World Simulation with Video Foundation Models for Physical AI

NVIDIA[1]

Abstract

We introduce [Cosmos-Predict2.5], the latest generation of the Cosmos World Foundation Models for Physical AI. Built on a flow-based architecture, [Cosmos-Predict2.5] unifies Text2World, Image2World, and Video2World generation in a single model and leverages [Cosmos-Reason1], a Physical AI vision-language model, to provide richer text grounding and finer control of world simulation. Trained on 200M curated video clips and refined with reinforcement learning–based post-training, [Cosmos-Predict2.5] achieves substantial improvements over [Cosmos-Predict1] in video quality and instruction alignment, with models released at 2B and 14B scales. These capabilities enable more reliable synthetic data generation, policy evaluation, and closed-loop simulation for robotics and autonomous systems. We further extend the family with [Cosmos-Transfer2.5], a control-net style framework for Sim2Real and Real2Real world translation. Despite being 3.5× smaller than [Cosmos-Transfer1], it delivers higher fidelity and robust long-horizon video generation. Together, these advances establish [Cosmos-Predict2.5] and [Cosmos-Transfer2.5] as versatile tools for scaling embodied intelligence. To accelerate research and deployment in Physical AI, we release source code, pretrained checkpoints, and curated benchmarks under the NVIDIA Open Model License at https://github.com/nvidia-cosmos/cosmos-predict2.5 and https://github.com/nvidia-cosmos/cosmos-transfer2.5. We hope these open resources lower the barrier to adoption and foster innovation in building the next generation of embodied intelligence.

[1] A detailed list of contributors and acknowledgments can be found in App. A of this paper.

Contents

1. Introduction

Physical AI systems—embodied agents equipped with sensors and actuators—assist humans in carrying out physical tasks by interacting with the environment: their actuators act on the world in response to sensory inputs. Training such systems directly in the real world, however, is often slow, costly, and risky. This is particularly true in the early stages, when system imperfections may lead to unsafe actions that damage either the agent, the environment, or both. A world simulator that can generate high-quality, diverse visual environments based on the actions of a Physical AI agent can serve as a safe proxy for the physical world. Such simulators enable agents to acquire perception and control skills entirely in silicon before deployment in the real world (NVIDIA, 2025). In this paper, we introduce [Cosmos-Predict2.5], our latest world foundation model based on flow matching that significantly enhances simulation fidelity across diverse Physical AI domains.

[Cosmos-Predict2.5] leapfrogs the diffusion video world model in [Cosmos-Predict1] (NVIDIA, 2025) via three key improvements. First, we strengthen the data filtering pipeline to produce higher-quality pre-training datasets and manually curate specialized post-training data tailored for Physical AI. Second, we simplify the model architecture and combine Text2World, Image2World, and Video2World capabilities into a single model. Third, we improve the training recipe, leveraging model merging and a novel reinforcement learning algorithm for post-training, and replace the T5 text encoder used in [Cosmos-Predict1] with [Cosmos-Reason1] (NVIDIA, 2025), a modern decoder-only VLM architecture specialized for Physical AI, providing richer text representations and enabling finer-grained control over world generation. Through extensive experiments, we demonstrate that [Cosmos-Predict2.5] delivers substantial gains over [Cosmos-Predict1] in both output quality and prompt alignment.

We further demonstrate that these advancements yield broad and practical benefits across diverse downstream Physical AI applications. In particular, they enable more efficient synthetic data generation for Vision-Language-Action (VLA) model training (Jang et al., 2025), a key ingredient for scaling embodied intelligence. Beyond this, [Cosmos-Predict2.5] improves action-conditioned video world generation for policy validation, enhances coherent multi-view video world generation for autonomous driving simulation, and supports camera-controllable multi-view video world generation for robotic manipulation.

Beyond these use cases, we expand [Cosmos-Predict2.5] into a broader family of control-net models, termed [Cosmos-Transfer2.5], designed for diverse visual world-translation tasks. These include enhancing the photo-realism of physical simulator outputs (NVIDIA, 2025), augmenting real-world videos (Ren et al., 2025), and converting semantic world scenarios into realistic, multi-view sensory inputs for Physical AI agents (NVIDIA, 2025). Compared to its predecessor [Cosmos-Transfer1], the new framework delivers substantially higher quality while being 3.5× smaller in size. Moreover, [Cosmos-Transfer2.5] demonstrates the ability to generate robust long-horizon video translations and enable closed-loop simulation—two essential capabilities for advancing the next stage of Physical AI research and deployment.

To further accelerate progress in this domain, we are releasing our source code, pre-trained checkpoints, and curated post-training examples to the community. By providing these open resources, we aim to lower the barrier for practitioners to adapt and specialize the pre-trained models for their own targeted Physical AI setups—whether in robotics, autonomous systems, or embodied reasoning. Tab. 1 provides a clear mapping of the released models and their corresponding capabilities, offering a practical guide for researchers and developers. We hope that by sharing these assets, we can foster broader adoption, reproducibility, and innovation in Physical AI.

2. Data

We improve upon the data pipeline in NVIDIA (2025) in two aspects. First, we upgrade the components in the filtering pipeline for general data processing. Second, we curate a set of high-quality Physical AI data to strengthen the capability of our models in this domain.

Table 1: List of released models with their corresponding capabilities and inputs.

Model Name	Capability	Input
Cosmos-Predict2.5 base		
Cosmos-Predict2.5-2B/pre-trained	pre-trained base	text + image or video
Cosmos-Predict2.5-14B/pre-trained	pre-trained base	text + image or video
Cosmos-Predict2.5-2B/post-trained	post-trained base	text + image or video
Cosmos-Predict2.5-14B/post-trained	post-trained base	text + image or video
Cosmos-Predict2.5 domain specialized		
Cosmos-Predict2.5-2B/auto/multiview	driving, 7-camera view	text + image or video
Cosmos-Predict2.5-2B/robot/multiview	robotic, 3-camera view	text + third-person video
Cosmos-Predict2.5-2B/robot/multiview-agibot	robotic, AgiBot data, 3-camera view	text + head-view video
Cosmos-Predict2.5-2B/robot/action-cond	robotic, action-conditioned	action
Cosmos-Predict2.5-2B/robot/gr00tdream-gr1	robotic, GR00T GR1 data	text + image or video
Cosmos-Transfer2.5		
Cosmos-Transfer2.5-2B	controlnet	edge, blur, segmentation, depth
Cosmos-Transfer2.5-2B/auto/multiview	driving, multiview controlnet	world scenario map

2.1. Video Curation Pipeline

Our video curation pipeline is shown in Fig. 1. It comprises seven stages: 1) shot-aware video splitting, 2) GPU-based transcoding, 3) video cropping, 4) filtering, 5) captioning, 6) semantic deduplication, and 7) sharding. Each stage is designed for high-throughput processing of enormous amounts of video data, with an emphasis on obtaining large-scale, high-quality, and semantically diverse data.

Building on the foundation established in [Cosmos-Predict1] (NVIDIA, 2025), we refine and substantially scale up our video data curation pipeline in [Cosmos-Predict2.5]. We processed over 200 million raw videos sourced from both proprietary datasets and open internet platforms. These videos cover domains such as driving, object manipulation, spatial navigation, human interaction, and nature scenes, among others, ensuring broad generalization across Physical AI use cases.

The pipeline begins by segmenting long-form videos into shots using high-accuracy boundary detection models, ensuring that raw shot transitions are excluded. Each segment is then GPU-accelerated, transcoded, and cropped to eliminate black borders and spatial padding. Very short clips under 5 seconds are discarded, while the remaining segments yield over 6 billion curated clips ranging from 5 to 60 seconds in length.

A multi-stage filtering process then removes low-quality or unsuitable data. Filters target motion artifacts, distortion, visual noise, overlay text, content that is unsuitable for training, and mismatched video types. A deduplication step is further applied to remove videos that are semantically similar. Only about 4% of the initial clips pass all filters, producing a curated dataset of approximately 200 million trainable clips. These 200 million clips form our pre-training dataset.

This multi-stage filtering pipeline comprises several key components, each serving a unique purpose. To begin with, we apply an aesthetic scoring filter, which grades the inputs by their aesthetic quality. Following this, we apply a motion filter, which quantifies and removes clips based on their degree of motion. The third stage is an OCR filter that attempts to remove clips with excessive text overlay. In the fourth stage, we apply a perceptual quality filter (akin to DOVER (Wu et al., 2023)) to weed out clips with technical distortions and perceptual artifacts. Next, we use a "semantic artifacts" filter (akin to VTSS (Wang et al., 2025)) that aims to filter out clips with semantic artifacts (video-in-video, poor transitions, *etc*.). Finally, we use a vision language model (VLM) (Bai et al., 2025) to further remove clips with a set of undesirable issues with higher precision. We apply the VLM at the very end of filtering because it is computationally more expensive. Surviving clips are subsequently categorized via a video content-type classifier, which enables structured downstream use of the dataset. At this stage, we further exclude content depicting physically unrealistic phenomena—such as video

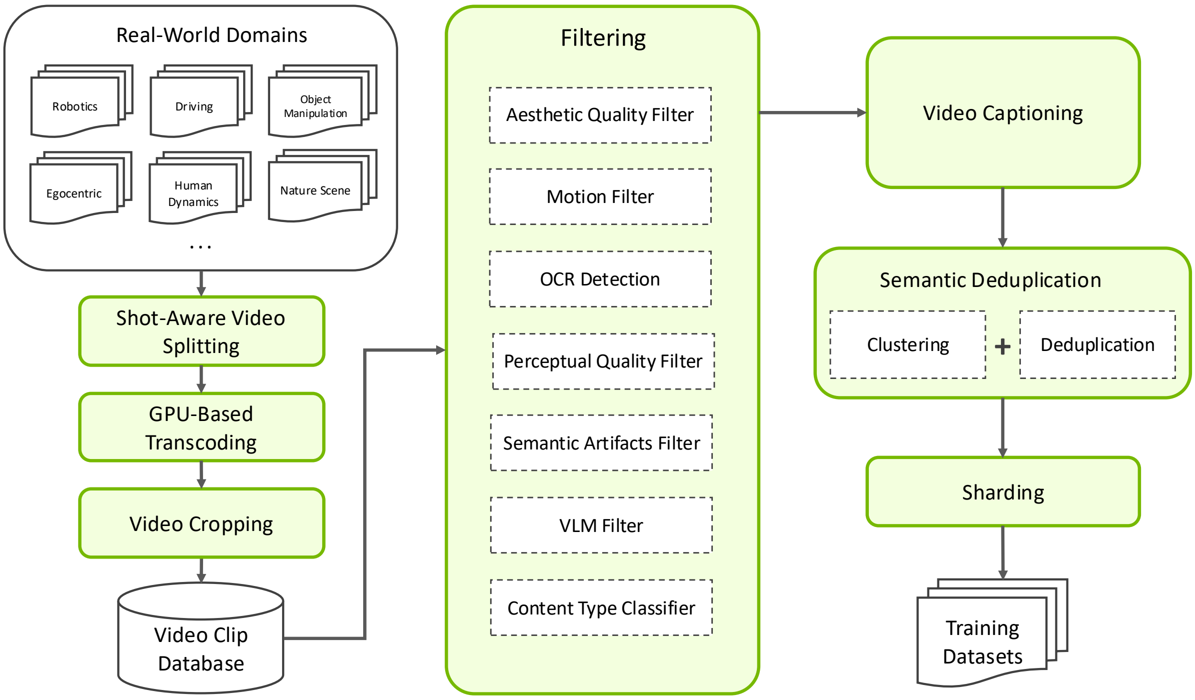


Figure 1: Our video curation pipeline transforms raw, unstructured video data from diverse real-world sources into a high-quality, annotated, deduplicated, and sharded dataset optimized for large-scale training of video world foundation models.

games, synthetic visual patterns, animations, or cartoons—to maintain alignment with the physical world distributions.

In the video captioning stage, we segment each clip into 5-second windows and caption each window using a Qwen2.5-VL-7B vision-language model (Bai et al., 2025), prompting it to generate factual, context-aware captions. We apply targeted prompt engineering to guide the model toward descriptions that emphasize the primary object, its motion, and key semantic details in the scene. Captions are produced at multiple lengths (short, medium, and long) to support diverse use cases, serving as both supervision signals and conditioning prompts for the model training.

In the semantic deduplication stage, we first assign video clips to clusters using embedding-based similarity. Within each cluster, clips are compared pairwise to detect semantically similar content, and the highest-resolution version is retained, as higher resolution preserves finer visual details and provides a richer signal for training. To support incremental and large-scale data curation, we adopt an online deduplication strategy: each new clip is compared against previously retained clips, with preference given to older and higher-resolution clips during tie-breaking. This enables scalable deduplication across growing datasets while maintaining semantic consistency across the full corpus.

To support scalable and flexible training, we implement a top-down sharding strategy. Using internally trained content type classifiers, each clip is assigned a semantic label from a custom-built taxonomy of 26 video types. The dataset is then sharded along multiple axes: content type, resolution, aspect ratio, and length. This structured sharding enables efficient sampling, curriculum-based training, and fine-grained domain balancing.

At the same time, the underlying infrastructure has been upgraded to handle petabyte-scale data processing, providing the capacity required for massive video corpora. The pipeline is built on highly parallelized workflows with dynamic auto-scaling of CPU and GPU worker allocation, ensuring workloads are efficiently balanced across heterogeneous resources. To further improve throughput, we employ video chunking and frame-rate control during inference, which reduces redundant computation while preserving semantic fidelity. Beyond ingestion and processing, the infrastructure integrates with a Delta Lake–based lakehouse (Databricks, 2019) for large-scale SQL analytics and Milvus (Zilliz, 2019), an open-source vector database for embedding-based search, enabling advanced semantic video-content similarity search and caption-level text embedding retrieval. Together, these analytical capabilities not only improve current training efficiency but also lay the foundation for large-scale dataset exploration, retrieval-augmented training, and fine-grained knowledge mining.

In contrast to the pipeline in [Cosmos-Predict1], the [Cosmos-Predict2.5] pipeline scales to a much larger volume, processing 35 million hours of raw video compared to 20 million hours, and producing over 6 billion clips from which 200 million high-quality clips are retained. At the same time, it achieves improved data quality control through a far stricter multi-stage filtering pipeline, which reduces survival from 30% of clips to only 4%. This pipeline systematically removes motion artifacts, distortions, overlay text, semantic artifacts, and other undesirable issues, with a final high-precision pass by a vision-language model. The pipeline further introduces finer content granularity by segmenting clips into shorter temporal windows, generating captions at multiple levels of detail, and structuring the dataset through semantic deduplication and sharding, resulting in richer and more precise supervision signals. These advances are supported by a more robust and scalable infrastructure, designed for petabyte-scale processing, flexible resource allocation, and advanced analytics. Together, these advances yield a dataset that is larger, cleaner, and semantically richer, underpinned by scalable infrastructure that facilitates enhanced pre-training efficiency and improved downstream generalization across diverse Physical AI domains.

2.2. Domain Specific Data

To curate high-quality data across diverse Physical AI domains, we design domain-specific pipelines that collect and annotate visual data tailored to each domain. We focus on five target domains: Robotics, Autonomous Driving, Smart Spaces, Human Dynamics, and Physics. The combined output is added to the general pre-training data.

Each domain follows a curation process similar to that used in pre-training (Fig. 1), but with two key differences in filtering and captioning. For filtering, we omit the VLM filter and instead apply a domain-specific subset of filters with adjusted hyperparameter values. For captioning, we employ a larger VLM model (Bai et al., 2025), incorporating customized prompts tailored to each domain. The following sub-sections provide detailed descriptions of the curation process for each domain.

2.2.1. Robotics

We sourced robotics datasets spanning diverse settings. For each dataset, we filtered out low-resolution and near-static videos. To ensure a consistent pace of action across the datasets, we increased the playback speed for videos featuring overly slow robotic movements. The resulting statistics are summarized in Tab. 2.

Table 2: Overview of high-quality robotics datasets with video counts by camera perspective.

Dataset	Embodiment	Central-view	Left-view	Right-view
AgiBot-Beta (Bu et al., 2025)	Bimanual	194k	30k	30k
Bridge (Walke et al., 2023)	Single-arm	36k	-	-
DROID (Khazatsky et al., 2024)	Single-arm	39k (wrist)	51k	51k
GR00T (Bjorck et al., 2025)	Bimanual	3k	-	-
1X (Technologies, 2025)	Bimanual	17k	-	-
OpenX (Vuong et al., 2023)	Single-arm	500	-	-
RoboMIND (Wu et al., 2024)	Dual-arm/Humanoid	16k	6k	7k

We design dataset-aware caption prompts that enforce task-centric, grounded descriptions while normalizing viewpoint and embodiment. Prompts require enumerating the initial scene, then describing the robot's actions chronologically with explicit motion types (e.g., linear, rotational), involved parts (arm, wrist, gripper), camera motion, and fine-grained object attributes. To improve caption accuracy and reduce hallucination, we inject available dataset-specific metadata into the prompt. For example, we include task description with human-labeled success ratings for GR00T, step-level instructions for Bridge, initial scene description for AgiBot, and unified camera perspectives across multiple dataset sources.

2.2.2. Autonomous Driving

We built a proprietary dataset consisting of approximately 3.1M 20-second surround-view video clips collected using NVIDIA's internal driving platform. Each clip includes recordings from seven synchronized cameras: front-wide, front-tele, left, right, rear, rear-left, and rear-right.

The dataset is sampled from a large-scale corpus to align with a target distribution of diverse driving attributes. The selected attributes encompass a wide range of conditions, including geographic regions (e.g., USA and Europe), traffic density (e.g., light and heavy), ego-vehicle speed (e.g., local roads and highways), ego-vehicle acceleration (e.g., constant and fast acceleration), ego-vehicle maneuvers (e.g., slow curves and sharp turns), road types (e.g., urban and rural), uncommon road structures (e.g., tunnels and tollbooths), visibility conditions (e.g., clear and foggy), weather (e.g., dry and snowy), and illumination (e.g., daytime and nighttime).

We design captioning prompts for autonomous driving by explicitly defining task requirements and emphasizing driving-relevant information. The prompts require the enumeration of:

1. various agents (e.g., vehicles, pedestrians, cyclists) and traffic elements (e.g., traffic lights, traffic signs) that the ego vehicle should be aware of for safe navigation,
2. global environmental factors (e.g., weather, time of day, road conditions) that could influence driving behavior,
3. meta actions in both longitudinal and lateral of ego vehicle and surrounding vehicles,
4. speed of ego vehicle and surrounding vehicles,
5. dynamic actions or state transitions of other objects, and
6. interactions between key objects.

To capture varying levels of detail, captions of each video are produced in three lengths: short, medium, and long.

2.2.3. Smart Spaces

We curate videos featuring scenarios that take place in warehouses, factories, construction sites, and other similar settings. We use the same pipeline for splitting these videos into individual shots as that used for the pretraining dataset. We use search keywords to find an initial set of videos that may be relevant to a smart space. For each video, we used a VLM (Bai et al., 2025) to verify its relevance. After clipping and filtering, approximately 40K video clips survive. These clips are then captioned by the VLM (Bai et al., 2025). We prompt the VLM by specifying that the videos focus on smart spaces, such as factories, warehouses, industrial facilities, automobiles, and other manufacturing environments, so that it can tailor its language and style accordingly.

2.2.4. Human Dynamics

We curated a human-dynamics video dataset by retaining clips of at least 5 seconds and processing each video with YOLOX (Ge et al., 2021) for human detection and RTMPose (Jiang et al., 2023) for full-body keypoints and facial landmark estimation. A clip is included only when people appear in more than 40% of its frames, no more than eight individuals are visible in any frame, and at least one person occupies 3% percent or more of the image area. We generated captions with the VLM using prompts that emphasize detailed descriptions of human motion and dynamics. We include this dataset to enhance the simulation capabilities of Physical AI agents, enabling them to simulate human behavior for improved action planning.

2.2.5. Physics

We curate a dataset aimed at improving the physical plausibility of generated videos by systematically emphasizing real-world dynamics. To achieve this, we first define a taxonomy of visually observable physical phenomena spanning core domains such as classical mechanics and fluid mechanics. This taxonomy provides a principled framework for identifying and categorizing key behaviors and interactions—such as shattering glass, colliding rolling balls, or flowing water. Using this structure, we curate a diverse set of videos that foreground the dynamic properties of these phenomena. In addition, we design tailored captioning prompts that guide the VLM to generate accurate, detailed descriptions of both the underlying physical processes and the associated object interactions. Together, these elements produce a dataset that is systematically organized and tightly aligned with the goal of advancing physically grounded video generation.

3. Method

In this section, we first discuss our flow-matching formulation and then present the network architecture.

3.1. Flow Matching

We adopt flow matching (FM)(Lipman et al., 2022) for training diffusion models because of its conceptual simplicity and practical effectiveness. While FM and the Elucidated Diffusion Model (EDM)(Karras et al., 2022), which was used in [Cosmos-Predict1] (NVIDIA, 2025), are mathematically equivalent in terms of their forward and backward diffusion processes, they differ in how the denoising network is parameterized (Gao et al., 2025). In EDM, the preconditioning coefficients are chosen so that both the inputs and outputs of the denoising network are approximately standardized Gaussians, which simplifies training stability. In contrast, FM selects coefficients that make the denoising network predict the velocity of the diffusion trajectory. This velocity-based formulation not only provides a more direct training target but also tends to yield smoother optimization and improved sample quality in practice.

Formally, given a data sample $\mathbf{x}$ (image or video), a noise vector $\epsilon \sim \mathcal{N}(0, I)$, and a timestep $t \in [0, 1]$ drawn from a logit-normal distribution, the interpolated latent $\mathbf{x}_t$ is defined as

$$\mathbf{x}_t = (1 - t)\mathbf{x} + t\epsilon. \tag{1}$$

The corresponding ground-truth velocity is

$$\mathbf{v}_t = \epsilon - \mathbf{x}. \tag{2}$$

The model is trained to predict $\mathbf{v}_t$ by minimizing the mean squared error (MSE) between the prediction and ground truth:

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{x},\epsilon,\mathbf{c},t} \left\| \mathbf{u}(\mathbf{x}_t, t, \mathbf{c}; \theta) - \mathbf{v}_t \right\|^2, \tag{3}$$

where $\mathbf{c}$ denotes conditioning information associated with $\mathbf{x}$ (*e.g.* text embeddings, reference frames, and other conditional inputs), θ represents the model parameters, and $\mathbf{u}(\cdot; \theta)$ is the predicted velocity function.

High-resolution content often contains significant redundancy, since nearby pixels are highly correlated. As a result, if the level of injected noise is too small, the model may fail to “break apart” this correlation, making it harder for the FM model to learn meaningful structure (Esser et al., 2024; Hoogeboom et al., 2023; Chen, 2023; Atzmon et al., 2024). To address this, we deliberately bias the training process toward higher noise levels. Specifically, we adopt the shifted logit-normal distribution (Esser et al., 2024). In practice, we first sample t from a logit-normal distribution, and then apply the monotone transformation

$$t_s = \frac{\beta t}{1 + (\beta - 1)t} \tag{4}$$

where β is a shift hyper-parameter. This transformation reweights the distribution so that t_s values are skewed

Table 3: Configuration details of [Cosmos-Predict2.5] models.

Configuration	Cosmos-Predict2.5-2B	Cosmos-Predict2.5-14B
Number of Layers	32	36
Model Dimension	2,048	5,120
FFN Hidden Dimension	8,192	20,480
AdaLN-LoRA Dimension	256	256
Number of Attention Heads	16	40
Head Dimension	128	128
MLP Activation	GELU	
Positional Embedding	3D RoPE	

toward higher noise. Intuitively, increasing β pushes the model to encounter noisier inputs more frequently, which helps it learn to reconstruct signals even when correlations are heavily disrupted. When $\beta = 1$, no shift is applied and $t_s = t$.

3.2. Network Architecture

In [Cosmos-Predict2.5], we largely reuse the denoising network $\mathbf{u}(\cdot, \theta)$ introduced in [Cosmos-Predict1]'s DiT (NVIDIA, 2025), which is based on a latent diffusion model. The main architectural change is the removal of the absolute positional embeddings and only keeping the relative positional embeddings. While absolute embeddings provide a fixed spatial or temporal reference, they limit the model's ability to generalize to resolutions or sequence lengths not seen during training. By removing them, [Cosmos-Predict2.5] gains greater flexibility for handling higher-resolution content and longer video sequences during post-training. This design choice is motivated by recent progress in long-context large language models, where alternative positional encoding strategies (Peng et al., 2023; bloc97, 2023) have proven effective at extending context length without sacrificing performance. The overall velocity prediction network design is illustrated in Fig. 2.

We adopt a different set of auxiliary models in [Cosmos-Predict2.5] compared to [Cosmos-Predict1], with improvements in both visual and textual representations. For the visual tokenizer, we use WAN2.1 VAE (Wan et al., 2025), a causal variational autoencoder that compresses video sequences with a compression rate of $4 \times 8 \times 8$ across the time, height, and width dimensions, respectively. This compression greatly reduces the computational cost while preserving essential spatiotemporal structure. On top of this representation, we apply the same $1 \times 2 \times 2$ patchification strategy to compress latent features further. We train our model to generate 93 frames, which corresponds to 24 latent frames, at a time using 16 fps videos. Each of the generated videos is about 5.8 seconds long.

For the text encoder, we leverage [Cosmos-Reason1] (NVIDIA, 2025) instead of the T5 encoder used in [Cosmos-Predict1]. Unlike standard approaches that rely on the output of a single transformer layer, we concatenate activations across multiple blocks for each token and project them into a 1024-dimensional space inspired by Wang et al. (2025). This yields a sequence of embedding vectors that more faithfully captures both local and global linguistic context. During training, these embeddings are integrated into the denoising process via cross-attention layers, enabling textual prompts to directly guide video generation. Moreover, the vision encoder in [Cosmos-Reason1] supports additional visual conditional inputs for style control, which we leave as an exciting direction for future exploration.

Each [Cosmos-Predict2.5] model is designed to operate in three modes: Text2World, Image2World, and Video2World. In the Text2World setting, generation is guided solely by a text prompt. In Image2World, the model receives both a text prompt and a reference image, allowing it to ground the generated video in specific visual content. In Video2World, the model further extends this conditioning to video sequences, enabling temporally coherent continuation or transformation of input clips.

For both Image2World and Video2World, we employ a frame-replacement strategy, where the initial frames of

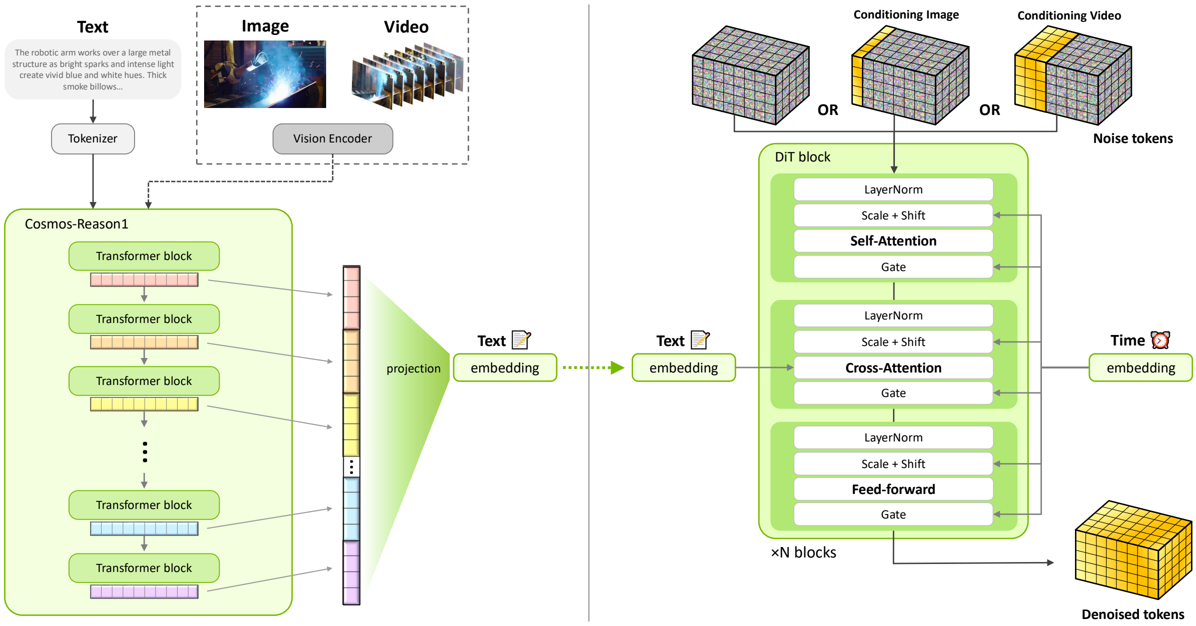


Figure 2: **Overall architecture of [Cosmos-Predict2.5]**. As shown on the right, in the latent space, the model applies repeated blocks of self-attention, cross-attention, and feed-forward MLP layers, modulated by adaptive layer normalization (scale, shift, gate) for a given time step t. We leverage [Cosmos-Reason1] as the text encoder (shown on the left). [Cosmos-Reason1] can also accommodate visual inputs (image and video) beyond text, which we leave for future work.

the generated sequence are consistently substituted with the conditioned frames. This approach serves two purposes: (1) it provides flexibility, since the number of conditioned frames can be adjusted depending on the task, and (2) it strengthens temporal consistency by ensuring that early frames remain faithful to the conditioning input. As a result, visual cues from the input image or video propagate more smoothly across subsequent frames, leading to more coherent world generation.

4. Training

We employ a progressive training strategy that balances efficiency with model quality. The process begins with multi-stage pretraining, where training difficulty is gradually increased along two axes: pixel resolution and task diversity. After pretraining, we perform supervised fine-tuning (SFT) on carefully curated, high-quality Physical AI datasets to strengthen the model's capabilities in specialized domains, before merging them into a unified model. Finally, we further enhance generation quality by applying a reinforcement learning (RL) algorithm on top of the merged model.

4.1. Pre-training

We describe the multi-stage pretraining procedure in Tab. 4. Training begins with the Text2Image task at a resolution of 256p. This stage allows the model to learn to generate high-quality individual frames before addressing motion and temporal consistency. We then introduce the Image2World and Video2World tasks to support joint image–video training. In this setting, we randomly sample either 1 or 5 conditioning frames and require the model to generate the remaining 92 or 88 frames, respectively (for a total of 93 pixel frames, corresponding to 24 latent video frames). The DiT is conditioned by concatenating ground-truth frames with noisy frames. To specify which inputs are conditional, we apply a masking scheme: each input token is formed by concatenating the original token with a mask token, where the mask serves as a binary flag indicating whether the inputs are conditional inputs. The denoising loss is applied only to the designated frames, ensuring gradients propagate correctly. After this, we progressively increase the resolution from 256p to 480p and then

Table 4: Stages of progressive pretraining and their specifications.

Task	Resolution	Number of Frames
Text2Image	256p (320×192)	1
Text2Image | Video2World	256p (320×192)	1 | 93
Text2Image | Video2World	480p (832×480)	1 | 93
Text2Image | Video2World	720p (1280×704)	1 | 93
Text2Image | Video2World | Text2World	720p (1280×704)	1 | 93 | 93

to 720p, advancing to the next stage once the model converges and visual quality plateaus. Finally, we add the Text2World task, where zero conditioning frames are provided. At this stage, we sample 0, 1, or 2 condition frames with probabilities of 0.5, 0.25, and 0.25, respectively.

We draw training timesteps from a logit-normal distribution, which, as shown in (Esser et al., 2024), places higher probability mass in the middle range of $[0, 1]$. Consistent with their approach, we apply a progressive timestep shift that grows with training resolution—starting with a shift of $\beta = 1$ at 256p and gradually increasing to $\beta = 5$ at 720p. Despite these adjustments, we observed artifacts in the generated videos, specifically abrupt and unnatural transitions between frames. We hypothesized that this instability arose because the model received too few training examples in the high-noise region, leaving it underexposed to such conditions. To address this imbalance, we modified the scheduler so that 5% of training samples are drawn explicitly from the highest 2% of the noise distribution. This targeted sampling strategy significantly reduced the transition artifacts and improved temporal consistency across generated sequences.

We train using the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We set the learning rate to 3×10^{-5} for [Cosmos-Predict2.5-2B] and 1.3×10^{-5} for [Cosmos-Predict2.5-14B]. The weight decay is set as 0.001. To stabilize optimization, we apply a linearly decaying learning rate scheduler that includes an initial warmup phase with 2000 iterations.

4.2. Post-training

4.2.1. Supervised Fine-tuning

We further conduct supervised fine-tuning (SFT) on a collection of curated, high-quality Physical AI datasets. To construct these datasets, we train a multi-head classifier on InternVideo2 embeddings (Wang et al., 2024), which enables us to categorize samples into five domains: object permanence, high motion, complex scenes, driving, and robotic manipulation. The distribution of samples across these domains is summarized in Tab. 5.

Table 5: Video statistics across different post-train domains.

Domain	Object Permanence	High Motion	Complex Scenes	Driving	Robotic Manipulation	4K
#Videos	10.4M	1.0M	1.6M	3.1M	730K	388K

We fine-tune a separate model for each domain rather than training a single model jointly across all domains. This domain-specific strategy enables us to fully leverage the available data without the need to balance mixture ratios across a combined dataset. In practice, we find that domain-specific SFT substantially improves performance on specialized domains, while causing only minimal degradation on general-domain tasks. Moreover, these slight degradations can be further mitigated through the model-merging approach described later. Each specialized model is trained for 30k iterations with a batch size of 256, using the same hyperparameter settings as the final stage of pretraining.

To evaluate these models, we construct a domain-specific test set for each category and conduct human preference studies. As shown in Fig. 3, every SFT model achieves a significantly higher win rate than the pretrained baseline on its target domain.

Figure 3: Domain-specific SFT training improves the performance of the pretrained model on each domain.

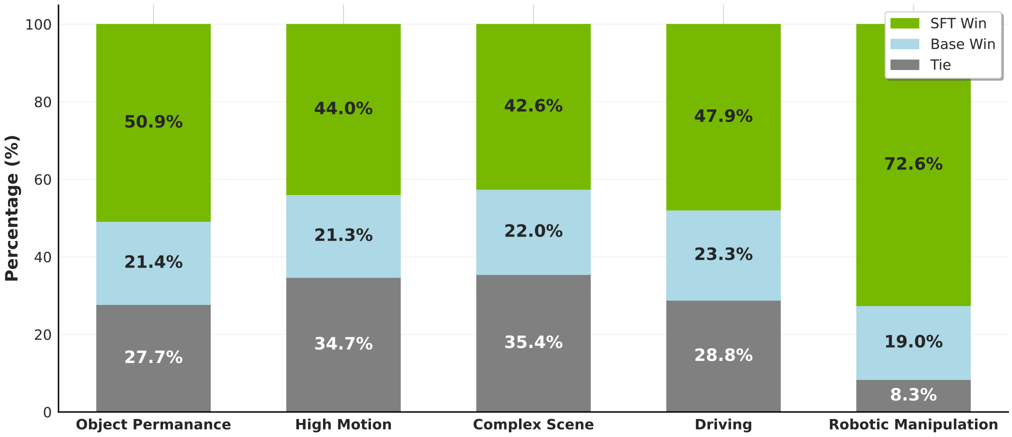
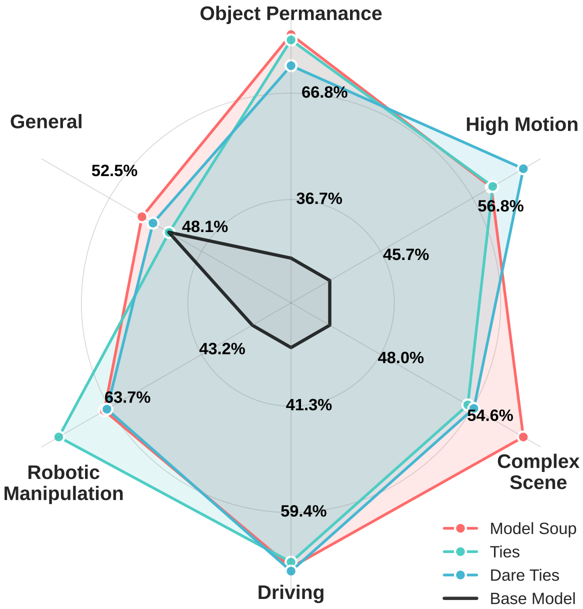


Figure 4: **Merged model gets the best of all the worlds while maintaining performance on the general domain.** Win rate for pretrained is average across three comparisons.

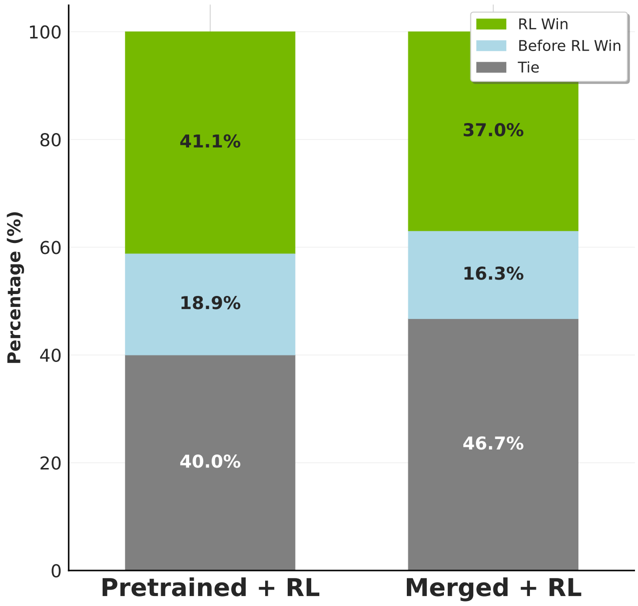


In addition, we apply a cooldown stage to the pretrained model using a curated set of high-quality 4K videos, where the learning rate is linearly decayed to zero. This step enhances fine-grained visual detail and produces smoother motion.

To unify the strengths of both the domain-specific SFT models and the cooldown model, we adopt model merging (Yang et al., 2024). We experiment with four approaches: model soup (Wortsman et al., 2022), TIES (Yadav et al., 2023), DARE-Linear (Yu et al., 2024), and DARE-TIES (Yu et al., 2024). For each method, we run hyperparameter sweeps and generate more than 20 merged models. From these candidates, we select the best-performing model based on quality assessments over a small, hand-picked set of challenging examples. We then validate the selected models on a larger evaluation set using human preference voting to ensure robust performance across both domain-specific and general tasks.

Interestingly, we find that simple grid search over hyperparameters consistently outperforms heuristic selection based on individual fine-tuned models' win rates. As illustrated in Fig. 4, all methods achieve comparable

Figure 5: Human voting shows that RL can effectively improve the quality of the generated videos.



performance with the exception of DARE-Linear. Given its effectiveness and simplicity, we select the model soup variant as our final post-trained model.

4.2.2. Reinforcement Learning

Reinforcement learning has been widely adopted in post-training to align model outputs with human preferences, either represented by human feedback (Ouyang et al., 2022) or by reward models (Schulman et al., 2017; Guo et al., 2025). For flow-based world generation, we can similarly view conditions as states and the entire denoising trajectories as actions, and leverage the reinforcement learning framework to post-train the model. Specifically, we adopt VideoAlign (Liu et al., 2025), a VLM-based reward model that evaluates text alignment, motion quality, and visual quality to post-train [Cosmos-Predict2.5-2B] (both the pre-trained and merged model). We generate eight outputs with 20 diffusion steps for each input condition and then compute the advantage of each output by normalizing the reward within its rollout group, following GRPO (Guo et al., 2025). Due to the GPU memory constraint, the probability of each trajectory is computed by decomposing it into the sum of conditional probabilities at each step, and in practice, we compute the gradient of every two conditional probabilities based on the advantages and accumulate the gradient of the probability over the entire trajectory (ten steps in total) for one parameter update. The model is trained for 256 steps with a batch size of 32. We additionally use a more fine-grained regularization beyond the KL divergence to alleviate the reward hacking phenomenon. We release the EMA weight after reinforcement learning on the merged model as our final [Cosmos-Predict2.5] post-train checkpoint.

We present the reward scores on PAI-Bench before and after RL post-training in Tab. 6. Both in Text2World and Image2World scenarios, and both for the pre-trained model and the merged model of the various SFT models, the reward increases by a large margin. We additionally conduct human voting between the videos generated by models before and after reinforcement learning, and the results are presented in Fig. 5. In all cases, videos generated by the RL models are preferred on average. In summary, reinforcement learning is proven effective in improving model quality, both in terms of the reward scores and of the human voting results.

4.3. Infrastructure

Hybrid Sharded Mode of FSDP2. We use FSDP2 as our primary distributed training framework because of its ability to shard model weights, gradients, and optimizer states while efficiently overlapping communication with

Table 6: Rewards of [Cosmos-Predict2.5-2B], before and after reinforcement learning on VideoAlign, for Text2World and Image2World settings.

Rewards Model	Text2World				Image2World			
	Text Alignment	Motion Quality	Visual Quality	Sum	Text Alignment	Motion Quality	Visual Quality	Sum
Predict2.5-2B [pre-train]	1.55	-0.43	-0.05	1.08	1.48	-0.76	-0.49	0.23
+ RL	1.69	-0.19	0.19	1.69	1.57	-0.70	-0.45	0.42
Predict2.5-2B [merged]	1.69	-0.46	-0.01	1.23	1.57	-0.82	-0.52	0.24
+ RL	1.75	-0.18	0.18	1.74	1.57	-0.68	-0.44	0.45

computation. Unlike FSDP1, which relies on a bucket-based sharding strategy, FSDP2 performs per-parameter sharding. This finer-grained design enables more efficient memory management by releasing memory promptly, thereby reducing overhead and improving utilization—an especially critical factor in video model training, where a single sequence can produce hundreds of thousands of tokens. These capabilities make FSDP2 a more scalable and flexible solution for large-scale distributed training. In addition, we incorporate several FSDP2-related optimizations from TorchTitan (Liang et al., 2025), including asynchronous distributed checkpointing and meta-device initialization, to further enhance training efficiency.

Flexible Context Parallelism. When training on high-resolution or long-duration videos, the input sequence length can easily grow to hundreds of thousands of tokens. To control per-GPU memory usage and distribute the computation of a single sample across multiple devices, we employ context parallelism. For added flexibility, we adopt the Ulysses-style parallelism approach (Rasley et al., 2020). Compared with the ring-attention strategy used in the diffusion world model of [Cosmos-Predict1], this method is both simpler and more communication-efficient, leveraging intra-node all-to-all collectives on NVIDIA GPUs. It also offers greater adaptability: for example, it better supports video post-training and diffusion distillation workloads that require advanced mechanisms such as NATTEN sparse attention (Hassani et al., 2025) and fused flash attention with Jacobian–vector product (JVP) support (Lu and Song, 2024). Achieving these capabilities with ring attention would be far more difficult while keeping computation balanced. To enable joint training across images and videos, we dynamically disable context parallelism during image iterations and re-enable it for video batches.

Selective Activation Checkpointing. To balance memory usage with computational efficiency, we apply torch Selective Activation Checkpointing (SAC) using a fine-grained policy. Lightweight operators—such as element-wise functions and normalization layers—are prioritized for recomputation, since they introduce minimal overhead while yielding significant memory savings. For large-scale video training workloads, we further extend checkpointing to portions of linear layers once all memory-intensive but computation-light operators have been covered, enabling additional reductions in memory consumption.

Elastic Reward Service. To handle a large amount of input and different reward models in the RL post-training, we rely on an efficient and flexible external service. The service supports VideoAlign and other reward functions, and can be dynamically scaled up or down according to the input traffic. Decoded latent is used to send the video for evaluation, enabling data compression during transfer. The service is pipelined in a producer-consumer fashion: a decode stage decodes the video from the received latent, while several reward models compute different rewards simultaneously in the inference stage. The decode and inference stages process different videos in a pipeline to fully utilize compute capacity. Each stage runs in a separate process to satisfy different environment requirements and support scalability. Data sharing between stages is achieved via CUDA inter-process communication (IPC) in a zero-copy manner, further enhancing efficiency. The reward calculation is handled in an asynchronous way. A task UUID is returned immediately after the video with its desired reward types is enqueued. A Redis server stores the computed rewards, which can be retrieved later using the UUID. Each task also supports batch processing of multiple videos. Between the interval of enqueue and result fetching, other actions can proceed asynchronously to maximize the system utilization.

Table 7: Training efficiency with 4096 NVIDIA H100 GPUs where the video resolution is 720p and number of frames is 93.

Model	Context Parallelism Size	MFU
Cosmos-Predict2.5-2B	2	36.49%
Cosmos-Predict2.5-14B	8	33.08%

Tab. 7 shows the Model Flops Utilization (MFU) of our video model training infrastructure. For [Cosmos-Predict2.5-2B], the MFU is 36.49%. For [Cosmos-Predict2.5-14B], the MFU drops to 33.08%. The drop is due to large context parallelism, which introduces more communication cost.

5. Results

Benchmarking. We report the performance of [Cosmos-Predict2.5-2B] models on PAI-Bench (Zhou et al., 2025), a recently proposed benchmark designed to assess physical AI generation and understanding capabilities. The [Cosmos-Predict2.5-14B] model is still undergoing post-training. The results will be updated in the updated version of the report.

We evaluate on PAI-Bench's predict task and report two main scores: the *Domain Score*, which measures performance on domain-specific physical AI tasks, and the *Quality Score*, which reflects the quality of generated videos. The *Quality Score* is derived from eight text-to-video and image-to-video metrics adapted from VBench. In contrast, the *Domain Score* is obtained through VQA-based evaluation across seven domains: av, common, human, industry, misc, physics, and robotics. The final *PAI-Bench Overall Score* is computed as the average of the Quality and Domain scores.

The PAI-Bench T2W and I2W quantitative results are shown in Tab. 8 and Tab. 9, respectively. The [Cosmos-Predict2.5-2B] post-trained model performs similarly to the larger Wan2.2-5B model in T2W, and is the best-performing model in I2W.

Figure 6: Despite being of smaller size, post-trained [Cosmos-Predict2.5-2B] is on par with Wan2.2-5B and Wan2.1-14B on a diverse set of prompts.

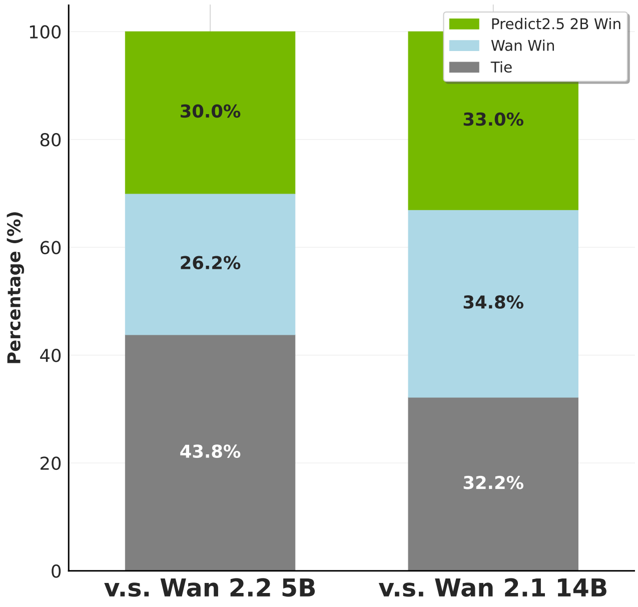


Table 8: Results on PAI-Bench-Predict-Text2World Benchmark.

Model	Domain Score	Quality Score	Overall Score
Cosmos-Predict2.5-2B [pre-train]	0.782	0.720	0.751
Cosmos-Predict2.5-2B [post-train]	0.804	**0.732**	0.768
Cosmos-Predict2.5-14B [pre-train]	0.791	0.722	0.757
Cosmos-Predict2.5-14B [post-train]		coming soon	
Wan2.1-1.3B	0.786	0.726	0.756
Wan2.1-14B	0.794	0.727	0.761
Wan2.2-5B	0.797	0.730	0.764
Wan2.2-A14B	**0.810**	0.728	**0.769**

Table 9: Results on PAI-Bench-Predict-Image2World benchmark.

Model	Domain Score	Quality Score	Overall Score
Cosmos-Predict2.5-2B [pre-train]	0.824	0.775	0.799
Cosmos-Predict2.5-2B [post-train]	0.840	**0.779**	**0.810**
Cosmos-Predict2.5-14B [pre-train]	0.835	0.777	0.806
Cosmos-Predict2.5-14B [post-train]		coming soon	
Wan2.1-14B	0.827	0.768	0.797
Wan2.2-5B	0.834	0.774	0.804
Wan2.2-A14B	**0.841**	0.772	0.806

Human Evaluation. Alongside automated metrics, we include human evaluation to capture aspects of video quality that are difficult to quantify and that better reflect human preference. Annotators compare pairs of generated videos, assessing criteria such as realism, visual quality, temporal consistency, and alignment with conditioning inputs. In Fig. 6, results are summarized using win ratios, defined as the proportion of comparisons in which a model's output is preferred over a baseline. Despite being 60.0% and 85.7% smaller compared to Wan 2.2 5B and Wan 2.1 14B, human voting shows that [Cosmos-Predict2.5-2B] is comparable with them on PAI-Bench I2W and T2W settings.

Qualitative Examples. Evaluation of generative video models requires both quantitative and qualitative perspectives. Automated benchmarks and human evaluation yield measurable results, but qualitative inspection reveals model behaviors that are difficult to capture numerically. We present high-quality sample videos generated by [Cosmos-Predict2.5-2B], focusing on physical AI. These examples complement benchmark results by illustrating the model's ability to generate realistic, high-quality, and physically coherent world simulations.

We show visual samples in Fig. 7 as representative examples of physical AI scenarios. The [Cosmos-Predict2.5-2B] post-trained model is able to simulate accurate behaviors in driving, generate realistic industrial and robotics scenes, and produce physically coherent motion.

6. Applications

We demonstrate the versatility of [Cosmos-Predict2.5] across multiple Physical AI applications. First, we introduce [Cosmos-Transfer2.5], which provides control-net style generation capability to Physical AI applications (Sec. 6.1). Compared to [Cosmos-Transfer1], the new model is substantially more effective while being 3.5× smaller. We further show that [Cosmos-Transfer2.5] enables Real2Real augmentation for policy learning (Sec. 6.2), and that the same paradigm applies to autonomous driving, where we construct multiview world models conditioned on world scenario maps for realistic driving simulation (Sec. 6.3).

We also extend [Cosmos-Predict2.5] to support camera-pose–controllable multiview generation (Sec. 6.4) and

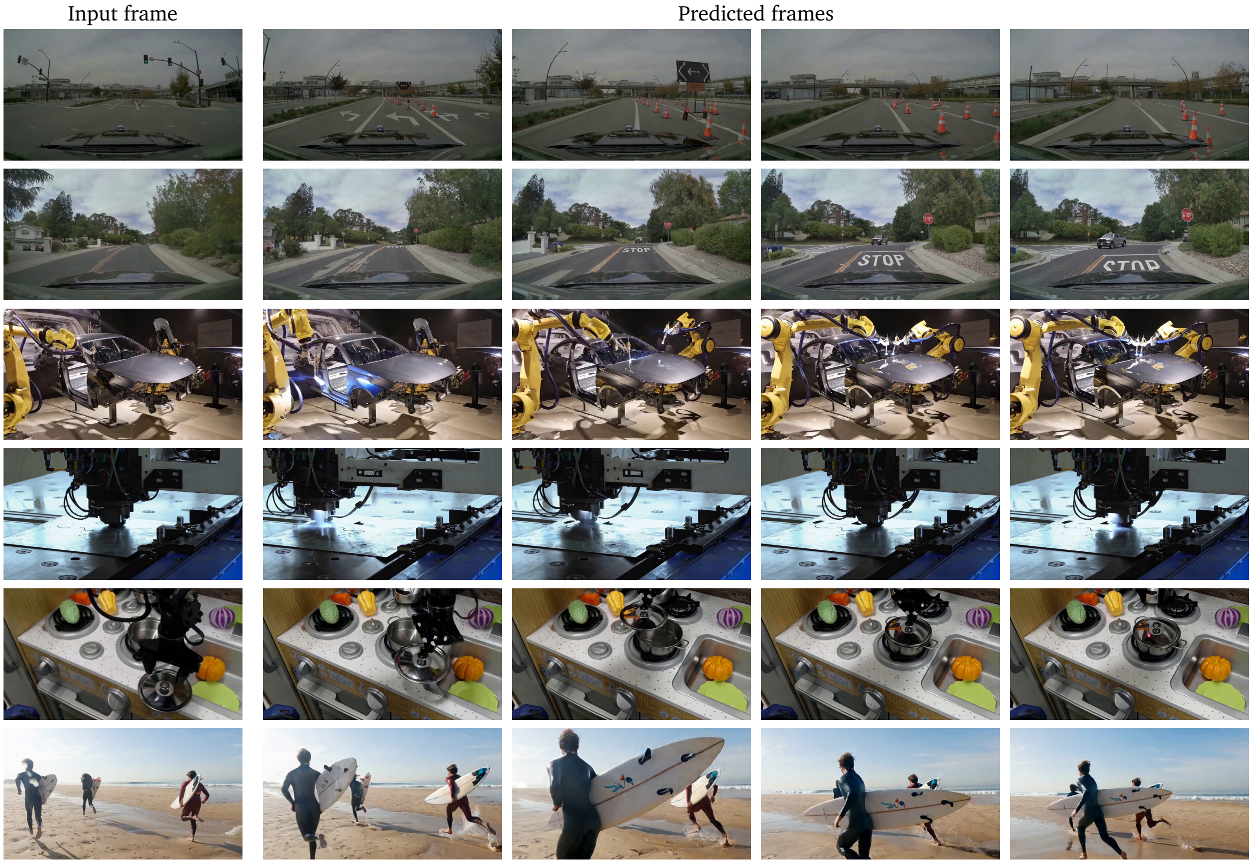


Figure 7: [Cosmos-Predict2.5-2B] post-trained prediction samples on the PAI-Bench dataset.

apply it to synthetic data generation for VLA training (Sec. 6.5). Finally, in Sec. 6.6, we post-train [Cosmos-Predict2.5] into an action-conditioned world model that is particularly well-suited for policy evaluation.

6.1. Cosmos-Transfer2.5

We develop a conditional world generation model, [Cosmos-Transfer2.5-2B], built on top of [Cosmos-Predict2.5-2B], that produces high-quality world simulations conditioned on *multiple spatial control inputs*. These inputs can take different modalities—including edges, blurred video, segmentation maps, and depth maps—and may originate from either a physics simulation engine, such as NVIDIA IsaacSim, or from real-world video data.

In terms of architecture, [Cosmos-Transfer2.5-2B] follows the general design of [Cosmos-Transfer1-7B] (NVIDIA, 2025), but with a key modification. Whereas [Cosmos-Transfer1-7B] inserts four control blocks sequentially at the start of the main branch, [Cosmos-Transfer2.5-2B] distributes its four control blocks more evenly by inserting one after every seven blocks in the main branch. This design preserves the total number of control blocks while integrating conditioning information more gradually throughout the network. For additional architectural details, please refer to [Cosmos-Transfer1](NVIDIA, 2025).

To train [Cosmos-Transfer2.5-2B], we curate high-quality, control-condition data from our pre-training video dataset, with a particular emphasis on Physics AI domains such as autonomous driving, robotics, smart spaces, and physics. World generations in these domains require precise spatial and temporal understanding, making them ideal for testing the effectiveness of different control modalities.

Depth information is crucial for capturing geometric structure and 3D reasoning. We use Video Depth Anything (Chen et al., 2025) to generate depth maps for 10 million videos for depth conditioning. Semantic segmentation provides fine-grained object-level and region-level cues that are essential for tasks like robotics and scene interaction. We apply SAMv2 (Ravi et al., 2024) on 3 million videos for segmentation conditioning. In addition, following the pipeline of [Cosmos-Transfer1-7B] (NVIDIA, 2025), we curate 14 million videos with edge and blur conditions. Edge maps highlight object boundaries that aid perception, while blurred videos serve as a robust training signal, forcing the model to recover sharp details.

Each control branch corresponding to a modality is trained independently for 100,000 iterations with an effective batch size of 64, allowing the model to specialize in extracting useful representations from each type of input before integration. For all other hyperparameters, we adopt the same settings as those used in [Cosmos-Predict2.5-2B], ensuring consistency across models.

Table 10: **Quantitative evaluation on transfer models for various configurations.** We compare single control models (each conditioned on a single modality) and multi-modal variants that use spatially uniform weights. For the multi-modal cases, “Uniform Weights” denotes the full model that integrates all four control modalities (each weighted at 0.25). Best results are in bold; second-best are underlined.

Model	Blur Alignment	Edge Alignment	Depth Alignment	Segmentation Alignment	Overall Quality
	Blur SSIM ↑	Edge F1 ↑	Depth si-RMSE ↓	Mask mIoU ↑	Quality Score ↑
Cosmos-Transfer1-7B [Blur]	0.89	0.20	0.66	0.73	6.56
Cosmos-Transfer1-7B [Edge]	0.77	0.38	0.85	0.73	6.76
Cosmos-Transfer1-7B [Depth]	0.67	0.15	0.76	0.71	6.89
Cosmos-Transfer1-7B [Seg]	0.62	0.11	1.13	0.70	6.02
Cosmos-Transfer1-7B Uniform Weights	0.82	0.26	0.70	0.74	9.24
Cosmos-Transfer2.5-2B [Blur]	**0.91**	0.27	**0.56**	0.76	**9.41**
Cosmos-Transfer2.5-2B [Edge]	0.79	**0.49**	0.76	0.75	8.73
Cosmos-Transfer2.5-2B [Depth]	0.71	0.19	0.70	0.73	8.85
Cosmos-Transfer2.5-2B [Seg]	0.68	0.14	1.02	0.71	8.81
Cosmos-Transfer2.5-2B Uniform Weights	0.87	0.41	0.67	**0.76**	9.31

6.1.1. Results

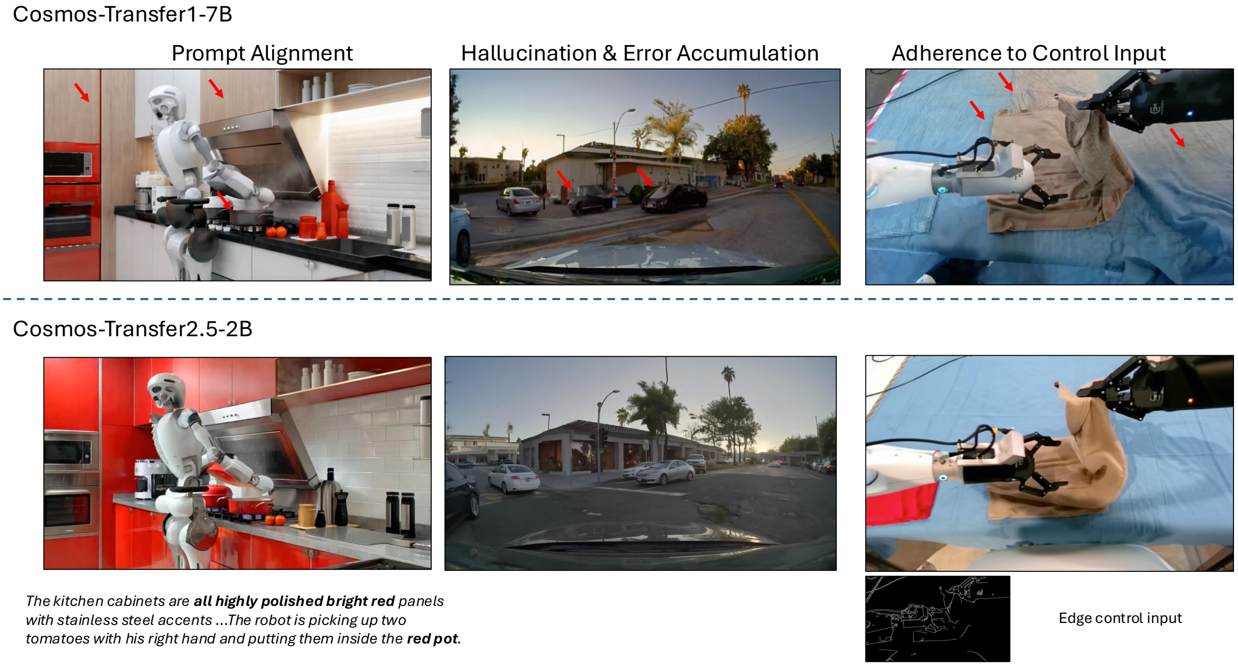
For evaluation, we use PAIBench-Transfer (Zhou et al., 2025), a benchmark dataset containing 600 videos spanning diverse domains such as driving and robotics. The evaluation is structured around two key dimensions: adherence to control inputs (how well the generated video follows the provided conditions) and overall video quality (measuring realism and consistency). The quantitative results are summarized in Tab. 10.

As shown in the table, [Cosmos-Transfer2.5-2B] outperforms [Cosmos-Transfer1-7B] on both metrics, despite being 3.5 times smaller in size. This improvement can be attributed to two factors: (1) stronger [Cosmos-Predict2.5-2B] as the base model, and (2) the use of more carefully curated, Physics-AI-focused training data, which better aligns with the benchmark domains. Visual comparisons highlighting these gains are provided in Fig. 8.

6.1.2. Long Video Generation

In addition, we introduce a new metric designed to evaluate error accumulation in long-video generation. Since DiT-based video generation models are constrained by limited context length, they typically generate long videos autoregressively, producing one chunk at a time. This chunked generation process inevitably leads to error accumulation, where artifacts and inconsistencies increase as the video length grows

To study this effect, we curate a set of 17 evaluation videos ranging from 30 to 120 seconds in length. We then propose the averaged Relative Normalized Dover Score (RNDS) as a quantitative measure of how video quality



*The kitchen cabinets are **all highly polished bright red** panels with stainless steel accents ...The robot is picking up two tomatoes with his right hand and putting them inside the **red pot**.*

Figure 8: **Sample comparison results of [Cosmos-Transfer2.5-2B]**. Compared to [Cosmos-Transfer1-7B], [Cosmos-Transfer2.5-2B] has better prompt alignment, better adherence to control input, and less hallucination and error accumulation (especially for long videos).

degrades across chunks. RNDS is defined as a curve over chunk indices:

$$\mathrm{RNDS}[i] = \left(\frac{\mathrm{DOVER}[i]}{\mathrm{DOVER}_{\mathrm{GT}}[i]} \right) / \left(\frac{\mathrm{DOVER}[1]}{\mathrm{DOVER}_{\mathrm{GT}}[1]} \right), \tag{5}$$

where $i = 1, \ldots, T$ denotes the chunk index, $\mathrm{DOVER}[i]$ is the Dover score (Wu et al., 2023) of the i-th generated chunk, and $\mathrm{DOVER}_{\mathrm{GT}}[i]$ is the corresponding Dover score for the ground-truth video. This normalization ensures that the RNDS curve always starts at $(1, 1)$, making it easy to compare degradation trends across models. The averaged RNDS is then obtained by averaging curves over all evaluation videos.

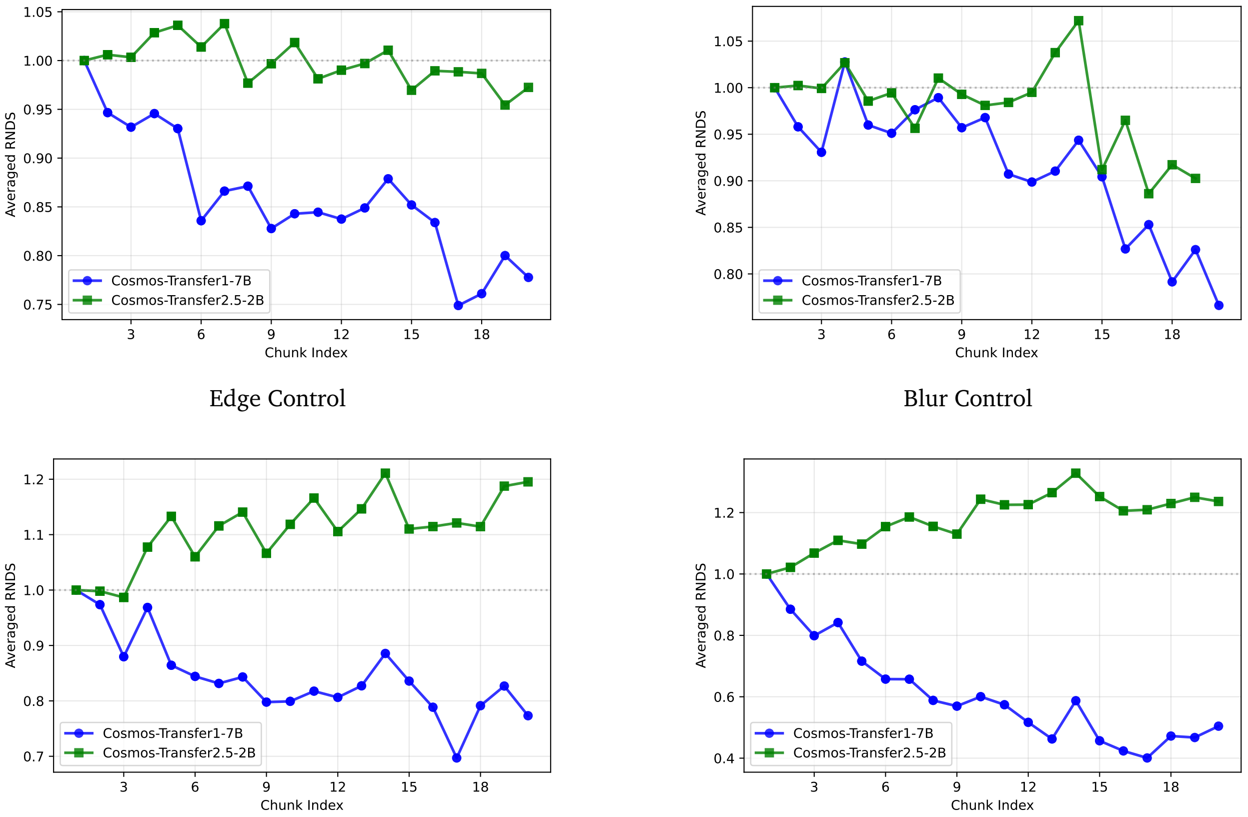
As shown in Fig. 9, the RNDS curves reveal that [Cosmos-Transfer2.5-2B] exhibits far less reduction in RNDS over time compared to [Cosmos-Transfer1-7B]. This indicates that our smaller model accumulates fewer errors, demonstrates less hallucination, and maintains higher fidelity over long video sequences.

6.2. Cosmos-Transfer2.5 for Robot Policy Learning

We aim to investigate the following question: Can [Cosmos-Transfer2.5-2B] be used as a visual synthetic data generator to augment robot policy training and enable generalization to unseen visual scenarios?

Our setup follows a standard real-world imitation learning pipeline. Using a bimanual robot equipped with an egocentric camera, we first collect human teleoperation demonstrations for table-top manipulation tasks. From these demonstrations, we train a vision-based policy that maps image observations and proprioception to action chunks using state-of-the-art behavioral cloning techniques. The trained policy is then deployed back on the same platform for evaluation.

Unlike conventional imitation learning benchmarks, however, we introduce adversarial visual perturbations during evaluation—for example, modifying object appearances, changing scene backgrounds, or placing distractor objects on the table. This setting reflects a realistic deployment scenario, where a policy must operate



Edge Control

Blur Control

Depth Control

Segmentation Control

Figure 9: **Error accumulation for long video generations.** These plots show the Normalized Relative Dover Score vs Chunk Index for auto-regressive multi-trunk long video generation where each trunk is 93 frames. As shown, for all four control modalities (edge/blur/depth/seg), compared to [Cosmos-Transfer1-7B] (blue curves), [Cosmos-Transfer2.5-2B] (green curves) has much less reduction in RNDS along the chunk index dimension, which shows less hallucination and error accumulation for long videos.

in environments that differ drastically from the conditions in which the demonstrations were collected. Such domain shifts often involve structured visual changes that cannot be easily synthesized using standard image augmentation methods.

Here, [Cosmos-Transfer2.5-2B] offers a unique advantage: it not only enables the generation of these structured variations for visual data augmentation, but also provides controllability through text prompts that specify the desired visual conditions. This enables the systematic simulation of challenging out-of-domain scenarios and the testing of policy robustness in a controlled yet flexible manner.

6.2.1. System and Task Settings

We conduct our experiments on a semi-humanoid robotic platform equipped with two 7-DoF Kinova Gen3 arms, each fitted with a Robotiq 2F-140 gripper. An Intel RealSense D455 camera is mounted on the robot's head to capture egocentric image observations. The robot's base is fixed in front of a table and remains stationary throughout all experiments to ensure consistency.

For teleoperation, we use Meta Quest 2 controllers to track the 6D target poses of the left and right end effectors. These 6D poses are converted into target joint positions and velocities via a GPU-accelerated model predictive control (MPC) framework from cuRobo (Sundaralingam et al., 2023). The resulting commands are then executed by the robot's low-level joint impedance controller, enabling smooth and responsive teleoperation.

The demonstration task is a bimanual pick-and-place scenario involving two objects: an apple and a bowl, placed randomly on the table at the start of each trial. The task requires the robot to grasp the apple and the bowl with separate arms, place the apple into the bowl while holding it, and finally set the bowl back on the table as if serving. Across trials, only the positions of the apple and bowl are varied, while the objects themselves (a gray apple and bowl), the table surface, and the background remain fixed.

In total, we collect 100 human teleoperation demonstrations of this task. Using these demonstrations, we train a UNet-based Diffusion Policy (Ren et al., 2025; Chi et al., 2023), which takes the single-image observation (processed via a small ViT) with gripper joint state and predicts chunks of actions consisting of the target end-effector poses and gripper commands for both arms. Each chunk spans a horizon of 8 timesteps sampled at 10 FPS. Examples of egocentric image observations recorded during the demonstrations are shown in Fig. 10, illustrating the consistency of the setup and the controlled variability introduced by object placement.

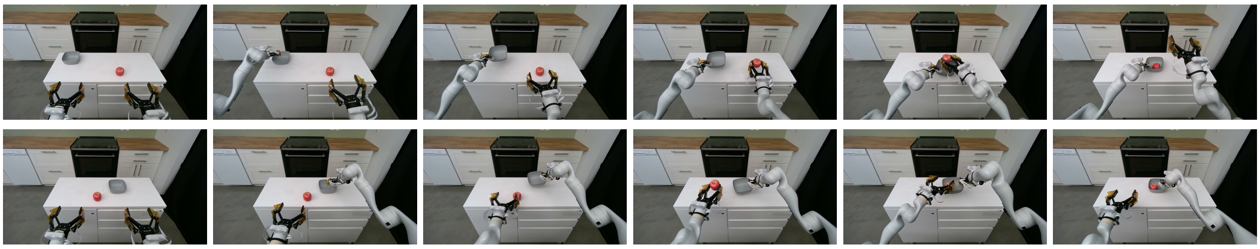


Figure 10: **Real-Robot Teleoperation Samples.** Two episodes of image observations captured from the egocentric camera during demonstration collection. We keep the object instances and scene fixed and only change the objects' (apple and bowl) poses.

6.2.2. Data Augmentation Strategy

We use [Cosmos-Transfer2.5-2B] to generate diverse and realistic visually augmented videos that expand the training set, improving the diffusion policy's ability to generalize to test-time variations. Our augmentation strategy applies global edge control across the entire image, while restricting blur control to robot pixels. To isolate the robot in each frame, we combine Grounding DINO and SAMv2 (Liu et al., 2023; Ravi et al., 2024; Ren et al., 2024) for detection and pixel-level segmentation. We set the edge threshold to medium, the blur threshold to very low, and the classifier-free guidance scale to 3, while keeping all other parameters at their default values.

We design a prompt template that diversifies the appearance of synthetic videos while preserving the underlying scene and task structure. The process begins by providing an example video to a VLM, which generates a detailed caption of the scene. We then iteratively refine this caption by prompting [Cosmos-Transfer2.5-2B] and checking whether the generated video faithfully resembles the original.

From the refined caption, we construct a formatted prompt that marks which components can vary. An LLM is then used to generate candidate variations for these components. Below is the full formatted prompt:

The scene depicts a bright, modern kitchen with plenty of ambient light. From a first-person perspective, a robot faces [TABLE]. On the table rest [COLOR_APPLE] apple and [COLOR_BOWL] bowl. [SENTENCE_LIGHT] In the background are a black cooking range featuring a black stovetop, wooden countertops, and cabinetry with white doors and drawers, including a built-in white dishwasher on the left. [SENTENCE_BACKGROUND] A wide black curtain hangs vertically on the right side, covering a large portion of the space. As the video progresses, the robot picks up the apple, then the bowl, places the apple into the bowl, and sets the bowl down on the table.

In Fig. 11 (bottom two rows), we present a few examples of diverse and realistic synthetic videos used for visual augmentation. These examples illustrate variations in apple and bowl colors, table appearances with realistic textures, as well as diverse lighting conditions, object shadows, and background changes. For each of

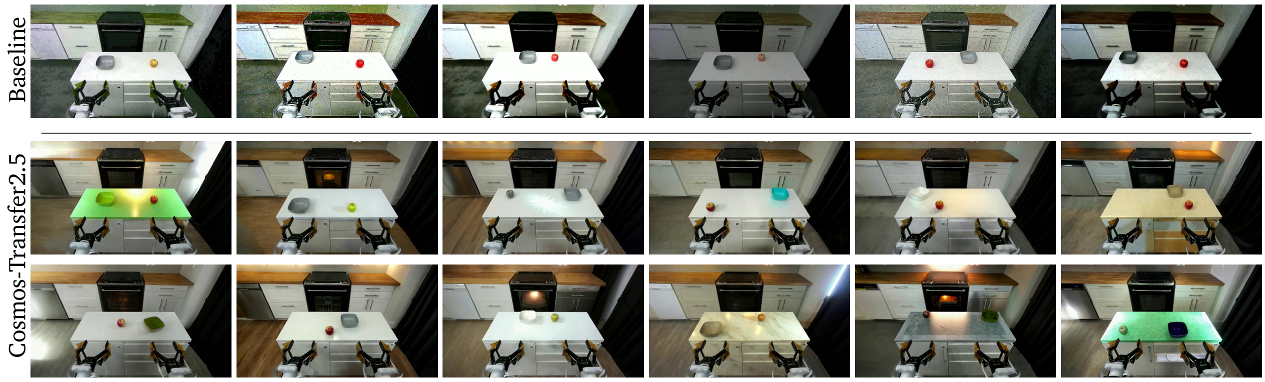


Figure 11: **Real-Robot Data Augmentation Gallery.** We show the baseline (top row) and [Cosmos-Transfer2.5-2B] (bottom two rows) data augmentation samples.

the 100 original demonstration videos, we randomly generate five synthetic variants for augmentation. The rest of the training data (i.e., actions and joint states) remain unchanged, while only the input images are augmented.

6.2.3. Experiments and Results

We perform real-robot experiments under varied test-time object and environment conditions. Beyond the base setting, which mirrors the training configuration, we evaluate nine novel scenarios: (1) replacing the apple with a purple mangosteen, (2) replacing the gray bowl with an orange bowl, (3) placing a beige tablecloth, (4) placing a black tablecloth, (5) adding a spotlight to the robot's left, (6) adding distractor objects on the table, (7) changing the left-side background cabinet to black, (8) opening the background drawers and oven door, and (9) a challenging combination of the first three modifications. Notably, while the first five variations may fall within the range of our diverse prompt augmentations, the subsequent three represent clear out-of-distribution shifts, and the final composite condition poses an especially challenging test. See Fig. 12 (leftmost column) for an overview of all ten test settings.

Table 11: **Real-Robot Quantitative Evaluation.** We test the base, baseline, and proposed (a policy trained with [Cosmos-Transfer2.5-2B] augmented observations) on 10 test scenarios.

	Base	Mangosteen	Orange Bowl	Beige Table	Black Table	Light On	Distractors	Black Cabinet	Open Drawers	Combo	**Total**
Base	1/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	1/30
Baseline	**3/3**	0/3	2/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	5/30
Proposed	**3/3**	**3/3**	**3/3**	**1/3**	**1/3**	**2/3**	**3/3**	**2/3**	**3/3**	**3/3**	**24/30**

We compare our trained [Cosmos-Transfer2.5-2B] diffusion policy against two policies:

1. a base policy trained solely on 100 teleoperation videos, and
2. a baseline policy trained with standard image-based data augmentation techniques (*e.g.*, random adjustments of brightness, contrast, saturation, and hue; gamma correction; salt-and-pepper noise; histogram equalization; random blurring or sharpening).

For the baseline policy, augmentations are applied on-the-fly during training to maximize input diversity. Example augmented images are shown in Fig. 11 (top row). While standard image-based augmentation is a commonly used technique to improve test-time visual robustness, it cannot perform semantic edits such as changing object colors, environment appearances, or lighting conditions, which [Cosmos-Transfer2.5-2B] can naturally address.

Tab. 11 summarizes our policy performance against the base and baseline policies. For each test scenario, we perform three trials and fix the object pose and environment configuration to ensure fair comparisons.

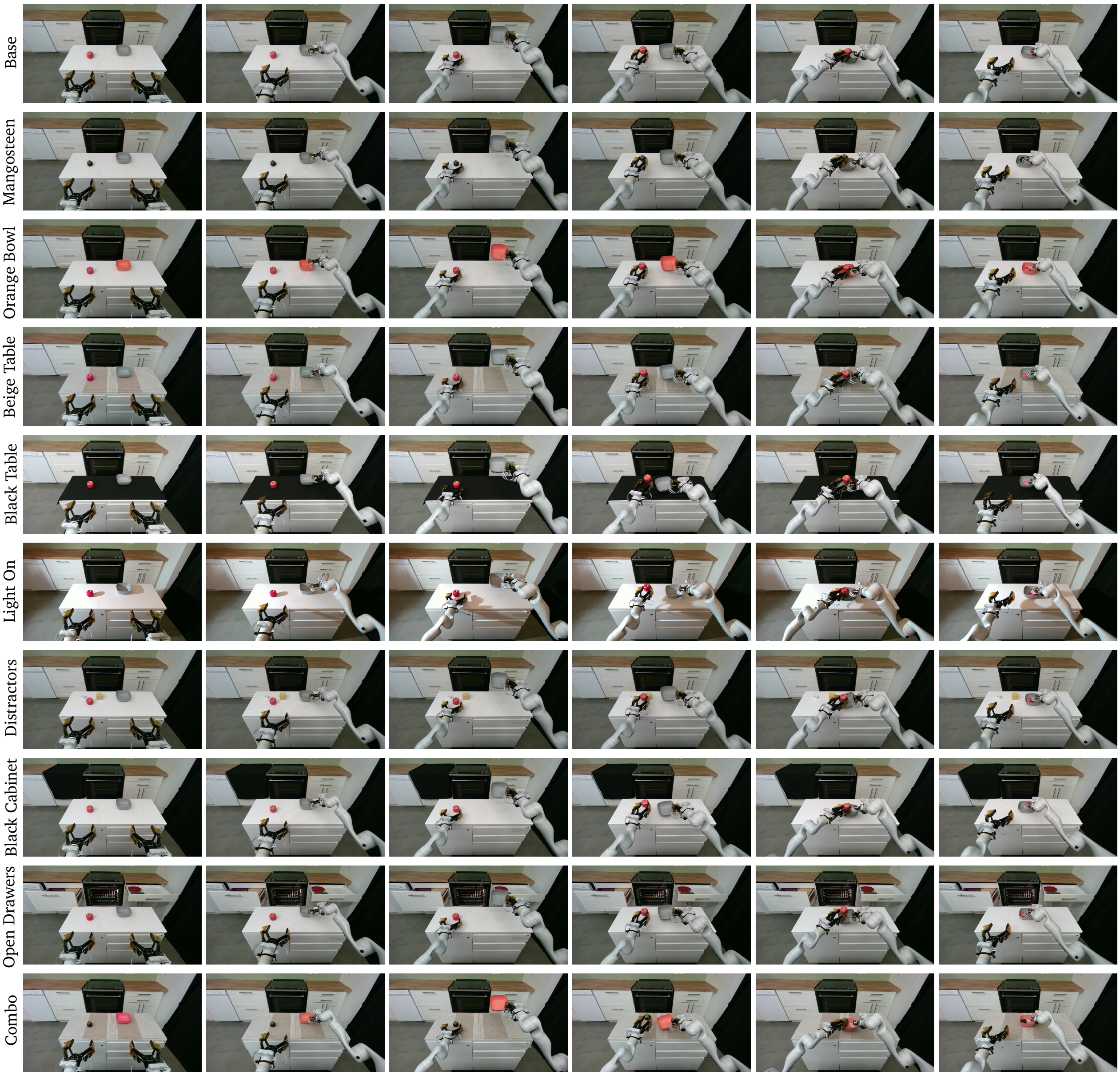


Figure 12: **[Cosmos-Transfer2.5] Real-Robot Policy Rollouts.** We present sample [Cosmos-Transfer2.5-2B] policy rollouts under the base setting and nine unseen test-time scenarios.

The [Cosmos-Transfer2.5-2B]-augmented policy achieves 24 successes out of 30 trials, clearly outperforming both baselines. It demonstrates markedly higher robustness and generalization to novel test-time object and environment changes.

The base policy, trained only on the base setting, fails to generalize to novel settings and performs poorly even on the base setting due to subtle, human-imperceptible scene variations. The baseline policy, using standard image augmentations, succeeds in just one case, highlighting the limitations of basic transformations for these challenging scenarios.

Fig. 12 visualizes successful rollouts across all ten test cases. Despite occasional failures (*e.g.*, imprecise grasps), the results indicate that [Cosmos-Transfer2.5] provides a promising, lightweight, and effective pipeline for synthetic data generation in robotics.

6.3. Cosmos-Transfer2.5 for Driving Simulation

We extend [Cosmos-Predict2.5-2B] from single-view to multi-view world generation, resulting in [Cosmos-Predict2.5-2B/auto/multiview]. In addition, just as we extended [Cosmos-Predict2.5-2B] into [Cosmos-Transfer2.5-2B] by adding a control branch, we similarly augment the multi-view version. This yields [Cosmos-Transfer2.5-2B/auto/multiview], a conditional, multi-view world generation model capable of generating consistent scenes across multiple viewpoints.

6.3.1. Model Architecture

To generate multiple 720p views, we re-purpose the latent temporal dimension by concatenating multiple views along it, effectively treating views as sequential frames. To remain within memory limits while still benefiting from FSDP and context parallelism, we reduce the latent temporal dimension to 8, which allows us to fit up to 7 views simultaneously.

Each view is encoded (and decoded) independently by the tokenizer. Once encoded, we concatenate in the latent channel dimension a compact per-view learnt embedding (of size 7) before passing it through the DiT network. We apply 3D-factorized Rotary Position Embeddings (RoPE) and cross-attention with text embeddings, following the same design as in [Cosmos-Predict2.5-2B]. Although we concatenate the views in the latent temporal dimension, we construct the RoPE embeddings separately per view. Each view can also be conditioned by one or more frames, and in the case of [Cosmos-Transfer2.5-2B/auto/multiview], each view can additionally be controlled by a separate control signal, as shown in Fig. 13.

Table 12: Evaluation of visual metrics of on generated multi-view videos from RDS-HQ-HL dataset (Ren et al., 2025). We use FVD StyleGAN, FVD I3D, and FID for visual quality (Skorokhodov et al., 2021) and TSE and CSE (Sampson, 1982) for multi-view consistency.

Model	FVD StyleGAN ↓	FVD I3D ↓	FID ↓	TSE ↓	CSE ↓
Predict2.5-2B/auto/mv	**23.060**	**25.308**	**12.095**	0.948	**1.903**
Predict1-7B-Sample-AV	63.685	69.613	25.341	0.930	2.631
Transfer2.5-2B/auto/multiview	**24.222**	**25.692**	**20.022**	1.246	2.310
Transfer1-7B-Sample-AV	56.606	60.660	22.633	1.017	1.835
Real Videos (Reference)	-	-	-	1.193	1.832

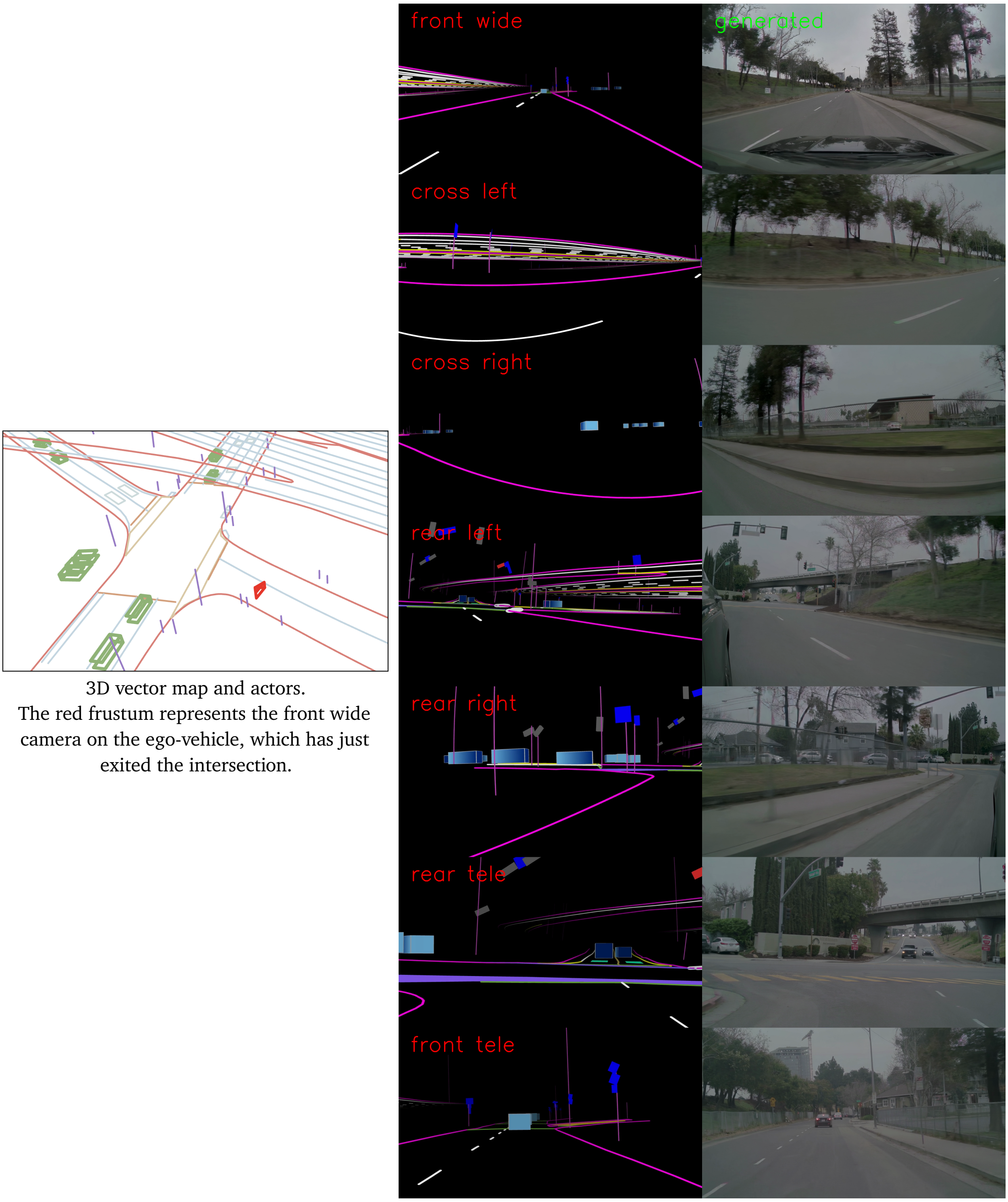
Table 13: Evaluation of lane and bounding box detection on multi-view generated videos from RDS-HQ-HL dataset (Ren et al., 2025). We use LET-AP/APL/APH for cuboid metrics (Hung et al., 2024), and F1, x-coordinate rMSE and accuracy for detection, regression and classification scores of lane detection.

Model	Cuboids			Lanes		
	LET-AP ↑	LET-APL ↑	LET-APH ↑	F1 ↑	x-error (far) ↓	Category Acc. ↑
Transfer2.5-2B/auto/multiview	**0.394**	**0.254**	**0.383**	**0.637**	**0.487**	**0.904**
Transfer1-7B-Sample-AV	0.243	0.154	0.236	0.604	0.524	0.899
Real Videos (Reference)	0.476	0.319	0.462	0.637	0.480	0.905

6.3.2. Training Datasets

For [Cosmos-Predict2.5-2B/auto/multiview], we curate a multi-view captioned dataset of 1.5 million clips, each containing 20-second-long scenarios with 7 synchronized cameras recording at 30FPS (front-wide, front-tele, front-left, front-right, rear-left, rear-right, rear-tele). To facilitate training with text conditioning, we generate captions at 150-frame intervals using Qwen2.5-7B-Instruct with three different lengths (short, medium, and long).

For [Cosmos-Transfer2.5-2B/auto/multiview], we project HD maps and dynamic objects in the scene onto the seven camera views as the control input, and we name it “world scenario map” Fig. 13. The world scenario map



3D vector map and actors.
The red frustum represents the front wide camera on the ego-vehicle, which has just exited the intersection.

Figure 13: Generated multi-view frames from [Cosmos-Transfer2.5-2B/auto/multiview]. The multi-view 720p control videos for driving simulation consist of HD map elements like lanes, road markings, poles, traffic signals, traffic lights (with state), all of which can represent complex road topologies (including overpasses) as well as actors represented as cuboids. Each cuboid is color-coded based on a coarse class ontology (e.g., truck, vehicle, pedestrian), and is also shaded to differentiate between the front and back.

includes map elements like lane lines, poles, road boundaries, traffic lights, etc., and is augmented with dynamic 3D bounding boxes that indicate the positions of vehicles and pedestrians. Each object type is color-coded, and

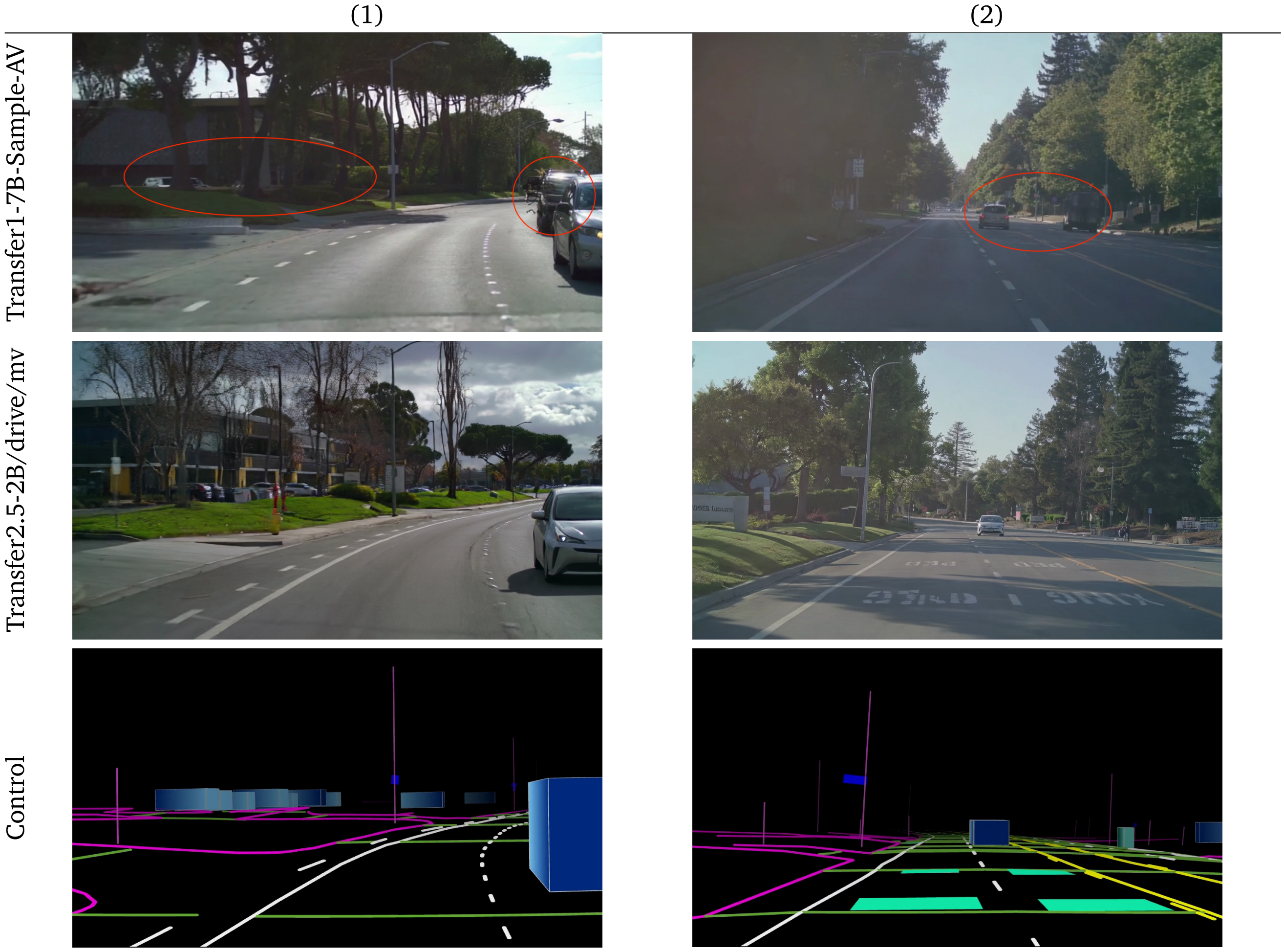


Figure 14: Comparative controlled generations between [Cosmos-Transfer1-7B-Sample-AV] and [Cosmos-Transfer2.5-2B/auto/multiview]. In example (1), we can observe that [Cosmos-Transfer1-7B-Sample-AV] hallucinates a distorted black car behind the silver vehicle, which is described neither in the text prompt nor in the control video. We can also observe the lack of alignment to the control signal when generating the parked vehicles behind the grassy mounds. In example (2), we can observe that [Cosmos-Transfer1-7B-Sample-AV] renders the vehicle in the central lane driving on the wrong side of the street with an incorrect orientation, as well as a truck instead of a pedestrian close to the sidewalk. All these inconsistencies are resolved in [Cosmos-Transfer2.5-2B/auto/multiview].

the bounding boxes are shaded according to the direction of motion, providing both semantic and motion cues.

To train the control net, we use the RDS-HQ dataset (Ren et al., 2025), which consists of 140,000 20-second multi-view driving scenes and HD map metadata covering a diverse set of traffic scenarios. Compared to the original control videos used in this work, the new world scenario map improves the following aspects: firstly, it has fine-grained controls of lane line types (e.g., dashed line, dotted line, double yellow line), whose colors and geometry patterns are directly rendered into the control video. Secondly, the bounding boxes of dynamic objects are occlusion-aware and heading-aware, providing more accurate control signals for the model learning.

6.3.3. Experiments and Results

We train [Cosmos-Predict2.5-2B/auto/multiview] for 2 epochs on the 1.5m clip dataset, using a global batch size of 64 and context parallelism of 8. We denoise 203 frames (29 per view) using 30 FPS video. For [Cosmos-Transfer2.5-2B/auto/multiview], we subsample the video and control inputs to 10FPS.

For evaluation, we use a 1000 multi-view clip dataset in RQS-HQ (Ren et al., 2025), with HD map, as well as human-labeled lanes and cuboids. These clips are disjoint from the prior two datasets used in training. As shown in Tab. 12, we observe a significant boost (up to 2.3x) in FVD/FID scores while remaining competitive in temporal and cross-camera Sampson error.

To test adherence to the control signals, we measure the detection performance of 3D-cuboid and lane detection models on generated videos, and compare these with the ground truth labels. Following the protocol described in (Ren et al., 2025), we use a monocular 3D lane detector, LATR (Luo et al., 2023), for evaluating 3D lane detection tasks, and a temporal 3D object detector, BEVFormer (Li et al., 2022), for evaluating 3D cuboid detection tasks. As shown in Tab. 13, we observe a substantial improvement (up to 60%) in detection metrics compared to Transfer1-7B-Sample-AV (NVIDIA, 2025). See Fig. 14 for visual comparisons of [Cosmos-Transfer-7B-AV-Sample] versus [Cosmos-Transfer2.5-2B/auto/multiview].

6.4. Multi-view Generation with Camera Control

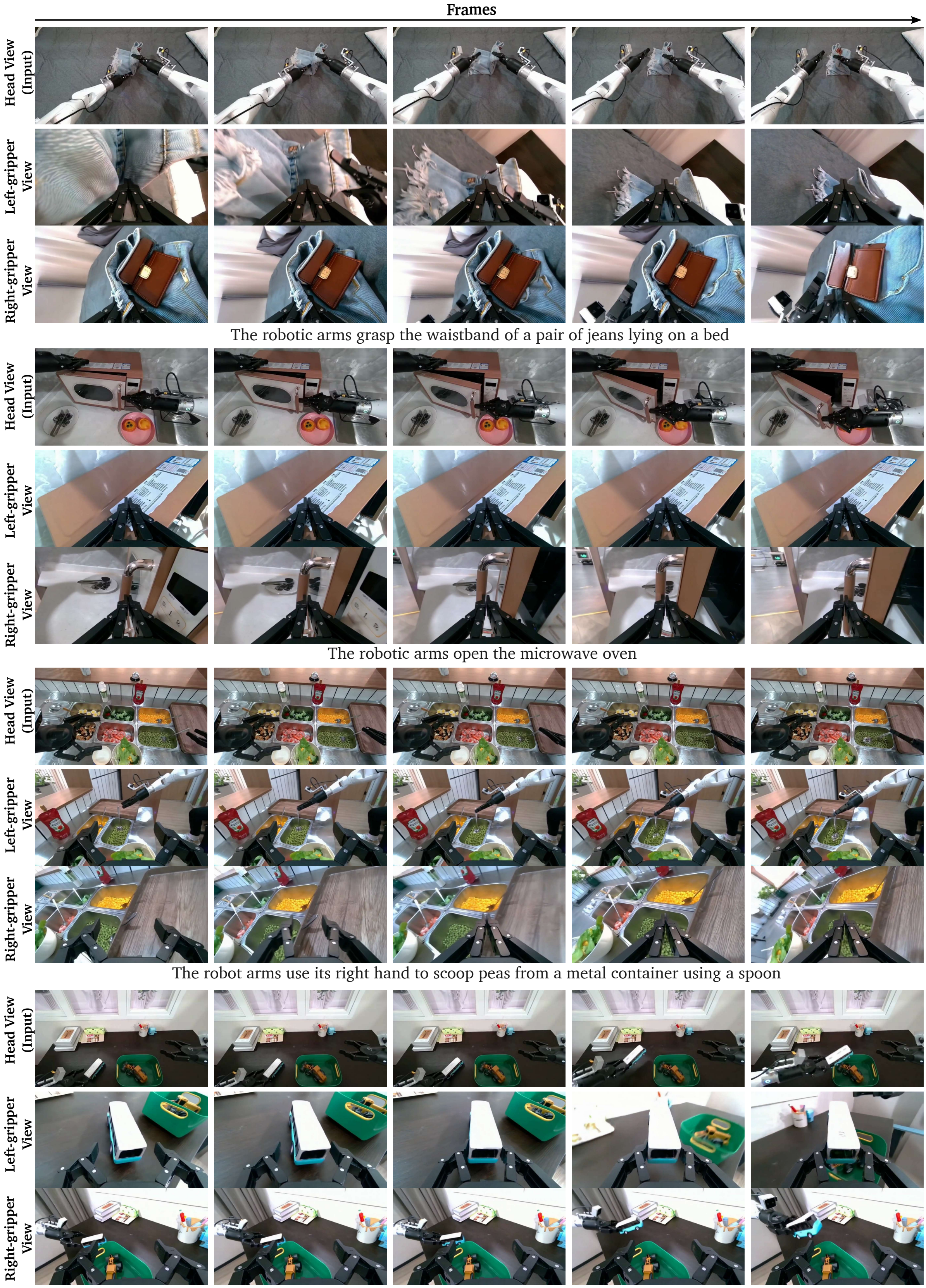
We develop [Cosmos-Predict2.5-2B/robot/multiview], a camera-controllable multi-view world generation model built on top of [Cosmos-Predict2.5-2B]. Unlike standard single-view generation, this model takes a video from a reference view and synthesizes additional videos from multiple target viewpoints defined by camera trajectories. Such a setting is especially valuable in robotics, where it enables mapping a humanoid robot's egocentric head-camera view to the gripper views on its two hands, useful for robotic manipulation simulation, where the robot must reason about objects beyond its direct line of sight. By generating consistent views that fill in occluded regions, the model provides a richer and more complete representation of the scene, enabling more reliable perception, planning, and control in real-world settings.

Table 14: Camera Control Comparison between [Cosmos-Predict2.5] and [Cosmos-Predict1].

Model	Camera Views	Condition	Type	Resolution
Cosmos-Predict1	1	text + image condition	future prediction	720p
Cosmos-Predict2.5	3	text + video condition	video re-rendering	720p

Given a source video and a set of N target camera trajectories, each specified by extrinsic-intrinsic parameters, our objective is to synthesize N target videos, each corresponding to a distinct virtual camera trajectory. We assume a standard pinhole camera model to project 3D scene points into 2D image coordinates. A comparison of camera control between [Cosmos-Predict1] and [Cosmos-Predict2.5] is provided in Tab. 14.

Architecture. We tokenize both source and target videos and concatenate their tokens along the temporal dimension. Since the encoder downsamples videos temporally by a factor of 4, we also sample camera parameters (intrinsic and extrinsic) every 4 frames to maintain temporal alignment with the latent features.



The robotic arms grasp the waistband of a pair of jeans lying on a bed

The robotic arms open the microwave oven

The robot arms use its right hand to scoop peas from a metal container using a spoon

The robot arms use its left hand to pick up the toy train placed on the wooden surface

Figure 15: [Cosmos-Predict2.5-2B/robot/multiview-agibot] generates temporally synchronized robotic manipulation videos from the left and right gripper viewpoints, conditioned on the head-view input.

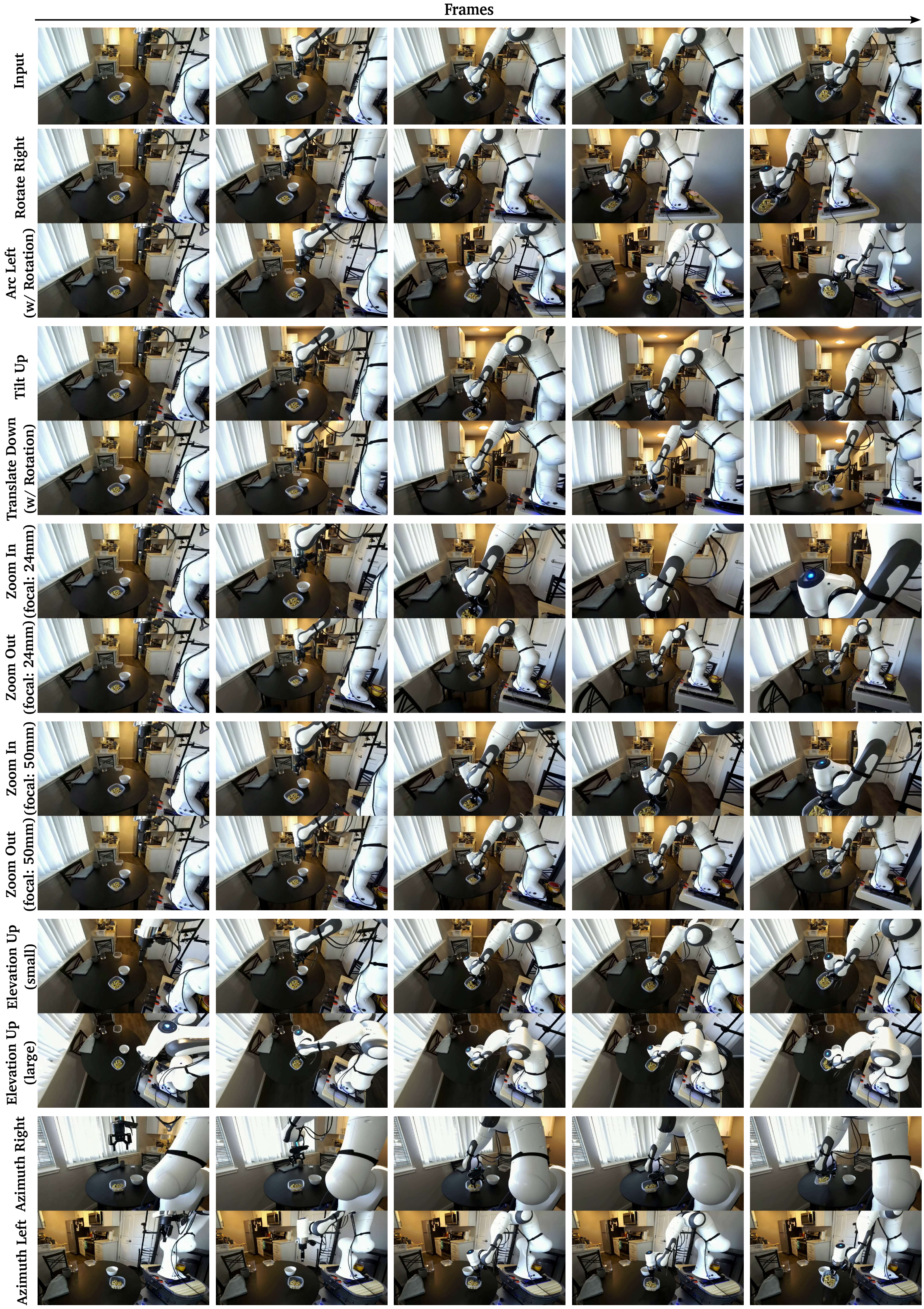


Figure 16: [Cosmos-Predict2.5-2B/robot/multiview] synthesizes synchronized videos under basic dynamic and static camera transformations, conditioned on the third-view robotic manipulation input.

Target cameras are represented as Plücker raymaps (Sitzmann et al., 2021), where pixels are mapped to 6D ray representations and subsequently patchified. A camera projection layer is introduced to align the raymap representation with the dimensionality of the video latents. The resulting raymap tokens are then added to the video tokens prior to the self-attention operation, enabling the DiT to incorporate camera pose information. During training, we freeze all the layers except the self-attention layers and the camera projection layer.

Training Datasets. We train our model on the following datasets:

- Agibot (Bu et al., 2025): A robot dataset contain ∼1,000,000 episodes. We sample 145,820 episodes, each providing 3 video views with precise camera pose information.
- MultiCamVideo (Bai et al., 2025): A large-scale synthetic dataset comprising 136,000 episodes of human motion captured with dynamic camera trajectories.
- SynCamVideo (Bai et al., 2025): A complementary synthetic dataset containing 34,000 episodes similar in content to MultiCamVideo but with fixed novel camera viewpoints, enabling evaluation under static multi-view settings.

Experiments. We adopt [Cosmos-Predict2.5-2B] as the backbone model and further post-train two variants as follows. Both variants generate outputs at 720p resolution. To address out-of-memory issues during training and inference, we employ context parallelism across multiple GPUs.

- **[Cosmos-Predict2.5-2B/robot/multiview-agibot]**: Fine-tuned on the Agibot dataset. Given a head-view robotic manipulation video as input, it synthesizes synchronized videos from the left and right gripper perspectives, as illustrated in Fig. 15.
- **[Cosmos-Predict2.5-2B/robot/multiview]**: Fine-tuned on MultiCamVideo and SynCamVideo datasets. Conditioned on a third-view video, it generates two synchronized videos under basic camera transformations, such as left/right rotations, left/right arcs, zoom in/out, azimuth shifts, elevation changes, and distance variations, while allowing for dynamic focal length adjustments, as shown in Fig. 16.

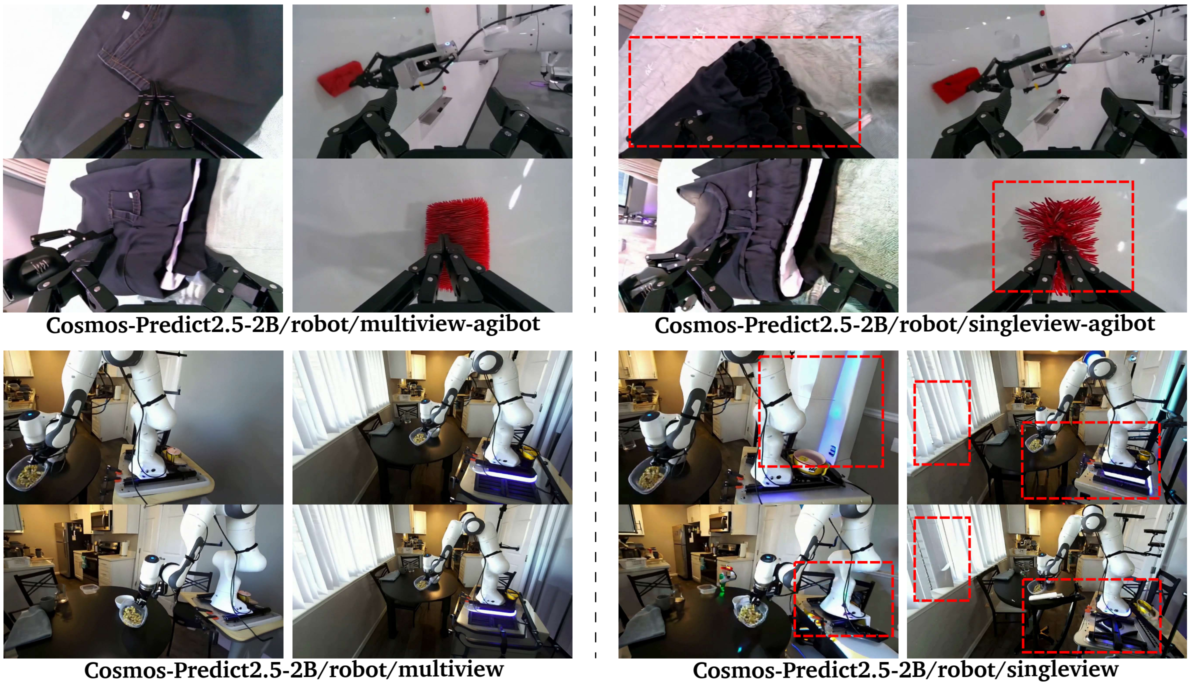
We further evaluate the generated synchronized videos along two dimensions: (1) camera trajectory error. including rotation error and translation, which measures the error between predicted camera poses from ViPE (Huang et al., 2025) on the generated videos and the corresponding ground-truth poses, and (2) cross-view consistency, quantified by the Sampson error between pairs of generated views (NVIDIA, 2025). Specifically, we conduct experiments on 80 validation videos with 16 camera trajectories using [Cosmos-Predict2.5-2B/robot/multiview]. For the baseline, we implement a single-view-to-single-view variant ([Cosmos-Predict2.5-2B/robot/singleview]) by restricting the number of target views to a single view. As illustrated in Fig. 17 and Tab. 15, [Cosmos-Predict2.5-2B/robot/multiview] achieves significantly better cross-view consistency than its single-view counterpart, while maintaining comparable camera trajectory accuracy.

Table 15: **Multi-Camera Video Generation Evaluation**. We evaluate both our model and the baseline on 80 in-the-wild robotic manipulation videos across 16 diverse camera trajectories. Best is bolded.

	Camera Accuracy		View Synchronization
Model	TransErr ↓	RotErr (rad) ↓	Sampson Error (px) ↓
Cosmos-Predict2.5-2B/robot/singleview	**0.08**	**0.19**	26.61
Cosmos-Predict2.5-2B/robot/multiview	**0.08**	0.20	**19.73**

6.5. Synthetic Data Generation for VLA training

World models show significant potential as planners and simulators for robotic manipulation. After post-training on a large video dataset of real demonstrations where robots perform actions from natural language instructions, [Cosmos-Predict2.5] can generate realistic videos of robots executing unseen commands. We can then extract pseudo-action sequences from these videos using either a latent action model or an inverse-dynamics model



Cosmos-Predict2.5-2B/robot/multiview-agibot

Cosmos-Predict2.5-2B/robot/singleview-agibot

Cosmos-Predict2.5-2B/robot/multiview

Cosmos-Predict2.5-2B/robot/singleview

Figure 17: **View synchronization comparison.** [Cosmos-Predict2.5-2B/robot/multiview] generates more coherent videos across multiple viewpoints, compared with the single-view targeted baseline (the red dotted box highlights the inconsistent parts).

(IDM) (Jang et al., 2025). This renders samples with vision (generated videos), language (instructions), and action (generated pseudo actions) annotations for VLA training. We can leverage this paradigm to generate diverse synthetic VLA training data that augments real demonstrations, thereby improving the generalization capabilities of a VLA model.

We post-train [Cosmos-Predict2.5-14B] and evaluate its performance on the synthetic VLA training data generation task using the DreamGen benchmark (Jang et al., 2025). DreamGen examines three key dimensions of generalization—object, behavior, and environment—and employs automated evaluation with vision-language models such as Qwen-VL-2.5 (Bai et al., 2025) and GPT-4o (Hurst et al., 2024). The benchmark specifically measures whether the generated videos accurately follow task instructions involving unseen objects, novel behaviors, or new environments.

From Tab. 16, we found that the resulting post-trained model, [Cosmos-Predict2.5-14B/robot/gr00tdream-gr1], achieved the highest instruction-following scores on the GR1 humanoid robot dataset (Jang et al., 2025). It outperformed competing models including Hunyuan (Kong et al., 2024), CogVideoX (Yang et al., 2024), and WAN 2.1 (Wan et al., 2025), particularly in object and environment generalization. These results highlight [Cosmos-Predict2.5-14B]'s adaptability through post-training and its ability to generate contextually accurate robot videos that faithfully realize natural language instructions.

6.6. Action-Conditioned World Generation

We extend [Cosmos-Predict2.5] from pure video generation to action-conditioned video generation, resulting in [Cosmos-Predict2.5-2B/robot/action-cond]. The model takes as input a single conditional image together with a sequence of robot actions, and generates a chunk of future frames that follow the provided action sequence. To produce full trajectories, generation is carried out in an autoregressive manner, where each chunk is predicted conditioned on the last generated frame.

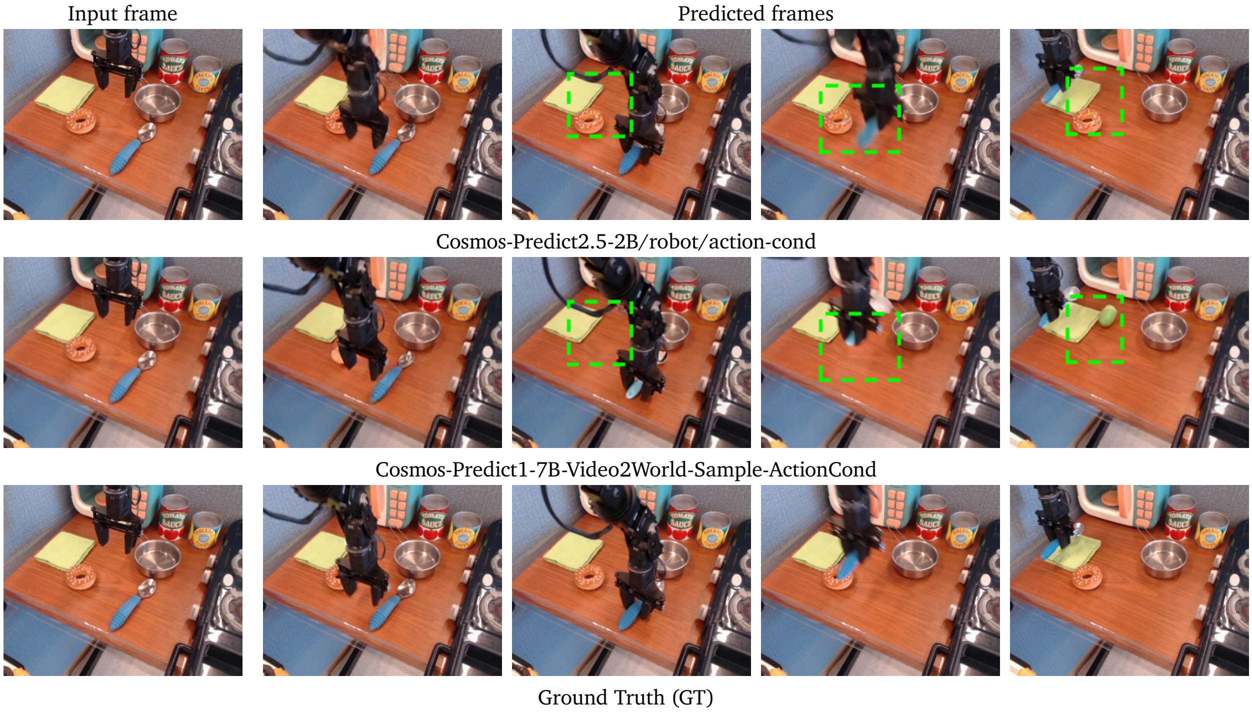
Table 16: **DreamGen Bench Statistics and Results**. GPT represents the evaluation from GPT4o, and Qwen represents the evaluation from Qwen2.5VL. All the models are SFT models. The best is bold and the second best is underlined. [Cosmos-Predict2-14B/robot/gr00tdream-gr1] is an earlier version of [Cosmos-Predict], which did not use [Cosmos-Reason1] for text embedding.

DreamGen Bench GR1 Instruction Following						
	Object		**Behavior**		**Env**	
	GPT	Qwen	GPT	Qwen	GPT	Qwen
Hunyuan	38.0	26.0	38.3	10.6	27.6	27.6
CogVideoX	72.0	38.0	44.0	28.0	55.2	41.4
WAN2.1	72.0	58.0	**72.3**	55.3	48.3	65.5
Cosmos-Predict2-14B/robot/gr00tdream-gr1	90.0	62.0	59.6	**61.7**	**69.0**	65.5
Cosmos-Predict2.5-14B/robot/gr00tdream-gr1	**91.8**	**69.4**	70.2	59.6	**69.0**	**69.0**

Because actions represent a new modality not present during pre-training, we introduce additional modules for conditioning. Specifically, we add an action embedder MLP that maps each action into a tensor. Instead of injecting this tensor directly, we incorporate it by adding it to the timestamp embeddings of the DiT modules.

Table 17: Evaluation of action-conditioned video prediction on Bridge dataset.

Method	PSNR ↑	SSIM ↑	Latent L2 ↓	FVD ↓
Cosmos-Predict1-7B-Video2World-Sample-ActionCond	21.14	0.82	0.32	190
Cosmos-Predict2.5-2B/robot/action-cond	**24.95**	**0.85**	**0.28**	**146**


Input frame

Predicted frames

Cosmos-Predict2.5-2B/robot/action-cond

Cosmos-Predict1-7B-Video2World-Sample-ActionCond

Ground Truth (GT)

Figure 18: **Action-conditioned video prediction samples on the Bridge dataset.** Comparison of predicted rollouts from [Cosmos-Predict2.5-2B/robot/action-cond] and [Cosmos-Predict1-7B-Video2World-Sample-ActionCond] against the ground-truth frames. [Cosmos-Predict2.5-2B/robot/action-cond] demonstrates better object permanence. The green dotted box highlights the parts with the object permanence issues.

Experiments. We conduct experiments using the public Bridge dataset (Walke et al., 2023) following prior work (Zhu et al., 2024). The dataset contains approximately 20,000 episodes of third-person videos capturing a robot arm performing various tasks in a kitchen environment. Each video has a resolution of 320×256 and is recorded at 5 FPS. Corresponding to each frame, the robot action is represented as a 7-dimensional vector in the gripper coordinate space: $(\Delta x, \Delta y, \Delta z, \Delta\theta_r, \Delta\theta_p, \Delta\theta_y, \text{GripperWidth})$, which specifies the relative displacement, rotation, and width of the gripper.

To evaluate the quality of video generation, we randomly sample 100 episodes from the official Bridge test set and generate videos for them, comparing the results against the ground-truth videos. We use [Cosmos-Predict1-7B-Video2World-Sample-ActionCond] as a baseline for comparison.

The quantitative metrics, summarized in Tab. 17, include PSNR, SSIM, Latent L2 (Zhu et al., 2024), and FVD. As shown, the [Cosmos-Predict2.5-2B/robot/action-cond] models outperform the baseline across all metrics. Selected predicted video frames are presented in Fig. 18, highlighting the high quality of the predictions relative to the ground-truth frames.

Table 18: Ablation study on the Bridge dataset. The results show that incorporating action conditioning with time embeddings yields better action-conditioned video generation performance.

Method	PSNR ↑	SSIM ↑	Latent L2 ↓	FVD ↓
Cosmos-Predict2.5-2B/robot/action-cond with TimeEmbedding (proposed)	**24.95**	**0.85**	**0.28**	**146**
Cosmos-Predict2.5-2B/robot/action-cond with CrossAtten	24.41	0.84	**0.28**	159
Cosmos-Predict2.5-2B/robot/action-cond with ChannelConcat	23.11	0.78	0.35	267

We further investigate various methods for incorporating action conditioning. In addition to applying it through time embeddings, we also explore two alternatives: (1) cross-attention within the DiT blocks and (2) channel concatenation. The results are presented in Tab. 18.

7. Related Work

World Models. Recent years have seen growing interest in world models that learn to predict future states from current observations and potential actions, enabling more efficient decision-making and planning (Ha and Schmidhuber, 2018). Research in this area has evolved into two primary paradigms for modeling world dynamics. The first focuses on learning predictive models in abstract, latent representation spaces (Ha and Schmidhuber, 2018; Hafner et al., 2019; Assran et al., 2025; Chen et al., 2025). These approaches aim to compress high-dimensional sensory inputs into compact, learned state representations that preserve the essential structure of the environment, thereby enabling efficient and tractable planning. In contrast, the second paradigm, the one we adopt, centers on modeling world dynamics directly in pixel space through high-fidelity video prediction as a video generative model (OpenAI, 2024; NVIDIA, 2025; Ball et al., 2025). These models simulate future observations frame-by-frame and can be extended to incorporate various control signals, such as camera pose, action sequences, and spatially dense inputs like a world scenario map. This retains rich, high-fidelity information, making our models effective synthetic data generators for downstream policy learning, while also remaining flexible enough to be extended to support diverse control signals. In addition to these two dominant approaches, a third, emerging direction explores native 3D and 4D representations of world states, using either neural scene representations or physically grounded simulators (Singer et al., 2023; Watson et al., 2024; Zhao et al., 2024; Liu et al., 2025; Li et al., 2025; Nasiriany et al., 2024). These models aim to provide a more structured and geometric-aware understanding of the environment.

Video Generative Models. Video generative models represent a rapidly advancing frontier in generative

AI. In recent years, several powerful closed-source systems—such as Sora (OpenAI, 2024), Kling (KuaiShou, 2024), (Runway, 2024), Hailuo (MiniMax, 2024), MovieGen (Polyak et al., 2024), Seedance (Gao et al., 2025), Veo (DeepMind, 2025), and Waver (Zhang et al., 2025)—have demonstrated remarkable progress in general-purpose video generation. Despite their impressive capabilities, the proprietary nature of these models poses a significant barrier to research and downstream applications. The lack of access to model weights and training code prevents the broader community from fine-tuning, extending, or adapting these models for specialized use cases such as autonomous driving and robotics. In contrast, the emergence of open-source video generation models, including Wan (Wan et al., 2025), LTX (HaCohen et al., 2024), and Hunyuan (Kong et al., 2024), has fostered greater transparency and accessibility. These models enable reproducible research and community-driven innovation. However, most remain optimized for general-purpose content creation and often fall short in domains requiring precise, fine-grained control over object dynamics, interactions, and physical consistency—capabilities essential for advancing physical AI. [Cosmos-Predict1] (NVIDIA, 2025) represents the first open-source video generative model explicitly tailored for physical AI applications. In this work, we further enhance its capabilities by training it on a high-quality, domain-specific dataset curated for the complexities of physical reasoning. Additionally, we integrate a text encoder based on [Cosmos-Reason1] (NVIDIA, 2025), our vision-language foundation model designed specifically for physical AI tasks. This integration significantly improves the model's ability to generate physically plausible and controllable video sequences conditioned on natural language descriptions.

Foundation World Model for Physical AI. Most existing world models, whether closed-source or open-source, regardless of technical approaches, focus on general content generation in the digital world, e.g., movies and computer games. The introduction of [Cosmos-Predict1] (NVIDIA, 2025) and [Cosmos-Transfer1] (NVIDIA, 2025) brings the first batch of open-source world models in Physical AI. It has facilitated the development of open evaluation benchmarks for both general-purpose world generation (Duan et al., 2025; Zhou et al., 2025; Zhao et al., 2025) and specialized domains such as physics (Bansal et al., 2024; Motamed et al., 2025; Guo et al., 2025; Bansal et al., 2025; Bordes et al., 2025) and Embodied AI (Yang et al., 2025; Liao et al., 2025; Yue et al., 2025). As foundation world models, [Cosmos-Predict1] was post-trained to enable new capabilities, including camera control (Ren et al., 2025), motion trajectory control (Wang et al., 2025), and video relighting (He et al., 2025). It has also been used as a synthetic data generation engine for robot policy training (Jang et al., 2025; Bjorck et al., 2025) and development of autonomous driving systems (Ren et al., 2025; Fu et al., 2025). In this work, we demonstrated the enhanced capabilities of [Cosmos-Predict2.5] for VLA model training, robot policy training/validation, autonomous driving simulation, and robotic manipulation. [Cosmos-Transfer2.5] also improved upon its predecessors for long-horizon video translations and closed-loop simulation. We hope the open-source of [Cosmos-Predict2.5] and [Cosmos-Transfer2.5] can continue facilitating development and innovation within the Physical AI community.

8. Conclusion

We presented [Cosmos-Predict2.5] and [Cosmos-Transfer2.5], the latest Cosmos video world foundation models for Physical AI. Leveraging large-scale curated video datasets, flow-matching training, improved text embedding, domain-specific post-training, and reinforcement learning, our models achieve leading results on Physical AI benchmarks. Beyond benchmarks, we demonstrated their effectiveness in robotics and autonomous driving, where high-fidelity synthetic video is essential. By releasing models and code, we aim to establish Cosmos as a world foundation model platform for a simulation-first ecosystem that advances Physical AI and bridges the gap between simulation and real-world deployment.

A. Contributors and Acknowledgments

A.1. Contributors

Arslan Ali
Junjie Bai
Maciej Bala
Yogesh Balaji
Aaron Blakeman
Tiffany Cai
Jiaxin Cao
Tianshi Cao
Elizabeth Cha
Yu-Wei Chao
Prithvijit Chattopadhyay
Mike Chen
Yongxin Chen
Yu Chen
Shuai Cheng
Yin Cui
Jenna Diamond
Jiaojiao Fan
Linxi Fan
Liang Feng
Francesco Ferroni
Sanja Fidler
Xiao Fu
Ruiyuan Gao
Yunhao Ge
Jinwei Gu
Aryaman Gupta
Siddharth Gururani
Imad El Hanafi
Ali Hassani
Zekun Hao
Jacob Huffman
Joel Jang
Pooya Jannaty
Jan Kautz
Grace Lam
Xuan Li
Zhaoshuo Li
Maosheng Liao
Chen-Hsuan Lin
Tsung-Yi Lin
Yen-Chen Lin
Huan Ling
Ming-Yu Liu
Xian Liu
Yifan Lu
Alice Luo
Qianli Ma
Hanzi Mao
Kaichun Mo
Seungjun Nah
Yashraj Narang
Abhijeet Panaskar
Lindsey Pavao
Trung Pham
Morteza Ramezanali
Fitsum Reda
Scott Reed
Xuanchi Ren
Haonan Shao
Yue Shen
Stella Shi
Shuran Song
Bartosz Stefaniak
Shangkun Sun
Shitao Tang
Sameena Tasmeen
Lyne Tchapmi
Jibin Varghese
Andrew Z. Wang
Hao Wang
Haoxiang Wang
Heng Wang
Ting-Chun Wang
Fangyin Wei
Jiashu Xu
Dinghao Yang
Xiaodong Yang
Haotian Ye
Seonghyeon Ye
Xiaohui Zeng
Jing Zhang
Qinsheng Zhang
Kaiwen Zheng
Andrew Zhu
Yuke Zhu

A.2. Acknowledgments

Chris Alexiuk
Jon Allen
Charles Anderson
Vince Auletta
Joshua Bapst
Sanchit Bhattacharjee
Alexis Bjorlin
Dan Blick
Aditi Bodhankar
Jeremy Bottleson
HJ Chen
Rui Chen
Tiffany Chen
Wenkai Chen
Nuttapong Chentanez
Himanshu Chodhary
Tae Eun Choe
Jeana Choi
Ashley Chow
Guillermo Garcia Cobo
Marek Dabek
John Dickinson
Naomi Eigbe
Damien Fagnou
Dominik Farhan
Amol Fasale
Mohamed Fawzy
Sergiy Fefilatyev
TJ Galda
Abhinav Garg
Vlad Getselevich
Pengfei Guo
Aryaman Gupta
Brett Hamilton
Mohammad Harrim
Milos Hasan
Nathan Hayes-Roth
Yufan He
Logan Herche
Sophia Huang
Spencer Huang
Ryan Ji
Yangqing Jia
Brendan Johnson
Jaehyun Jung
Sagar Karale
Artur Kasymov
Arnav Khanna
Will Kim
Carsten Kolve
Reka Kovacs
Christopher Labis
Christian Laforte
Ti Leggett
Alice Li
Xiang Li
Edy Lim
Daniel Lindsey
Aditya Mahajan
Nikolay Matveiev
Amanda Moran
Jashojit Mukherjee
Apurv Naman
Nigel Nelson
Merlin Nimier-David
Sangmin Oh
Ruben Ohana
Shubham Pachori
Sravan Patchala
Mahesh Patekar
Mitesh Patel
Joel Pennington
Sahil Ramani
Vaibhav Ranglani
Wojciech Rymer
Amirmojtaba Sabour
Jun Saito
Akul Santhosh
Chris Schultz
Alexander Schwarz
Gautham Sholingar
Mateusz Sieniawski
Rajat Vikram Singh
Hariharan Srinivasan
Javier Gamazo Tejero
Ruchik Thaker
Peter Udvardi
Gandhi Vaithilingam
George Vine
Thomas Volk
Raju Wagwani
Mengmeng Xiao
Ning Xu
Lixiao Yang
Wei Yang
Jianhe Yuan
Itai Zadok
Zheng Zeng
Suyue Zhang
Hongning Zhao
Andrew Zhu

References

[1] 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. 33

[2] Yuval Atzmon, Maciej Bala, Yogesh Balaji, Tiffany Cai, Yin Cui, Jiaojiao Fan, Yunhao Ge, Siddharth Gururani, Jacob Huffman, Ronald Isaac, et al. Edify image: High-quality image generation with pixel space laplacian diffusion models. *arXiv preprint arXiv:2411.07126*, 2024. 8

[3] Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. In *ICCV*, 2025. 30

[4] Jianhong Bai, Menghan Xia, Xintao Wang, Ziyang Yuan, Xiao Fu, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Di Zhang. Syncammaster: Synchronizing multi-camera video generation from diverse viewpoints. In *ICLR*, 2025. 30

[5] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 4, 5, 6, 7, 31

[6] Philip J Ball, J Bauer, F Belletti, et al. Genie 3: A new frontier for world models, 2025. 33

[7] Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. Videophy: Evaluating physical commonsense for video generation. *arXiv preprint arXiv:2406.03520*, 2024. 34

[8] Hritik Bansal, Clark Peng, Yonatan Bitton, Roman Goldenberg, Aditya Grover, and Kai-Wei Chang. Videophy-2: A challenging action-centric physical commonsense evaluation in video generation. *arXiv preprint arXiv:2503.06800*, 2025. 34

[9] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. 6, 34

[10] bloc97. Ntk-aware scaled rope allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation. https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/, 2023. Reddit post, r/LocalLLaMA. 9

[11] Florian Bordes, Quentin Garrido, Justine T Kao, Adina Williams, Michael Rabbat, and Emmanuel Dupoux. Intphys 2: Benchmarking intuitive physics understanding in complex synthetic environments. *arXiv preprint arXiv:2506.09849*, 2025. 34

[12] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *IROS*, 2025. 6, 30

[13] Delong Chen, Theo Moutakanni, Willy Chung, Yejin Bang, Ziwei Ji, Allen Bolourchi, and Pascale Fung. Planning with reasoning using vision language world model. *arXiv preprint arXiv:2509.02722*, 2025. 33

[14] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. In *CVPR*, 2025. 17

[15] Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023. 8

[16] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *RSS*, 2023. 21

[17] Databricks. Delta lake: Open-source storage framework that enables building lakehouses. https://delta.io/, 2019. Open-source project, Delta Lake. 6

[18] Google DeepMind. Veo 3, 5 2025. URL https://deepmind.google/technologies/veo/veo-3/. 34

[19] Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. Worldscore: A unified evaluation benchmark for world generation. *arXiv preprint arXiv:2504.00983*, 2025. 34

[20] 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 *ICML*, 2024. 8, 11

[21] Yongjie Fu, Ruijian Zha, Pei Tian, and Xuan Di. Llm-based realistic safety-critical driving video generation. *arXiv preprint arXiv:2507.01264*, 2025. 34

[22] Ruiqi Gao, Emiel Hoogeboom, Jonathan Heek, Valentin De Bortoli, Kevin Patrick Murphy, and Tim Salimans. Diffusion models and gaussian flow matching: Two sides of the same coin. In *The Fourth Blogpost Track at ICLR 2025*, 2025. 8

[23] Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025. 34

[24] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 7

[25] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 13

[26] Xuyang Guo, Jiayan Huo, Zhenmei Shi, Zhao Song, Jiahao Zhang, and Jiale Zhao. T2vphysbench: A first-principles benchmark for physical consistency in text-to-video generation. *arXiv preprint arXiv:2505.00337*, 2025. 34

[27] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. 33

[28] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024. 34

[29] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019. 33

[30] Ali Hassani, Fengzhe Zhou, Aditya Kane, Jiannan Huang, Chieh-Yun Chen, Min Shi, Steven Walton, Markus Hoehnerbach, Vijay Thakkar, Michael Isaev, et al. Generalized neighborhood attention: Multi-dimensional sparse attention at the speed of light. *arXiv preprint arXiv:2504.16922*, 2025. 14

[31] Kai He, Ruofan Liang, Jacob Munkberg, Jon Hasselgren, Nandita Vijaykumar, Alexander Keller, Sanja Fidler, Igor Gilitschenski, Zan Gojcic, and Zian Wang. Unirelight: Learning joint decomposition and synthesis for video relighting. *arXiv preprint arXiv:2506.15673*, 2025. 34

[32] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *ICML*, 2023. 8

[33] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, et al. Vipe: Video pose engine for 3d geometric perception. *arXiv preprint arXiv:2508.10934*, 2025. 30

[34] Wei-Chih Hung, Vincent Casser, Henrik Kretzschmar, Jyh-Jing Hwang, and Dragomir Anguelov. Let-3d-ap: Longitudinal error tolerant 3d average precision for camera-only 3d detection, 2024. URL https://arxiv.org/abs/2206.07705. 24

[35] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 31

[36] Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025. 3, 31, 34

[37] Tao Jiang, Peng Lu, Li Zhang, Ningsheng Ma, Rui Han, Chengqi Lyu, Yining Li, and Kai Chen. Rtmpose: Real-time multi-person pose estimation based on mmpose. *arXiv preprint arXiv:2303.07399*, 2023. 7

[38] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *NeurIPS*, 2022. 8

[39] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. 6

[40] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 31, 34

[41] KuaiShou. Kling, 2024. URL https://klingai.com/. 34

[42] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers, 2022. URL https://arxiv.org/abs/2203.17270. 27

[43] Zizhang Li, Hong-Xing Yu, Wei Liu, Yin Yang, Charles Herrmann, Gordon Wetzstein, and Jiajun Wu. Wonderplay: Dynamic 3d scene generation from a single image and actions. *arXiv preprint arXiv:2505.18151*, 2025. 33

[44] 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 *ICLR*, 2025. 14

[45] 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. 34

[46] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 8

[47] Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Wenyu Qin, Menghan Xia, et al. Improving video generation with human feedback. *arXiv preprint arXiv:2501.13918*, 2025. 13

[48] Jinxiu Liu, Shaoheng Lin, Yinxiao Li, and Ming-Hsuan Yang. Dynamicscaler: Seamless and scalable video generation for panoramic scenes. In *CVPR*, 2025. 33

[49] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 21

[50] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024. 14

[51] Yueru Luo, Chaoda Zheng, Xu Yan, Tang Kun, Chao Zheng, Shuguang Cui, and Zhen Li. Latr: 3d lane detection from monocular images with transformer, 2023. URL `https://arxiv.org/abs/2308.04583`. 27

[52] MiniMax. Hailuo, 2024. URL `https://hailuoai.com/video`. 34

[53] Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? *arXiv preprint arXiv:2501.09038*, 2025. 34

[54] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024. 33

[55] NVIDIA. Cosmos-reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*, 2025. 3, 9, 34

[56] NVIDIA. Cosmos-transfer1: Conditional world generation with adaptive multimodal control. *arXiv preprint arXiv:2503.14492*, 2025. 3, 17, 18, 27, 34

[57] NVIDIA. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. 3, 4, 8, 9, 30, 33, 34

[58] OpenAI. Sora, 2024. URL `https://openai.com/sora/`. 33, 34

[59] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *NeurIPS*, 2022. 13

[60] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023. 9

[61] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 34

[62] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *KDD*, 2020. 14

[63] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 18, 21

[64] Allen Z. Ren, Justin Lidard, Lars Lien Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *ICLR*, 2025. 21

[65] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 21

[66] 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. 3, 24, 27, 34

[67] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *CVPR*, 2025. 34

[68] Runway. Gen 3, 2024. URL https://runwayml.com/research/introducing-gen-3-alpha. 34

[69] Paul D Sampson. Fitting conic sections to “very scattered” data: An iterative refinement of the bookstein algorithm. *Computer Graphics and Image Processing*, 1982. ISSN 0146-664X. doi: https://doi.org/10.1016/0146-664X(82)90101-0. URL https://www.sciencedirect.com/science/article/pii/0146664X82901010. 24

[70] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 13

[71] Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, et al. Text-to-4d dynamic scene generation. *arXiv preprint arXiv:2301.11280*, 2023. 33

[72] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *NeurIPS*, 2021. 30

[73] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2, 2021. 24

[74] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. cuRobo: Parallelized collision-free minimum-jerk robot motion generation. *arXiv preprint arXiv:2310.17274*, 2023. 20

[75] 1X Technologies. 1x technologies | safe humanoids for the home, 2025. URL https://www.1x.tech/. 6

[76] Quan Vuong, Sergey Levine, Homer Rich Walke, Karl Pertsch, Anikait Singh, Ria Doshi, Charles Xu, Jianlan Luo, Liam Tan, Dhruv Shah, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 2023. 6

[77] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *CoRL*, 2023. 6, 33

[78] 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. 9, 31, 34

[79] Andrew Z Wang, Songwei Ge, Tero Karras, Ming-Yu Liu, and Yogesh Balaji. A comprehensive study of decoder-only llms for text-to-image generation. In *CVPR*, 2025. 9

[80] Boyang Wang, Xuweiyi Chen, Matheus Gadelha, and Zezhou Cheng. Frame in-n-out: Unbounded controllable image-to-video generation. *arXiv preprint arXiv:2505.21491*, 2025. 34

[81] Qiuheng Wang, Yukai Shi, Jiarong Ou, Rui Chen, Ke Lin, Jiahao Wang, Boyuan Jiang, Haotian Yang, Mingwu Zheng, Xin Tao, et al. Koala-36m: A large-scale video dataset improving consistency between fine-grained conditions and video content. In *CVPR*, 2025. 4

[82] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *ECCV*, 2024. 11

[83] Daniel Watson, Saurabh Saxena, Lala Li, Andrea Tagliasacchi, and David J Fleet. Controlling space and time with diffusion models. *arXiv preprint arXiv:2407.07860*, 2024. 33

[84] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *ICML*, 2022. 12

[85] Haoning Wu, Erli Zhang, Liang Liao, Chaofeng Chen, Jingwen Hou, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *CVPR*, 2023. 4, 19

[86] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024. 6

[87] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *NeurIPS*, 2023. 12

[88] Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*, 2024. 12

[89] Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025. 34

[90] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 31

[91] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *ICML*, 2024. 12

[92] Hu Yue, Siyuan Huang, Yue Liao, Shengcong Chen, Pengfei Zhou, Liliang Chen, Maoqing Yao, and Guanghui Ren. Ewmbench: Evaluating scene, motion, and semantic quality in embodied world models. *arXiv preprint arXiv:2505.09694*, 2025. 34

[93] 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. 34

[94] Yuyang Zhao, Chung-Ching Lin, Kevin Lin, Zhiwen Yan, Linjie Li, Zhengyuan Yang, Jianfeng Wang, Gim Hee Lee, and Lijuan Wang. Genxd: Generating any 3d and 4d scenes. *arXiv preprint arXiv:2411.02319*, 2024. 33

[95] Zecheng Zhao, Selena Song, Tong Chen, Zhi Chen, Shazia Sadiq, and Yadan Luo. Are synthetic videos useful? a benchmark for retrieval-centric evaluation of synthetic videos. *arXiv preprint arXiv:2507.02316*, 2025. 34

[96] Fengzhe Zhou, Jiannan Huang, Jialuo Li, and Humphrey Shi. Physical ai bench: A comprehensive benchmark for physical ai generation and understanding, 2025. URL `https://github.com/SHI-Labs/physical-ai-bench`. 15, 18

[97] Shijie Zhou, Alexander Vilesov, Xuehai He, Ziyu Wan, Shuwang Zhang, Aditya Nagachandra, Di Chang, Dongdong Chen, Xin Eric Wang, and Achuta Kadambi. Vlm4d: Towards spatiotemporal awareness in vision language models. *arXiv preprint arXiv:2508.02095*, 2025. 34

[98] Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: Learning interactive real-robot action simulators. *arXiv preprint arXiv:2406.14540*, 2024. 33

[99] Zilliz. Milvus: An open-source vector database for scalable similarity search. `https://milvus.io/`, 2019. Open-source project, Milvus. 6