REASONING CANVAS: SPATIAL-TEMPORAL VISUAL REASONING FOR EMBODIED VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Video generation models are increasingly used as visual world models for embodied AI, but they are not explicitly trained for spatial-temporal reasoning required by multi-object, multi-step manipulation tasks. Such models may generate visually plausible videos while manipulating the wrong object, misordering subtasks, or placing objects at incorrect targets. To this end, we propose **Reasoning Canvas**, a dual-stream video diffusion framework that introduces an explicit auxiliary Thought stream for spatial-temporal structure. The Thought stream renders object-following bounding boxes and step-wise text prompts on a black canvas, providing a lightweight representation of what should happen before high-fidelity pixels are fully synthesized. A **Timestep-Bias Reasoning Routing (TBRR)** schedule resolves this simpler Thought stream earlier than the RGB stream, enabling subsequent RGB denoising to condition on a more stable plan. We also construct **HierCanvas Dataset**, a hierarchical manipulation dataset with *task→subtask→step* programs, bounding boxes, and rendered step prompts. Experiments show that Reasoning Canvas yields consistent gains in visual quality, instruction alignment, and tracking, with stronger robustness to ambiguous instructions and OOD task compositions, validating ahead-of-time visual reasoning as an effective interface for video generation based world models.

1 INTRODUCTION

Video generation based world models have become a dominant paradigm in embodied AI, playing an indispensable role in synthetic data construction (GigaBrain Team et al., 2025; Jang et al., 2025; Shang et al., 2025), closed-loop interaction with policies (Guo et al., 2026; Jiang et al., 2026b; Liu et al., 2026; Yang et al., 2026), and world action model (Hu et al., 2025; Kim et al., 2026; Li et al., 2026; Ye et al., 2026; Yuan et al., 2026; Zhu et al., 2025a). However, conventional video generators are pre-trained on web-scale corpora that overwhelmingly reward visual fidelity and aesthetics, and as such are not optimized for the *planning* ability that downstream world models actually require, *e.g.*, deciding which objects matter in the current scene, where they should move, and in what order the events should unfold. In complex compositional scenarios, where tasks typically involve multiple objects and multiple sequential steps, pre-trained video generators frequently fail in characteristic ways: they manipulate the wrong object, drop intermediate steps, swap the order of subtasks, or hallucinate a plausible-looking but task-violating placement. What is missing is not visual capacity, but an explicit handle for spatial-temporal planning.

When a large language model (Bai et al., 2023; Guo et al., 2025a; Touvron et al., 2023) encounters a complex task, it does not produce the final answer all at once. Instead, it externalizes intermediate reasoning steps as Chain-of-Thought (Wei et al., 2022; Guo et al., 2025a; Shao et al., 2024; Shen et al., 2025) and uses these intermediate results to guide the final answer. This also mirrors how humans reason, as we typically decompose a final answer into coarse intermediate results such as *which* entity is involved, *what* the high-level step is, and *where* things should go before arriving at the final solution. Video generation based world models lack such an explicit reasoning mechanism for multi-object, multi-step tasks, even though they must identify which entity should be manipulated, segment the global instruction into executable atomic steps, and arrange those steps temporally before generating any high-fidelity pixels. Currently, all of this planning occurs *inside* the same denoising trajectory as pixel generation, with only a text instruction as anchor, so that the plan and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the pixels co-evolve under the same noisy budget, and there is nowhere for the model to commit to a spatial-temporal decision *before* visual content generation begins.

Building on this observation, we introduce an explicit *reasoning stage* into video generation, as shown in Figure 1. Before the main denoising phase, we first generate a structured but simpler companion stream that contains explicitly the task-critical information. This companion stream encodes a *plan* rather than photo-realistic visual details, it can be denoised faster and committed earlier. Once the companion stream completes its denoising earlier than the RGB stream, all remaining RGB denoising stages can condition on this resolved plan, rather than inventing the plan and visual details in lockstep from pure noise. We instantiate this idea as **Reasoning Canvas**, a dual-stream video diffusion framework that augments the visual stream with a parallel *thought channel* encoding the spatial-temporal plan, where object-following bounding boxes for the manipulated entity provide *spatial grounding* and the step-wise rendered text provides *temporal planning*. The two streams share a single diffusion backbone, but a *Timestep-Bias Reasoning Routing (TBRR)* schedule lets the thought stream be completed before the visual stream exits its high-noise phase, so that the visual stream is guided by a clean, committed spatial-temporal plan.

To evaluate the effectiveness of Reasoning Canvas, we use robot manipulation videos as a testbed for multi-object, multi-step video generation and construct the **HierCanvas Dataset**. Each video is represented as a hierarchical *[task → subtask → step]* program, where each step is annotated with bounding boxes as spatial grounding signals and step-wise prompt rendered on a black canvas as temporal planning signals, yielding paired `(RGB, Thought)` streams. With precise annotations of object identities, spatial relations, and task progression, HierCanvas provides a comprehensive benchmark for video generation based world models. Rather than focusing solely on visual generation quality, we argue that metrics such as the accuracy of object motion trajectories, alignment with specific task instructions, and robustness to ambiguous task instructions are more meaningful for evaluating how well generative models transfer to embodied AI.

Extensive experiments show that Reasoning Canvas consistently improves video generation across visual quality (Unterthiner et al., 2019; Wang et al., 2004; Nilsson & Akenine-Möller, 2020), instruction alignment (Hessel et al., 2021), and tracking accuracy (Karaev et al., 2024). Beyond standard evaluation, our method also exhibits stronger robustness under ambiguous instructions (Jiang et al., 2026a) and better generalization to out-of-distribution task compositions. These results suggest that explicitly co-generating spatial and temporal reasoning with visual content provides a more robust interface for complex video generation in embodied domains.

Our contributions are:

1. We identify the missing piece between generic video generators and embodied world models as an explicit spatial-temporal *planning* stage, externalizing it as a dedicated reasoning channel prior to guiding subsequent RGB denoising.
2. We propose **Reasoning Canvas**, a dual-stream video diffusion framework that co-generates this explicit thought channel with the RGB stream, together with TBRR and complementary guidance/training strategies that enforce strict visual adherence to the generated thoughts.
3. We construct **HierCanvas Dataset**, a hierarchical *task → subtask → step* reasoning dataset in a spatially and temporally complex robot manipulation testbed, with bounding-box and rendered-text annotations.
4. We show consistent gains in visual quality, instruction alignment, and tracking accuracy, together with stronger robustness to ambiguous instructions and OOD task compositions.

2 Related Work

Video generation based reasoning. Recent work has introduced a new perspective on video generation models as visual reasoners rather than merely visual synthesizers. Wiedemer et al. (2025) formulate video generation as a general-purpose task-solving paradigm and introduce the Chain-of-Frames (CoF) reasoning framework, where intermediate frames serve as visual reasoning steps. Wang et al. (2026b) further show that the reasoning process of video diffusion models is not primarily carried by frame-to-frame transitions, but emerges along the denoising trajectory. Following

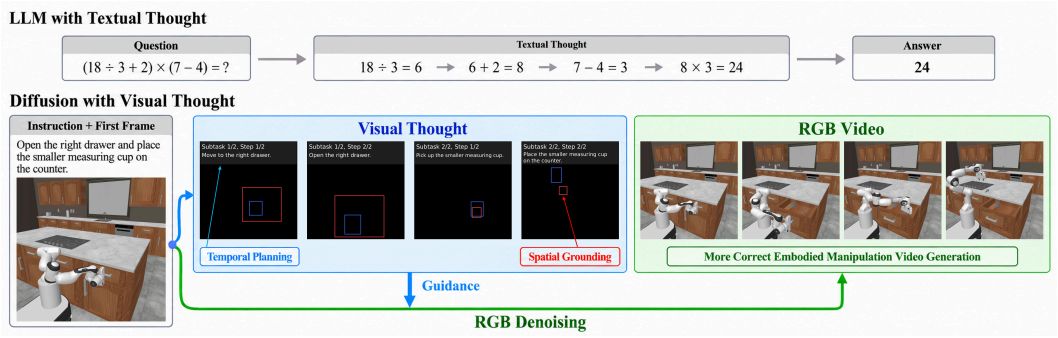

Figure 1: **Textual and visual Thought.** Analogous to textual Chain-of-Thought in LLMs, Reasoning Canvas generates a spatial-temporal Thought stream alongside RGB. The Thought is finalized early to guide the remaining RGB denoising. See Appendix A.1 for Thought details.

this view, a series of benchmarks have been proposed to evaluate video-generation-based reasoning across commonsense, structural, symbolic, mathematical, physical, and process-level tasks (Guo et al., 2025b; Li et al., 2025; Liu et al., 2025; Qi et al., 2026; Wang et al., 2026a; Wu et al., 2026b;a; Yang et al., 2025a). Beyond evaluation, recent works also leverage such reasoning abilities to improve downstream tasks that require structured reasoning, including reward modeling, text-to-image generation, video editing, and navigation (Mi et al., 2025; Tong et al., 2026; Yang et al., 2025b; Xiao et al., 2025).

Visual instructions for video generation. A growing body of work shows that visual-domain signals provide more direct generative control than text-only prompts, ranging from rendered text, arrows, and trajectories embedded in frames (Fang et al., 2025; Yi et al., 2026) to dense visual motion signals such as motion fields and physics-conditioned goal motions (Xu et al., 2026; Gillman et al., 2026). All of these treat the visual instruction as an *input* that must be supplied at inference time. Reasoning Canvas instead makes it an *output*: the spatial-temporal plan, consisting of object-following bounding boxes and on-screen step prompts, is co-generated with the RGB video, so the model produces its own visual reasoning signal rather than consuming an externally provided one.

Video generation based world models for embodied AI. Video-generation-based world models have become a dominant paradigm in embodied AI and roughly fall into three lines of work. (i) *Video-WMs as data generators*, where pretrained or fine-tuned video generators synthesize large-scale, action-aligned interaction videos to scale up downstream policy learning (GigaBrain Team et al., 2025; Jang et al., 2025; Shang et al., 2025). (ii) *Closed-loop video-WMs with policies*, where a video generator is rolled out together with a VLA-style policy as a learned simulator for planning, value estimation, or evaluation in the loop (Guo et al., 2026; Jiang et al., 2026b; Liu et al., 2026; Yang et al., 2026). (iii) *World action models*, where action prediction and video prediction are jointly modeled inside a single generative world model (Hu et al., 2025; Kim et al., 2026; Li et al., 2026; Ye et al., 2026; Yuan et al., 2026; Zhu et al., 2025a). However, none of these works systematically address the *transition* from a generic video generator to an embodied world model: they implicitly assume the underlying video generator can already plan over multi-object, multi-step spatial-temporal structure, while generic videogen pre-training in fact only optimizes for visual fidelity and aesthetics.

3 METHOD

We first specify the form of the Thought channel (Sec. 3.1) and embed it into a dual-stream video diffusion backbone (Sec. 3.2). To finalize the Thought before RGB enters its main denoising phase, we introduce the *Timestep-Bias Reasoning Routing* schedule (Sec. 3.3). Sec. 3.4 describes two-stage training and the attention-based Thought CFG used at inference, and Sec. 3.5 presents *Thought Forcing*, an in-training Thought rollout that closes the train–inference gap.

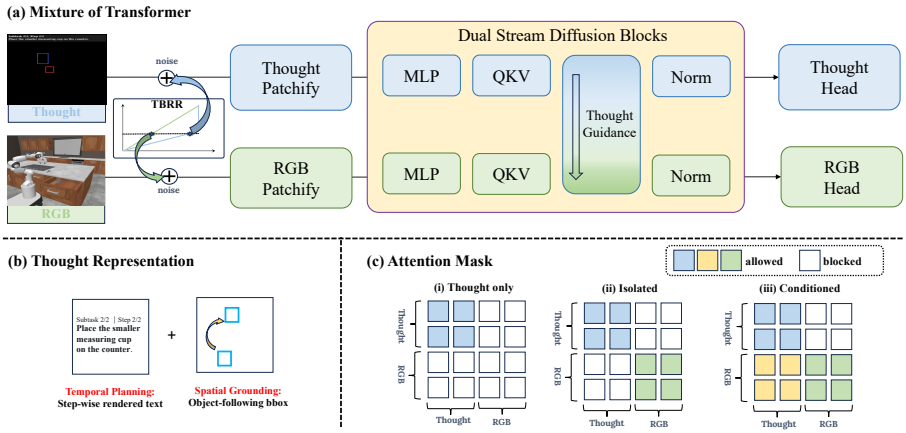

Figure 2: **Reasoning Canvas pipeline.** A dual-stream backbone (Sec. 3.2) co-generates the Thought channel (Sec. 3.1) with RGB; the TBRR scheduler (Sec. 3.3) finalizes Thought before RGB's low-noise stages. The three attention masks (Sec. 3.4) are **c-(i) Thought-only** for Stage 1, **c-(ii) isolated** (Stage-2 dropout, enables standalone RGB), and **c-(iii) conditioned** (RGB attends to both streams, default at training and inference).

3.1 Thought Channel: Spatial Grounding and Temporal Planning

We instantiate the Thought channel as a black canvas with two complementary visual primitives (Fig. 1): object-following bounding boxes for spatial grounding and step-wise rendered text for temporal planning.

Object-following bounding boxes for spatial grounding. For every frame we render colour-coded boxes of the gripper and the manipulated entity (object or fixture); boxes follow these entities across the entire episode, so a single Thought frame conveys *which* entities matter and *where* they are. This provides a modality-orthogonal spatial prior: even when the instruction is vague (e.g., "make a quick cheese snack with the bread"), the boxes still anchor the entity to manipulate and its destination.

Step-wise rendered text for temporal planning. We render the step-wise prompt near the top of the canvas, switching the active prompt at subtask/step boundaries (e.g. for `ChooseMeasuringCup`: *"Move to the right drawer"* $\rightarrow$ *"Open the right drawer"* $\rightarrow$ *"Pick up the smaller measuring cup"* $\rightarrow$ *"Place the smaller measuring cup on the counter"*). Step prompts encode the temporal program: which step is currently being executed and, by extension, what comes next, which is critical for long-horizon composite tasks.

3.2 Dual-Stream Video Diffusion Architecture

As shown in Fig.2, we place RGB and Thought inside a single shared DiT with stream-specific input/time embeddings and modulation, so the two streams share a visual prior yet carry *independent* timesteps as required by Sec. 3.3.

Preliminaries. We build on rectified flow (Lipman et al., 2023; Liu et al., 2022): a video $\mathbf{x}_0$ is encoded to a latent $\mathbf{z}_0 = \mathcal{E}(\mathbf{x}_0)$, the forward process linearly interpolates with $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{z}_t = (1-\sigma_t)\mathbf{z}_0 + \sigma_t\epsilon$, and v_θ is trained with $\mathcal{L}_{\mathrm{FM}}(\mathbf{z}_0, \mathbf{c}) = \mathbb{E}_{t,\epsilon}\|v_\theta(\mathbf{z}_t, t, \mathbf{c}) - (\epsilon - \mathbf{z}_0)\|^2$.

Two streams, one backbone. As shown in Fig. 2, We factorize each episode into an *RGB stream* $\mathbf{x}^{\mathrm{rgb}}$ and a *Thought stream* $\mathbf{x}^{\mathrm{th}}$, encoded by the same VAE to latents $\mathbf{z}_0^{\mathrm{rgb}}, \mathbf{z}_0^{\mathrm{th}}$. The two streams carry independent timesteps $t_{\mathrm{rgb}}, t_{\mathrm{th}}$ and noise samples; the only cross-stream interaction is a shared self-attention with a configurable mask $\mathbf{M} \in \{\mathrm{Thought_only}, \mathrm{isolated}, \mathrm{conditioned}\}$ (Sec. 3.4). The training loss sums the two flow-matching losses,

$$\mathcal{L} = \mathcal{L}_{\mathrm{FM}}(\mathbf{z}_0^{\mathrm{rgb}}, \mathbf{c}^{\mathrm{rgb}}) + \lambda_{\mathrm{th}}\,\mathcal{L}_{\mathrm{FM}}(\mathbf{z}_0^{\mathrm{th}}, \mathbf{c}^{\mathrm{th}}). \tag{1}$$

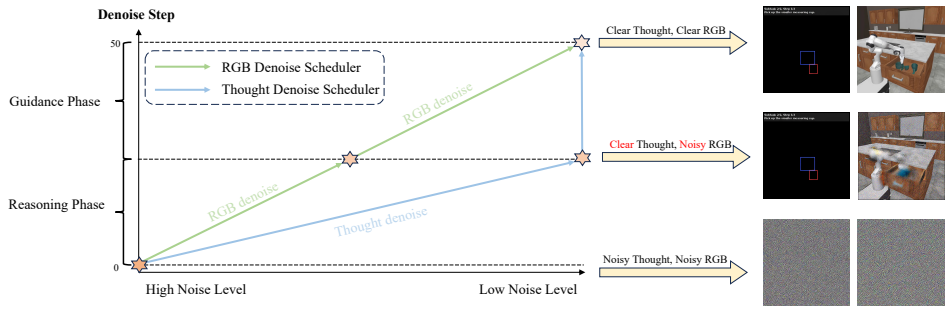

Figure 3: **Timestep-Bias Reasoning Routing (TBRR).** After the Reasoning phase the Thought stream is fully denoised; in the Guidance phase the clean Thought conditions the noisy RGB stream.

3.3 Timestep-Bias Reasoning Routing

If both streams share the same timestep, plan and pixels are denoised together and RGB never sees a clean Thought. The Thought stream is structurally simple (a black canvas with a few boxes and short captions) and admits high-quality generation in far fewer denoising steps than RGB. **We exploit this asymmetry by finalizing the Thought stream early and feeding the clean plan back into the RGB stream as a stable structured condition throughout the rest of its denoising trajectory.**

TBRR schedule. Given an RGB timestep t_{rgb}, we set

$$t_{\mathrm{th}} = g(t_{\mathrm{rgb}}) = \begin{cases} \dfrac{t_{\mathrm{rgb}} - \tau}{1 - \tau} & \text{if } t_{\mathrm{rgb}} > \tau, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

with $\tau = 0.9$ (i.e. $t_{\mathrm{rgb}} = 900$ on a 1000-step scale).

This compresses $t_{\mathrm{th}} \in [0, 1]$ into the high-noise sliver $t_{\mathrm{rgb}} \in [\tau, 1]$ and pins $t_{\mathrm{th}} = 0$ once $t_{\mathrm{rgb}} \le \tau$, partitioning the joint trajectory into two phases (Fig. 3): a **Reasoning phase** ($t_{\mathrm{rgb}} \in (\tau, 1]$) where the Thought completes its compressed schedule, followed by a **Guidance phase** ($t_{\mathrm{rgb}} \in [0, \tau]$) where the clean Thought conditions the remaining RGB denoising.

3.4 Two-Stage Training and Thought CFG

Aiming to stabilize training and provide an inference-time knob, we introduce a two-stage training strategy and a randomized cross-stream masking mechanism, the latter of which functions as Thought CFG at test time.

Stage 1: Thought-expert pretraining. We first train the backbone on the Thought channel only with the Thought-only attention mask (Fig. 2c-(i)), optimizing $\mathcal{L}_{\mathrm{FM}}(\mathbf{z}_0^{\mathrm{th}}, \mathbf{c}^{\mathrm{th}})$. This produces a strong Thought generator before joint optimization.

Stage 2: joint training with masked attention. Initialized from Stage 1, we jointly optimize Eq. (1) under the TBRR scheduler. Thought tokens attend only to Thought tokens in both modes. We randomize RGB's access to Thought via the mask $\mathbf{M}$ at each step:

$$\mathbf{M} \sim \begin{cases} \text{isolated} & \text{w.p. } 0.1 : \text{RGB self-attention only (Fig. 2c-(ii))}, \\ \text{conditioned} & \text{w.p. } 0.9 : \text{RGB attends to both streams (Fig. 2c-(iii))}. \end{cases} \tag{3}$$

The conditioned mode is the deployment regime; the isolated mode trains the Thought-unconditional branch needed for CFG.

Inference: combined text and Thought CFG. We compute three RGB velocity predictions:

$$v_\theta^{(\varnothing)} := v_\theta^{\mathrm{rgb}}(\,\cdot\,;\, \mathbf{c}_{\mathrm{txt}}^{-},\, \mathbf{M} = \text{isolated}) \qquad \text{(no text, w/o Thought)}, \tag{4}$$

$$v_\theta^{(\mathrm{txt})} := v_\theta^{\mathrm{rgb}}(\,\cdot\,;\, \mathbf{c}_{\mathrm{txt}}^{+},\, \mathbf{M} = \text{isolated}) \qquad \text{(text only)}, \tag{5}$$

$$v_\theta^{(\mathrm{txt,th})} := v_\theta^{\mathrm{rgb}}(\,\cdot\,;\, \mathbf{c}_{\mathrm{txt}}^{+},\, \mathbf{M} = \text{conditioned}) \qquad \text{(text and Thought)}, \tag{6}$$

with positive/negative prompts $\mathbf{c}^{\pm}_{\text{txt}}$. The guided velocity combines text-CFG and *Thought CFG* terms: $\hat{v}^{\text{rgb}}_\theta = v^{(\varnothing)}_\theta + s_{\text{txt}}\big(v^{(\text{txt})}_\theta - v^{(\varnothing)}_\theta\big) + s_{\text{th}}\big(v^{(\text{txt,th})}_\theta - v^{(\text{txt})}_\theta\big)$.

The Thought-CFG residual is the velocity change induced by toggling $\mathbf{M}$ from $\mathrm{isolated}$ to $\mathrm{conditioned}$: $s_{\text{th}} = 0$ recovers the RGB-only model and $s_{\text{th}} > 0$ amplifies the Thought-to-RGB guidance.

3.5 Thought Forcing

Even with TBRR, the Thought condition still suffers an exposure bias: in the Guidance phase ($t_{\text{rgb}} \le \tau$), training feeds RGB the clean ground-truth Thought $\mathbf{z}^{\text{th}}_0$, but inference feeds the model-generated $\hat{\mathbf{z}}^{\text{th}}_0$. *Thought Forcing* closes this gap with an in-training Thought rollout. Self Forcing (Huang et al., 2025) also addresses exposure bias through self-generated context, but uses autoregressive video rollouts and a video-level distribution-matching objective. We instead roll out the auxiliary Thought stream under stop-gradient and retain the RGB velocity loss.

For each sample with $t_{\text{rgb}} \le \tau$, we draw $t^{\text{init}}_{\text{th}} \sim \mathcal{U}[0,1]$ and $\epsilon^{\text{th}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, construct $\mathbf{z}^{\text{th}}_{t^{\text{init}}_{\text{th}}} = (1 - \sigma_{t^{\text{init}}_{\text{th}}})\mathbf{z}^{\text{th}}_0 + \sigma_{t^{\text{init}}_{\text{th}}}\epsilon^{\text{th}}$, and roll the Thought stream forward with K Euler steps under stop-gradient: $\hat{\mathbf{z}}^{\text{th}}_0 = \text{sg}\big[\Phi^{(K)}_\theta(\mathbf{z}^{\text{th}}_{t^{\text{init}}_{\text{th}}};\, t^{\text{init}}_{\text{th}} \to 0)\big]$.

With probability p_{TF} we replace the clean Thought in the RGB branch with $\hat{\mathbf{z}}^{\text{th}}_0$ and minimize

$$\mathcal{L}^{\text{rgb}}_{\text{TF}} = \mathbb{E}\left\| v^{\text{rgb}}_\theta(\mathbf{z}^{\text{rgb}}_{t_{\text{rgb}}}, t_{\text{rgb}};\, \mathbf{c}^{+}_{\text{txt}},\, \hat{\mathbf{z}}^{\text{th}}_0) - (\epsilon^{\text{rgb}} - \mathbf{z}^{\text{rgb}}_0) \right\|^2. \tag{7}$$

The Thought-channel loss in Eq. (1) is unchanged, so v^{th}_θ stays supervised on its full trajectory while v^{rgb}_θ sees the same noisy Thought it will see at inference.

4 Experiments

4.1 Experimental Setup

Models and training. All variants use Wan2.2-TI2V-5B (Team Wan et al., 2025) with frame-bucket sampling for variable-length videos. No Thought denotes the RGB-only Wan2.2-TI2V-5B baseline without the auxiliary Thought stream. Stage 1 trains a Thought expert in isolation; Stage 2 trains the visual stream with Thought guidance under TBRR. Training details are in Appendix E.

Dataset, metrics and benchmarks. The Thought-annotated **HierCanvas** dataset comprises simulation videos from RoboCasa (Nasiriany et al., 2024) (56,613 episodes across 311 task classes) and real-world videos (9,679 training and 641 validation episodes). We assess visual quality with FVD, PSNR, and SSIM, instruction alignment with CLIPScore, and tracking accuracy with mATE and MvATE. Real-world-trained models are also evaluated zero-shot on 174 **PAI-Bench-G Robotics** (Zhou et al., 2025) image–instruction pairs with eight reference-free quality metrics. Construction and protocols are detailed in Appendices A and F.

4.2 Does Thought Benefit Embodied Video Generation?

We compare Reasoning Canvas with existing video generators, evaluate it on real-robot data, and examine task-specific gains. Full Thought-variant ablations and VLM evaluation are in Appendix B.1.

Comparison with Embodied Video Models. Table 1 includes four embodied-pretrained models: WoW-14B (Chi et al., 2025), Large Video Planner (LVP-14B) (Chen et al., 2026), LingBot-Video-MoE-30B-A3B (Ma et al., 2026), and RoboMaster-5B (Fu et al., 2026). These models are evaluated without HierCanvas adaptation; Cosmos-Predict2.5-2B (NVIDIA et al., 2025), Wan2.2-TI2V-5B, and Reasoning Canvas are adapted on the same HierCanvas training set. Reasoning Canvas leads on FVD, PSNR, and both tracking metrics. Against the adapted Wan baseline, it additionally improves instruction alignment while matching SSIM. Other models retain the highest SSIM and CLIPScore overall, so the clearest strengths are distributional quality and trajectory fidelity rather than uniform superiority on every metric.

Table 1: **Comparison with embodied video models and HierCanvas-adapted baselines.** Models are grouped by whether they are adapted on the HierCanvas training set; all rows use no ground-truth future guidance. Bold marks the best reported value in each column. †LingBot uses its official structured prompt rewriting.

	Visual Quality			Instr. Align.	Tracking Accuracy	
Method	FVD↓	PSNR↑	SSIM↑	CLIPScore↑	mATE↓	MvATE↓
Embodied pretrained						
WoW-14B	99.68	16.17	0.7640	18.80	21.29	7.51
LVP-14B	102.53	17.33	0.8248	17.94	33.61	13.19
LingBot-Video-MoE-30B-A3B†	130.18	16.88	0.7982	18.52	20.18	6.86
RoboMaster-5B	283.34	14.11	0.7064	**18.82**	31.80	12.04
HierCanvas adapted						
Cosmos-Predict2.5-2B	63.53	19.20	**0.8400**	18.77	17.11	6.18
Wan2.2-TI2V-5B	24.67	19.49	0.8300	18.37	12.21	5.56
Reasoning Canvas	**21.47**	**19.81**	0.8300	18.60	**10.26**	**4.59**

Evaluation on Real-robot Data. On the real-world subset of HierCanvas, Reasoning Canvas and the RGB-only baseline use the same training and validation episodes. Reasoning Canvas improves all six metrics; the separately trained simulation counterpart shows the same pattern (Table 2), supporting Spatial-Temporal Thought supervision beyond simulator-rendered videos.

Table 2: **Real-world and simulation results on HierCanvas.** Best per setting in bold.

Method	FVD↓	PSNR↑	SSIM↑	CLIPScore↑	mATE↓	MvATE↓
Real World						
No Thought	29.88	18.56	0.7928	20.63	14.53	6.72
Reasoning Canvas	**28.91**	**19.16**	**0.7938**	**20.74**	**12.29**	**5.98**
Simulation						
No Thought	25.05	18.77	0.8234	18.17	11.42	4.83
Reasoning Canvas	**22.75**	**19.00**	**0.8282**	**18.48**	**10.29**	**4.17**

We also evaluate these models on 174 **PAI-Bench-G Robotics** pairs, using official condition images as first frames and unchanged instructions without further adaptation. Reasoning Canvas improves seven of eight reference-free dimensions, most notably I2V subject/background consistency and imaging quality, with a slight overall-consistency decrease (Table 3). These results complement the within-domain evaluation by showing improved condition fidelity on an external robotics benchmark.

Table 3: **Zero-shot quality on PAI-Bench-G Robotics.** Best values in bold.

Method	I2V BG↑	I2V Subj.↑	BG Aesth.↑	BG Cons.↑	Imaging↑	Motion↑	Overall Cons.↑	Subj. Cons.↑
No Thought	0.9501	0.9319	0.4399	0.9262	0.5977	0.9846	**0.1987**	0.8885
Reasoning Canvas	**0.9593**	**0.9467**	**0.4466**	**0.9295**	**0.6078**	**0.9881**	0.1966	**0.8946**

Disentangling Spatial Grounding and Temporal Planning. Atomic manipulation requires locating the relevant objects and targets, while composite tasks additionally require coordinating multiple actions. To probe these demands, we compare Spatial Thought on atomic tasks grouped by action type and Temporal Thought on non-navigation composite tasks grouped by subtask count, each against the w/o Thought baseline.

On atomic tasks, Spatial Thought provides broadly positive gains across action types, consistent with boxes supplying object-level spatial guidance (Table 4). Temporal Thought gives its largest gains across all reported metrics on the longest compositions ($n \geq 3$), suggesting that rendered step text

is particularly useful for coordinating longer sequences of subtasks. Together, these observations support complementary spatial and temporal roles.

Table 4: **Disentangling spatial grounding and temporal planning.** Each cell reports the per-metric improvement of the corresponding Thought variant over the Wan2.2 (w/o Thought) baseline; positive is better. SSIM is reported $\times 100$.

Subset	ΔPSNR	ΔSSIM	ΔCLIPScore	ΔMvATE	ΔATE	ΔEPE
Spatial Grounding (Subtask Type)						
Pick & Place	+0.30	+0.83	+0.61	+2.21	+0.14	+0.26
Open & Close	+0.19	+0.67	+0.79	+1.37	+0.13	+0.04
Others	-0.18	+0.16	+0.52	+0.45	+0.00	+0.01
Temporal Planning (Number of Subtasks)						
n=1	+0.20	+0.41	-0.06	+1.33	+0.05	-0.32
n=2	-0.02	-0.05	+0.09	+1.45	+0.01	-0.27
$n \geq 3$	**+0.28**	**+0.63**	**+0.41**	**+2.84**	**+0.26**	**+0.34**

4.3 Does Thought Improve Robustness Beyond Standard Metrics?

We test robustness to ambiguous instructions and unseen task compositions.

Vague Instructions. Starting from the original instruction (L0), an LLM produces progressively vaguer rewrites (L1–L3). With generation and evaluation settings fixed, we measure each Thought variant's gain over the w/o Thought baseline (Fig. 4).

Spatial Thought's alignment and tracking gains generally grow as instructions become vaguer, consistent with boxes providing a spatial reference when the text is underspecified. This pattern is not universal: Temporal Thought is non-monotonic on tracking, and Spatial-Temporal Thought does not consistently gain a larger tracking advantage. PSNR shows no consistent trend.

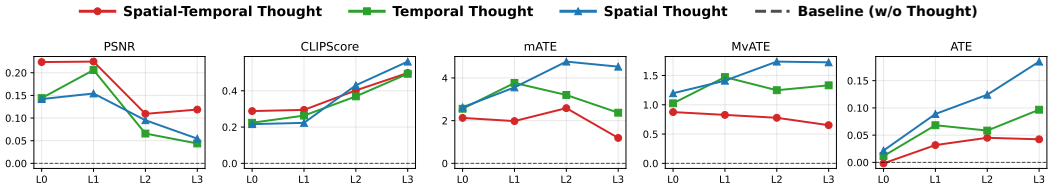

Figure 4: **Vague-instruction robustness.** Per-metric improvement of each Thought variant over the w/o Thought baseline as the instruction is progressively made more vague (L0→L3). Positive values indicate better than the baseline.

Out of Domain Task Compositions. We evaluate on 185 episodes that recombine atomic primitives into task compositions unseen during training.

Reasoning Canvas retains a consistent advantage over Wan2.2 (w/o Thought) across all five metrics on unseen task compositions (Table 5). These results support the role of explicit spatial grounding and temporal planning in compositional generalization, with benefits extending beyond the task combinations observed during training.

Table 5: **OOD compositions.** Evaluation on 185 episodes with unseen task compositions.

Method	FVD↓	PSNR↑	SSIM↑	CLIPScore↑	mATE↓
No Thought	35.45	18.10	0.7946	18.50	10.11
Reasoning Canvas	**32.33**	**18.39**	**0.8009**	**18.59**	**9.61**

4.4 What Makes Thought Guidance Effective?

We examine when Thought should guide RGB generation and how to handle the mismatch between clean training supervision and model-generated Thought at inference.

TBRR Threshold. TBRR determines when Thought is finalized relative to RGB: $\tau = 0$ synchronizes the streams, while larger thresholds finalize Thought earlier and leave more of RGB denoising under clean Thought guidance. In the sweep (Table 6), $\tau = 600$ gives the best FVD, PSNR, and tracking results, while SSIM and CLIPScore vary little. Compared with synchronized denoising ($\tau = 0$), this setting improves all four tracking metrics, although SSIM is slightly lower. Thus, a consistent gain in motion fidelity need not coincide with an improvement in every frame-level metric. The results favor moderate early guidance, not finalizing Thought as early as possible: the largest thresholds underperform this moderate setting on most metrics.

Table 6: TBRR threshold: Each row is a model trained with a different τ (1000-step scale).

τ	FVD↓	PSNR↑	SSIM↑	CLIPScore↑	mATE↓	MvATE↓	ATE↓	EPE↓
960	23.78	18.92	0.8266	<u>18.49</u>	9.95	4.18	0.788	1.038
900	<u>22.65</u>	19.00	0.8284	18.46	10.52	4.34	0.809	1.031
800	22.76	19.01	0.8285	**18.51**	9.90	4.07	0.791	1.039
600	**22.32**	**19.06**	0.8287	18.47	**9.06**	**3.84**	**0.763**	**0.962**
300	22.82	<u>19.04</u>	**0.8293**	18.48	9.50	3.90	<u>0.775</u>	<u>0.974</u>
0	23.03	19.03	<u>0.8289</u>	18.47	<u>9.43</u>	<u>3.85</u>	0.793	1.004

Train–Inference Gap: Thought Forcing. During the guidance phase, standard training supplies clean ground-truth Thought, whereas inference supplies the model's own predictions. Thought Forcing introduces stop-gradient model rollouts during training, exposing the visual stream to imperfect Thought. It improves five of six metrics, with a slight SSIM decrease (Table 7). This supports matching the training condition more closely to what is available at inference, rather than relying exclusively on clean Thought supervision. These improvements include FVD and both tracking metrics, so the benefit is not limited to instruction alignment. Taken together, the two ablations suggest that both the timing of guidance and the RGB stream's ability to use imperfect Thought matter for final video quality.

Table 7: Effect of Thought Forcing.

Method	FVD↓	PSNR↑	SSIM↑	CLIPScore↑	mATE↓	MvATE↓
No Thought	24.67	18.78	0.8238	18.18	12.65	5.22
Reasoning Canvas	<u>22.65</u>	<u>19.00</u>	**0.8284**	<u>18.46</u>	<u>10.52</u>	<u>4.34</u>
+ Thought Forcing	**21.47**	**19.05**	<u>0.8283</u>	**18.51**	**10.11**	**4.21**

5 Conclusion

We present **Reasoning Canvas**, a dual-stream video diffusion framework that decouples *what to do* from *how it looks* by co-generating visual content alongside an explicit Thought channel. This channel represents spatial-temporal reasoning through object-following bounding boxes and step-wise rendered text, which are resolved early via a TBRR scheduler to guide the subsequent RGB generation. Evaluated on a newly constructed, hierarchically annotated robot manipulation dataset, our approach significantly improves visual quality, instruction alignment, and tracking accuracy. Furthermore, it demonstrates stronger robustness to ambiguous instructions and better out-of-distribution generalization, validating a clear division of labor where bounding boxes anchor spatial grounding while text drives multi-step temporal planning.

Two limitations point to clear future directions. First, we evaluate Reasoning Canvas purely as a video generator and have not yet validated it on *downstream* world-model tasks. Second, even with TBRR and Thought Forcing, there remains a non-trivial gap between our *self-generated* Thought and directly conditioning the same RGB stream on *ground-truth* Thought (see Appendix D), indicating that Thought-channel quality is still a bottleneck. We leave both to future work.

AI USE STATEMENT

We used AI models to assist with step-caption generation, instruction rewriting for the vague-instruction benchmark, real-world pseudo-annotation, and auxiliary video evaluation, following the protocols described in the appendix. Generative AI tools also assisted with manuscript revision, interpretation and presentation of experimental results, and preparation of conceptual figures. The authors reviewed the AI-assisted content and checked it against the underlying data, experimental records, and source materials. The authors take full responsibility for the final manuscript, including its data, analyses, claims, and figures.

REFERENCES

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report, 2023. URL https://arxiv.org/abs/2309.16609.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

Boyuan Chen, Tianyuan Zhang, Haoran Geng, Caiyi Zhang, Peihao Li, Kiwhan Song, William T. Freeman, Jitendra Malik, Pieter Abbeel, Russ Tedrake, Vincent Sitzmann, and Yilun Du. Large video planner enables generalizable robot control, 2026. URL https://arxiv.org/abs/2512.15840.

Xiaowei Chi, Peidong Jia, Chun-Kai Fan, Xiaozhu Ju, Weishi Mi, Kevin Zhang, Zhiyuan Qin, Wanxin Tian, Kuangzhi Ge, Hao Li, Zezhong Qian, Anthony Chen, Qiang Zhou, Yueru Jia, Jiaming Liu, Yong Dai, Qingpo Wuwu, Chengyu Bai, Yu-Kai Wang, Ying Li, Lizhang Chen, Yong Bao, Zhiyuan Jiang, Jiacheng Zhu, Kai Tang, Ruichuan An, Yulin Luo, Qiuxuan Feng, Siyuan Zhou, Chi min Chan, Chengkai Hou, Wei Xue, Sirui Han, Yike Guo, Shanghang Zhang, and Jian Tang. Wow: Towards a world omniscient world model through embodied interaction, 2025. URL https://arxiv.org/abs/2509.22642.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. In-video instructions: Visual signals as generative control. *arXiv preprint arXiv:2511.19401*, 2025.

Xiao Fu, Xintao Wang, Xian Liu, Jianhong Bai, Runsen Xu, Pengfei Wan, Di Zhang, and Dahua Lin. Learning video generation for robotic manipulation with collaborative trajectory control, 2026. URL https://arxiv.org/abs/2506.01943.

GigaBrain Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jie Li, Jiagang Zhu, Lv Feng, Peng Li, Qiuping Deng, Runqi Ouyang, Wenkang Qin, Xinze Chen, Xiaofeng Wang, Yang Wang, Yifan Li, Yilong Li, Yiran Ding, Yuan Xu, Yun Ye, Yukun Zhou, Zhehao Dong, Zhenan Wang, Zhichao Liu, and Zheng Zhu. Gigabrain-0: A world model-powered vision-language-action model, 2025. URL https://arxiv.org/abs/2510.19430.

Nate Gillman, Yinghua Zhou, Zitian Tang, Evan Luo, Arjan Chakravarthy, Daksh Aggarwal, Michael Freeman, Charles Herrmann, and Chen Sun. Goal force: Teaching video models to accomplish physics-conditioned goals, 2026. URL https://arxiv.org/abs/2601.05848.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang,

Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025a. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Yanjiang Guo, Tony Lee, Lucy Xiaoyang Shi, Jianyu Chen, Percy Liang, and Chelsea Finn. Vlaw: Iterative co-improvement of vision-language-action policy and world model, 2026. URL https://arxiv.org/abs/2602.12063.

Ziyu Guo, Xinyan Chen, Renrui Zhang, Ruichuan An, Yu Qi, Dongzhi Jiang, Xiangtai Li, Manyuan Zhang, Hongsheng Li, and Pheng-Ann Heng. Are video models ready as zero-shot reasoners? an empirical study with the mme-cof benchmark. *arXiv preprint arXiv:2510.26802*, 2025b.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.595. URL https://aclanthology.org/2021.emnlp-main.595/.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations, 2025. URL https://arxiv.org/abs/2412.14803.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion, 2025. URL https://arxiv.org/abs/2506.08009.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loic Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. Dreamgen: Unlocking generalization in robot learning through video world models, 2025. URL https://arxiv.org/abs/2505.12705.

Chenxi Jiang, Chuhao Zhou, and Jianfei Yang. Rei-bench: Can embodied agents understand vague human instructions in task planning?, 2026a. URL https://arxiv.org/abs/2505.10872.

Zhennan Jiang, Shangqing Zhou, Yutong Jiang, Zefang Huang, Mingjie Wei, Yuhui Chen, Tianxing Zhou, Zhen Guo, Hao Lin, Quanlu Zhang, Yu Wang, Haoran Li, Chao Yu, and Dongbin Zhao. Wovr: World models as reliable simulators for post-training vla policies with rl, 2026b. URL https://arxiv.org/abs/2602.13977.

Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos, 2024. URL https://arxiv.org/abs/2410.11831.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning, 2026. URL https://arxiv.org/abs/2601.16163.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control, 2026. URL https://arxiv.org/abs/2601.21998.

Yifan Li, Yukai Gu, Yingqian Min, Zikang Liu, Yifan Du, Kun Zhou, Min Yang, Wayne Xin Zhao, and Minghui Qiu. Viper: Process-aware evaluation for generative video reasoning. *arXiv preprint arXiv:2512.24952*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Xiaokang Liu, Zechen Bai, Hai Ci, Kevin Yuchen Ma, and Mike Zheng Shou. World-vla-loop: Closed-loop learning of video world model and vla policy, 2026. URL https://arxiv.org/abs/2602.06508.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. URL https://arxiv.org/abs/2209.03003.

Xinxin Liu, Zhaopan Xu, Ming Li, Kai Wang, Yong Jae Lee, and Yuzhang Shang. Can world simulators reason? gen-vire: A generative visual reasoning benchmark. *arXiv preprint arXiv:2511.13853*, 2025.

Shuailei Ma, Jiaqi Liao, Xinyang Wang, Jingjing Wang, Chaoran Feng, Zijing Hu, Chong Bao, Zichen Xi, Yuqi Gan, Weisen Wang, Yanhong Zeng, Qin Zhao, Zifan Shi, Wei Wu, Hao Ouyang, Qiuyu Wang, Shangzhan Zhang, Jiahao Shao, Yipengjing Sun, Liangxiao Hu, Lunke Pan, Nan Xue, Kecheng Zheng, Yinghao Xu, Xing Zhu, Yujun Shen, and Ka Leong Cheng. Scaling mixture-of-experts video pretraining for embodied intelligence, 2026. URL https://arxiv.org/abs/2607.07675.

Xiaoyue Mi, Wenqing Yu, Jiesong Lian, Shibo Jie, Ruizhe Zhong, Zijun Liu, Guozhen Zhang, Zixiang Zhou, Zhiyong Xu, Yuan Zhou, et al. Video generation models are good latent reward models. *arXiv preprint arXiv:2511.21541*, 2025.

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, 2024. URL https://arxiv.org/abs/2406.02523.

Jim Nilsson and Tomas Akenine-Möller. Understanding ssim, 2020. URL https://arxiv.org/abs/2006.13846.

NVIDIA, 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, Yifan Ding, 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, Wei-Cheng Tseng, 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, and Yuke Zhu. World simulation with video foundation models for physical AI. *arXiv preprint arXiv:2511.00062*, 2025. URL https://arxiv.org/abs/2511.00062.

Yu Qi, Xinyi Xu, Ziyu Guo, Siyuan Ma, Renrui Zhang, Xinyan Chen, Ruichuan An, Ruofan Xing, Jiayi Zhang, Haojie Huang, et al. Mme-cof-pro: Evaluating reasoning coherence in video generative models with text and visual hints. *arXiv preprint arXiv:2603.20194*, 2026.

Yu Shang, Xin Zhang, Yinzhou Tang, Lei Jin, Chen Gao, Wei Wu, and Yong Li. Roboscape: Physics-informed embodied world model, 2025. URL https://arxiv.org/abs/2506.23135.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. Vlm-r1: A stable and generalizable r1-style large vision-language model, 2025. URL https://arxiv.org/abs/2504.07615.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu,

Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Chengzhuo Tong, Mingkun Chang, Shenglong Zhang, Yuran Wang, Cheng Liang, Zhizheng Zhao, Ruichuan An, Bohan Zeng, Yang Shi, Yifan Dai, et al. Cof-t2i: Video models as pure visual reasoners for text-to-image generation. *arXiv preprint arXiv:2601.10061*, 2026.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges, 2019. URL https://arxiv.org/abs/1812.01717.

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, Abraham Lee, Kuan Fang, Chelsea Finn, and Sergey Levine. BridgeData V2: A dataset for robot learning at scale. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1723–1736. PMLR, 2023. URL https://proceedings.mlr.press/v229/walke23a.html.

Maijunxian Wang, Ruisi Wang, Juyi Lin, Ran Ji, Thaddäus Wiedemer, Qingying Gao, Dezhi Luo, Yaoyao Qian, Lianyu Huang, Zelong Hong, et al. A very big video reasoning suite. *arXiv preprint arXiv:2602.20159*, 2026a.

Ruisi Wang, Zhongang Cai, Fanyi Pu, Junxiang Xu, Wanqi Yin, Maijunxian Wang, Ran Ji, Chenyang Gu, Bo Li, Ziqi Huang, et al. Demystifying video reasoning. *arXiv preprint arXiv:2603.16870*, 2026b.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. URL https://eceweb.uwaterloo.ca/~z70wang/publications/ssim.html.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800. URL https://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. Video models are zero-shot learners and reasoners. *arXiv preprint arXiv:2509.20328*, 2025.

Jialong Wu, Xiaoying Zhang, Hongyi Yuan, Xiangcheng Zhang, Tianhao Huang, Changjing He, Chaoyi Deng, Renrui Zhang, Youbin Wu, and Mingsheng Long. Visual generation unlocks human-like reasoning through multimodal world models. *arXiv preprint arXiv:2601.19834*, 2026a.

Yuhao Wu, Maojia Song, Yihuai Lan, Lei Wang, Zhiqiang Hu, Yao Xiao, Heng Zhou, Weihua Zheng, Dylan Raharja, Soujanya Poria, et al. From perception to action: An interactive benchmark for vision reasoning. *arXiv preprint arXiv:2602.21015*, 2026b.

Zeqi Xiao, Yiwei Zhao, Lingxiao Li, Yushi Lan, Ning Yu, Rahul Garg, Roshni Cooper, Mohammad H Taghavi, and Xingang Pan. Video4spatial: Towards visuospatial intelligence with context-guided video generation. *arXiv preprint arXiv:2512.03040*, 2025.

Tianshuo Xu, Zhifei Chen, Leyi Wu, Hao Lu, and Ying cong Chen. Motion forcing: A decoupled framework for robust video generation in motion dynamics, 2026. URL https://arxiv.org/abs/2603.10408.

Cheng Yang, Haiyuan Wan, Yiran Peng, Xin Cheng, Zhaoyang Yu, Jiayi Zhang, Junchi Yu, Xinlei Yu, Xiawu Zheng, Dongzhan Zhou, et al. Reasoning via video: The first evaluation of video models' reasoning abilities through maze-solving tasks. *arXiv preprint arXiv:2511.15065*, 2025a.

Jiazhi Yang, Kunyang Lin, Jinwei Li, Wencong Zhang, Tianwei Lin, Longyan Wu, Zhizhong Su, Hao Zhao, Ya-Qin Zhang, Li Chen, Ping Luo, Xiangyu Yue, and Hongyang Li. Rise: Self-improving robot policy with compositional world model, 2026. URL https://arxiv.org/abs/2602.11075.

Xiangpeng Yang, Ji Xie, Yiyuan Yang, Yan Huang, Min Xu, and Qiang Wu. Unified video editing with temporal reasoner. *arXiv preprint arXiv:2512.07469*, 2025b.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies, 2026. URL https://arxiv.org/abs/2602.15922.

Junchao Yi, Rui Zhao, Jiahao Tang, Weixian Lei, Linjie Li, Qisheng Su, Zhengyuan Yang, Lijuan Wang, Xiaofeng Zhu, and Alex Jinpeng Wang. Flowinone: Unifying multimodal generation as image-in, image-out flow matching. *arXiv preprint arXiv:2604.06757*, 2026.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination?, 2026. URL https://arxiv.org/abs/2603.16666.

Fengzhe Zhou, Jiannan Huang, Jialuo Li, Deva Ramanan, and Humphrey Shi. PAI-Bench: A comprehensive benchmark for physical AI, 2025. URL https://arxiv.org/abs/2512.01989.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets, 2025a. URL https://arxiv.org/abs/2504.02792.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: A fine-grained world model for robot manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025b. URL https://arxiv.org/abs/2406.14540.

A Datasets and Thought Construction

To train Reasoning Canvas we need video clips paired with two complementary annotations: per-frame bounding boxes for the gripper and the manipulated entities (the *spatial* signal), and per-step natural-language captions (the *temporal* signal). We construct the simulation subset of **HierCanvas** on top of RoboCasa (Nasiriany et al., 2024), exploiting the simulator's privileged physics state to recover task→subtask→step structure *directly* from interaction signals; the LLM is invoked only afterwards to *caption* that pre-extracted structure, so it cannot hallucinate subtasks that are not physically present.

A.1 Thought Channel Specification

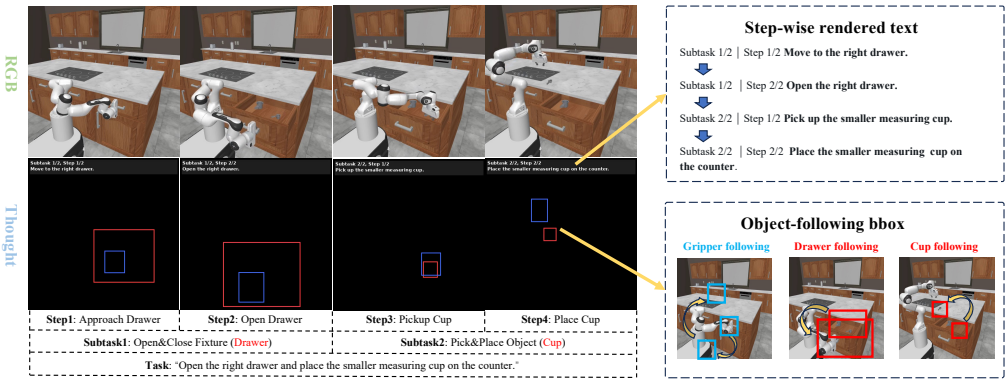

Figure 5: **Reasoning Canvas** (*best viewed zoomed in*). Two visual primitives share the same black canvas: **(i) Step-wise rendered text** along the top, naming the active subtask/step for temporal planning, and **(ii) Object-following bounding boxes** for spatial grounding (**blue** = gripper, **red** = manipulated entity of the current subtask).

The Thought channel is rendered as a regular video on a black canvas at the same spatial resolution and frame rate as the source RGB. To disentangle the contribution of each sub-modality, we render three variants over an identical underlying skeleton (subtasks, objects, captions, and boxes are byte-identical across variants; only which layers are drawn differs):

- **Spatial-Temporal Thought** (the default Reasoning Canvas channel). Per-frame bounding boxes for the gripper and the manipulated entity (object or fixture), *plus* the active step caption rendered as on-screen text near the top of the frame. Bounding-box colours encode role (*blue* for the gripper, *red* for the operated entity); the caption switches at subtask/step boundaries.
- **Spatial Thought** (boxes only). Same bounding boxes as Spatial-Temporal, with no rendered text.
- **Temporal Thought** (step prompts only). Same step captions as Spatial-Temporal, with no bounding boxes.

Because all three variants share the identical underlying skeleton, switching among them (or switching one off entirely as the RGB-only baseline) lets us isolate the contribution of bounding boxes vs. rendered text in a controlled way; the resulting ablations are reported in Sec. 4.2.

A.2 Simulation

A.2.1 Construction Pipeline

The pipeline processes one episode at a time in three stages, all keyed off the simulator's privileged state rather than LLM guesses, and is run sharded across many workers (on the order of $\sim$100 shards for the segmentation stage and $\sim$30-60 shards for the rendering stage) to scale across the full corpus.

Stage 1 – Subtask segmentation from physical signals. We replay every demonstration in the simulator and scan it frame-by-frame, combining three signals. **(i) Gripper grasp/release events.** A grasp/release primitive provided by the simulator marks every $(\textit{grasp} \rightarrow \textit{release})$ pair as one *pick-and-place* subtask; concurrent grasps within a 10-frame window are merged. The grasp routine differs slightly between the composite and atomic environments (different gripper joint ranges), but the segmentation logic is identical otherwise. **(ii) MuJoCo contact + joint q.** For each pick-and-place subtask we look back at the preceding interval; any non-object geometry that the gripper touches for ≥ 3 frames is traced up the body tree to its nearest jointed parent (excluding the robot itself). If $|\Delta q| > 0.05$ between contact start and end we register a *fixture-operation* subtask. Because this is purely a contact/joint test, it generalizes across fridges, microwaves, hinge cabinets, drawers, stove knobs, and faucets without any per-fixture special case. **(iii) Base displacement** is used to label each pick-and-place subtask with a navigation pattern in $\{pp, npp, pnp, npnp\}$. The same code path also selects an *adaptive camera* per subtask via grouped-visibility scoring—each operated entity contributes a small fixed bonus and a much larger bonus is awarded when *all* operated entities are simultaneously visible, which prevents a single large fixture from dominating the camera choice. For each episode this stage produces ordered subtasks (type, operated entity, target fixture, navigation pattern, frame ranges, and per-step actions), aligned per-frame bounding boxes for the gripper and operated entities, and a unified-viewpoint replay video at the frame stride and fps used for Thought rendering.

Stage 2 – LLM step captioning. A frozen open-source 9B-class instruction-tuned LLM takes the structural skeleton from Stage 1 and rewrites each step action (e.g., *"move to drawer"*) into a natural-language step caption (e.g., *"Move to the right drawer."*). Critically, the LLM is constrained to language rewriting only: it cannot add, remove, or reorder subtasks/steps, and on any parsing failure we automatically fall back to a deterministic action→template map (*"pick X"* → *"Pick up the X."*, *"operate Y"* → *"Operate the Y."*, etc.). This guarantees full coverage even for atomic-tail tasks while still avoiding hallucinated subtasks. The output for each episode is the Stage-1 skeleton augmented with a per-subtask list of step captions.

Stage 3 – Thought-video rendering. The renderer reads the Stage-1 / Stage-2 outputs and writes the three Thought variants on a black 512×512 canvas at the same fps as the unified-viewpoint RGB (we use frame-stride 6 / 20 fps). The default *Spatial-Temporal Thought* draws each frame's gripper box in blue, the operated-entity box in red, and stamps the active step caption at the top of the canvas in a fixed-width banner that switches at subtask/step boundaries (Fig. 1). *Spatial Thought* and *Temporal Thought* are obtained by toggling off the text and box layers respectively, leaving the underlying skeleton untouched.

Reproducibility. All random episode-level splits use a fixed seed ($= 42$). The full data-construction code and the resulting split definitions will be released with the paper.

A.2.2 Statistics, Splits, and OOD Design

Totals. The simulation subset of HierCanvas covers **311 task classes** (**247 composite + 64 atomic**) and **56,613 episodes** in total (**40,238** composite **+ 16,375** atomic). After the OOD hold-out and the 95/5 episode split below, the training pool is **48,935** episodes (33,371 composite + 15,564 atomic).

Composite difficulty grading. We grade each composite task along two axes: an **instruction-ambiguity score** $\in \{0, 1, 2\}$—0 if the instruction explicitly names the target object/location (e.g., *"Put the two ice cubes in the blender."*), 1 if it requires semantic understanding or implicit grouping (e.g., *"Place the drinks on the dining counter."*), 2 if it requires state reasoning or conditional judgement (e.g., *"Clear the non-vegetable object off the cutting board."*)—and a **structural-complexity score** $\in \{0, 1\}$ (1 iff the task has ≥ 3 subtasks or ≥ 2 navigations). Their sum yields three difficulty bins: *Easy* (86 tasks; ∼35%), *Medium* (100 tasks; ∼40%), *Hard* (61 tasks; ∼25%).

OOD selection. To probe whether the Thought channel helps with *novel atomic-primitive compositions*, we hold out the highest-composition-complexity tasks as an OOD validation split. Each composite task receives a *composition-complexity score* (`comp_score`) that rewards (i) more subtasks (2→+1, 3→+2, ≥4→+3); (ii) coexistence of pick-and-place and fixture-operation primitives

(+2) plus a +1 for every type alternation in the subtask sequence; (iii) longer navigation chains ($\geq$ 2→+1, $\geq$ 4→+2); and (iv) global rarity of the subtask-type sequence ($\leq$ 3 occurrences in the corpus →+2, $\leq$ 7→+1). Within each difficulty bin we mark the top **15% by `comp_score`** as OOD, yielding **37 OOD tasks** (13 Easy + 15 Medium + 9 Hard) and **5,140 OOD episodes** (1,862 + 2,361 + 917). All episodes of these 37 tasks form the OOD validation set; their task identities are never seen during training. The full OOD task list is released alongside the dataset.

Train / Val split on the remaining 274 tasks. On the remaining **274 task classes** (210 composite + 64 atomic), we randomly split episodes *within each task* using a fixed seed (`seed` = 42): 95% form the training set and 5% form the in-distribution validation set. This makes the per-class distribution identical between train and in-distribution val, so an in-distribution val gap reflects generalization to unseen episodes of seen task classes, while an OOD val gap additionally reflects generalization to unseen task classes.

Resulting splits. Table 8 summarizes the three splits used in the experiments. The **311-sample evaluation subset** reported in the main paper is *one episode per task* drawn from the in-distribution val pool (247 composite + 64 atomic), and the **185-sample OOD subset** is five episodes per OOD task drawn from the OOD val pool.

Table 8: **HierCanvas simulation splits.** OOD validation tasks are the top-15% composition-complexity tasks within each difficulty bin (37 task classes); their episodes are never seen during training. The remaining 274 task classes are split 95/5 by episode for in-distribution training and validation.

Split	Task classes	Episodes	Notes
Train (in-dist.)	274 (210 comp. + 64 atomic)	48,935 (33,371 + 15,564)	95% of episodes per task
Val (in-dist.)	274 (210 comp. + 64 atomic)	2,538 (1,727 + 811)	5% of episodes per task
OOD Val	37 composite	5,140	100% of held-out task episodes
Total	311	56,613	—

A.3 Real World

We construct the real-world subset of HierCanvas from BridgeData V2 (Walke et al., 2023) trajectories processed by IRASim (Zhu et al., 2025b). Unlike the simulation subset, these recordings do not provide simulator contact and object-pose signals. We therefore recover object-following boxes and interaction phases from RGB observations, task instructions, and the recorded gripper action. The outputs are automatically generated, quality-filtered pseudo-annotations, rather than frame-by-frame human ground truth.

Object grounding and interaction localization. We parse the manipulated-object phrase from each instruction and query SAM 3 (Carion et al., 2025) with progressively simplified noun phrases and object synonyms. Generic prompts such as "object" or "thing" are not accepted as sufficient semantic grounding. When a usable gripper-action transition is available, we threshold the action at the midpoint of its observed range and use the longest closed interval to identify grasp and release. The seed frame is chosen two frames before closure. If the action trace is missing or uninformative, we instead infer an interaction interval from object–gripper proximity and coordinated motion, recording this fallback in the annotation metadata.

Object and gripper tracking. We obtain up to five seed-mask candidates for the operated object and gripper, and place grid queries inside each mask. CoTracker3 (Karaev et al., 2024) tracks these queries bidirectionally through the episode. Visible points and a robust partial-affine fit propagate each seed box; at least six visible points and a RANSAC inlier ratio of 0.35 are required for direct tracking reliability. Candidate object–gripper pairs are scored jointly using semantic agreement, contact proximity, motion-direction agreement, speed correlation, displacement, and separation from

the next-best pair. Short tracking gaps inside the interaction interval can be bridged using the paired object's or gripper's motion; these inferred boxes are explicitly distinguished from directly tracked observations.

Quality filtering. The first pass checks semantic specificity, interaction coverage of the object and gripper (at least 0.85 and 0.75, respectively), direct object-track reliability (at least 0.60), and normalized contact gap (at most 0.12). It also requires at least two of three motion/candidate checks: joint motion, nontrivial object motion, and a sufficiently separated winning candidate. This pass accepts 12,997 of 28,935 source trajectories across the original train/validation/test splits. A visual audit of 50 accepted training examples identifies recurrent false accepts, including wrong-object selection, static distractors, and late box loss. A second, metadata-based pass excludes transport trajectories with negligible motion or a semantic mismatch, severe trailing box loss, incomplete open/close tracks, and tiny incomplete object boxes. It removes 2,024 trajectories and retains 10,973 in total. These acceptance rates describe filtering, not annotation accuracy.

Thought rendering. The retained tracks are rendered on a black canvas, with the manipulated object in red and the gripper in blue. Interaction boundaries additionally define deterministic pre-interaction, interaction, and post-interaction step prompts. The real-robot experiments use full Spatial-Temporal Thought supervision, rendering both these boxes and the active step prompt on the same canvas. The original task instruction is also supplied as text conditioning. RGB and Thought videos have identical resolution, frame count, and frame rate; rendering does not resample time or change the retained annotations.

Final splits and integrity checks. We preserve the source IRASim train/validation/test membership and apply the final retained-episode manifest as the sole inclusion rule. The RGB-only and Reasoning Canvas training lists contain exactly the same 9,679 training episode IDs in the same order, and both evaluations use the same 641 validation IDs. The retained test split contains another 653 episodes and is not used in the reported comparison; we do not claim an environment- or task-held-out split for the real-world subset. Table 9 summarizes the train/validation videos used here. All retained RGB/Thought pairs are verified for matching resolution, frame count, and frame rate, with no missing or extra rendered episodes relative to the manifest. Validation and test episodes are excluded from training.

Table 9: **The real-world subset of HierCanvas used in the experiments.** All videos are 640×480 at 3 fps. Duration is computed from the raw frame count; frame statistics precede the model's temporal-length alignment.

Split	Episodes	Frames	Duration	Frames: min / median / p95 / max
Train	9,679	335,413	31.06 h	11 / 34 / 49 / 79
Validation	641	22,336	2.07 h	10 / 34 / 49 / 79

A.4 Vague-Instruction Robustness Benchmark

The 311 in-distribution evaluation prompts described in Sec. A.2.2 are deliberately templated during data construction: they explicitly name every target object and destination (e.g., *"Pick chicken drumstick from the fridge and place it on the plate on the counter."*). Such prompts cannot tell us how a video world model behaves when the human instruction is *vague*—a setting that is unavoidable in real embodied use, where users routinely under-specify their requests. To probe this, we derive a **vague-instruction benchmark** on top of the 311 in-distribution evaluation episodes by grading each prompt at four levels of vagueness (L0–L3), while keeping the ground-truth video, Thought channel, first-frame conditioning, and target-action sequence *unchanged*. Any quality gap across the four levels is therefore attributable to instruction ambiguity rather than to a different visual target.

Vagueness levels. For each of the 292 unique original prompts we generate three rewritten variants with *Claude Opus 4.7* and then manually verify and edit them. Together with the original prompt this yields four parallel evaluation prompts per episode:

- **L0 – Original.** The original template-driven prompt produced during data construction. All target objects, destinations, and actions are explicitly named.
- **L1 – Information-preserving paraphrase.** A syntactic rewrite with synonymous verbs; *all noun keywords are preserved*. E.g., *"Take the chicken drumstick out of the fridge and put it on the plate on the counter."* A robust language understander should be near-invariant to L0→L1.
- **L2 – Lexical drift to intent.** Specific noun phrases are replaced by hypernyms or short intent-level expressions, removing roughly a third of the surface lexical detail. E.g., *"Move the chicken onto the plate."* The action and the operated entity remain inferrable from world knowledge but are no longer matched word-for-word against the training distribution.
- **L3 – Implicit reference.** Explicit object names are replaced by anaphoric pronouns (*it / that / them*) so that target identity can only be recovered from the first-frame visual context that the model already conditions on. E.g., *"Put it on the plate."* This is the closest analogue to under-specified everyday instructions and stresses the model's ability to ground language in the observed scene.

All four levels share the same ground-truth videos and the same first frame; only the textual prompt changes. This deliberately decouples *language ambiguity* from *visual difficulty*, so the resulting L0→L3 degradation curve measures exactly how robust a model is to under-specified instructions.

Construction protocol. For every unique original prompt we (i) prompt *Claude Opus 4.7* with per-level rubrics that explicitly forbid changing any operated entity, destination, or action and require each rewrite to be a single self-contained imperative sentence; (ii) manually inspect every L1/L2/L3 triple and rewrite or reject any candidate that adds, drops, or alters a target object, deviates from the gripper trajectory in the ground-truth video, or introduces entities not visible in the first frame; and (iii) propagate the verified rewrites to all three Thought variants by aligning them row-by-row with the 311-episode in-distribution subset, so that for every $(\text{Thought variant}, \text{level})$ pair we obtain a 311-row evaluation set with the visual target fixed and only the textual prompt varied. The resulting benchmark is the basis for the vague-instruction robustness study reported in Sec. 4.3.

B Thought Ablations and Guidance Analyses

We first report the full Thought-variant and VLM comparisons complementing the main-paper experiments. We then ask whether Thought actually *guides* the visual stream by intervening on its strength and asking whether the visual stream responds. The complementary question of *how* this guidance reshapes the denoising process is deferred to Sec. C.

B.1 Thought-Variant Ablation and VLM Evaluation

Thought variants. On the same video diffusion backbone, we toggle the two layers of the Thought channel: **w/o Thought** (RGB-only baseline), **Spatial Thought** (boxes only), **Temporal Thought** (step prompts only), and **Spatial-Temporal Thought** (the default Reasoning Canvas with both).

Thought Variants Ablation. We compare Thought variants on the same backbone, with self-generated Thought at inference. All three improve the six video metrics over the RGB-only baseline (Table 10). Spatial Thought attains the best PSNR and SSIM; Temporal Thought attains the best CLIPScore and mATE; Spatial-Temporal Thought attains the best FVD and MvATE. These complementary strengths support combining the two cues, although no variant is uniformly best.

Table 10: Thought ablation: Four Thought variants on the in-distribution validation set.

Method	FVD↓	PSNR↑	SSIM↑	CLIPScore↑	mATE↓	MvATE↓
Wan2.2 (w/o Thought)	25.05	18.77	0.8234	18.17	11.42	4.83
Spatial Thought	<u>23.30</u>	**19.04**	**0.8294**	18.40	10.30	4.23
Temporal Thought	24.35	18.81	0.8249	**18.50**	**10.05**	<u>4.22</u>
Spatial-Temporal Thought	**22.75**	<u>19.00</u>	<u>0.8282</u>	<u>18.48</u>	<u>10.29</u>	**4.17**

We observe that the combined Spatial-Temporal Thought does not Pareto-dominate all single-channel variants. Spatial Thought tends to better preserve low-level reconstruction metrics (PSNR, SSIM), while Temporal Thought can improve instruction alignment (CLIPScore) and some tracking metrics (mATE). The combined variant mainly improves aggregate distributional quality (FVD) and motion consistency (MvATE), but the results suggest that spatial and temporal reasoning channels may introduce different trade-offs.

Table 11: Gemini 3.1 Pro scores (0–5) on five axes, on the same generation results as Table 10.

Method	Semantic	Temporal	Geometric	Instr. Ground.	Phys. Plaus.	Avg5
Wan2.2 (w/o Thought)	2.904	2.945	2.415	3.379	1.643	2.657
Spatial Thought	2.881	2.904	2.437	3.341	1.656	2.644
Temporal Thought	**3.006**	**3.019**	2.421	**3.405**	**1.714**	2.713
Spatial-Temporal Thought	2.990	3.006	**2.479**	3.379	**1.714**	**2.714**

In Table 11, we further evaluate the same generation results with Gemini 3.1 Pro. The VLM is asked to score each video on semantic correctness, temporal consistency, geometric accuracy, instruction grounding, and physical plausibility. Spatial-Temporal Thought attains the best Avg5 and the best geometric score, and is tied with Temporal Thought on physical plausibility. Temporal Thought is strongest on semantic, temporal, and instruction-grounding axes. Spatial Thought is slightly below the RGB-only baseline on language-heavy axes (semantic, temporal, instruction grounding) but marginally exceeds it on geometric and physical plausibility, consistent with the metric-level reading that rendered text mainly contributes temporal and semantic information while bounding boxes contribute spatial grounding.

B.2 Global View: Structural-CFG Sweep

Structural CFG provides a direct knob s_{struct} that controls the strength of Thought-to-visual guidance. Sweeping this knob lets us intervene on the Thought signal while keeping the model and input fixed.

Table 12: **Structural CFG sweep — full metric breakdown.** Visual quality (FVD/PSNR/SSIM), instruction alignment (CLIPScore), and tracking accuracy (ATE/mATE/MvATE/EPE; thresholded $\delta@k$ for $k \in \{5, 10, 20\}$). Bold / underline mark the best / second-best value per metric across the sweep. CLIP-I, MDA and Survival have negligible variation across the sweep and are omitted.

s_{struct}	Visual quality			Alignment	Tracking error				Tracking $\delta@k$		
	FVD↓	PSNR↑	SSIM↑	CLIPScore↑	ATE↓	mATE↓	MvATE↓	EPE↓	δ_5 ↑	δ_{10} ↑	δ_{20} ↑
0.0	33.51	18.60	0.8216	18.53	0.780	10.63	4.51	0.968	0.9752	0.9852	0.9914
0.2	33.66	18.70	0.8229	18.53	0.775	9.75	4.24	0.966	0.9750	0.9852	0.9918
0.5	35.17	18.91	0.8247	18.62	0.757	8.53	3.54	0.968	**0.9769**	**0.9873**	**0.9934**
0.7	32.59	19.01	0.8251	**18.63**	**0.747**	**7.94**	**3.42**	**0.963**	**0.9769**	0.9865	0.9929
0.9	27.15	**19.02**	0.8259	18.56	0.771	8.71	3.69	0.977	0.9742	0.9849	0.9924
1.0	22.53	19.01	**0.8279**	18.44	0.800	10.24	4.15	1.016	0.9722	0.9831	0.9910
1.2	23.98	18.98	0.8268	18.47	0.801	10.20	4.20	0.986	0.9729	0.9830	0.9907
1.5	22.66	18.94	0.8269	18.46	0.811	10.57	4.26	1.038	0.9718	0.9826	0.9904
1.7	**22.37**	18.91	0.8268	18.41	0.794	10.54	4.35	0.989	0.9721	0.9824	0.9903
2.0	22.68	18.88	0.8267	18.41	0.809	10.77	4.41	1.020	0.9711	0.9818	0.9901
2.5	23.12	18.83	0.8262	18.39	0.811	10.91	4.41	1.036	0.9715	0.9815	0.9899

Changing s_{struct} produces a clear, controllable response in the visual stream. **Every reported metric improves over the no-guidance setting** ($s_{\text{struct}} = 0$) at some non-zero s_{struct}, and the improvements span all three axes: visual quality (FVD $33.51 \to 22.37$ at $s_{\text{struct}} = 1.7$, PSNR $18.60 \to 19.02$ at 0.9, SSIM $0.8216 \to 0.8279$ at 1.0), instruction alignment (CLIPScore $18.53 \to 18.63$ at 0.7), and tracking accuracy at every granularity (ATE $0.780 \to 0.747$, mATE $10.63 \to 7.94$, MvATE $4.51 \to 3.42$, EPE $0.968 \to 0.963$, all at 0.7; thresholded $\delta_5/\delta_{10}/\delta_{20}$ peak at 0.5–0.7). Different

metrics peak at different s_{struct} values—visual-quality metrics that prefer sharp pixels keep improving at higher CFG, while alignment- and trajectory-faithfulness metrics peak earlier in the 0.5–0.7 band—but the overall picture is unambiguous: *strengthening the Thought condition consistently helps the visual stream*. Pushing s_{struct} much beyond this band degrades the trajectory-faithful metrics again, which is expected in co-generation: the self-generated Thought is useful but imperfect, so excessive guidance also amplifies its errors. The non-monotonic response, together with the fact that no metric peaks at $s_{\text{struct}} = 0$, rules out the trivial explanation that the Thought channel is merely decorative.

B.3 Sample View: Per-Sample IoU vs. Advantage

We complement the global CFG sweep with a per-sample analysis on the 311-sample val split. For each sample we measure Thought quality on the dual-stream output by the mean bounding-box IoU of the rendered gripper / operated-object boxes against ground truth, and define the per-sample *RC-ST advantage* on every metric as Spatial-Temporal Thought minus the w/o Thought baseline (sign-corrected so that a positive value always means RC-ST wins). We then ask whether samples whose Thought channel is drawn more accurately are also the samples on which RC-ST wins by more.

Figure 6 shows three representative scatter plots, one per metric category. The trend is clearly stronger on the metrics that depend on object identity, position and subtask order than on pure pixel reconstruction: SSIM advantage stays nearly flat as Thought IoU grows (r=+0.10), whereas CLIPScore advantage (r=+0.15, p<0.01) and ATE advantage (r=+0.15, p<0.01) both rise visibly with better Thought. The same pattern holds across the rest of the tracking suite (δ_{20} r=+0.12, p<0.05; survival rate r=+0.12, p<0.05), and is even stronger if we condition CLIPScore on the operated-object IoU (r=+0.20, p=0.001). Stratifying by IoU quartile makes the same trend visible from a different angle: between the lowest- and highest-IoU quartiles, the RC-ST advantage on mATE grows by +2.07, MvATE by +0.40, CLIPScore by +0.30, and ATE by +0.14, while SSIM and PSNR advantages stay essentially constant. This rules out a pure dataset-level regularization explanation—holding the instruction fixed, samples whose Thought is more accurate are exactly the samples on which RC-ST wins by more on geometry- and alignment-driven metrics—and is consistent with the claim that Thought injects a spatial-temporal program into the visual stream rather than acting as a perceptual super-resolver.

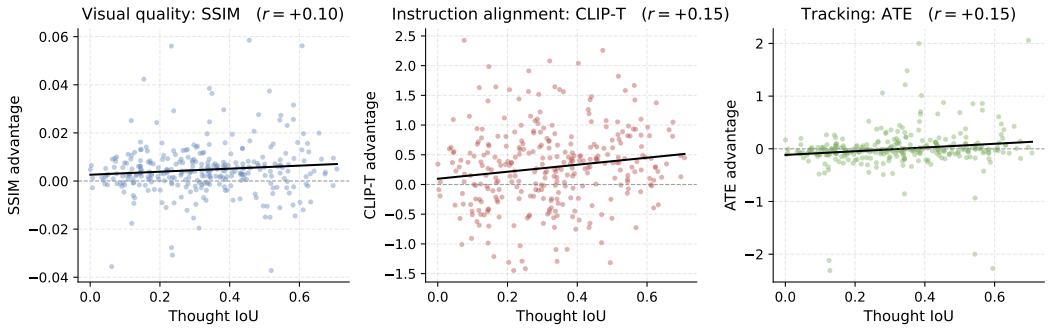

Figure 6: **Per-sample: Thought quality vs. RC-ST advantage.** Each point is one of the 311 val samples; x is mean thought-bbox IoU of the dual-stream output, y is the per-sample advantage of RC-ST over w/o Thought on the metric (positive ⇒ RC-ST wins). Black line is the linear fit. SSIM stays roughly flat with Thought quality, while CLIPScore and ATE advantages rise visibly.

C Mechanism Analysis

The previous section establishes *that* Thought guides the visual stream. This section asks *why* the two key design choices behind that guidance actually work: (i) the asymmetric TBRR scheduler, which lets the Thought stream finish denoising in only a small fraction of the schedule before the RGB stream takes over; and (ii) the joint co-generation of Thought and RGB inside a single back-

bone, which we show reduces trajectory-superposition and accelerates the RGB stream's convergence.

C.1 WHY TBRR WORKS: A LATENT-RMS ASYMMETRY VIEW

TBRR partitions denoising into a *reasoning phase*—during which only the Thought stream is actively denoised while the RGB stream stays at high noise—followed by a *rendering phase* where the Thought is treated as clean conditioning and the RGB stream is denoised to completion. Our default switch (Sec. 4.4) sits at τ=900 in the canonical $[0, 1000]$ diffusion-timestep convention; equivalently t_{rgb}=0.9 on the normalised flow timeline, so the Thought stream is forced to converge during the first 10% of the schedule and the remaining 90% is spent on RGB conditioned on the now-frozen Thought. The post-hoc threshold sweep (Table 6) finds a marginally better operating point at τ=600, but both values fall in the same well-posed regime that we characterise below. Why is this asymmetric direction (Thought ahead of RGB) the right one in the first place, and why does even a very small reasoning-phase budget such as 10% already work? Here we show that both fall out of an empirical asymmetry between the two streams' VAE latents.

Latent-RMS measurement. We encode 100 (RGB video, Thought video) pairs from the training distribution with the off-the-shelf Wan 2.2 VAE that is shared by both streams, and measure the per-stream root-mean-square activation of the resulting latents:

- $\text{RMS}_{\text{rgb}} = 0.836$
- $\text{RMS}_{\text{thought}} = 1.495$
- $\alpha \triangleq \text{RMS}_{\text{rgb}} \,/\, \text{RMS}_{\text{thought}} = 0.559.$

The Thought latent has $1.79\times$ larger per-element amplitude than the RGB latent. Standard deviation and mean absolute value give the same ratio to two decimal places, so this is a robust property of the latent representation rather than an artefact of any one summary statistic. This is consistent with the structural simplicity of the Thought signal—a mostly-black canvas with a small set of high-amplitude bbox edges and a single text banner—which after VAE encoding concentrates its energy into a few sharp activations rather than spreading it across natural-image textures.

Equivalent noise-schedule shift. Because both streams share the same Gaussian noise schedule $z_t = (1-t)\, z_0 + t\, \epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$, this RMS asymmetry directly shifts the effective signal-to-noise ratio: at any nominal timestep t, the Thought stream sees $\alpha^{-2} \approx 3.2\times$ higher SNR than the RGB stream. Equating the noise-to-signal ratio of the two streams gives a per-stream timestep remapping (i.e. the RGB-scale time at which RGB has the same noise-to-signal ratio as Thought at t):

$$t^{\text{eff}}_{\text{thought}}(t) \;=\; \frac{\alpha\, t}{1 + (\alpha - 1)\, t}.$$

Plugging in α=0.559 (Table 13), the Thought stream is consistently ahead on the denoising trajectory: at every nominal t, the Thought signal sits at the noise level RGB only reaches at a smaller (cleaner) effective t. The asymmetry grows as denoising progresses, peaking near the middle of the schedule.

Implication for the deployed τ=900 and the sweep optimum τ=600. At our default τ=900 (t_{rgb}=0.9) the Thought stream's effective RGB-time is 0.834, i.e. a lead of 0.066 on the RGB scale. In noise-to-signal terms, the Thought side is already $1.79\times$ closer to clean than the RGB stream at the same nominal t, so even at this very high noise level the phase switch is well-posed: when TBRR forces t_{thought}=0 at τ, the underlying VAE latent is the easier-to-denoise of the two and the conditioning the RGB branch sees in the rendering phase is the cleanest one available given the schedule. As t_{rgb} decreases, this lead widens (Table 13) and peaks near t_{rgb}=0.6 at 0.144, which is exactly where the post-hoc TBRR-threshold sweep (Sec. 4.4, Table 6) finds the marginal optimum: τ=600 pushes the phase switch to roughly the maximum-lead point, trading a small amount of late-stage RGB compute for a Thought that is allowed to denoise for 40% of the schedule rather than 10%. Pushing τ much higher (e.g. τ=950, lead ≈ 0.036) cuts the reasoning phase down to a 5% sliver where the Thought's natural lead has barely accumulated; pushing τ much lower (e.g. τ=300,

Table 13: **Equivalent noise-schedule shift induced by the latent-RMS asymmetry.** For each nominal RGB timestep t_{rgb}, $t^{\mathrm{eff}}_{\mathrm{thought}}$ is the RGB-scale timestep at which the RGB stream reaches the same noise/signal ratio as the Thought stream at t_{rgb}. *Lead* $= t_{\mathrm{rgb}} - t^{\mathrm{eff}}_{\mathrm{thought}}$ measures how far ahead the Thought stream is on the RGB scale. The bold row marks our default phase-switch τ=0.9; the lead grows further into the schedule and peaks near t_{rgb}=0.6, which the empirical sweep selects as the marginal optimum.

t_{rgb}	$t^{\mathrm{eff}}_{\mathrm{thought}}$	Lead
0.95 (very noisy)	0.914	0.036
0.90 *(=τ, default)*	**0.834**	**0.066**
0.80	0.691	0.109
0.60 *(empirical opt.)*	0.456	0.144
0.50	0.359	0.141
0.30	0.193	0.107

lead ≈ 0.107) keeps spending compute on a Thought that has already crossed the "noise $\leq$ signal" threshold and stops improving. Both extremes fall outside the well-posed band predicted here, and both are penalised in the empirical sweep.

This also explains why we do *not* have to pay the cost of true Latent-Forcing-style RMS rescaling or per-stream schedule rescheduling. TBRR converts the SNR asymmetry from a problem (mismatched difficulty between streams could in principle starve cross-stream learning) into a benefit: a single shared noise schedule plus a phase-switch threshold, chosen at the magnitude predicted by the RMS measurement, is enough.

C.2 Why Joint Reasoning Generation Works: Reduced Superposition and Faster Convergence

The previous subsection explains why the asymmetric TBRR *schedule* is well-posed; this subsection explains why the dual-stream architecture (Thought and RGB co-generated by the same backbone) actually changes what the visual stream does, beyond just acting as an external condition. Our hypothesis is that without an explicit Thought channel, the visual stream remains in a *trajectory-superposition* state: different seeds implicitly commit to different object choices, motion paths, or subtask orders, and the model spends a non-trivial fraction of the denoising budget collapsing this ambiguity. Co-generating an explicit Thought collapses that ambiguity earlier, so the visual stream can refine a committed future rather than spend steps deciding which future to follow.

We test this on the top-10 samples where Spatial-Temporal Thought most improves over the baseline. For each model, we save intermediate clean-video predictions $\hat{x}_0$ at denoising steps $\{1, 3, 5, 10, 20, 40\}$ under four random seeds, and report two quantities: *cross-seed diversity*, the mean pairwise MSE between $\hat{x}_0$ predictions across seeds at the same step; and *step-to-step convergence*, the MSE between adjacent $\hat{x}_0$ predictions within the same seed.

Table 14: **Denoising dynamics.**

Step	Cross-Seed Diversity↓			Convergence MSE↓	
	Wan2.2 (w/o Thought)	Spatial-Temporal Thought	Reduction	Wan2.2 (w/o Thought)	Spatial-Temporal Thought
1	0.0112	0.0117	−3.9%	—	—
3	0.0132	0.0120	9.0%	0.0128	0.0142
5	0.0131	0.0125	5.1%	0.0077	0.0056
10	0.0138	0.0125	9.2%	0.0032	0.0023
20	0.0148	0.0124	15.9%	0.0027	0.0023
40	0.0152	0.0125	17.7%	0.0013	0.0012

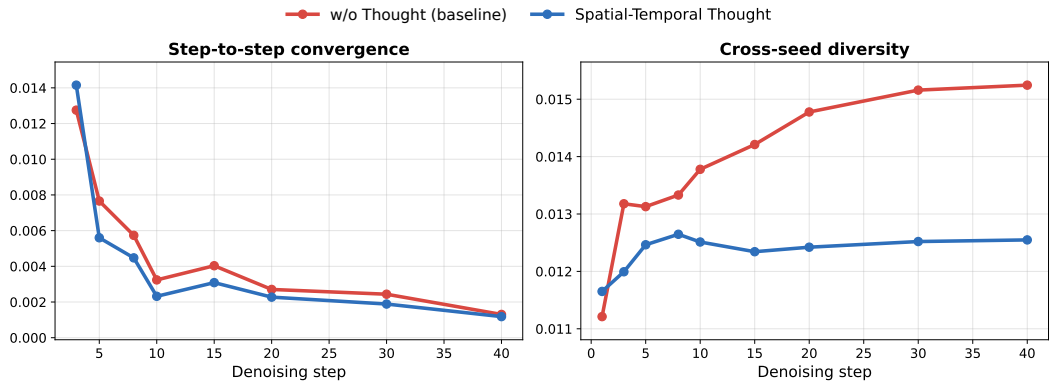

Figure 7: **Denoising dynamics on top-10 Thought-improvement samples.** *Left:* step-to-step convergence MSE within each seed. *Right:* cross-seed $\hat{x}_0$ diversity. With Spatial-Temporal Thought (blue) the visual trajectory both stabilizes earlier and ends up less seed-dependent than the w/o Thought baseline (red).

At step 1, Spatial-Temporal Thought is not less diverse than w/o Thought, so the gain is not simply reduced sampling randomness. From step 3 onward, cross-seed diversity becomes consistently lower with the Thought stream, and by step 40 the reduction reaches 17.7%. From step 5 onward, convergence MSE is also lower, indicating that the generated trajectory stabilizes earlier. These findings, visualised in Fig. 7, support the reasoning-as-planning interpretation: the Thought stream provides an explicit spatial-temporal program, and the visual stream spends fewer denoising steps resolving which future to follow. Note that this complements the TBRR analysis above: TBRR explains why the Thought *can be* cheaply produced first; the superposition analysis explains why having that Thought available actually *changes* the RGB stream's denoising behaviour, rather than only saving compute.

D Thought as External Condition: Upper Bound and Editing

The main paper studies Thought as a *co-generated* signal: the model produces both the Thought channel and the RGB channel from the same instruction in a single forward pass. A natural sibling question is what happens when Thought is supplied as an *external condition*—e.g. written by an upstream planner or a human user—instead of being co-generated. This experiment serves two purposes: it (i) gives a clean upper bound for how much our co-generated Thought is leaving on the table, and (ii) makes the Thought channel directly editable, so the same backbone can be steered to produce different videos by rewriting only the Thought.

D.1 Co-generated vs. ground-truth Thought

We compare three Thought variants in their co-generated form (used in the main paper) against the same variants when their Thought is replaced by the ground-truth Thought of the validation episode (**GT**). Across all three variants, GT-Thought conditioning consistently dominates co-generated Thought on every metric except CLIPScore (where the two are essentially tied), with the largest gaps on tracking-driven metrics: mATE drops from $10{-}11$ to roughly 4.4 and MvATE from ~ 4.2 to ~ 2.2 (Table 15). This sets a concrete upper bound: with a perfect Thought, the visual stream of the same Reasoning Canvas backbone roughly halves the trajectory error of the w/o Thought baseline, and roughly halves it again relative to its own co-generated Thought.

We further sweep Structural CFG with the GT Thought held as the conditioning signal (Table 16). Already at $s_{\text{struct}}{=}1.0$ all metrics enter their best regime, and the curve is essentially flat for $s_{\text{struct}}{\geq}1.0$—a clean signal that the visual stream willingly absorbs strong Thought guidance when the Thought itself is reliable. This is a useful counterpart to the co-gen sweep in Table 12: under co-gen the metrics peak in the 0.5–0.7 band and degrade beyond, because amplifying noisy Thought also amplifies its errors; under GT Thought there is no such trade-off.

Table 15: **Co-generated Thought vs. ground-truth Thought (the upper bound).** The three Thought variants are Spatial-Temporal, Temporal, and Spatial Thought, as defined in the main paper. *co-gen* = Thought is co-generated by our model from the instruction (no privileged information); *GT* = the validation-episode ground-truth Thought is supplied as the conditioning signal. GT-Thought consistently dominates co-gen Thought on tracking-driven metrics, defining the headroom for stronger Thought generators or upstream planners.

Method	FVD↓	PSNR↑	SSIM↑	CLIPScore↑	ATE↓	mATE↓	EPE↓	MvATE↓
Wan2.2 (w/o Thought)	24.67	18.78	0.8238	18.18	0.807	12.65	0.938	5.22
Spatial-Temporal Thought (co-gen)	22.65	19.00	0.8284	18.46	0.809	10.52	1.031	4.34
Spatial-Temporal Thought (GT)	16.44	20.69	0.8450	18.43	0.675	4.44	0.869	2.16
Temporal Thought (co-gen)	21.82	18.92	0.8275	18.39	0.785	10.03	1.005	4.02
Temporal Thought (GT)	15.77	20.70	0.8456	18.39	0.662	4.40	0.842	2.15
Spatial Thought (co-gen)	23.60	18.92	0.8264	18.40	0.795	10.09	0.989	4.19
Spatial Thought (GT)	22.16	19.04	0.8277	18.41	0.774	8.71	0.958	3.88

Table 16: **Structural CFG sweep with ground-truth Thought.** The visual stream saturates near $s_{\text{struct}}=1.0$ (**best per metric in bold**) and stays there as the knob is pushed further—no co-gen-style degradation, because the Thought signal is exact.

Condition	s_{struct}	FVD↓	PSNR↑	SSIM↑	CLIPScore↑	ATE↓	mATE↓	EPE↓	MvATE↓
GT	0.0	33.48	18.60	0.8216	18.54	0.779	10.48	0.985	4.50
GT	0.2	34.04	18.76	0.8237	18.54	0.775	9.91	0.971	4.19
GT	0.5	36.45	19.16	0.8278	**18.60**	0.719	7.28	0.933	3.20
GT	1.0	16.20	20.64	0.8442	18.45	**0.656**	4.35	0.848	2.18
GT	1.5	16.23	20.57	0.8430	18.46	0.657	**4.17**	**0.834**	2.10
GT	2.0	**15.50**	20.68	0.8446	18.42	0.661	4.24	0.851	**2.07**
GT	2.5	16.31	**20.70**	**0.8450**	18.38	0.661	4.21	0.839	2.09

D.2 Editable Thought as a Steering Wheel

Because the Thought channel is a self-contained, human-readable signal, exposing it as an external condition turns the same backbone into an *editable* video generator: rewriting the Thought rewrites what the robot does, with the first-frame RGB held fixed. Figure 8 demonstrates this on a kitchen scene containing a bowl, a spoon, and a cloth. From a single first frame we synthesize four different Thought channels—changing which object is grasped, where it is placed, and the linguistic action wording—and the same model then renders four distinct, instruction-faithful videos. Each row of Fig. 8 corresponds to one such edit: we plot five evenly sampled frames of the rendered Thought (left) next to five evenly sampled frames of the predicted RGB (right), so the bbox trajectory and the on-canvas caption can be read off the Thought half and the resulting motion read off the RGB half. We provide the full interactive workflow (SAM-assisted bbox, click-based waypoint editing, Thought rendering, and one-click inference) as a Gradio demo together with the paper.

Simplified Thought form for the editing demo. To keep the user-facing Gradio interaction lightweight, the demo uses a deliberately simplified Thought layout rather than the full multi-step Thought of the main paper: the canvas contains *a single rendered text line for the whole task instruction* (no per-step caption switching), and the bounding boxes only mark *the object to be operated on and its destination region* (no per-frame fine-grained tracking). This form is enough to express "move this to here, while saying that"—which is the user-facing primitive we want to expose—and lets the user place each scene in ~ 10 seconds: one click on the source object (SAM auto-segments it into a bbox), one click on the destination, a few optional waypoint clicks, and a free-text instruction. Because the same backbone is then conditioned on this simplified Thought, the demo's video quality is naturally a strict subset of what the full Thought enables, but it suffices for the purpose of showing that the model is steerable.

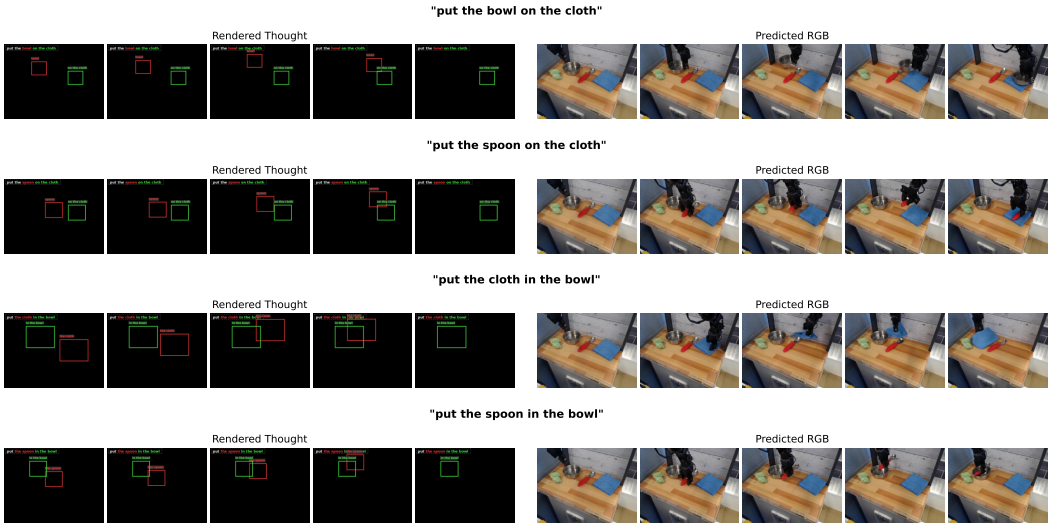

Figure 8: **Editable Thought.** Same first frame, four different Thoughts (the instruction in quotes above each row is the only thing that changes). For each edit we show three rendered-Thought frames at t=0, $T/2$, T on the left (bbox trajectory grows over time, on-canvas caption is fixed throughout the clip) and the predicted RGB at the same three time steps on the right. Rewriting the bbox trajectory and the action words on the Thought canvas redirects the same model to grasp a different object, place it at a different target, or swap the action verb—without retraining or touching the RGB stream.

E Training Details

We summarise the training-time configuration of the four main experiments on the simulation subset of HierCanvas side-by-side in Table 17. The Thought-Forcing extension (Sec. E.2) inherits column "RC–ST" of that table verbatim and is described separately below; the Stage 1 Thought-expert pretraining (Sec. E.1) is shared by all three Reasoning-Canvas columns and is also described separately.

Reading the table. The four columns correspond, in order, to: (i) the single-stream RGB-only **Baseline**, which trains the off-the-shelf backbone on RGB videos with the original prompt and ignores all Thought annotations; (ii) **RC–ST**, the default Reasoning-Canvas configuration with Spatial-Temporal Thought; (iii) **RC–T**, with Temporal-only Thought; and (iv) **RC–S**, with Spatial-only Thought. All three Reasoning-Canvas variants jointly train their Thought and RGB branches in Stage 2.

Joint optimisation. Stage 2 uses the objective in the main paper (Sec. 3.4, Eq. 1), $\mathcal{L} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{th}} \mathcal{L}_{\text{th}}$, with both branches trainable. Table 17 records the per-variant loss settings.

E.1 Stage 1 – Thought-Expert Pretraining

Each Reasoning-Canvas column in Table 17 initialises its Thought branch from a Stage 1 *Thought expert* that we pretrain once per Thought variant. The expert reuses the backbone's single-stream architecture and conditioning (RGB first frame + text prompt) but is trained with the corresponding Thought video as the target. Each Thought expert is trained for 24 epochs and the resulting weights are used both (i) as the Stage 2 Thought-branch initialisation and (ii) as a standalone Thought generator for the visualisations in the main paper.

Table 17: **Training-time configuration of the four main experiments.** *Settings shared across all four columns* (omitted from the table to save space): off-the-shelf Wan 2.2 TI2V-5B backbone (only the DiT is trained, text encoder and VAE are frozen); AdamW, lr $= 1 \times 10^{-5}$, `bf16` mixed precision, DeepSpeed ZeRO-2 sharding, gradient checkpointing; 512×512 video at 20 fps with up to 401 frames per clip; HierCanvas frame-bucket sampler (Sec. A.2.2) with boundaries $\{20, 40, 60, 80, 100, 130, 160, 200, 250, 300, 400\}$; 20 training epochs (the launch scripts schedule 50 but we observe validation metrics saturate by epoch 20, with two early-stopped runs reported at epoch 15 where flagged in the main-paper tables). The three Reasoning-Canvas columns additionally replicate *every* parameterised module of the DiT (norm, qkv, ffn, cross-attn, time, text, modulation) into two independent copies. RC–ST is our default Reasoning-Canvas configuration.

	Baseline	**Reasoning Canvas (dual-stream MoT)**		
Configuration	**(RGB only)**	**RC–ST** *(default)*	**RC–T**	**RC–S**
Stream count	1	2	2	2
Thought-channel target	—	Spatial-Temporal	Temporal	Spatial
Thought branch init. (Stage 2)	—	Stage 1 Thought expert (Sec. E.1)		
Thought branch in Stage 2	—	jointly trained	jointly trained	jointly trained
Thought-side velocity loss	—	MSE (λ_{th}=1)	MSE (λ_{th}=1)	MSE (λ_{th}=1)
Cross-stream attention	—	split (Thought→RGB causal)		
TBRR threshold T_{TBRR}	—	900		
Trainable parameters	full DiT	both branches	both branches	both branches

E.2 Thought Forcing – Implementation Details

This subsection details the implementation of *Thought Forcing* (Sec. 3.5 of the main paper). The default RC–ST model exposes a train/inference distribution gap on the Thought channel during the late RGB phase of TBRR ($t_{\text{Thought}} < 10^{-3}$): training feeds the clean ground-truth Thought, while inference fills that slot with the model's own rollout. We close this gap by replacing, on the fly, the clean ground-truth Thought with a self-rolled one whenever the sampled $t_{\text{Thought}} < 10^{-3}$: we run a short (4-step) Euler rollout of the Thought stream from pure noise down to $t = 0$ *using the current step's noisy RGB as cross-stream context*, pin the first frame to the first-frame latent at every substep, and then use this self-rolled Thought as the conditioning when computing the RGB velocity loss. The rollout is wrapped in a no-grad context, so gradients do not backpropagate through its sampling steps; this stop-gradient operation does not freeze the Thought branch during the Stage 2 parameter updates. We initialise this run from the RC–ST checkpoint and continue training for the same epoch budget, so that train and inference now see Thought channels drawn from the same self-rollout distribution.

F Evaluation Protocol

For the simulation studies on HierCanvas, we evaluate generated videos along two complementary tracks. **Track 1** reuses well-established *traditional metrics* that score pred-vs-GT fidelity at three semantic granularities: *visual quality*, *instruction alignment*, and *tracking accuracy*. **Track 2** uses a strong VLM (*Gemini 3.1 Pro*) as a paired-video judge along six axes, in the spirit of recent VLM-as-judge protocols for video generation. Both tracks share the identical evaluation samples described in Sec. A.2.2 (and the vague-instruction variants of Sec. A.4). We document every metric in full so that they can be selectively retained or dropped in the main text.

F.1 Traditional Metrics

For every $(\text{prompt}, \text{first frame}, \text{GT video})$ triple in the evaluation subset, the model generates one video at the same resolution and frame count as the GT. Predictions wider than the RGB resolution—

e.g. dual-stream outputs that horizontally concatenate the Thought channel and the RGB channel—are right-cropped to a 512×512 RGB before computing pixel-, feature-, or motion-level metrics; the Thought half-frame is consumed only by the Thought-channel metric below.

Visual quality. These metrics quantify per-pixel and per-feature fidelity to the GT video while ignoring the textual instruction.

- **PSNR.** Per-frame peak signal-to-noise ratio between aligned pred/GT frames after resizing both to 512×512, averaged within each sample and then across the dataset. Higher is better.
- **SSIM** (Wang et al., 2004). Per-frame structural similarity between aligned pred/GT frames, also at 512×512, averaged in the same way as PSNR. Higher is better.
- **FVD.** We extract spatiotemporal features with a Kinetics-pretrained I3D backbone from 224×224 inputs at 16-frame clips, segment each video into non-overlapping 16-frame chunks (rather than a single uniform clip per video) so that short manipulation videos still yield enough samples for a stable estimate, and report the Frechet distance between the resulting Gaussian feature distributions of the pred corpus and the GT corpus. Lower is better.

Instruction alignment. We use **CLIPScore** (Hessel et al., 2021) (text↛pred) as the sole instruction-alignment metric: for every frame of the pred video we embed both the prompt and the frame with a CLIP backbone and report $100 \cdot \cos\big(\mathrm{CLIP}_{\mathrm{text}}(\mathrm{prompt}),\ \mathrm{CLIP}_{\mathrm{img}}(\mathrm{frame})\big)$, averaged over frames. Higher is better.

Tracking accuracy. To capture manipulation-relevant motion that pixel- and feature-level metrics miss, we use **point-trajectory consistency via CoTracker3**: a fixed grid of query points is sampled on the *shared* first frame and tracked independently on the pred and GT videos to obtain two sets of T-step 2D trajectories anchored on the same query points, which we then compare point-by-point. We report the following metrics:

- **ATE** — Average Trajectory Error in pixels, averaged over time and over all query points (lower is better).
- **mATE** — motion-weighted ATE that re-weights each query point by its GT motion magnitude, suppressing the contribution of stationary background points and acting as the most informative single number for manipulation videos (lower is better).
- **MvATE** — ATE on moving points. We select points whose maximum GT displacement from the first frame exceeds 2 pixels, average each point's Euclidean pred–GT error over frames where both tracker visibility scores exceed 0.5, and then average equally over these points. Dataset results average valid per-video values, excluding videos with no jointly visible moving points. Unlike mATE, selected points receive equal, not motion-proportional, weights (pixels; lower is better).
- **EPE** — endpoint error on the last frame only (lower is better).
- **MDA** — motion direction accuracy, defined as the fraction of (point, time) entries whose pred and GT instantaneous motion vectors have positive cosine similarity (higher is better).
- $\boldsymbol{\delta@k}$ — the fraction of query points whose ATE is below k pixels, reported for $k \in \{5, 10, 20\}$ (higher is better).
- **Survival rate** — the fraction of query points that the tracker keeps confidently visible up to the last frame (higher is better).

F.2 VLM-as-Judge Evaluation with Gemini 3.1 Pro

The traditional metrics above all reduce a video to a fixed scalar (or a small set of scalars) by comparing pred-vs-GT in a fixed feature space. They cannot, however, cleanly capture higher-order properties such as "the gripper picks up the correct object but the object then phases through the drawer" or "the timing of the second sub-action is wrong". We therefore complement them with a paired-video judge that scores six interpretable axes.

Inputs. For every evaluation episode the judge receives **(i)** the textual instruction, **(ii)** the GT simulator rollout as the *reference* video, and **(iii)** the model's generated video as the *candidate* video. To keep each request inside the API's video upload budget while preserving temporal feel, both videos are pre-processed identically: we evenly sample at most 40 frames per video, re-encode at 8 fps with H.264 (CRF 26), and submit the two clips together with the instruction as a single multimodal prompt.

Six-axis rubric. The judge is asked to assign an integer score in $\{1, 2, 3, 4, 5\}$ along each of six axes (higher is better) and to return a single JSON object as the response. The full system prompt with per-score rubrics is reproduced verbatim in the prompt card below.

System prompt given to Gemini 3.1 Pro for paired-video evaluation

Setup. You are an expert evaluator for robot manipulation video generation. You are shown two videos of the same initial scene attempting the same instruction: VIDEO 1 is the REFERENCE (ground-truth simulator rollout, what should happen) and VIDEO 2 is the GENERATED candidate to be judged. Reward the GENERATED video for matching the REFERENCE in semantics, timing and geometry, and for being a physically plausible execution of the text instruction. Do *not* penalise stylistic differences (lighting, exact texture) that are unrelated to manipulation correctness.

Score the GENERATED video on an integer 1–5 scale along each axis:

- **Semantic Consistency** (vs. reference) – does it reproduce the same high-level action / sub-tasks?
  1: completely different action; 2: largely wrong but touches a related object; 3: roughly the right action but a key sub-step is missing or replaced; 4: the same action with only minor sub-step deviations; 5: every meaningful sub-action present and in the right order.
- **Temporal Consistency** (vs. reference) – same timing and ordering of motions?
  1: motion ordering is incoherent or chaotic; 2: many sub-motions occur in the wrong order or with wildly wrong duration; 3: rough order preserved, timing noticeably off (too slow / too fast / freezes); 4: timing matches reference within minor speed differences; 5: nearly frame-aligned timing of all sub-motions.
- **Geometric Consistency** (vs. reference) – faithful spatial trajectories and end-effector / object poses?
  1: trajectories bear no relation to reference; 2: large trajectory deviation, gripper visits very different regions; 3: trajectory shape matches but with substantial position errors; 4: small trajectory / pose deviations; 5: trajectory and poses closely match reference.
- **Instruction Grounding** (vs. text instruction; reference *not* considered) – does GENERATED physically act on the object/location named in the instruction?
  1: acts on a completely wrong object or does nothing relevant; 2: acts on a visually similar but wrong object; 3: targets the right object briefly but is not committed; 4: clearly targets the correct object/location, minor confusion; 5: unambiguous, focused interaction with the object/location stated in the instruction.
- **Physical Plausibility** (GENERATED only) – penalise melting / morphing, teleporting or disappearing/appearing objects, severe flicker, broken contacts, gripper passing through solids; do not penalise stylistic blur or minor lighting drift.
  1: severe artefacts throughout; 2: frequent but localised artefacts; 3: a few clear artefacts, mostly plausible; 4: occasional minor artefacts only; 5: physically plausible throughout.
- **Overall** – holistic judgement of GENERATED as a robot world-model prediction conditioned on the first frame and the instruction, reflecting a weighted summary of the five axes above.

Response format (strict JSON, no extra text):

```
{
  "semantic": <1-5>,
```

```
    "temporal": <1-5>,
    "geometric": <1-5>,
    "instruction_grounding": <1-5>,
    "physical_plausibility": <1-5>,
    "overall": <1-5>,
    "reason": "<one short sentence summarising the main
reason>"
  }
```

Aggregation and reproducibility. For each $(\text{model}, \text{axis})$ pair we report the dataset-level mean over the in-distribution evaluation samples and, separately, over the OOD evaluation samples. Because the VLM is non-deterministic, every sample whose API call rejects (rate-limit, finish-reason failure, or malformed JSON output) is automatically retried up to four times with exponential backoff; persistent failures are marked as errors and excluded from the per-axis mean (we observe that this affects $< 1\%$ of samples in our runs). The same downsampled videos, identical system prompt, and identical user prompt are used for all models and all conditions, so that the only variable across evaluation runs is the candidate video.

F.3 Real-robot Evaluation Protocols

HierCanvas real-world validation. We evaluate the RGB-only baseline and Reasoning Canvas trained with full Spatial-Temporal Thought (boxes and rendered step text) on all 641 validation trajectories. Both are conditioned only on the first RGB frame and the original task instruction; Reasoning Canvas generates its own Thought stream, without future ground-truth guidance. The reported checkpoints are the RGB baseline's `epoch-5` and Reasoning Canvas's Stage-2 `epoch-5`, with the latter using its Stage-1 `epoch-18` Thought expert. Generation uses 640×480 resolution, 3 fps, 50 RGB denoising steps, CFG 5, sigma shift 5, and seed 42; Reasoning Canvas uses joint TBRR with $\tau = 600$.

For an episode with N raw frames, generation uses the largest positive $4k + 1$ length not exceeding N, giving 9–77 generated frames and 21,353 frames in total per method. Evaluation includes every generated RGB frame and uniformly maps the complete raw ground-truth trajectory to that native output length. Thus, the alignment slightly normalizes temporal speed rather than discarding the end of the ground-truth episode. FVD uses non-overlapping 16-frame clips, padding a final partial clip by repeating its last frame, and is computed once from globally pooled I3D features rather than averaged across shards. PSNR, SSIM, CLIPScore (text–video), mATE, and MvATE are aggregated over all 641 episodes; tracking errors are in pixels. Thought videos are not included in the RGB metrics.

PAI-Bench-G Robotics. The same checkpoints trained on the real-world subset of HierCanvas are evaluated on all 174 official Robotics condition-image/instruction pairs, without benchmark-specific training or instruction rewriting. Each method generates 81 frames at 15 fps from the official condition image and original instruction, using 50 RGB denoising steps, CFG 5, and seed 42. Both follow the same aspect-ratio buckets; when the model requires an aligned internal resolution, decoded frames are center-cropped without interpolation to the requested output size. Only generated RGB videos are supplied to the official quality evaluator.

The eight reference-free dimensions are I2V background consistency, I2V subject consistency, aesthetic quality, background consistency, imaging quality, motion smoothness, overall consistency, and subject consistency. Higher is better on every dimension. These scores evaluate condition fidelity and visual quality; they do not establish successful execution of a robotic task or physical correctness. In Table 3, I2V-BG/Subj. denote I2V background/subject consistency, and BG/Subj. Cons. denote background/subject consistency.

G Qualitative Comparison

The numerical evidence so far—global metrics, structural-CFG sweeps, and the per-sample IoU correlation—all point to the same mechanism, but the failure modes that these scores summarise are

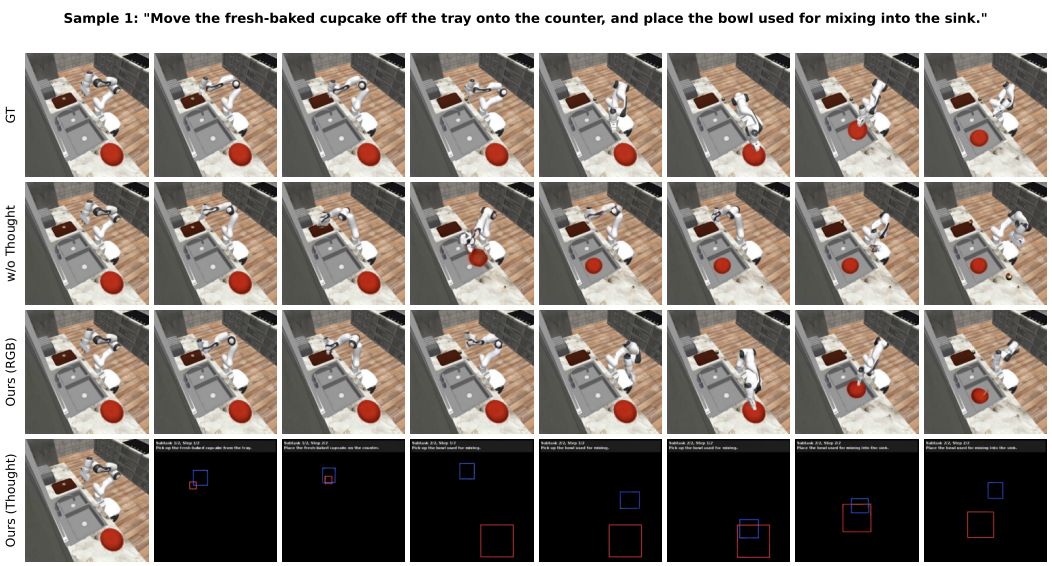

Figure 9: **Sample 1 (CUPCAKECLEANUP, two-object two-destination).** The baseline grasps the cupcake but releases it into the wrong region of the counter and then never moves the mixing bowl, dropping the second subtask entirely; with co-generated Thought the gripper visits both objects in the correct order and reaches both destinations.

easier to grasp visually. We therefore include a small gallery of qualitative co-generation comparisons against the w/o Thought baseline.

What the baseline gets wrong. On multi-object, multi-step instructions, the w/o Thought baseline has no explicit handle on *which* object the current step should act on or *where* to place it; it has to decide both inside the same RGB diffusion process that is also responsible for pixel rendering. The recurring failure modes we observe are exactly the ones this lack of an explicit plan predicts: **(i) object hallucinations** (objects appear out of nowhere, or the wrong object is summoned mid-trajectory), **(ii) wrong-category placement** (the right object is moved to a plausible-looking but semantically wrong destination), and **(iii) instruction-intent confusion** on prompts that require state-conditional or compositional reasoning, where the baseline simplifies the instruction to a single visible action and silently drops the rest. By co-generating an explicit plan—per-frame bounding boxes for spatial grounding and step-wise text for temporal planning—Reasoning Canvas exposes the "what to grasp / where to put / which step we are in" decision *before* the RGB stream commits to a future, which (as analysed in Sec. C.2) collapses cross-seed diversity earlier and accelerates step-to-step convergence. The end effect is that the visual stream no longer has to plan and render at once, and the failure modes above largely disappear.

Sample gallery. Figures 9–13 show one representative episode each, all rendered at 20 fps and uniformly sub-sampled to eight frames per row (top to bottom: GT, w/o Thought baseline, our co-generated RGB, our co-generated Thought). The five samples are picked to span the failure modes above: multi-object pick-and-place to different destinations (Samples 1, 3, 5), fixture-state preparation (Sample 2), and instruction-intent reasoning over object categories (Sample 4). Eight-frame strips inevitably hide most of the temporal dynamics, so the failure modes (object hallucinations mid-trajectory, momentary phase-throughs, mis-ordered subtasks) are noticeably easier to see in motion—we therefore include the full-length 20 fps videos for all five samples in the supplementary material and strongly encourage the reader to watch them alongside Figures 9–13.

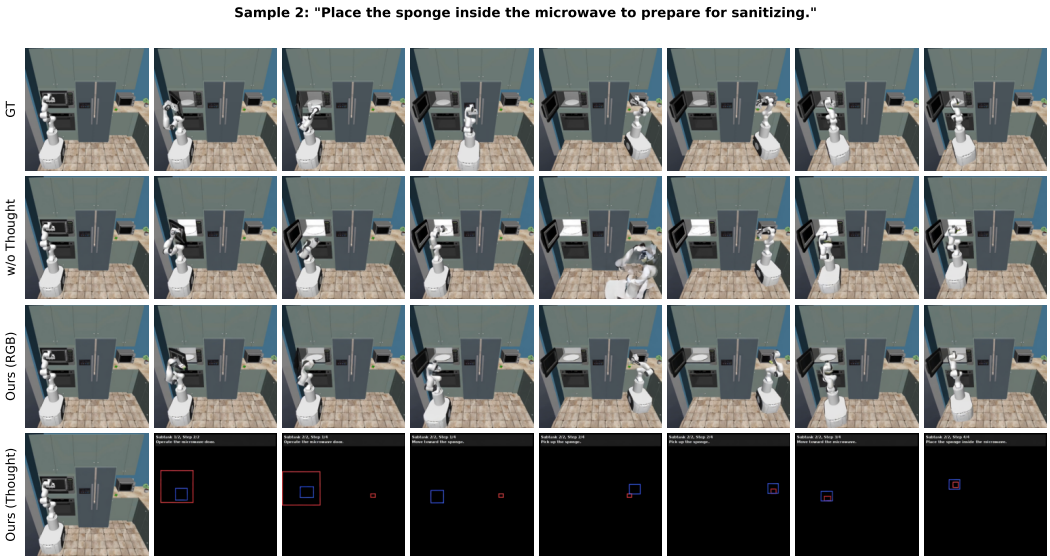

Figure 10: **Sample 2 (CLEANMICROWAVE, object hallucination inside a fixture).** The baseline correctly opens the microwave, but once the cavity is exposed it hallucinates a sponge *inside* the microwave and reaches in to grasp the phantom object, completely ignoring the real sponge sitting on the counter. With co-generated Thought, the bbox channel anchors the sponge to its actual on-counter location, so the RGB stream picks up the real sponge and only then places it into the opened microwave.

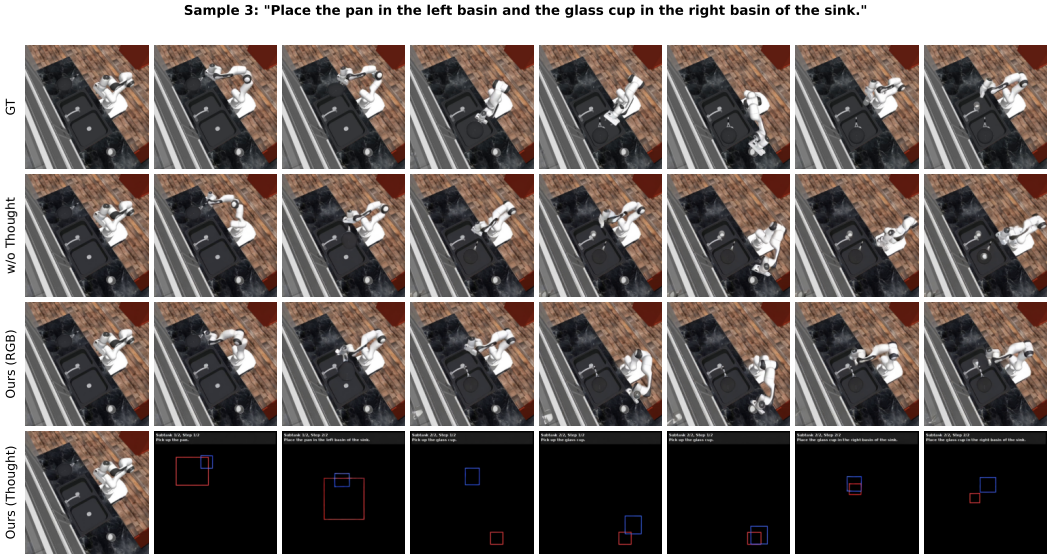

Figure 11: **Sample 3 (PRERINSESTATION, two-object two-basin).** The baseline confuses the two basins and tends to drop both items into the same side; the Thought channel encodes the per-step destination explicitly, so the pan ends up in the left basin and the glass cup in the right basin.

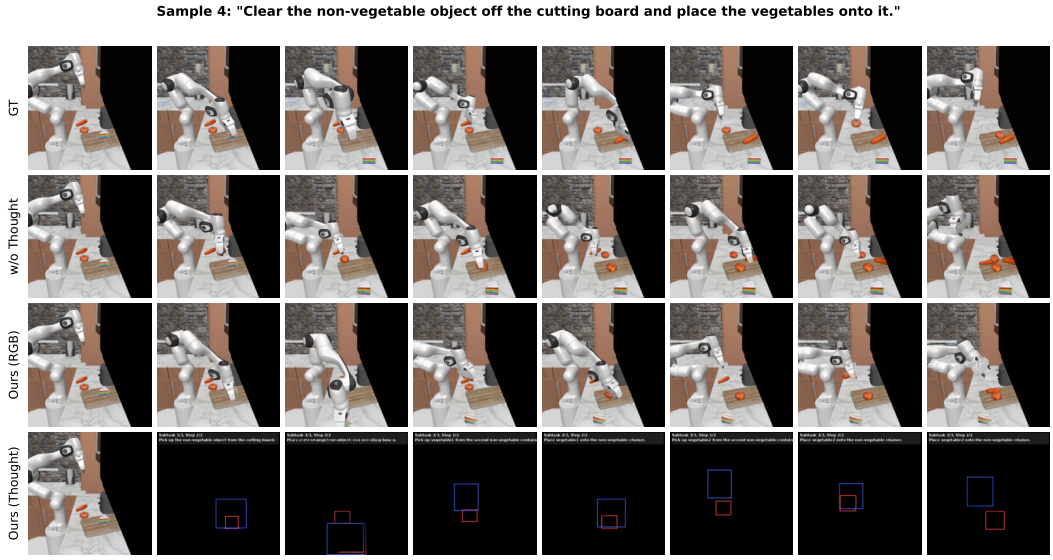

Figure 12: **Sample 4 (CLEARCUTTINGBOARD, instruction-intent reasoning).** The instruction implicitly partitions the scene into "vegetable" vs. "non-vegetable" objects. The baseline collapses the instruction into a single grasp action and acts on the wrong object (or hallucinates a missing one); the Thought channel pre-commits to the correct categorisation and the RGB stream follows it.

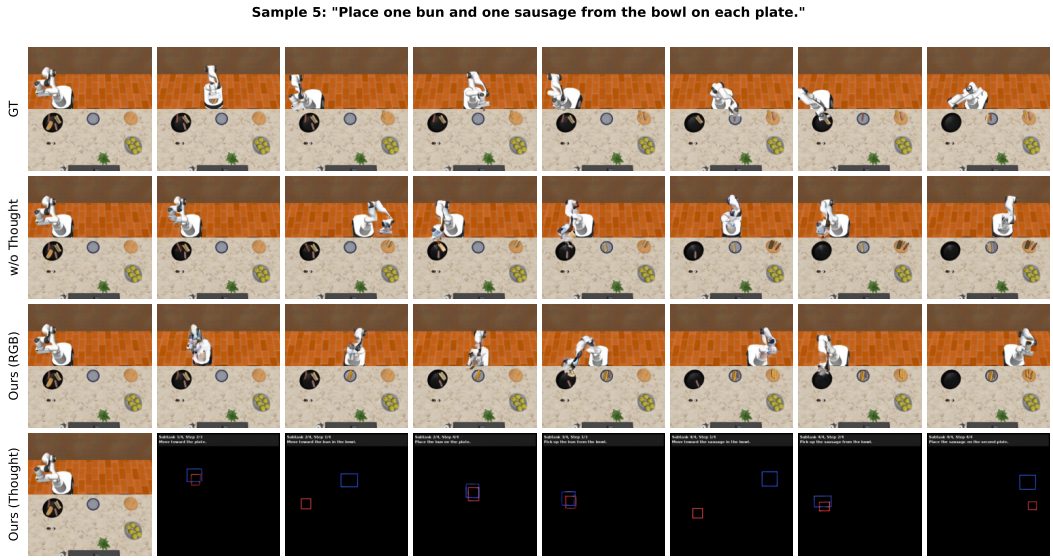

Figure 13: **Sample 5 (PORTIONHOTDOGS, multi-object distribution to multiple targets).** "One bun and one sausage on *each* plate" requires the model to alternate object types and destinations across four sub-pick-and-place steps. The baseline drops items onto a single plate and produces phantom buns; our co-generated Thought enumerates the four subtasks and the RGB stream visits each plate exactly once with the correct content.