconds of motion. Generated videos are compared against ground-truth physical continuations

along four complementary axes: spatial overlap, temporal alignment, magnitude-weighted spatial agreement, and pixel-level error. We evaluate LingBot-Video under the I2V setting, which matches our image-conditioned generation interface. As shown in Fig. 16(b), LingBot-Video obtains a Physics-IQ Verified score of 40.4, ranking first among the evaluated open-source models and narrowly surpassing Cosmos 3 (39.5). The margin over Hunyuan Video 1.5 (33.4) and Wan 2.2 A14B (32.2) is more pronounced, indicating stronger predictive consistency on real physical processes. Together with the RBench results, this suggests that LingBot-Video's world-modeling capability extends from robot-centric interaction scenarios to broader physical dynamics.

6.3 User Study

We conduct a Good-Same-Bad (GSB) human evaluation to benchmark LingBot-Video against leading open-source and commercial video generation models. The evaluation adopts the same domain as our internal benchmark. For each prompt, human raters are presented with a pair of videos in a randomized and anonymized side-by-side format, and are asked to judge whether the first video is *Good* (better), *Same* (comparable), or *Bad* (worse) than the second. This blind pairwise setup forces a direct quality comparison, mitigating subjective scoring bias. In this study, we compare LingBot-Video against six open-source models (NVIDIA Cosmos 3, Wan 2.2 5B, Wan 2.2 A14B, LongCat-Video, HunyuanVideo 1.5, and LTX-2.3) and four commercial models (Kling-V3, Wan 2.7, Seedance 2.0, and HappyHorse 1.0). In total, we collect human ratings across 400 prompts for each comparison pair. The full GSB breakdown for text-to-video (T2V) and text-and-image-to-video (TI2V) generation is reported in Fig. 17. For T2V generation, LingBot-Video clearly outperforms several open-source baselines, with the *Good* rate exceeding the *Bad* rate against Wan 2.2 5B, LongCat-Video, Wan 2.2 A14B, and LTX-2.3. The comparison is closer against stronger open-source models such as HunyuanVideo 1.5 and Cosmos 3, indicating that LingBot-Video remains a competitive setting. For TI2V generation, the advantage becomes more consistent: LingBot-Video obtains higher *Good* than *Bad* rates against all evaluated open-source baselines, with especially large margins over Wan 2.2 5B and clear gains over Cosmos 3, LTX-2.3, Wan 2.2 A14B, LongCat-Video, and HunyuanVideo 1.5. This stronger TI2V result suggests LingBot-Video as a strong open-source model while trailing the stronger commercial models.



(a). T2V. **(b).** TI2V.

Figure 17. User study results. Good-Same-Bad human evaluation results for T2V and TI2V generation.

6.4 Action-to-Video Post-Training

We include the results of LingBot-Video-A2V on EgoDex Eval and DreamDojo-HV Eval in Fig. 18 to examine whether the post-trained model generalizes beyond the GR-1 trajectories used for training. Both evaluation datasets contain novel objects and actions that are absent from the GR-1 post-training dataset [26], making them suitable for testing out-of-distribution action following rather than memorization of training rollouts. The results shown in Fig. 18 demonstrate that our model adheres better to physical laws and follows actions more accurately.

Figure 18. Compared with DreamDojo [26], our model demonstrates better adherence to physical laws, such as preserving the yellow apple in the first example, and stronger action following, as shown by the hand pose relative to the sandwich.

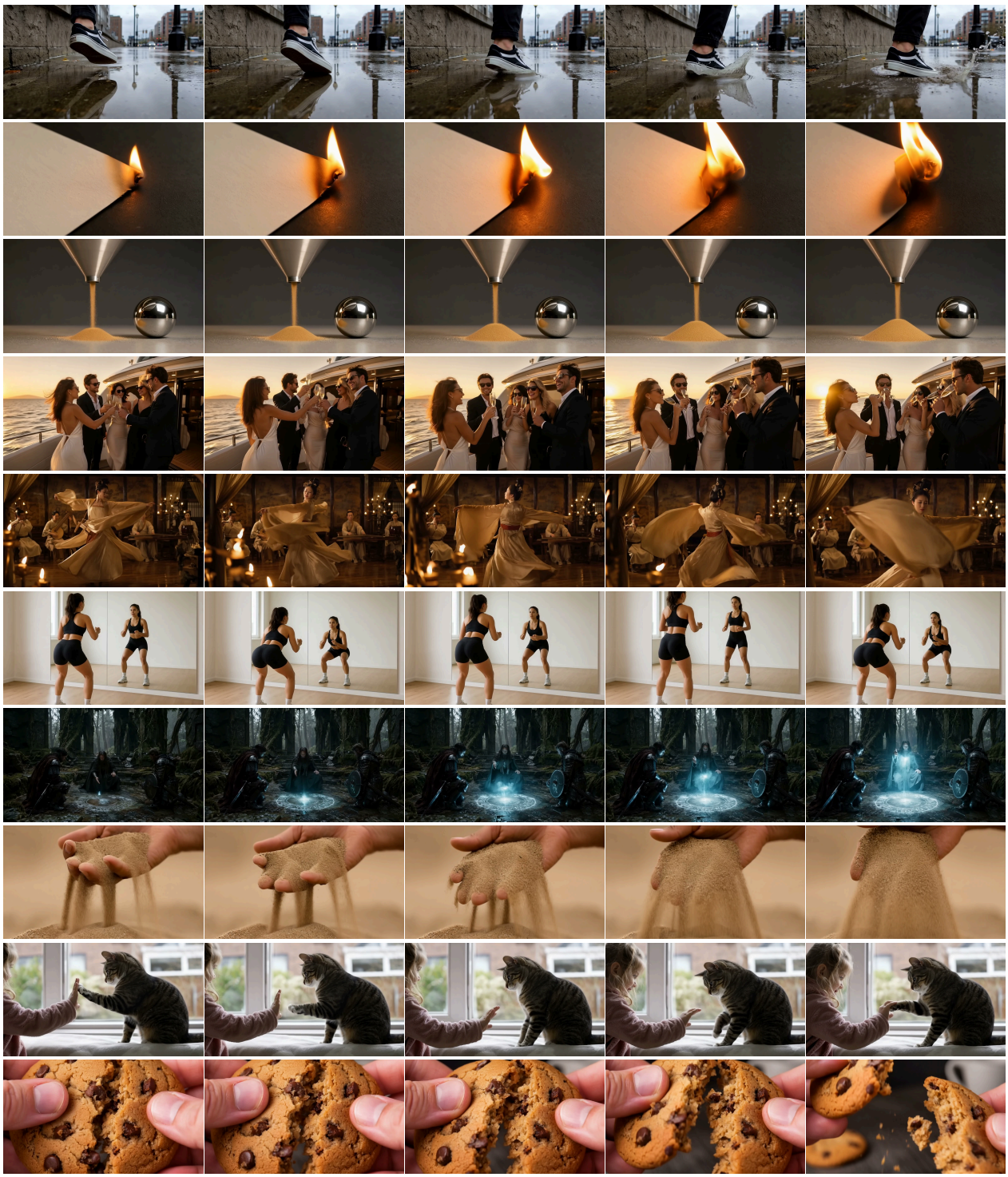


Figure 19. Qualitative results of `LingBot-Video` on text-and-image-to-video generation. Each row shows five keyframes uniformly sampled from one generated video; time flows from left to right.

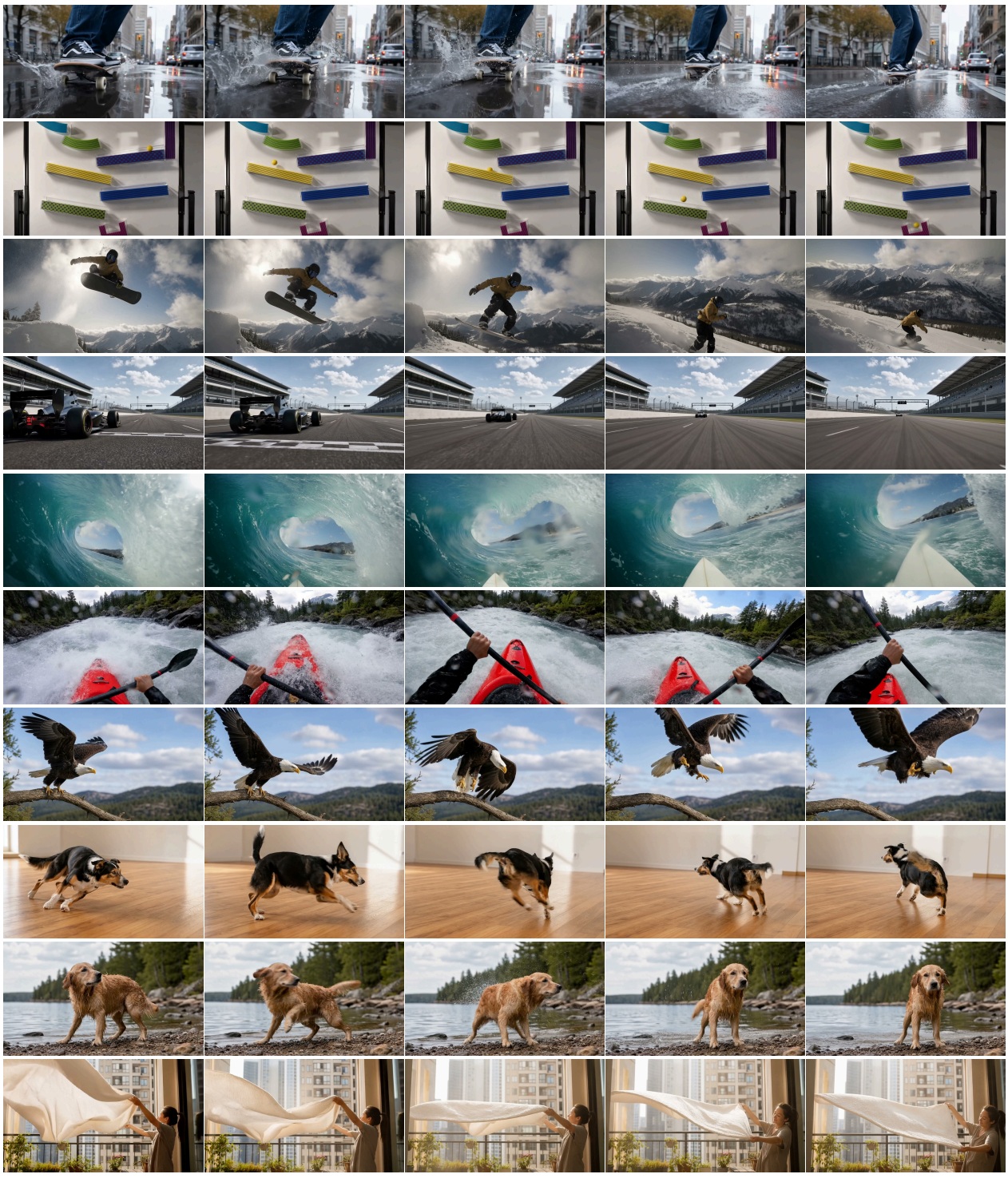


Figure 20. Qualitative results of LingBot-Video on text-and-image-to-video generation.

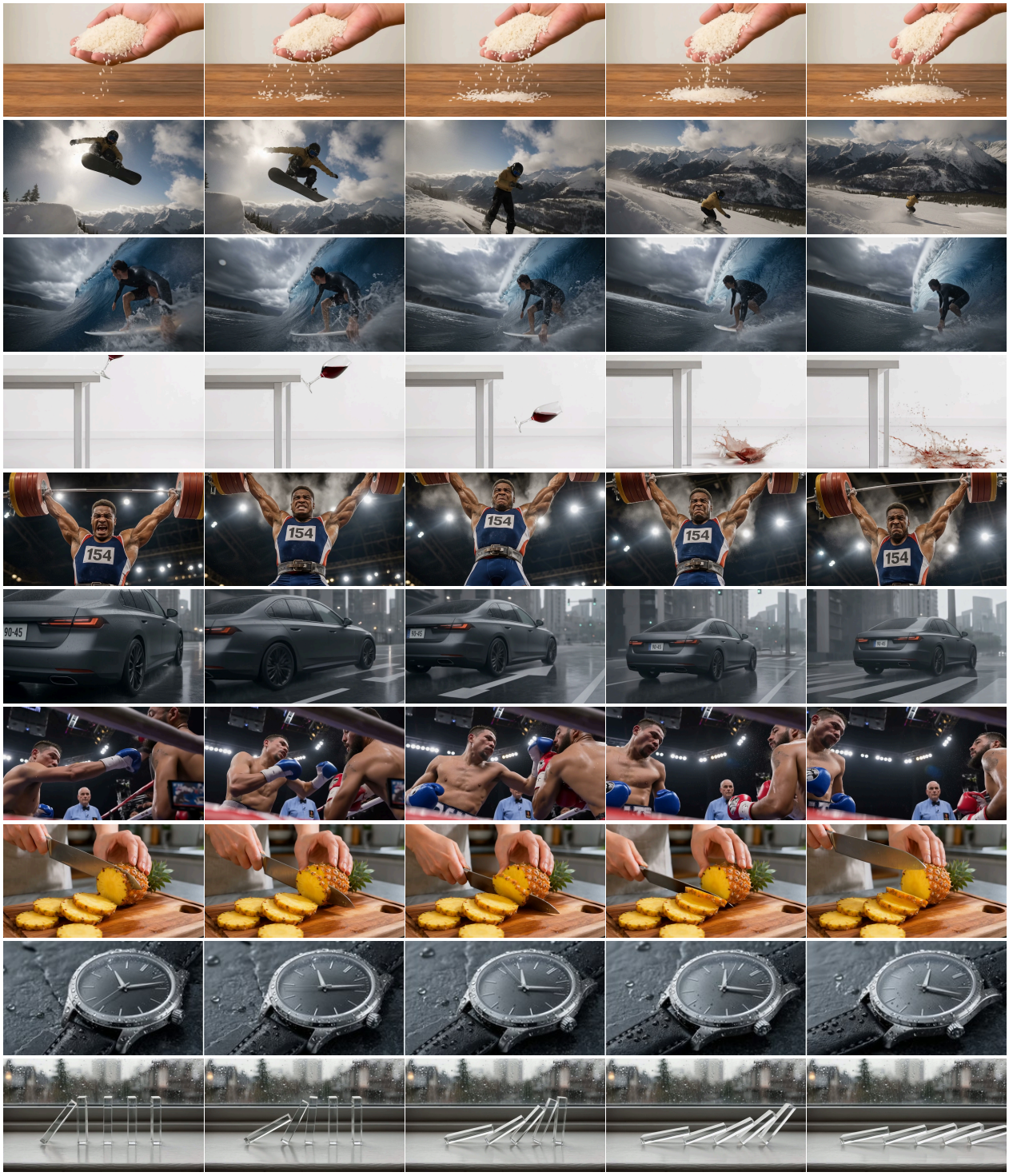


Figure 21. Qualitative results of LingBot-Video on text-and-image-to-video generation.

Figure 22. Qualitative results of LingBot-Video on text-and-image-to-video generation.

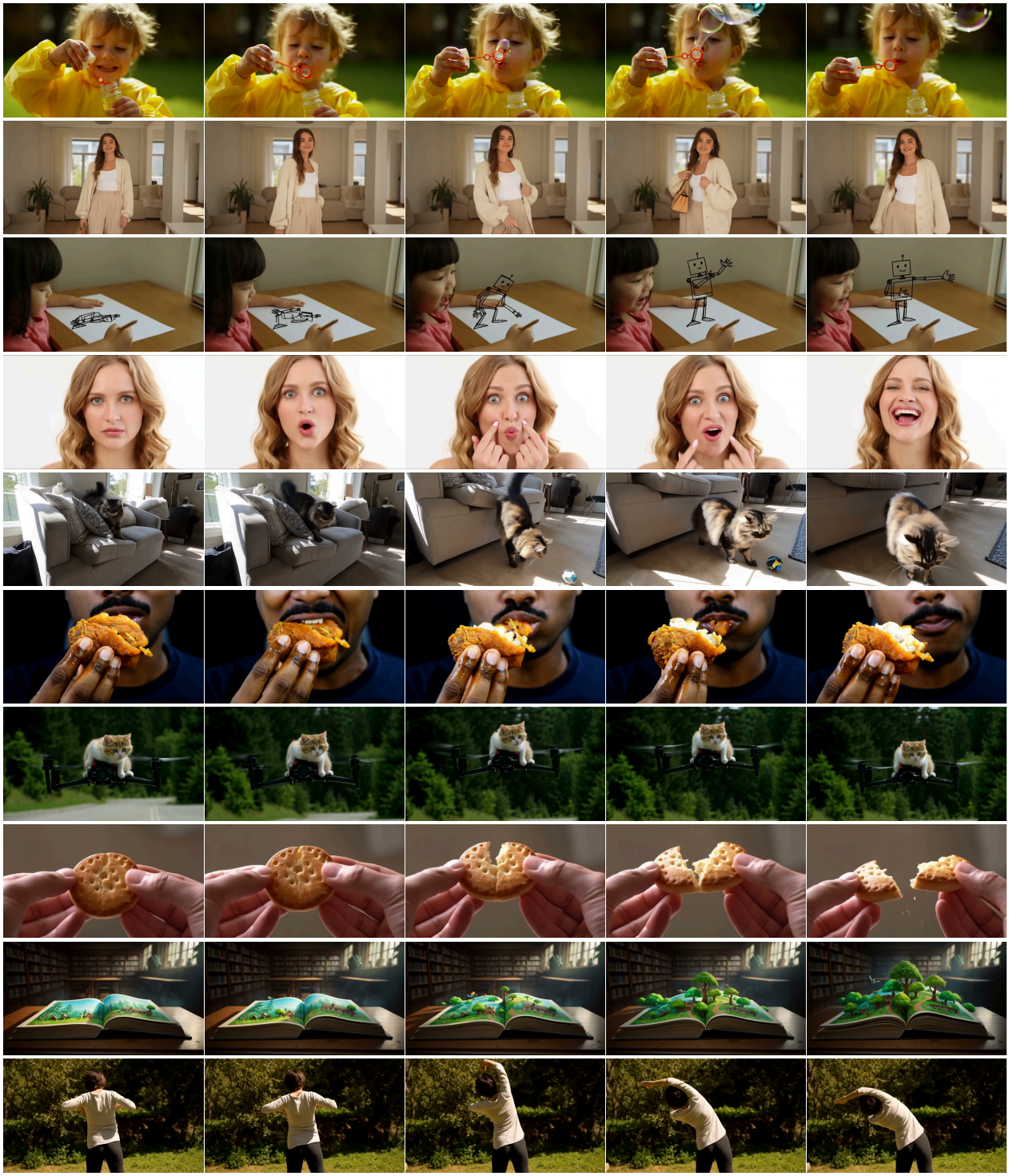


Figure 23. Qualitative results of LingBot-Video on text-to-video generation. Each row shows five keyframes uniformly sampled from one generated video; time flows from left to right.

Conclusion and Discussion

In this work, we present LingBot-Video, a pioneering MoE-based video foundation model tailored specifically for embodied intelligence, successfully bridging the gap between digital creativity and physical actuation. By scaling up the MoE-based Diffusion Transformer from scratch, we achieve an optimal balance between modeling capacity and inference efficiency, paving the way for the development of the next-generation robot brain. We aim for it to serve as an embodied video simulator that plays critical roles for the robotics community:

- **Data Engine:** Synthesizes high-fidelity, low-cost training data at scale to mitigate data scarcity in robotics.
- **Policy Evaluator:** Serves as a visual simulator to evaluate robot policies safety-critically without real-world risks.
- **Action Planner:** Predicts "what happens next" to assist the robot's real-time decision-making and planning.

By open-sourcing LingBot-Video, we hope to inspire and collaborate with the community to collectively push the boundaries of embodied physical engines and next-generation robot brains.

Contributors

Pre-training: Shuailei Ma, Jingjing Wang, Kecheng Zheng, Yinghao Xu

Data Infra: Xinyang Wang, Jingjing Wang, Jiaqi Liao, Shuailei Ma, Yuqi Gan, Weisen Wang, Wei Wu, Jiahao Shao, Hao Ouyang, Qiuyu Wang, Yipengjing Sun, Liangxiao Hu

Post-training: Jiaqi Liao, Chaoran Feng, Zijing Hu, Zichen Xi, Yanhong Zeng, Qin Zhao, Zifan Shi, Shangzhan Zhang, Nan Xue

Refiner: Chong Bao

Serving Infra: Shuailei Ma

Evaluation: Jingjing Wang*, Zijing Hu*, Chaoran Feng, Lunke Pan

Project Sponsor: Xing Zhu, Yujun Shen

Project Lead: Ka Leong Cheng

*Equal Contribution

Acknowledgments

We thank Qingyan Bai, Jingye Chen, Jingyue Chen, Ka Yu Cheng, Xiaoyue Duan, Xiaoqian Ma, Yihao Meng, Fan Fan, Biao Gong, Bo Jiang, Yangyan Li, Yixuan Li, Zichen Liu, Fan Lu, Yichong Lu, Fangqing Teng, Hanlin Wang, Jiahao Wang, Junke Wang, Ruonan Wang, Wenfei Xie, Jingmei Zhao Shuai Zhou, Jiapeng Zhu, Jiayi Zhu (***listed alphabetically by last name***) for their valuable discussions and assistance.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

[3] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, et al. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pages 929–947, 2024.

[4] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

[5] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[6] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[7] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[8] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

[9] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[10] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, pages 4603–4623, 2024.

[11] Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Xin Jin, Liangchen Li, et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

[12] Xunliang Cai, Qilong Huang, Zhuoliang Kang, Hongyu Li, Shijun Liang, Liya Ma, Siyu Ren, Xiaoming Wei, Rixu Xie, and Tong Zhang. Longcat-video technical report. *arXiv preprint arXiv:2510.22200*, 2025.

[13] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-$alpha$: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International conference on learning representations*, volume 2024, pages 57611–57640, 2024.

[14] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

[15] Guo Cheng, Danni Yang, Ziqi Huang, Jianlou Si, Chenyang Si, and Ziwei Liu. Realdpo: Real or not real, that is the preference. *arXiv preprint arXiv:2510.14955*, 2025.

[16] Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local all-pair correspondence for point tracking. In *European conference on computer vision*, pages 306–325. Springer, 2024.

[17] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of machine learning research*, 24(240):1–113, 2023.

[18] Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1280–1297, 2024.

[19] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pages 16344–16359, 2022.

[20] DeepSeek-AI. Deepep: An efficient expert-parallel communication library. https://github.com/deepseek-ai/DeepEP, 2025. Accessed: 2026-07-08.

[21] Mostafa Dehghani, Alexey Gritsenko, Aurelien Sun, Sherjil Uesato, Yi Tay, Basil Mustafa, Joao Carreira, Christian Szegedy, and Xiaohua Zhai. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512, 2023.

[22] Yufan Deng, Zilin Pan, Hongyu Zhang, Xiaojie Li, Ruoqing Hu, Yufei Ding, Yiming Zou, Yan Zeng, and Daquan Zhou. Rethinking video generation model for the embodied world. In *Forty-third International Conference on Machine Learning*, 2026.

[23] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7346–7356, 2023.

[24] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, pages 12606–12633, 2024.

[25] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[26] Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

[27] Xiangbo Gao, Mingyang Wu, Siyuan Yang, Jiongze Yu, Pardis Taghavi, Fangzhou Lin, and Zhengzhong Tu. The pulse of motion: Measuring physical frame rate from visual dynamics. *arXiv preprint arXiv:2603.14375*, 2026.

[28] Yanjiang Guo, Tony Lee, Lucy Xiaoyang Shi, Jianyu Chen, Percy Liang, and Chelsea Finn. Vlaw: Iterative co-improvement of vision-language-action policy and world model. *arXiv preprint arXiv:2602.12063*, 2026.

[29] Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.

[30] Yuncheng Guo, Junyan Ye, Chenjue Zhang, Hengrui Kang, Haohuan Fu, Conghui He, and Weijia Li. Omniaid: Decoupling semantic and artifacts for universal ai-generated image detection in the wild. *arXiv preprint arXiv:2511.08423*, 2025.

[31] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *The Twelfth International Conference on Learning Representations*, 2024.

[32] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Fei-Fei Li, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. In *European Conference on Computer Vision*, pages 393–411. Springer, 2024.

[33] Eyal Gutflaish, Eliran Kachlon, Hezi Zisman, Tal Hacham, Nimrod Sarid, Alexander Visheratin, Saar Huberman, Gal Davidi, Guy Bukchin, Kfir Goldberg, and Ron Mokady. Generating an image from 1,000 words: Enhancing text-to-image with structured captions. *arXiv preprint arXiv:2511.06876*, 2025.

[34] Xiaoxuan He, Siming Fu, Zeyue Xue, Weijie Wang, Ruizhe He, Yuming Li, Dacheng Yin, Shuai Dong, Haoyang Huang, Hongfa Wang, et al. Flash-grpo: Efficient alignment for video diffusion via one-step policy optimization. *arXiv preprint arXiv:2605.15980*, 2026.

[35] Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. Tempflow-grpo: When timing matters for grpo in flow models. *arXiv preprint arXiv:2508.04324*, 2025.

[36] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[37] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

[38] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, pages 30016–30030, 2022.

[39] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*, 2022.

[40] Jian Hu, Xibin Wu, Zilin Zhu, Weixun Wang, Dehao Zhang, Yu Cao, et al. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*, 6, 2024.

[41] Changho Hwang, Wei Cui, Yifan Xiong, Ziyue Yang, Ze Liu, Han Hu, Zilong Wang, Rafael Salas, Jithin Jose, Prabhat Ram, et al. Tutel: Adaptive mixture-of-experts at scale. In *Proceedings of Machine Learning and Systems*, pages 269–287, 2023.

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

[43] Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models. *arXiv preprint arXiv:2309.14509*, 2023.

[44] Zhennan Jiang, Shangqing Zhou, Yutong Jiang, Zefang Huang, Mingjie Wei, Yuhui Chen, Tianxing Zhou, Zhen Guo, Hao Lin, Quanlu Zhang, et al. Wovr: World models as reliable simulators for post-training vla policies with rl. *arXiv preprint arXiv:2602.13977*, 2026.

[45] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

[46] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[47] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

[48] Vijay Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing activation recomputation in large transformer models. In *Proceedings of Machine Learning and Systems*, pages 341–353, 2023.

[49] Mario Michael Krell, Matej Kosec, Sergio P. Perez, and Andrew Fitzgibbon. Efficient sequence packing without cross-contamination: Accelerating large language models without impacting performance. *arXiv preprint arXiv:2107.02027*, 2021.

[50] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In *The Ninth International Conference on Learning Representations*, 2021.

[51] Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Yiming Cheng, Miles Yang, Zhao Zhong, and Liefeng Bo. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025.

[52] Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

[53] Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, and Soumith Chintala. Pytorch distributed: Experiences on accelerating data parallel training. *Proceedings of the VLDB Endowment*, 13(12):3005–3018, 2020.

[54] Wanchao Liang, Tianyu Liu, Less Wright, Will Constable, Andrew Gu, Chien-Chin Huang, Iris Zhang, Wei Feng, Howard Huang, Junjie Wang, Sanket Purandare, Gokul Nadathur, and Stratos Idreos. Torchtitan: One-stop pytorch native solution for production ready LLM pretraining. In *The Thirteenth International Conference on Learning Representations*, 2025.

[55] Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. *arXiv preprint arXiv:2508.05635*, 2025.

[56] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

[57] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[58] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. In *Advances in neural information processing systems*, pages 40783–40818, 2026.

[59] Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, et al. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *arXiv preprint arXiv:2601.05242*, 2026.

[60] 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*, 2023.

[61] Qianli Ma, Yaowei Zheng, Zhelun Shi, Zhongkai Zhao, Bin Jia, Ziyue Huang, Zhiqi Lin, Youjie Li, Jiacheng Yang, Yanghua Peng, Zhi Zhang, and Xin Liu. Veomni: Scaling any modality model training with model-centric distributed recipe zoo. *arXiv preprint arXiv:2508.02317*, 2025.

[62] Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15086–15095, 2025.

[63] Zhiyu Mei, Wei Fu, Kaiwei Li, Guangju Wang, Huanchen Zhang, and Yi Wu. Real: Efficient rlhf training of large language models with parameter reallocation. In *Proceedings of Machine Learning and Systems*, 2025.

[64] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. Ray: A distributed framework for emerging {AI} applications. In *13th USENIX symposium on operating systems design and implementation (OSDI 18)*, pages 561–577, 2018.

[65] Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2026.

[66] Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15, 2021.

[67] Steven J. Nowlan and Geoffrey E. Hinton. Evaluation of adaptive mixtures of competing experts. In *Advances in Neural Information Processing Systems 3 (NIPS 1990)*, pages 774–780. Morgan Kaufmann, 1990.

[68] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking : Bringing order to the web. In *The Web Conference*, 1999.

[69] Byeongjun Park, Hyojun Go, Jin-Young Kim, Sangmin Woo, Seokil Ham, and Changick Kim. Switch diffusion transformer: Synergizing denoising tasks with sparse mixture-of-experts. In *European Conference on Computer Vision*, pages 461–477. Springer, 2024.

[70] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[71] Qi Qin, Le Zhuo, Yi Xin, Ruoyi Du, Zhen Li, Bin Fu, Yiting Lu, Xinyue Li, Dongyang Liu, Xiangyang Zhu, et al. Lumina-image 2.0: A unified and efficient image generative framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20031–20042, 2025.

[72] Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model. https://qwen.ai/blog?id=qwen3.6-27b, 2026. Accessed: 2026-07-08.

[73] Tim Rädsch, Yuki M Asano, Hilde Kuehne, Stefan Bauer, Priyank Jaini, Robert Geirhos, and Carsten T Lüth. Physics-iq verified. *arXiv preprint arXiv:2606.18943*, 2026.

[74] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, pages 53728–53741, 2023.

[75] Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale. In *International Conference on Machine Learning*, pages 18332–18346. PMLR, 2022.

[76] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16, 2020.

[77] Xuanchi Ren, Yifan Lu, Tianshi Cao, Ruiyuan Gao, Shengyu Huang, Amirmojtaba Sabour, Tianchang Shen, Tobias Pfaff, Jay Zhangjie Wu, Runjian Chen, et al. Cosmos-drive-dreams: Scalable synthetic driving data generation with world foundation models. *arXiv preprint arXiv:2506.09042*, 2025.

[78] Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. Gaia-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.

[79] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4713–4726, 2022.

[80] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[81] Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

[82] Haiying Sha and Yan Zheng. Sparse mixture-of-experts routing in visual diffusion transformers: Diagnosis, boundary calibration and evolutionary roadmap from routing collapse to selective deadlock. *arXiv preprint arXiv:2605.19378*, 2026.

[83] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[84] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[85] Noam Shazeer, Youlong Cheng, Niki Parmar, Dustin Tran, Ashish Vaswani, Penporn Koanantakool, Peter Hawkins, HyoukJoong Lee, Mingsheng Hong, Cliff Young, Ryan Sepassi, and Blake Hechtman. Mesh-tensorflow: Deep learning for supercomputers. In *Advances in Neural Information Processing Systems*, 2018.

[86] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

[87] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297, 2025.

[88] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

[89] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*, 2023.

[90] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *The Ninth International Conference on Learning Representations*, 2021.

[91] Tomás Soucek and Jakub Lokoc. Transnet v2: An effective deep network architecture for fast shot transition detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11218–11221, 2024.

[92] Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint arXiv:2512.14614*, 2025.

[93] Zhenyu Tang, Chaoran Feng, Yufan Deng, Jie Wu, Xiaojie Li, Rui Wang, Yunpeng Chen, and Daquan Zhou. Enhancing spatial understanding in image generation via reward modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27249–27259, 2026.

[94] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[95] Gemini Robotics Team, Krzysztof Choromanski, Coline Devin, Yilun Du, Debidatta Dwibedi, Ruiqi Gao, Abhishek Jindal, Thomas Kipf, Sean Kirmani, Isabel Leal, et al. Evaluating gemini robotics policies in a veo world simulator. *arXiv preprint arXiv:2512.10675*, 2025.

[96] Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, Yihang Chen, Jie Liu, Yansong Cheng, Yao Yao, Jiayi Zhu, Yihao Meng, Kecheng Zheng, Qingyan Bai, Jingye Chen, Zehong Shen, Yue Yu, Xing Zhu, Yujun Shen, and Hao Ouyang. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026.

[97] THUDM. slime: An LLM post-training framework for reinforcement learning at scale. https://github.com/THUDM/slime, 2025.

[98] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

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

[100] Wei-Cheng Tseng, Gashon Hussein, Yuzhu Dong, Allen Z Ren, Lucy X Shi, XuDong Wang, Sergey Levine, Zhaoshuo Li, Jinwei Gu, Florian Shkurti, et al. Sc3-eval: Evaluating robot foundation models via self-consistent video generation. *arXiv preprint arXiv:2606.18610*, 2026.

[101] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. In *International Conference on Learning Representations (ICLR)*, 2023.

[102] Huy V Vo, Vasil Khalidov, Timothée Darcet, Théo Moutakanni, Nikita Smetanin, Marc Szafraniec, Hugo Touvron, Camille Couprie, Maxime Oquab, Armand Joulin, et al. Automatic data curation for self-supervised learning: A clustering-based approach. *Transactions on Machine Learning Research*, 2024.

[103] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. Accessed: 2026-07-08.

[104] Chi Wan, Kangrui Wang, Yuan Si, Pingyue Zhang, and Manling Li. Worldagen: Unified state-action prediction with test-time world model training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18584–18592, 2026.

[105] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[106] Wan-AI Team. Wan2.2: A text-to-video and image-to-video generative framework. https://github.com/Wan-Video/Wan2.2, 2025. Accessed: 2026-07-08.

[107] Feng Wang and Zihao Yu. Coefficients-preserving sampling for reinforcement learning with flow matching. *arXiv preprint arXiv:2509.05952*, 2025.

[108] Liang Wang, Huazuo Gao, Ruixin Zhao, Xiaoting Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *arXiv preprint arXiv:2408.15664*, 2024.

[109] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 1905–1914, 2021.

[110] Wikimedia Foundation. Wikimedia downloads. https://dumps.wikimedia.org, 2026. Accessed: 2026-07-08.

[111] Samuel Williams, Andrew Waterman, and David Patterson. Roofline: An insightful visual performance model for multicore architectures. *Communications of the ACM*, 52(4):65–76, 2009.

[112] Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, et al. Hunyuanvideo 1.5 technical report. *arXiv preprint arXiv:2511.18870*, 2025.

[113] Chenyuan Wu, Jiahao Wang, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, et al. Omnigen2: Towards instruction-aligned multimodal generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21964–21975, 2026.

[114] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

[115] Yuanzhong Xu, HyoukJoong Lee, Dehao Chen, Blake Hechtman, Yanping Huang, Rahul Joshi, Maxim Krikun, Dmitry Lepikhin, Andy Ly, Marcello Maggioni, Ruoming Pang, Noam Shazeer, Shibo Wang, Tao Wang, Yonghui Wu, and Zhifeng Chen. Gspmd: General and scalable parallelization for ml computation graphs. *arXiv preprint arXiv:2105.04663*, 2021.

[116] Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning rl with pretraining in diffusion models. *arXiv preprint arXiv:2509.25050*, 2025.

[117] Zeyue Xue, Siming Fu, Jie Huang, Shuai Lu, Haoran Li, Yijun Liu, Yuming Li, Xiaoxuan He, Mengzhao Chen, Haoyang Huang, et al. A systematic post-train framework for video generation. *arXiv preprint arXiv:2604.25427*, 2026.

[118] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

[119] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

[120] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in neural information processing systems*, pages 47455–47487, 2024.

[121] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations*, 2023.

[122] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4791–4800, 2021.

[123] Liujie Zhang, Benzhe Ning, Rui Yang, Xiaoyan Yu, Jiaxing Li, Lumeng Wu, Jia Liu, Minghao Li, Weihang Chen, Weiqi Hu, et al. Relax: An asynchronous reinforcement learning engine for omni-modal post-training at scale. *arXiv preprint arXiv:2604.11554*, 2026.

[124] Shilong Zhang, Wenbo Li, Shoufa Chen, Chongjian Ge, Peize Sun, Yifu Zhang, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Ping Luo. Flashvideo: Flowing fidelity to detail for efficient high-resolution video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12735–12743, 2026.

[125] Yifu Zhang, Hao Yang, Yuqi Zhang, Yifei Hu, Fengda Zhu, Chuang Lin, Xiaofeng Mei, Yi Jiang, Zehuan Yuan, and Bingyue Peng. Waver: Wave your way to lifelike video generation. *arXiv preprint arXiv:2508.15761*, 2025.

[126] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. Pytorch fsdp: Experiences on scaling fully sharded data parallel. *Proceedings of the VLDB Endowment*, 16(12):3848–3860, 2023.

[127] Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations (ICLR)*, 2026.

[128] Lianmin Zheng, Zhuohan Li, Hao Zhang, Yonghao Zhuang, Zhifeng Chen, Yanping Huang, Yida Wang, Yuanzhong Xu, Danyang Zhuo, Eric P. Xing, Joseph E. Gonzalez, and Ion Stoica. Alpa: Automating inter- and intra-operator parallelism for distributed deep learning. In *16th USENIX Symposium on Operating Systems Design and Implementation*, pages 559–578, 2022.

[129] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. Sglang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems*, pages 62557–62583, 2024.

[130] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025.

A Structured Caption Details

A.1 Example Structured Captions

We show one representative caption per data category, reproduced verbatim from the training data.

Example structured caption for image data.


```
{
 "comprehensive_description": "This image captures the opulent interior of a grand palace hall, likely the Grand Kremlin Palace. The room is characterized by a lavish Baroque or Neoclassical style, featuring a high vaulted ceiling adorned with intricate gold patterns and a central medallion. The walls are covered in deep red fabric with gold embroidery, punctuated by tall, fluted white columns with ornate gold capitals. At the far end of the hall, a raised dais holds a large, framed portrait of a historical figure and a single red upholstered throne. The floor is a masterpiece of polished wood parquet, featuring complex geometric and circular inlaid patterns in various shades of brown. A massive, multi-tiered crystal chandelier hangs from the upper left, casting a warm, golden glow throughout the space. The overall atmosphere is one of immense wealth, power, and historical significance, with a symmetrical composition that emphasizes the room's scale and architectural detail.",
 "camera_info": {
  "color": "Warm",
  "frame_size": "Extreme Wide",
  "shot_type_angle": "Low angle",
  "lens_size": "Wide",
  "composition": "Balanced",
  "lighting": "Hard light",
  "lighting_type": "Artificial light"
 },
 "world_knowledge": [],
 "prominent_elements": [
  {
   "name": "vaulted ceiling",
   "description": "A high, arched ceiling featuring a complex pattern of gold medallions and intricate carvings.",
   "location": "spanning the top half of the frame",
   "relative_size": "dominant",
   "shape_and_color": "arched; gold and cream",
   "texture": "carved and metallic",
   "appearance_details": "Features a central large medallion and radiating patterns of smaller circular motifs.",
   "relationship": "Forms the upper boundary of the entire scene.",
   "orientation": "arched",
   "pose": "",
   "expression": "",
   "clothing": "",
   "gender": "",
   "skin_tone_and_texture": ""
  },
  {
   "name": "red wall panels",
   "description": "Large sections of wall covered in deep red fabric with gold embroidery.",
   "location": "spanning the middle and background walls",
   "relative_size": "dominant",
   "shape_and_color": "rectangular; deep red and gold",
   "texture": "fabric and metallic embroidery",
   "appearance_details": "The fabric is decorated with repeating gold floral and heraldic patterns.",
   "relationship": "Provides the primary background for the columns and the throne area.",
   "orientation": "vertical",
   "pose": "",
   "expression": "",
   "clothing": "",
   "gender": "",
   "skin_tone_and_texture": "",
   "is_cluster": true,
   "number_of_objects": "several"
  },
  {
   "name": "fluted columns",
   "description": "Tall, white columns with vertical grooves and ornate gold capitals.",
   "location": "distributed along the walls and at the far end",
   "relative_size": "large",
   "shape_and_color": "cylindrical; white and gold",
   "texture": "smooth and metallic",
   "appearance_details": "The capitals are highly decorative with gold leaf and classical motifs.",
   "relationship": "They support the upper architectural elements and frame the throne area.",
   "orientation": "upright",
   "pose": "",
   "expression": "",
   "clothing": "",
   "gender": "",
```

```
    "skin_tone_and_texture": "",
    "is_cluster": true,
    "number_of_objects": "many"
   },
   {
    "name": "parquet floor",
    "description": "A highly polished wooden floor with intricate geometric and circular inlaid patterns.",
    "location": "spanning the bottom half of the frame",
    "relative_size": "dominant",
    "shape_and_color": "flat; various shades of brown and tan",
    "texture": "smooth and glossy",
    "appearance_details": "Features complex interlocking geometric shapes and circular medallions.",
    "relationship": "Reflects the light from the chandelier and the colors of the walls.",
    "orientation": "horizontal",
    "pose": "",
    "expression": "",
    "clothing": "",
    "gender": "",
    "skin_tone_and_texture": ""
   },
   {
    "name": "large portrait",
    "description": "A framed painting depicting a historical figure, likely a monarch.",
    "location": "center background, above the throne",
    "relative_size": "medium",
    "shape_and_color": "rectangular; dark tones with gold frame",
    "texture": "matte",
    "appearance_details": "The figure is dressed in elaborate historical attire and is seated.",
    "relationship": "Positioned as the focal point of the room's back wall.",
    "orientation": "upright",
    "pose": "",
    "expression": "",
    "clothing": "",
    "gender": "",
    "skin_tone_and_texture": ""
   },
   {
    "name": "ornate throne",
    "description": "A single, high-backed chair upholstered in red fabric with gold trim.",
    "location": "center background, on a raised platform",
    "relative_size": "small",
    "shape_and_color": "rectangular; red and gold",
    "texture": "fabric and metallic",
    "appearance_details": "Features a high back and gold-colored armrests and legs.",
    "relationship": "Sits on a small red-carpeted dais in front of the portrait.",
    "orientation": "upright",
    "pose": "",
    "expression": "",
    "clothing": "",
    "gender": "",
    "skin_tone_and_texture": ""
   },
   {
    "name": "crystal chandelier",
    "description": "A large, multi-tiered chandelier with numerous light sources.",
    "location": "top-left corner",
    "relative_size": "medium",
    "shape_and_color": "complex; gold and clear",
    "texture": "metallic and glass",
    "appearance_details": "Features multiple arms and hanging crystal elements.",
    "relationship": "Hangs from the ceiling and provides the main light source for the left side of the room.",
    "orientation": "hanging",
    "pose": "",
    "expression": "",
    "clothing": "",
    "gender": "",
    "skin_tone_and_texture": ""
   }
  ]
}
```


Example structured caption for video data.


```
{
 "comprehensive_description": {
```

```
  "scene_content_description": "The video captures an outdoor cooking scene where several meat rolls are being prepared on a large, black, circular metal griddle. The griddle is positioned over an open fire, with visible flames and glowing embers at the bottom. The meat rolls, made of thin slices of red meat stuffed with green herbs and orange carrot pieces, are held together by wooden toothpicks. Throughout the video, wisps of white smoke rise from the hot surface. A person's hand periodically enters the frame from the top right to place additional meat rolls onto the griddle. The background is softly blurred, showing hints of greenery and a wooden structure, suggesting a garden or backyard setting. The lighting is bright and natural, creating a warm and rustic atmosphere.",
  "camera_movement_description": "The camera is stationary throughout the video, maintaining a steady, eye-level close-up shot of the griddle and the cooking process."
},
"camera_info": {
  "color": "Saturated",
  "frame_size": "Wide",
  "shot_type_angle": "High angle",
  "lens_size": "Medium",
  "composition": "Center",
  "lighting": "Hard light",
  "lighting_type": "Daylight"
},
"world_knowledge": [],
"prominent_elements": [
 {
   "name": "meat rolls",
   "description": "Several cylindrical rolls made of thin red meat slices, stuffed with green herbs and orange carrot pieces, secured with wooden toothpicks.",
   "actions": [
     {
       "timestamp": "[0.0s - 7.1s]",
       "action": "The rolls are placed onto the griddle and sizzle as they cook, with smoke rising from them."
     }
   ],
   "location": "Center of the frame on the griddle",
   "relative_size": "large",
   "shape_and_color": "Cylindrical shapes with red, green, and orange colors",
   "texture": "Fleshy and moist",
   "appearance_details": "Visible toothpicks and stuffing ingredients like parsley and carrots.",
   "relationship": "Placed on the black griddle to be cooked.",
   "orientation": "Horizontal",
   "pose": "",
   "expression": "",
   "clothing": "",
   "gender": "",
   "skin_tone_and_texture": "",
   "is_cluster": true,
   "number_of_objects": "several"
 },
 {
   "name": "black griddle",
   "description": "A large, circular, flat metal cooking surface with a slightly textured, matte black finish.",
   "actions": [
     {
       "timestamp": "[0.0s - 7.1s]",
       "action": ""
     }
   ],
   "location": "Occupies most of the lower and middle frame",
   "relative_size": "dominant",
   "shape_and_color": "Circular and black",
   "texture": "Matte and slightly rough",
   "appearance_details": "Shows signs of heat and oil from cooking.",
   "relationship": "Serves as the cooking surface for the meat rolls.",
   "orientation": "Horizontal",
   "pose": "",
   "expression": "",
   "clothing": "",
   "gender": "",
   "skin_tone_and_texture": ""
 },
 {
   "name": "person's hand",
   "description": "A human hand that appears periodically to place meat rolls on the griddle.",
   "actions": [
     {
       "timestamp": "[0.0s - 0.67s]",
       "action": "Enters from the top right and places a roll."
     },
     {
       "timestamp": "[2.67s - 3.67s]",
       "action": "Enters from the top right and places a roll."
```

```
      },
      {
        "timestamp": "[5.33s - 6.0s]",
        "action": "Enters from the top right and places a roll."
      }
    ],
    "location": "Top right corner of the frame",
    "relative_size": "medium",
    "shape_and_color": "Flesh-toned hand",
    "texture": "Smooth skin",
    "appearance_details": "Only the hand and part of the forearm are visible.",
    "relationship": "Interacts with the meat rolls and the griddle.",
    "orientation": "Reaching downward",
    "pose": "Reaching and grasping",
    "expression": "",
    "clothing": "",
    "gender": "male",
    "skin_tone_and_texture": "Light skin tone"
  },
  {
    "name": "smoke",
    "description": "Wisps of white and grey smoke rising from the hot griddle.",
    "actions": [
      {
        "timestamp": "[0.0s - 7.1s]",
        "action": "Continuously rises and drifts toward the left side of the frame."
      }
    ],
    "location": "Upper left and center of the frame",
    "relative_size": "medium",
    "shape_and_color": "Amorphous and white/grey",
    "texture": "Wispy and translucent",
    "appearance_details": "Thin, rising plumes.",
    "relationship": "Produced by the heat of the griddle and the cooking meat.",
    "orientation": "Rising upward",
    "pose": "",
    "expression": "",
    "clothing": "",
    "gender": "",
    "skin_tone_and_texture": ""
  }
 ]
}
```


Example structured caption for VLA data.


```
{
 "comprehensive_description": {
  "scene_content_description": "The video presents a first-person perspective of a robotic workspace, likely a simulated grocery store or automated sorting station. In the foreground, a metal wire shopping cart with red handles is positioned, containing a clear plastic bag with red printed text. Behind the cart is a wooden display shelf divided into several compartments, neatly stocked with a variety of colorful artificial produce. The items include yellow bell peppers, purple eggplants, red pumpkins, white mushrooms, brown potatoes, green cucumbers, yellow corn, red tomatoes, and red chili peppers. Two robotic arms are visible in the frame. The left robotic arm, featuring a grey body and a black two-finger gripper, remains completely stationary in the upper left corner throughout the video. The right robotic arm, which has a white body and a black two-finger gripper, is the active subject. It begins by holding a green cucumber, moves downwards to position the cucumber over the plastic bag in the shopping cart, and then opens its gripper to release the cucumber into the bag. After dropping the item, the right robotic arm retracts upwards and slightly to the right, returning to a higher position above the display shelf. The lighting is bright and even, highlighting the vibrant colors of the artificial vegetables.",
  "camera_movement_description": "The camera remains completely stationary throughout the entire video, maintaining a fixed, slightly high-angle perspective looking down at the shopping cart and the display shelf."
 },
 "camera_info": {
  "color": "Saturated",
  "frame_size": "Wide",
  "shot_type_angle": "High angle",
  "lens_size": "Ultra Wide / Fisheye",
  "composition": "Balanced",
  "lighting": "Hard light",
  "lighting_type": "Artificial light"
 },
 "world_knowledge": [],
 "prominent_elements": [
  {
    "name": "right robotic arm",
```

```
    "description": "A robotic arm with a white cylindrical body and a black, two-finger gripper mechanism.",
    "actions": [
     {
      "timestamp": "[0.0s - 1.5s]",
      "action": "Moves downwards while grasping a green cucumber."
     },
     {
      "timestamp": "[1.5s - 5.5s]",
      "action": "Opens its gripper to release the cucumber."
     },
     {
      "timestamp": "[5.5s - 7.2s]",
      "action": "Moves upwards and to the right."
     }
    ],
    "location": "Originates in the upper right, moves to the lower center, and returns to the upper right.",
    "relative_size": "large",
    "shape_and_color": "Cylindrical and angular, white and black.",
    "texture": "Smooth, metallic and plastic.",
    "appearance_details": "Features joints, cables, and a distinct two-finger gripper.",
    "relationship": "Interacts with the green cucumber and moves above the shopping cart.",
    "orientation": "Tilted downwards initially, then retracts upwards.",
    "pose": "",
    "expression": "",
    "clothing": "",
    "is_cluster": false,
    "number_of_objects": ""
   },
   {
    "name": "left robotic arm",
    "description": "A robotic arm with a grey rectangular body and a black, two-finger gripper mechanism.",
    "actions": [
     {
      "timestamp": "[0.0s - 7.2s]",
      "action": "Remains stationary."
     }
    ],
    "location": "Upper left corner of the frame.",
    "relative_size": "large",
    "shape_and_color": "Angular, grey and black.",
    "texture": "Smooth, metallic and plastic.",
    "appearance_details": "Features a distinct two-finger gripper and visible joints.",
    "relationship": "Positioned above the display shelf, inactive.",
    "orientation": "Tilted downwards.",
    "pose": "",
    "expression": "",
    "clothing": "",
    "is_cluster": false,
    "number_of_objects": ""
   },
   {
    "name": "green cucumber",
    "description": "A long, green, artificial vegetable.",
    "actions": [
     {
      "timestamp": "[0.0s - 1.5s]",
      "action": "Grasped by the right robotic arm and moves downwards."
     },
     {
      "timestamp": "[1.5s - 5.5s]",
      "action": "Falls downwards into the plastic bag."
     },
     {
      "timestamp": "[5.5s - 7.2s]",
      "action": "Rests inside the plastic bag."
     }
    ],
    "location": "Starts in the right robotic arm's gripper, ends up in the plastic bag in the lower center.",
    "relative_size": "small",
    "shape_and_color": "Elongated, green.",
    "texture": "Smooth.",
    "appearance_details": "Looks like a standard cucumber.",
    "relationship": "Held by the right robotic arm, then dropped into the plastic bag.",
    "orientation": "Tilted downwards while held, horizontal when resting in the bag.",
    "pose": "",
    "expression": "",
    "clothing": "",
    "is_cluster": false,
    "number_of_objects": ""
```

```
    },
    {
      "name": "shopping cart",
      "description": "A metal wire shopping cart with red handles.",
      "actions": [
        {
          "timestamp": "[0.0s - 7.2s]",
          "action": "Remains stationary."
        }
      ],
      "location": "Lower half of the frame.",
      "relative_size": "dominant",
      "shape_and_color": "Rectangular wireframe, silver and red.",
      "texture": "Metallic.",
      "appearance_details": "Contains a clear plastic bag with red text.",
      "relationship": "Serves as the receptacle for the dropped cucumber.",
      "orientation": "Horizontal.",
      "pose": "",
      "expression": "",
      "clothing": "",
      "is_cluster": false,
      "number_of_objects": ""
    },
    {
      "name": "display shelf",
      "description": "A wooden shelf divided into compartments, holding various artificial vegetables.",
      "actions": [
        {
          "timestamp": "[0.0s - 7.2s]",
          "action": "Remains stationary."
        }
      ],
      "location": "Upper half of the frame, behind the shopping cart.",
      "relative_size": "dominant",
      "shape_and_color": "Rectangular, light brown wood with colorful items.",
      "texture": "Wood grain.",
      "appearance_details": "Contains neatly arranged yellow peppers, purple eggplants, red pumpkins, mushrooms, potatoes, corn,
        ↪ tomatoes, and chili peppers.",
      "relationship": "Provides the background context and source of the items.",
      "orientation": "Horizontal.",
      "pose": "",
      "expression": "",
      "clothing": "",
      "is_cluster": false,
      "number_of_objects": ""
    }
  ]
}
```


Example structured caption for egocentric data.


```
{
  "comprehensive_description": {
    "scene_content_description": "The video presents a first-person perspective of a person working on a piece of machinery, likely in
      ↪ a garage or workshop setting. The environment features a concrete floor scattered with small debris. The primary subject is
      ↪ a large, bright orange lawnmower positioned centrally in the frame. The lawnmower has a prominent black rear tire with
      ↪ deep treads on the left side, a black engine cover on the right, and a blue 'Kohler Professional' sticker on its side. The person
      ↪ 's left arm, wearing a black short-sleeved shirt, is visible resting flat on the concrete floor on the left side of the frame. The
      ↪ person's legs, clad in blue jeans, are visible at the bottom. The action begins with the person's right arm, also wearing a
      ↪ black short-sleeved shirt, entering the frame from the right side. The right hand reaches towards the center of the lawnmower
      ↪ , specifically targeting a black cable or wire near the engine area. The hand grasps the cable and pulls it upwards and
      ↪ towards the right, manipulating the component. The left hand remains stationary on the floor throughout the sequence,
      ↪ providing stability.",
    "camera_movement_description": "The camera is body-mounted, providing a first-person point of view. It exhibits slight,
      ↪ continuous panning and tilting movements that correspond to the natural head and body motions of the person working.
      ↪ The shot size remains a close-up on the machinery and the person's hands."
  },
  "camera_info": {
    "color": "Saturated",
    "frame_size": "Medium",
    "shot_type_angle": "High angle",
    "lens_size": "Ultra Wide / Fisheye",
    "composition": "Center",
    "lighting": "Hard light",
    "lighting_type": "Daylight"
```

```
},
"world_knowledge": [],
"prominent_elements": [
 {
  "name": "right hand",
  "description": "A human right hand and forearm, wearing a black short-sleeved shirt.",
  "actions": [
   {
    "timestamp": "[0.0s - 2.0s]",
    "action": ""
   },
   {
    "timestamp": "[2.0s - 5.0s]",
    "action": "Enters from the right, reaches towards the center, grasps a black cable, and pulls it upwards and to the right."
   }
  ],
  "location": "Moves from the right edge towards the center-right of the frame.",
  "relative_size": "medium",
  "shape_and_color": "Arm shape, skin tone, black sleeve.",
  "texture": "Smooth skin, fabric texture.",
  "appearance_details": "Wearing a black short-sleeved shirt.",
  "relationship": "Interacts with the black cable on the lawnmower.",
  "orientation": "Extended forward and slightly downwards.",
  "pose": "",
  "expression": "",
  "clothing": "Black short-sleeved shirt.",
  "is_cluster": false,
  "number_of_objects": ""
 },
 {
  "name": "left hand",
  "description": "A human left hand and forearm, wearing a black short-sleeved shirt, resting flat on the ground.",
  "actions": [
   {
    "timestamp": "[0.0s - 5.0s]",
    "action": "Remains stationary on the floor."
   }
  ],
  "location": "Bottom left of the frame.",
  "relative_size": "medium",
  "shape_and_color": "Hand shape, skin tone, black sleeve.",
  "texture": "Smooth skin, fabric texture.",
  "appearance_details": "Wearing a black short-sleeved shirt, a ring is visible on one finger.",
  "relationship": "Resting on the concrete floor, providing stability for the person.",
  "orientation": "Flat, horizontal.",
  "pose": "",
  "expression": "",
  "clothing": "Black short-sleeved shirt.",
  "is_cluster": false,
  "number_of_objects": ""
 },
 {
  "name": "lawnmower",
  "description": "A large piece of outdoor machinery, primarily orange with black components.",
  "actions": [
   {
    "timestamp": "[0.0s - 5.0s]",
    "action": "Remains stationary."
   }
  ],
  "location": "Occupies the center and right portions of the frame.",
  "relative_size": "dominant",
  "shape_and_color": "Complex mechanical shape, predominantly bright orange and black.",
  "texture": "Smooth painted metal, rubber tire.",
  "appearance_details": "Features a large black rear tire with deep treads, a black engine cover, and a blue 'Kohler Professional'
    ↪ sticker.",
  "relationship": "The object being worked on by the right hand.",
  "orientation": "Upright.",
  "pose": "",
  "expression": "",
  "clothing": "",
  "is_cluster": false,
  "number_of_objects": ""
 },
 {
  "name": "black cable",
  "description": "A thin, flexible black wire or cable attached to the lawnmower.",
  "actions": [
   {
```

```
        "timestamp": "[0.0s - 2.0s]",
        "action": "Remains stationary."
      },
      {
        "timestamp": "[2.0s - 5.0s]",
        "action": "Grasped by the right hand and pulled upwards and to the right."
      }
    ],
    "location": "Center of the frame, near the engine area of the lawnmower.",
    "relative_size": "small",
    "shape_and_color": "Thin, linear, black.",
    "texture": "Smooth, flexible.",
    "appearance_details": "Appears to be a control cable or wire.",
    "relationship": "Attached to the lawnmower, manipulated by the right hand.",
    "orientation": "Curved, extending from the engine area.",
    "pose": "",
    "expression": "",
    "clothing": "",
    "is_cluster": false,
    "number_of_objects": ""
  }
 ]
}
```