VR-JEPA: LEARNING CONTRASTIVE-STATE LATENT GUIDANCE FOR GENERATION-BASED VIDEO REASONING

Anonymous authors
Paper under double-blind review

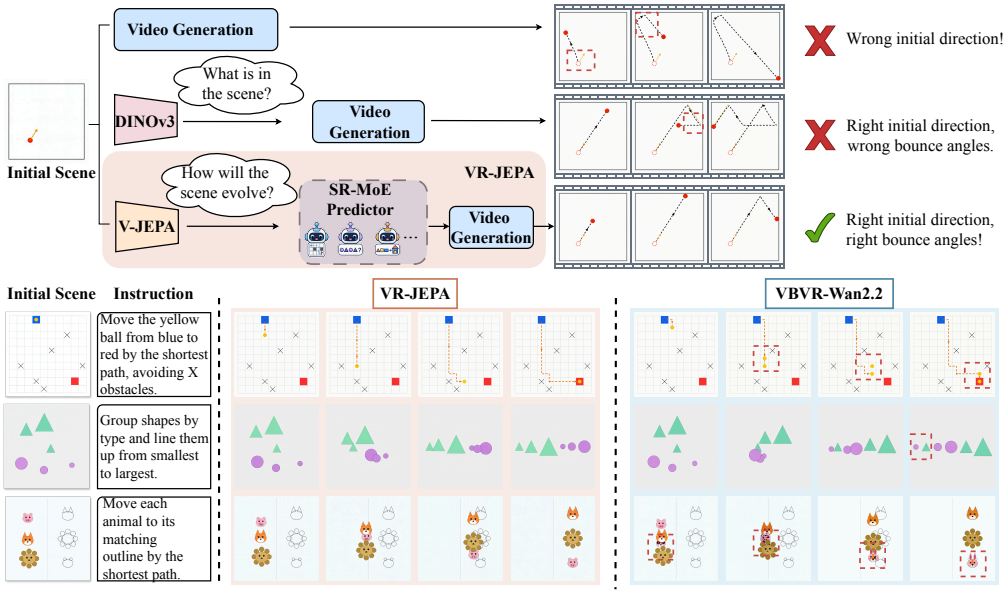

Figure 1: **Generation-based visual reasoning with VR-JEPA.** Top: Comparison of different video reasoning paradigms: (1) direct video generation; (2) video generation guided by static DINOv3 features of the initial state; and (3) VR-JEPA, which adapts V-JEPA's spatiotemporal priors to diverse reasoning skills and guides video generation through predicted latent trajectories. Bottom: Video reasoning results of VR-JEPA and VBVR-Wan2.2, showing that VR-JEPA generates trajectories that satisfy task requirements. Red dashed boxes highlight reasoning errors.

ABSTRACT

Reasoning through video generation offers a promising path toward visual intelligence by modeling latent visual states and their dynamics. However, current video generation models often lack explicit guidance on how these states should evolve, leaving generated trajectories prone to physical and structural inconsistencies that undermine reasoning reliability. While the Video Joint-Embedding Predictive Architecture (V-JEPA) provides rich spatiotemporal priors learned through latent prediction, these general priors do not naturally adapt to the logical reasoning capabilities required for complex visual tasks. To bridge this gap, we propose VR-JEPA, a framework that aligns the V-JEPA predictor with task-specific reasoning logic through localized contrastive-state learning and uses its predicted latent trajectories to guide video generation for visual reasoning. Specifically, (i) we pair successful trajectories with generated alternatives under the same input conditions and use discrepancies in their V-JEPA representations to identify informative states and tokens for localized contrastive supervision. (ii) We further equip the V-JEPA predictor with skill-specific experts trained on anchor-task data, allowing the model to adaptively specialize its shared spatiotemporal priors across

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.

diverse cognitive domains. Together with skill-specific experts, this contrastive supervision enables VR-JEPA to predict latent trajectories that provide task-specific logical guidance for video generation. Comprehensive experiments on the large-scale VBVR-Pro-Bench dataset demonstrate that VR-JEPA achieves an 11.33% relative improvement over the cutting-edge generation-based reasoning baseline, significantly mitigating physical artifacts and enhancing logical consistency. Our code and data will be released upon acceptance.

1 INTRODUCTION

Anticipating how the world will change is central to planning and problem solving, motivating world models that support prediction and decision making (Ha & Schmidhuber, 2018; Micheli et al., 2023; Du et al., 2024; Xiang et al., 2025). Video generation provides a setting for examining predicted scene evolution: it renders intermediate state transitions and final outcomes as a sequence. Diffusion models have made substantial progress in generating realistic and temporally coherent videos (Ho et al., 2022; Blattmann et al., 2023). However, their denoising objective does not by itself ensure that generated transitions satisfy a task's requirements. A video may look convincing while moving the wrong object or depicting an incorrect object relation. This gap motivates guidance that links generated state changes to task requirements (Liu et al., 2026; Cheng et al., 2026).

The Video Joint-Embedding Predictive Architecture (V-JEPA) offers pretrained spatiotemporal representations that can support such guidance. By predicting visual features rather than pixels, it captures scene content and motion while reducing emphasis on appearance details (Bardes et al., 2024). Yet representing how a scene evolves is not the same as predicting how it should evolve to satisfy an instruction. A predictor may track an object's motion while missing the spatial relation that motion must establish. Furthermore, different tasks also require distinct reasoning skills, such as understanding physical interactions, comparing quantities, and inferring abstract rules. A predictor built on V-JEPA features must therefore focus on task-relevant transitions and adapt to diverse reasoning requirements. Its predicted trajectories should guide video generation toward sequences that satisfy these requirements.

To address these challenges, we introduce *VR-JEPA*, a framework that predicts task-relevant trajectories in a latent space derived from V-JEPA features and uses them to guide video generation (Figure 1). Given an initial scene and an instruction, a learned predictor autoregressively produces a sequence of compact latent states describing the desired scene evolution. These states condition a video diffusion model, separating the prediction of task-relevant states from visual synthesis.

Specifically, to make latent prediction more sensitive to task-relevant changes, VR-JEPA trains the predictor with localized contrastive-state learning. A solution may depend on only a few object movements or changes in inter-object relations, while many other visual details are incidental. We pair successful reference trajectories with task-matched generated candidates and use discrepancies in their V-JEPA features as a heuristic for weighting contrastive supervision at the state and token levels. This encourages predicted rollouts to follow the reference evolution rather than merely capture generic scene motion. Complementing this focused supervision, VR-JEPA incorporates a *skill-routed mixture of experts* (SR-MoE) to model heterogeneous state transitions. Lightweight residual experts are first trained on representative anchor tasks; a sparse, state-conditioned router then learns to combine them as the rollout evolves, using the predictor's training objectives without expert-assignment labels.

On VBVR-Pro-Bench, VR-JEPA improves overall task success from 50.3% for VBVR-Wan2.2 to 56.0%. Component ablations support the contributions of localized contrastive-state learning and skill routed mixture of experts. Our contributions are threefold:

- We introduce VR-JEPA, a framework that predicts task-relevant latent trajectories from V-JEPA features and uses them to guide video generation, separating state evolution prediction from visual synthesis.
- We develop localized contrastive-state learning (LCL), which uses feature discrepancies between successful reference trajectories and task-matched generated candidates to focus contrastive supervision on informative states and tokens.

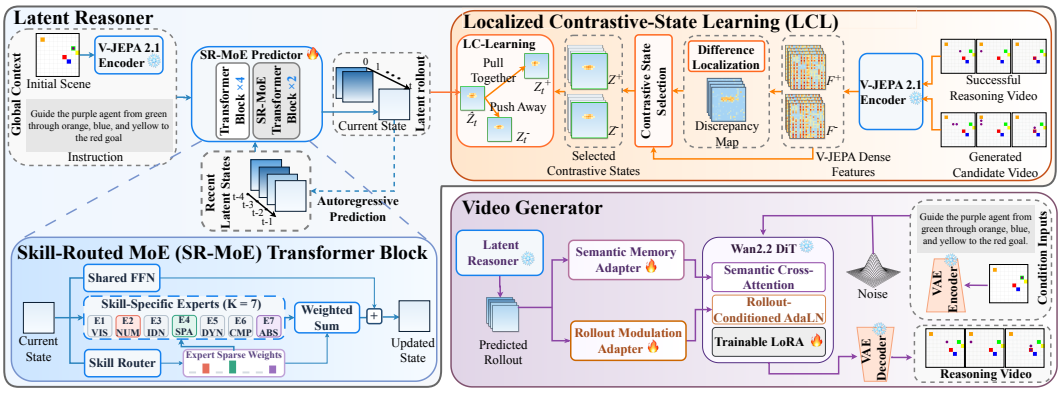

Figure 2: **Overview of VR-JEPA.** VR-JEPA comprises a Latent Reasoner and a Video Generator. The Latent Reasoner predicts a latent rollout from the initial scene and task instruction, with SR-MoE adaptively selecting expert combinations for different reasoning tasks. Its seven experts cover VIS (visual discrimination), NUM (numerical reasoning), IDN (object identity and persistence), SPA (spatial reasoning and planning), DYN (physical dynamics), CMP (compositional reasoning), and ABS (abstract rule inference). During predictor training, localized contrastive-state learning uses dense V-JEPA discrepancies between successful and generated videos to focus contrastive supervision on tokens corresponding to informative states and regions. The Video Generator conditions Wan2.2 DiT on the rollout to synthesize a reasoning video.

- We develop a skill-routed mixture of experts (SR-MoE) for latent prediction. Anchor-task-trained residual experts are combined by a sparse, state-conditioned router to adapt to heterogeneous state transitions without expert-assignment labels.

2 Related Work

2.1 Joint-Embedding Predictive Architectures (JEPA)

Joint-Embedding Predictive Architecture (JEPA) learns representations by predicting target embeddings from contextual observations. I-JEPA introduces this approach for images, and V-JEPA extends it to spatiotemporal representations (Assran et al., 2023; Bardes et al., 2024). V-JEPA 2 demonstrates latent prediction for video understanding and action-conditioned planning (Assran et al., 2025). Recent work extends JEPA to reasoning and generation. JEPA-Reasoner separates latent reasoning from text generation, while ThinkJEPA incorporates vision-language reasoning features into latent world modeling (Liu & Chen, 2025; Zhang et al., 2026). For video generation, WMReward uses V-JEPA prediction consistency for candidate selection and sampling guidance, while Off-Manifold Refinement refines generation trajectories with prediction-energy gradients (Yuan et al., 2026; Nguyen-Truong et al., 2026). PhysVideoGenerator predicts V-JEPA features from noisy diffusion latents and injects them into the generator (Satish et al., 2026). These generation approaches primarily target physical plausibility. VR-JEPA focuses on visual problem solving, adapting a JEPA predictor through localized contrastive-state learning and skill specialization to produce latent trajectories that condition video generation.

2.2 Visual Reasoning through Generation

Visual generation provides a medium for expressing intermediate states during problem solving. Wiedemer et al. (2025) demonstrate emergent zero-shot perception, manipulation, and reasoning capabilities in video models. Thinking with Video further examines generation-based reasoning through VideoThinkBench, covering both vision-centric and text-centric tasks (Tong et al., 2026). Moving toward systematic training and evaluation, VBVR introduces large-scale reasoning data, verifiable scoring, and scaling studies with VBVR-Wan2.2 (Wang et al., 2026). VBVR-Pro extends this direction to native visual reasoning across generation modalities and evaluates transfer to external benchmarks (Xu et al., 2026). Beyond establishing these capabilities, recent approaches im-

prove video reasoning through explicit feedback. Wan-R1 studies verifiable reinforcement learning for spatial reasoning and planning, while *VLMs are Good Teachers* uses a vision-language model to formulate rewards for test-time optimization of a video generator (Liu et al., 2026; Cheng et al., 2026). These approaches motivate supervision that evaluates task correctness beyond visual plausibility. Our work takes a complementary route by learning reasoning guidance within a latent predictor. Rather than applying correctness feedback only to the generator, VR-JEPA uses successful and contrastive trajectories to train intermediate state predictions, which subsequently guide the generation of visual solutions.

3 METHOD

3.1 FRAMEWORK OVERVIEW

As shown in Figure 2, VR-JEPA comprises a *Latent Reasoner* and a *Video Generator*. A frozen V-JEPA encoder and a learnable pooler encode the initial scene x_0 as a latent state Z_0. Given Z_0, the instruction c, and up to H recent states, the predictor P_θ produces a latent rollout that conditions the video generator G_ϕ:

$$\widehat{Z}_{t+1} = P_\theta\Big(Z_0, \widehat{Z}_{\max(0,t-H+1):t}, c\Big), \qquad \widehat{\mathcal{Z}} = (Z_0, \widehat{Z}_1, \ldots, \widehat{Z}_T), \qquad \widehat{V} = G_\phi(x_0, c, \widehat{\mathcal{Z}}). \quad (1)$$

Here, $\widehat{Z}_0 = Z_0$, T is the prediction horizon, $t = 0, \ldots, T-1$, and H is the maximum number of recent predicted states supplied to the predictor.

The rollout conditions the generator through two pathways. A *semantic memory adapter* applies temporal self-attention to rollout states and projects them into keys and values, which the DiT video tokens query through gated cross-attention. A *rollout modulation adapter* adds rollout-dependent residual offsets to the shift, scale, and residual-gate parameters of the DiT self-attention and MLP branches, following the adaptive modulation used in DiT (Peebles & Xie, 2023).

3.2 LOCALIZED CONTRASTIVE-STATE LEARNING

V-JEPA's general spatiotemporal priors may not emphasize the state transitions and regions critical to visual reasoning. To address this limitation, VR-JEPA introduces localized contrastive-state learning (LCL), which compares successful videos with task-matched generated candidates in dense V-JEPA feature space, identifies informative states and regions, and strengthens contrastive supervision on the corresponding latent tokens.

V-JEPA latent rollout prediction. For each successful training video, frozen V-JEPA 2.1 encoder E (Mur-Labadia et al., 2026) extracts features F_t^+ from local clip v_t^+. The learnable pooler Q_ψ maps them to M tokens of dimension d:

$$F_t^+ = E(v_t^+), \qquad Z_t^+ = Q_\psi(F_t^+) \in \mathbb{R}^{M\times d}. \quad (2)$$

We train the pooler and predictor with one- and two-step prediction (feeding back the first prediction) and VISReg regularization (Assran et al., 2025; Wu et al., 2026):

$$\mathcal{L}_j = \frac{1}{|\mathcal{T}_j| M d} \sum_{t\in\mathcal{T}_j} \Big\| \nu(\widehat{Z}_{t+j|t}) - \mathrm{sg}\big[\nu(Z_{t+j}^+)\big] \Big\|_F^2, \quad j \in \{1,2\},$$
$$\mathcal{L}_{\text{base}} = \mathcal{L}_1 + \lambda_2 \mathcal{L}_2 + \lambda_{\text{reg}} \mathcal{R}_{\text{VISReg}}. \quad (3)$$

Here, $\widehat{Z}_{t+j|t}$ is the j-step prediction from ground-truth history at t; $\mathcal{T}_j$ contains valid starting times. The operator ν applies token-wise ℓ_2 normalization, and sg stops gradients through the target. The coefficients λ_2 and λ_{reg} weight the two-step and regularization losses.

Trajectory differences localization. VR-JEPA treats each training video V^+ as a successful reasoning sample and generates a candidate video V^- using VBVR-Wan2.2 (Wang et al., 2026) under the same input conditions. Since the generated candidate may satisfy the reasoning requirements, it may not serve as a valid negative sample. VR-JEPA uses discrepancies between their dense V-JEPA features as a heuristic to select pairs and locate states and regions that merit stronger supervision. Figure 3 shows the selection and localization.

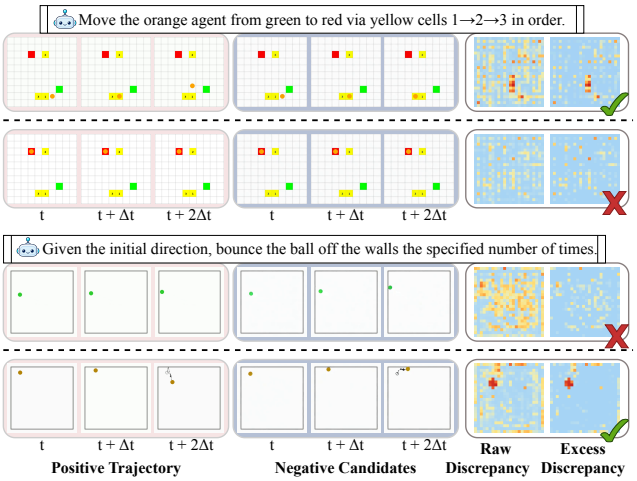

Figure 3: **State selection for localized contrastive-state learning.** Successful reference trajectories and task-matched generated candidates are compared in dense V-JEPA feature space. After state-specific quantile subtraction, large residual discrepancies select states for contrastive supervision (green checks), while small discrepancies exclude them (red crosses).

To measure differences between each training video and its candidate, the frozen V-JEPA encoder processes clips at the same temporal index t, yielding $F_t^{\pm} = E(v_t^{\pm})$. VR-JEPA computes cosine discrepancies at corresponding feature-grid positions n. To reduce the influence of scene-wide appearance differences, it subtracts the q-quantile of discrepancies within each state and retains the positive residuals:

$$D_{t,n} = 1 - \frac{\langle F_{t,n}^+, F_{t,n}^- \rangle}{\|F_{t,n}^+\|_2 \|F_{t,n}^-\|_2}, \qquad S_{t,n} = \left[D_{t,n} - \mathrm{Quantile}_q(D_{t,:}) \right]_+ . \tag{4}$$

Here, q is the quantile-level hyperparameter, and $[x]_+ = \max(x, 0)$. Averaging the residuals over the N dense positions yields the state score $s_t = N^{-1} \sum_n S_{t,n}$. VR-JEPA maps this score to a state-level weight $a_t = \mathrm{clip}((s_t - \tau_{\min})/(\tau_{\mathrm{full}} - \tau_{\min}), 0, 1)$, which scales the state's contribution to the contrastive-state loss. The thresholds $\tau_{\min}$ and τ_{full} specify the scores at which this weight is zero and one, respectively.

For each state with $a_t > 0$, VR-JEPA uses the pooler's attention maps to transfer localized discrepancies to latent tokens. Let $A_{t,h,m,n}^+$ denote attention from token m to dense position n in head h for the successful video. The normalized discrepancy map and token weights are computed as

$$\widetilde{S}_{t,n} = \frac{S_{t,n}}{\sum_{n'} S_{t,n'} + \epsilon}, \qquad b_{t,m} = \frac{1}{N_h} \sum_h \left[\sum_n A_{t,h,m,n}^+ \widetilde{S}_{t,n} - \frac{1}{N} \right]_+ , \tag{5}$$

where N_h is the number of attention heads, and $\epsilon > 0$ ensures numerical stability. The term $1/N$ provides a uniform-attention baseline over the N dense positions. A token receives positive weight only if its attention to the localized discrepancies exceeds this baseline in at least one head; states with no positively weighted tokens are excluded.

Contrastive-state supervision. Using the state and token weights, VR-JEPA encourages each one-step prediction to be closer to the successful target than to its paired generated candidate. Let Ω denote the selected state–token positions with $a_t > 0$ and $b_{t,m} > 0$. The weighted margin loss and overall training objective are

$$\mathcal{L}_{\mathrm{LCL}} = \frac{1}{|\Omega|} \sum_{(t,m) \in \Omega} a_t b_{t,m} \left[\mu + d(\widehat{Z}_{t,m}, Z_{t,m}^+) - d(\widehat{Z}_{t,m}, Z_{t,m}^-) \right]_+ ,$$
$$\mathcal{L}_{\mathrm{reason}} = \mathcal{L}_{\mathrm{base}} + \lambda_{\mathrm{LCL}} \mathcal{L}_{\mathrm{LCL}}. \tag{6}$$

Here, $\widehat{Z}_{t,m}$ denotes token m of the one-step prediction $\widehat{Z}_{t|t-1}$, while $Z^{+}_{t,m}$ and $Z^{-}_{t,m}$ are the corresponding pooled tokens from the successful and candidate videos. The function d is cosine distance, μ is the margin, and λ_{LCL} controls the contribution of contrastive-state supervision. Gradients are stopped through the weights and both target representations, and $\mathcal{L}_{\mathrm{LCL}} = 0$ when Ω is empty.

3.3 Skill-Routed Mixture of Experts

A compact V-JEPA predictor may lack the capacity to model the diverse reasoning patterns required across tasks. To address this limitation, VR-JEPA equips the predictor with a lightweight skill-routed mixture of experts, which adaptively selects different expert combinations for different tasks.

Learning skill-specific experts. VR-JEPA first trains experts on representative anchor tasks to encourage specialization and provide a skill-oriented basis for subsequent composition. Starting from the trained dense predictor, SR-MoE adds lightweight bottleneck adapters to the final two blocks of the six-layer predictor while retaining the shared feed-forward networks:

$$Y^{(\ell)}_{t,m} = F^{(\ell)}_{\mathrm{shared}}\left(H^{(\ell)}_{t,m}\right) + \alpha \sum_{k=1}^{K} \pi^{(\ell)}_{t,k} A^{(\ell)}_{k}\left(H^{(\ell)}_{t,m}\right). \tag{7}$$

Here, $H^{(\ell)}_{t,m}$ is the input token, $Y^{(\ell)}_{t,m}$ is its expert-augmented output, $F^{(\ell)}_{\mathrm{shared}}$ is the shared feed-forward network, and $A^{(\ell)}_{k}$ is the k-th expert adapter. The expert residual coefficient is fixed to $\alpha = 1$, so the weighted expert output is added directly to the shared output without an additional learned or scheduled scale; $\pi^{(\ell)}_{t,k}$ is the routing weight. The model uses $K = 7$ skill-specific experts. Please refer to Figure 2 for their abbreviations and corresponding skills.

For each anchor task, VR-JEPA directly activates the expert associated with its skill and updates only that expert's adapters, leaving the shared predictor fixed. This gives each expert an initial skill-specific capability before the router learns to combine them across tasks.

Learning adaptive routing. Different categories of reasoning tasks require different skills or combinations of skills. VR-JEPA therefore learns a router to select and combine the pretrained experts according to each task's reasoning requirements. The router conditions on the current state, the preceding state, their difference, and the task context:

$$r^{(\ell)}_{t} = \left[\bar{H}^{(\ell)}_{t}; \bar{H}^{(\ell)}_{t-1}; \bar{H}^{(\ell)}_{t} - \bar{H}^{(\ell)}_{t-1}; g^{(\ell)}\right], \qquad \pi^{(\ell)}_{t} = \mathrm{Entmax}_{1.5}\left(f^{(\ell)}_{\mathrm{route}}(r^{(\ell)}_{t})\right). \tag{8}$$

Here, $\bar{H}^{(\ell)}_{t}$ averages the hidden tokens within a state, $g^{(\ell)}$ summarizes the conditioning context, and $f^{(\ell)}_{\mathrm{route}}$ maps the concatenated features $r^{(\ell)}_{t}$ to expert scores. $\mathrm{Entmax}_{1.5}$ (Peters et al., 2019) converts these scores into normalized sparse weights, enabling the router to select and combine experts for different reasoning tasks.

The router is trained through the prediction and contrastive-state objectives, without expert-assignment annotations for the full training set or additional load-balancing losses. Training first optimizes the router with fixed experts, then jointly adapts the router and experts, and finally updates selected shared components. This progression learns how to combine the initialized experts before refining their capabilities and coordination beyond the anchor tasks. The resulting adaptive composition supports latent rollout prediction for tasks requiring multiple reasoning skills.

4 Experiments

4.1 Experimental Setup

Benchmark and Evaluation. We evaluate VR-JEPA on VBVR-Pro-Bench (Xu et al., 2026), which comprises 100 visual reasoning tasks spanning abstraction, knowledge, perception, spatial reasoning, and transformation. We evaluate five instances per task, totaling 500 instances. Given an initial image and an instruction, models generate a video depicting the solution. The benchmark

Table 1: **Main results on VBVR-Pro-Bench (Xu et al., 2026).** We report overall, in-domain (ID), and out-of-domain (OOD) scores, followed by results for abstraction (Abst.), knowledge (Know.), perception (Perc.), spatial reasoning (Spat.), and transformation (Trans.). Avg. denotes the mean of task scores weighted by the number of samples per task. All scores are reported as percentages. Higher is better. Bold and underlined values indicate the best and second-best results.

		In-Domain (ID) (%)						Out-of-Domain (OOD) (%)					
Models	**Overall (%)**	**Avg.**	**Abst.**	**Know.**	**Perc.**	**Spat.**	**Trans.**	**Avg.**	**Abst.**	**Know.**	**Perc.**	**Spat.**	**Trans.**
Open-source Video Models													
Wan2.2-I2V-A14B	18.2	15.7	12.3	15.2	15.0	17.0	23.3	20.7	29.2	17.9	13.8	19.5	31.8
LTX-2.3-I2AV	11.2	10.6	9.3	12.3	9.7	13.6	5.1	11.9	21.7	19.8	7.0	9.1	5.9
Wan2.1-I2V-14B-720P	10.0	10.5	8.7	14.6	12.4	9.9	4.0	9.5	13.9	8.7	7.6	12.3	5.2
Wan2.2-TI2V-5B	9.4	6.6	5.0	7.4	7.7	9.2	2.5	12.2	22.2	9.2	9.1	6.3	11.6
CogVideoX1.5-5B-I2V	8.6	9.7	9.2	14.2	7.8	7.9	5.5	7.5	16.5	5.6	5.0	3.9	2.8
HunyuanVideo-I2V	5.6	5.5	4.0	7.2	1.0	9.1	3.8	5.7	11.6	1.5	2.7	7.4	6.4
Proprietary Video Models													
Seedance 2.0	48.5	43.9	41.9	40.6	47.5	49.7	41.8	**53.0**	**43.7**	67.9	**52.4**	53.3	61.3
Kling VIDEO 3.0	38.7	36.1	28.7	34.6	48.2	40.3	34.6	41.3	34.6	**75.5**	40.1	24.1	47.7
Veo 3.1	14.9	16.3	15.1	20.5	19.6	13.6	9.7	13.4	21.2	14.8	10.1	12.9	9.1
Video Reasoning Models													
VBVR-Wan2.2	50.3	55.4	33.8	54.8	50.9	65.9	91.2	45.1	35.8	42.7	34.0	**72.2**	**77.7**
VR-JEPA	**56.0**	**65.9**	**56.3**	**63.8**	**61.8**	**69.1**	**91.4**	46.1	41.5	42.1	35.2	65.2	77.5

includes 50 in-domain (ID) tasks from task families represented in its training set and 50 held-out out-of-domain (OOD) tasks. Task-specific, rule-based evaluators assess generated outputs against the required outcomes and constraints. We report overall, ID/OOD, and category-level scores; absolute gains are expressed in percentage points.

The publicly released baseline outputs for VBVR-Pro-Bench differ in frame count and spatial resolution. We rerun VBVR-Wan2.2 on VBVR-Pro-Bench. Before evaluation, we temporally sample each generated video to match the frame count of its corresponding ground-truth video and resize the frames to 512×512 pixels. We apply this preprocessing to outputs from all compared methods and recompute their scores using the same evaluation pipeline, ensuring a consistent comparison.

Baselines. We compare VR-JEPA with six open-source video models: Wan2.2-I2V-A14B and Wan2.2-TI2V-5B (Wan Team, 2025), Wan2.1-I2V-14B-720P (Team Wan et al., 2025), LTX-2.3-I2AV (HaCohen et al., 2026; Lightricks, 2026), CogVideoX1.5-5B-I2V (Yang et al., 2024; Z.ai, 2024), and HunyuanVideo-I2V (Kong et al., 2024; Tencent, 2025). We also include three proprietary video models: Seedance 2.0 (Team Seedance et al., 2026), Kling VIDEO 3.0 (Kling AI, 2026), and Veo 3.1 (Gallegos & Iljic, 2025). Among video reasoning models, our primary baseline is VBVR-Wan2.2 (Wang et al., 2026), which adapts Wan2.2-I2V-A14B to the VBVR dataset without modifying the model architecture.

Implementation Details. VR-JEPA is trained in two stages. First, the Latent Reasoner is trained with the V-JEPA 2.1 encoder (Mur-Labadia et al., 2026) frozen. The learnable pooler and predictor are optimized using the prediction and localized contrastive-state objectives described in Section 3.2, with expert training and routing detailed in Section 3.3. Second, the Latent Reasoner is frozen, and the Wan2.2-14B generator, initialized from VBVR-Wan2.2, is adapted by optimizing only the LoRA parameters (Hu et al., 2022) and rollout-conditioning modules with a flow-matching loss (Lipman et al., 2023). Additional implementation details are provided in Appendix A.

4.2 Main Results

Quantitative Comparisons. Table 1 reports quantitative comparisons on VBVR-Pro-Bench. VR-JEPA achieves the highest overall score of 56.0% among the evaluated models, outperforming VBVR-Wan2.2 by 5.7 percentage points and Seedance 2.0, the highest-scoring proprietary model overall, by 7.5 points. Compared with VBVR-Wan2.2, it improves the ID average from 55.4% to

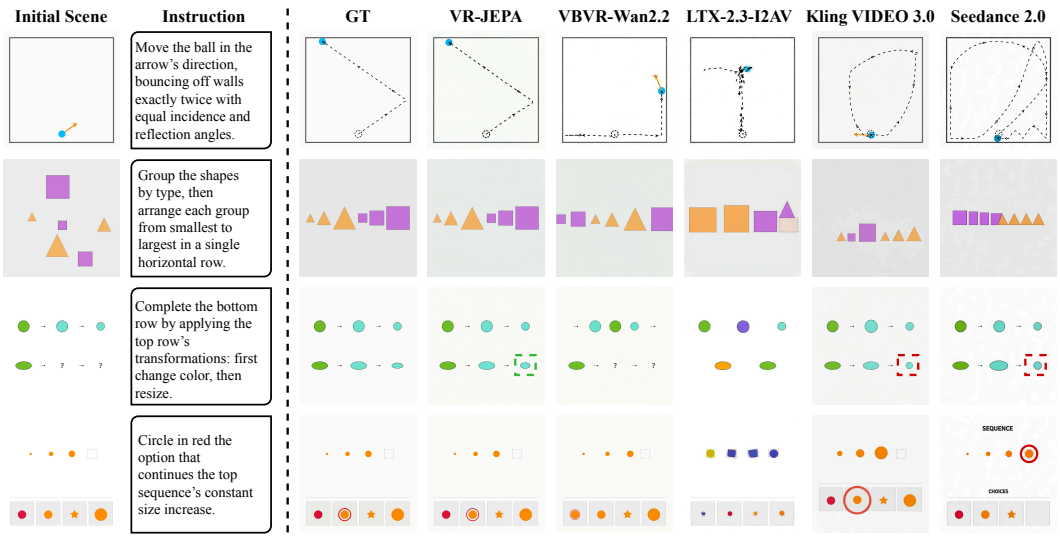

Figure 4: **Qualitative comparison on VBVR-Pro-Bench.** Given the initial state and instruction, we compare the reasoning results of VR-JEPA with those of a video reasoning model (VBVR-Wan2.2), an open-source video model (LTX-2.3-I2AV), and two proprietary video models (Kling VIDEO 3.0 and Seedance 2.0).

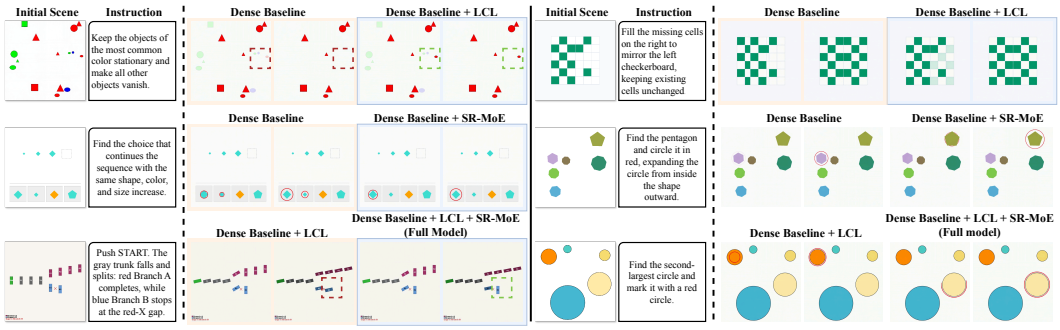

Figure 5: **Qualitative ablation of localized contrastive-state learning (LCL) and skill-routed mixture of experts (SR-MoE).** Top and middle: adding LCL and SR-MoE to the dense baseline, respectively. Bottom: adding SR-MoE to the LCL-equipped predictor to obtain the full model.

65.9% and the OOD average from 45.1% to 46.1%. In particular, ID abstraction and perception improve by 22.5 and 10.9 points, reaching 56.3% and 61.8%, respectively. ID knowledge and spatial reasoning also improve by 9.0 and 3.2 points, while transformation performance remains nearly unchanged at 91.4%. These gains over VBVR-Wan2.2 support the effectiveness of the proposed latent reasoning framework, which combines supervision on informative states and regions with adaptive skill-expert composition. Section 4.3 examines the contributions of localized contrastive-state learning and SR-MoE individually and jointly.

Qualitative Comparisons. Figure 4 compares VR-JEPA with VBVR-Wan2.2, LTX-2.3-I2AV, Kling VIDEO 3.0, and Seedance 2.0 on four visual reasoning examples. In the ball-bouncing task, VR-JEPA produces a trajectory closely matching the ground truth, whereas VBVR-Wan2.2 follows an incorrect initial direction and the other models produce curved or irregular paths. In the shape-sorting task, VR-JEPA preserves the objects and groups them by type in ascending size order. By contrast, VBVR-Wan2.2 interleaves the two shape groups, while LTX-2.3-I2AV alters object shapes and colors. In the transformation task, VR-JEPA correctly applies the demonstrated color change followed by size reduction to the bottom-row ellipses. VBVR-Wan2.2 leaves the answer positions unfilled, while Kling VIDEO 3.0 and Seedance 2.0 change the final ellipse into a circle. In the sequence-completion task, VR-JEPA circles the same option as the ground truth, whereas VBVR-

Table 2: **Ablation of localized contrastive-state learning (LCL) and skill-routed mixture of experts (SR-MoE).** Scores are reported as percentages; higher is better.

Variant	Components		Overall (%)	In-Domain (ID) (%)						Out-of-Domain (OOD) (%)					
	LCL	SR-MoE		Avg.	Abst.	Know.	Perc.	Spat.	Trans.	Avg.	Abst.	Know.	Perc.	Spat.	Trans.
Dense baseline	✗	✗	53.9	63.4	53.8	58.9	58.9	68.8	91.4	44.4	43.3	**44.5**	31.9	59.3	75.3
+ LCL	✓	✗	55.1	64.9	52.6	61.2	61.3	**71.6**	93.0	45.3	**44.1**	38.5	33.1	**65.4**	75.8
+ SR-MoE	✗	✓	55.5	65.2	54.0	61.4	60.7	69.8	**95.6**	45.9	42.8	42.3	34.3	64.2	77.3
Full model	✓	✓	**56.0**	**65.9**	**56.3**	**63.8**	**61.8**	69.1	91.4	**46.1**	41.5	42.1	**35.2**	65.2	**77.5**

Table 3: **Ablation of visual representations.** Scores are reported as percentages; higher is better.

Visual representation	Representation form	ID (%) ↑	OOD (%) ↑
DINOv3	Initial-state features	62.8	45.3
DINOv3	Predicted latent rollout	65.3	43.8
V-JEPA 2.1 (VR-JEPA)	Predicted latent rollout	**65.9**	**46.1**

Wan2.2 circles the red distractor and Seedance 2.0 modifies the sequence instead of marking an answer option. Together, these examples highlight VR-JEPA's advantage over the compared models in following task rules while preserving relevant object attributes, yielding reasoning outputs more consistent with the intended solutions.

4.3 Ablation Studies

Effectiveness of LCL and SR-MoE. The dense baseline guides Wan2.2 video generation with latent rollouts predicted by a trained V-JEPA predictor. The three variants add localized contrastive-state learning (LCL), SR-MoE, or both to this baseline. Table 2 shows that LCL and SR-MoE individually raise the overall score from 53.9% to 55.1% and 55.5%, respectively. Combining both achieves the best overall, ID, and OOD averages among the ablated variants: 56.0%, 65.9%, and 46.1%. Figure 5 illustrates these benefits: LCL corrects the mirrored checkerboard completion, while SR-MoE selects the instructed pentagon instead of a distractor. Combining both also enables the model to identify the second-largest circle, which the LCL-only variant fails to select. Together, these results support the complementary benefits of supervision focused on informative states and regions and skill-specific expert routing in improving VR-JEPA's video reasoning.

Visual Representation Space. Table 3 compares DINOv3 (Siméoni et al., 2025) and V-JEPA 2.1 (Mur-Labadia et al., 2026) as representation spaces for latent reasoning under the same framework, with initial-state DINOv3 features included as a reference. The selected DINOv3 and V-JEPA 2.1 encoders have comparable parameter counts. Using V-JEPA 2.1 for latent rollout prediction achieves video reasoning scores of 65.9% on ID tasks and 46.1% on OOD tasks, outperforming the DINOv3 rollout variant by 0.6 and 2.3 percentage points, respectively. These results favor V-JEPA 2.1 for video reasoning, consistent with its ability to capture spatiotemporal dynamics beyond the static visual semantics primarily encoded by DINOv3. Corresponding qualitative comparisons are provided in Figure 9 in Appendix C.3.

5 Conclusion

We presented VR-JEPA, a framework that predicts latent rollouts in V-JEPA representation space to guide video generation for visual reasoning. VR-JEPA uses localized contrastive-state learning to focus contrastive supervision on informative states and regions identified from successful reference videos and task-matched generated candidates. It further equips the predictor with a skill-routed mixture of experts that adaptively combines skill-specific experts for different reasoning tasks. On VBVR-Pro-Bench, VR-JEPA improves the overall score from 50.3% to 56.0% over VBVR-Wan2.2, an absolute gain of 5.7 percentage points. Component ablations support the complementary contributions of localized supervision and skill-expert composition. Together, these results highlight the potential of reasoning-guided latent prediction for reasoning through video generation.

6 AI USE STATEMENT

AI tools were used to assist with literature searches. We reviewed and selected the relevant papers and independently organized and wrote the related work section. We take full responsibility for the content and references of this paper.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *arXiv preprint arXiv:2301.08243*, 2023. URL https://arxiv.org/abs/2301.08243.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. URL https://arxiv.org/abs/2506.09985.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=QaCCuDfBk2.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22563–22575, 2023. doi: 10.1109/CVPR52729.2023.02161.

Junhao Cheng, Liang Hou, Tianxiong Zhong, Xin Tao, Pengfei Wan, Kun Gai, and Jing Liao. VLMs are good teachers for video reasoning via adaptive test-time optimization. *arXiv preprint arXiv:2606.02564*, 2026. URL https://arxiv.org/abs/2606.02564.

Yilun Du, Sherry Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Pack Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9pKtcJcMP3.

Jess Gallegos and Thomas Iljic. Introducing veo 3.1 and advanced capabilities in flow. Google blog post, 2025. URL https://blog.google/innovation-and-ai/products/veo-updates-flow/.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. URL https://arxiv.org/abs/1803.10122.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, Eitan Richardson, Guy Shiran, Itay Chachy, Jonathan Chetboun, Michael Finkelson, Michael Kupchick, Nir Zabari, Nitzan Guetta, Noa Kotler, Ofir Bibi, Ori Gordon, Poriya Panet, Roi Benita, Shahar Armon, Victor Kulikov, Yaron Inger, Yonatan Shiftan, Zeev Melumian, and Zeev Farbman. LTX-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026. URL https://arxiv.org/abs/2601.03233.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022. URL https://arxiv.org/abs/2204.03458.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Kling AI. Kling video 3.0 model user guide. Official model guide, 2026. URL https://kling.ai/quickstart/klingai-video-3-model-user-guide.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. URL https://arxiv.org/abs/2412.03603.

Lightricks. Ltx-2.3. Official model card, 2026. URL https://huggingface.co/Lightricks/LTX-2.3.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Bingyang Kelvin Liu and Ziyu Patrick Chen. JEPA-Reasoner: Decoupling latent reasoning from token generation. *arXiv preprint arXiv:2512.19171*, 2025. URL https://arxiv.org/abs/2512.19171.

Ming Liu, Yunbei Zhang, Shilong Liu, Liwen Wang, and Wensheng Zhang. Wan-R1: Verifiable-reinforcement learning for video reasoning. *arXiv preprint arXiv:2603.27866*, 2026. URL https://arxiv.org/abs/2603.27866.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vhFu1Acb0xb.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026. URL https://arxiv.org/abs/2603.14482.

Hai Nguyen-Truong, Tuan-Anh Vu, and Dang Huynh. Off-manifold refinement: Guiding video generators with a frozen world model. *arXiv preprint arXiv:2608.29904*, 2026. URL https://arxiv.org/abs/2608.29904.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Ben Peters, Vlad Niculae, and André F. T. Martins. Sparse sequence-to-sequence models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1504–1519, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1146. URL https://aclanthology.org/P19-1146/.

Siddarth Nilol Kundur Satish, Devesh Jaiswal, Hongyu Chen, and Abhishek Bakshi. PhysVideo-Generator: Towards physically aware video generation via latent physics guidance. *arXiv preprint arXiv:2601.03665*, 2026. URL https://arxiv.org/abs/2601.03665.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. URL https://arxiv.org/abs/2508.10104.

Team Seedance et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026. URL https://arxiv.org/abs/2604.14148.

Team Wan et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Tencent. Hunyuanvideo-i2v. Official model card, 2025. URL https://huggingface.co/tencent/HunyuanVideo-I2V.

Jingqi Tong, Yurong Mou, Hangcheng Li, Mingzhe Li, Yongzhuo Yang, Ming Zhang, Qiguang Chen, Tianyi Liang, Xiaomeng Hu, Yining Zheng, Xinchi Chen, Jun Zhao, Xuanjing Huang, and Xipeng Qiu. Thinking with video: Video generation as a promising multimodal reasoning paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 41121–41129, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Tong_Thinking_with_Video_Video_Generation_as_a_Promising_Multimodal_Reasoning_CVPR_2026_paper.html.

Wan Team. Wan2.2. Official model repository, 2025. URL https://github.com/Wan-Video/Wan2.2.

Maijunxian Wang, Ruisi Wang, Juyi Lin, Ran Ji, et al. A very big video reasoning suite. *arXiv preprint arXiv:2602.20159*, 2026. URL https://arxiv.org/abs/2602.20159.

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. URL https://arxiv.org/abs/2509.20328.

Haiyu Wu, Randall Balestriero, and Morgan Levine. VISReg: Variance-invariance-sketching regularization for JEPA training. *arXiv preprint arXiv:2606.02572*, 2026. URL https://arxiv.org/abs/2606.02572.

Kun Xiang, Terry Jingchen Zhang, Yinya Huang, Jixi He, Zirong Liu, Yueling Tang, Ruizhe Zhou, Lijing Luo, Youpeng Wen, Xiuwei Chen, Bingqian Lin, Jianhua Han, Hang Xu, Hanhui Li, Bin Dong, and Xiaodan Liang. Aligning perception, reasoning, modeling and interaction: A survey on physical AI. *arXiv preprint arXiv:2510.04978*, 2025. URL https://arxiv.org/abs/2510.04978.

Junxiang Xu, Ruisi Wang, Fanyi Pu, Maijunxian Wang, et al. VBVR-Pro: A scalable and verifiable suite for native visual reasoning. *arXiv preprint arXiv:2608.26105*, 2026. URL https://arxiv.org/abs/2608.26105.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihan Wang, Yean Cheng, Ting Liu, Bin Xu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. *ArXiv*, abs/2408.06072, 2024. URL https://api.semanticscholar.org/CorpusID:271855655.

Jianhao Yuan, Xiaofeng Zhang, Felix Friedrich, Nicolas Beltran-Velez, Melissa Hall, Reyhane Askari-Hemmat, Xiaochuang Han, Nicolas Ballas, Michal Drozdzal, and Adriana Romero-Soriano. Inference-time physics alignment of video generative models with latent world models. *arXiv preprint arXiv:2601.10553*, 2026. URL https://arxiv.org/abs/2601.10553.

Z.ai. Cogvideox1.5-5b-i2v. Official model card, 2024. URL https://huggingface.co/zai-org/CogVideoX1.5-5B-I2V.

Haichao Zhang, Yijiang Li, Shwai He, Tushar Nagarajan, Mingfei Chen, Jianglin Lu, Ang Li, and Yun Fu. ThinkJEPA: Empowering latent world models with large vision-language reasoning model. *arXiv preprint arXiv:2603.22281*, 2026. URL https://arxiv.org/abs/2603.22281.

APPENDIX CONTENTS

- **A Implementation Details**
  - A.1 Training Data
  - A.2 Architecture and Training Procedure
  - A.3 Eight-GPU Reproduction Configuration
  - A.4 Parameter Overhead
- **B Additional Ablation Studies**
  - B.1 Training Strategies
  - B.2 Expert Routing Strategies
  - B.3 Interventions on the Predicted Rollout
- **C Further Visualization**
  - C.1 Comparisons with Open-source Models
  - C.2 Comparisons with Proprietary Video Models
  - C.3 Visual Representation Space Ablation
  - C.4 Comparison of latent trajectory guidance
  - C.5 Task-Dependent Expert Routing
  - C.6 Expert Removal

A IMPLEMENTATION DETAILS

A.1 TRAINING DATA

Our training corpus comprises 100 tasks with 1,000 samples per task. Each sample contains an initial state, a task instruction, and a successful reference trajectory. These 100,000 reference samples support latent reasoner training in Stage 1 and video generator adaptation in Stage 2. For localized contrastive-state learning, we additionally generate one negative candidate per sample using VBVR-Wan2.2 (Wang et al., 2026) under the same initial state and instruction, yielding 100,000 positive–negative trajectory pairs. Since generated candidates may also satisfy the task, our localization mechanism identifies informative differences within these pairs for supervision. Of the 100 training tasks, 50 are included in the ID evaluation using held-out instances. The 50 OOD evaluation tasks are absent from the training corpus.

For expert pretraining, we first define the reasoning skill associated with each of the seven experts and manually select seven representative anchor tasks for that skill from the same corpus. The task subsets are disjoint across experts, yielding 49 anchor tasks and 49,000 reference samples in total. Each expert is pretrained on its corresponding subset to establish an initial skill specialization. Subsequent router training and joint refinement cover all 100 tasks without additional task-to-expert annotations.

A.2 ARCHITECTURE AND TRAINING PROCEDURE

Model initialization. We initialize the visual encoder from the V-JEPA 2.1 ViT-L (Mur-Labadia et al., 2026) checkpoint `vjepa2_1_vitl_dist_vitG_384.pt`, loading its `ema_encoder` weights. The encoder remains frozen throughout both training stages. The state-token pooler and six-layer task-relevant latent predictor are trained for our framework rather than initialized from the pretrained V-JEPA predictor. For Stage 2, we initialize the video generator from VBVR-Wan2.2 and adapt its high-noise branch, while retaining the low-noise branch.

Expert architecture and routing. Seven bottleneck-adapter experts are inserted into the final two predictor blocks. Their designated skills are visual discrimination (VIS), numerical reasoning (NUM), object identity and persistence (IDN), spatial reasoning and planning (SPA), physical dynamics (DYN), compositional reasoning (CMP), and abstract rule inference (ABS). Routing uses $\mathrm{Entmax}_{1.5}$ (Peters et al., 2019) and the objectives in Section 3.3.

Stage 1: Optimization settings. We use a history window of $H = 4$ states and represent each state with $M = 8$ tokens of dimension $d = 256$. The pooler has $N_h = 8$ attention heads. Each rollout contains ten states including the initial state, corresponding to a prediction horizon of $T = 9$ in Equation 1. Early stopping is disabled. The two-step prediction and VISReg weights in Equation 3 are $\lambda_2 = 2.0$ and $\lambda_{\text{reg}} = 10^{-4}$, respectively. The expert residual coefficient in Equation 7 is fixed to $\alpha = 1$; the implementation adds the routed expert output directly to the shared feed-forward output, without learning or scheduling a separate residual scale.

Latent reasoner training consists of base training, expert pretraining, router initialization, and two successive joint refinement phases. All phases use AdamW with $\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$, and an effective global batch size of 512. Learning rates follow a linear warmup followed by cosine decay. The V-JEPA encoder remains frozen in every phase.

LCL hyperparameters. For localized contrastive-state learning in Section 3.2, we use the quantile level $q = 0.75$ in Equation 4 and state-weight thresholds $\tau_{\min} = 0.1$ and $\tau_{\text{full}} = 0.2$. The numerical stability term for discrepancy normalization in Equation 5 is $\epsilon = 10^{-6}$. The margin and loss weight in Equation 6 are $\mu = 0.05$ and $\lambda_{\text{LCL}} = 0.05$, respectively.

Base latent reasoner training. We train the predictor with the one- and two-step objectives, VISReg regularization (Wu et al., 2026), and localized contrastive-state learning (LCL) defined in Section 3.2. We first train the shared latent reasoner for 20 epochs (3,720 optimizer updates). The learning rates are 10^{-5} for the predictor and other trainable modules, 3×10^{-5} for the state-token pooler, and 3×10^{-6} for the text-conditioning projection. We use weight decay of 10^{-3} and 200 warmup updates.

Anchor-task expert pretraining. Starting from the trained shared reasoner, we specialize the seven experts on their respective anchor-task subsets. For each subset, only the corresponding expert adapters are activated and optimized; the shared predictor, pooler, and other modules remain frozen. Expert pretraining runs for 20 epochs with a learning rate of 10^{-5}, weight decay of 10^{-3}, and 200 warmup updates. This phase establishes skill-specific specializations before learning how to route heterogeneous tasks.

Prediction-driven router initialization. We then freeze the expert adapters and shared modules and optimize only the router on the multi-task training split. Routing is learned through the latent prediction objective, without task-to-expert labels or an auxiliary load-balancing loss. We use a learning rate of 10^{-4}, zero weight decay, and 200 warmup updates. The cosine schedule is configured for 20 epochs; the checkpoint used to initialize joint refinement is selected after four epochs (744 optimizer updates).

Joint refinement. We first jointly optimize the router and expert adapters for two epochs (372 optimizer updates), using learning rates of 2×10^{-5} and 10^{-5}, respectively, while keeping the shared modules frozen.

We subsequently unfreeze the predictor's attention layers, shared feed-forward layers, and state-token pooler for three additional epochs (558 optimizer updates). The learning rates are 10^{-5} for the router, 5×10^{-6} for expert adapters, 5×10^{-7} for both attention and shared feed-forward layers, and 2×10^{-7} for the pooler. Both refinement phases use zero weight decay and 25 warmup updates. This progressive refinement first coordinates expert selection and specialization, then adapts the shared latent representation to support them.

Stage 2: Video generator adaptation. We freeze the trained latent reasoner and optimize the video generator using the flow-matching objective (Lipman et al., 2023) on reference videos, conditioned on the initial state, instruction, and predicted latent trajectory. Only the LoRA parameters (Hu et al., 2022) and trajectory-conditioning modules in the adapted Wan2.2 generator branch (Wan Team, 2025) are optimized. The trajectory-conditioning interface (TCI) comprises the semantic memory adapter and rollout modulation adapter described in Section 3.1. The former supplies rollout features through gated cross-attention; the latter produces offsets for the DiT modulation parameters.

We use AdamW with $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and weight decay of 0.01. The learning rates are 10^{-5} for LoRA and 10^{-4} for the trajectory-conditioning modules, with 50 warmup updates followed by constant learning rates. Training runs for 2,000 optimizer updates with an effective global batch size of 64. We use rank-32 LoRA on the attention projections and feed-forward linear layers, and clip the gradient norm to 1.0.

A.3 Eight-GPU Reproduction Configuration

For reproduction on eight H100 GPUs, the Stage 1 effective batch size of 512 can be obtained using a per-GPU batch size of 32 with two gradient accumulation steps for base training, or a per-GPU batch size of eight with eight accumulation steps for expert pretraining. Router initialization and both joint refinement phases use a per-GPU batch size of one with 64 accumulation steps. Stage 2 uses a per-GPU batch size of one with eight accumulation steps, yielding an effective batch size of 64.

A.4 Parameter Overhead

VR-JEPA is initialized from VBVR-Wan2.2 and inherits its 306.708M LoRA parameters across the two generator branches (153.354M per branch). The underlying Wan2.2 DiT contains 28.578B parameters, with approximately 14.289B per branch. Our latent predictor contains 38.093M parameters, including the SR-MoE experts and router. The trajectory-conditioning interface adds 17.181M parameters: 11.807M for the semantic memory adapter and 5.374M for the rollout modulation adapter. Relative to VBVR-Wan2.2, the predictor and conditioning modules therefore add 55.274M parameters, equivalent to approximately 0.19% of the underlying 28.578B-parameter DiT. Relative to the original Wan2.2 DiT, the inherited LoRA parameters and these new modules together account for 361.982M additional parameters, approximately 1.27% of the DiT parameter count.

As shown in Table 1, VR-JEPA improves the overall score from 50.3% to 56.0% over VBVR-Wan2.2 with only 55.274M additional predictor and conditioning parameters beyond that baseline; the inherited LoRA parameters are already present in VBVR-Wan2.2. This small overhead supports a design that emphasizes informative contrastive supervision through LCL and adaptive skill-expert composition through SR-MoE, rather than substantial model expansion.

B Additional Ablation Studies

B.1 Training Strategies

Table 4: **Ablation of the training strategy.** We compare tuning the trajectory-conditioning interface (TCI), LoRA parameters, or both, with VBVR-Wan2.2 as the baseline. ID and OOD task-success scores are reported on VBVR-Pro-Bench as percentages; higher is better.

Training strategy	Trainable modules		Task success (%)	
	TCI	LoRA	ID ↑	OOD ↑
VBVR-Wan2.2	✗	✗	55.4	45.1
LoRA-only tuning	✗	✓	57.4	45.5
TCI-only tuning	✓	✗	63.1	43.7
Joint tuning	✓	✓	**65.9**	**46.1**

Table 4 compares LoRA-only, TCI-only, and joint tuning. Joint tuning achieves 65.9%/46.1%, exceeding TCI-only tuning by 2.8 percentage points on ID tasks and 2.4 points on OOD tasks. It also exceeds LoRA-only tuning by 8.5 and 0.6 points, respectively. Training the conditioning interface alone is therefore sufficient to improve ID performance in this comparison, whereas the strongest scores on both splits are obtained when the interface and renderer LoRA parameters are adapted together.

Table 5: **Ablation of the SR-MoE routing strategy.** We compare learned routing with uniform routing, random routing, and a shared-FFN-only variant to assess the contribution of adaptive expert composition. Shared FFN only disables the expert branches of the trained full model at inference, retaining the shared FFNs. ID and OOD task-success scores are reported on VBVR-Pro-Bench as percentages; higher is better.

Routing strategy	Task success(%)	
	ID ↑	OOD ↑
Shared FFN only	64.2	44.7
Uniform routing	63.9	44.7
Random routing	64.8	44.8
Learned routing	**65.9**	**46.1**

B.2 Expert Routing Strategies

Table 5 compares learned routing with uniform and random expert combinations, together with a shared-FFN-only reference. The shared-FFN-only variant disables the expert branches of the trained full model at inference; the + LCL variant in Table 2 is trained without SR-MoE. Learned routing achieves the highest ID and OOD scores of 65.9% and 46.1%, exceeding uniform routing by 2.0 and 1.4 percentage points and random routing by 1.1 and 1.3 points, respectively. Uniform routing does not improve upon the shared-FFN-only reference, while random routing yields only small increases. These results suggest that the benefit of the expert set depends on how its outputs are combined and support learning the routing weights from the prediction context. Figure 6 provides qualitative comparisons of the same expert routing strategies.

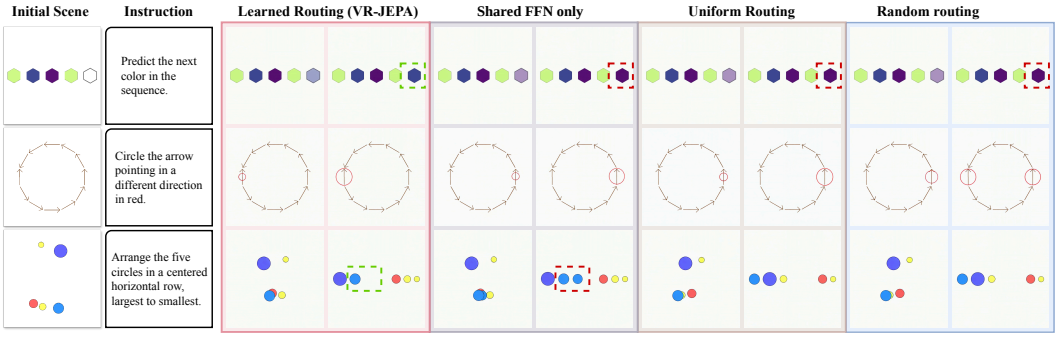

Figure 6: **Qualitative comparison of SR-MoE routing strategies.** We compare video reasoning results with learned routing, shared FFN only, uniform routing, and random routing. The shared-FFN-only variant disables expert branches at inference. Disrupting learned expert coordination degrades reasoning performance in the illustrated examples. Green and red dashed boxes highlight correct and incorrect results, respectively.

B.3 Predicted Trajectory

Table 6 examines whether the generator uses the content of the predicted rollout by changing its trajectory input while keeping the observation, instruction, and sampling seed fixed. Removing trajectory conditioning reduces ID performance from 65.9% to 51.5%, a drop of 14.4 percentage points. Replacing the predicted rollout with one from another sample yields 58.5%, a drop of 7.4 points despite retaining the conditioning pathway. **This comparison indicates that the ID benefit depends in part on the correspondence between the rollout and the current sample, beyond the mere presence of an additional conditioning signal.** The effect is smaller on OOD tasks: removing conditioning lowers the score from 46.1% to 44.9%, whereas cross-sample substitution

Table 6: **Ablation of latent trajectory guidance.** We test whether task success depends on the sample-specific temporal content of the rollout, rather than merely the presence of an additional conditioning signal.

Rollout condition	**Intervention**	**Task success (%)**	
		ID ↑	**OOD** ↑
Predicted rollout	Use the model-predicted trajectory	**65.9**	**46.1**
No conditioning	Remove all trajectory conditioning	51.5	44.9
Repeated initial state	Repeat the initial state across all rollout steps	62.5	43.3
Cross-sample rollout	Use the predicted trajectory from another sample	58.5	45.9

yields 45.9%, close to the original score. Thus, sample-matched rollout content contributes more clearly to ID performance, consistent with the smaller OOD gains in the main evaluation.

C Further Visualization

C.1 Comparisons with Open-source Models

Figure 7 compares VR-JEPA with VBVR-Wan2.2 and LTX-2.3-I2AV on additional reasoning tasks. In the shape-sorting example, VR-JEPA groups objects by type and orders them by size, while VBVR-Wan2.2 interleaves the groups and LTX-2.3-I2AV changes object attributes. In the selective-outlining example, VR-JEPA marks the target circle, whereas the baselines additionally select an outside circle or outline the enclosing region, illustrating errors in applying spatial selection constraints.

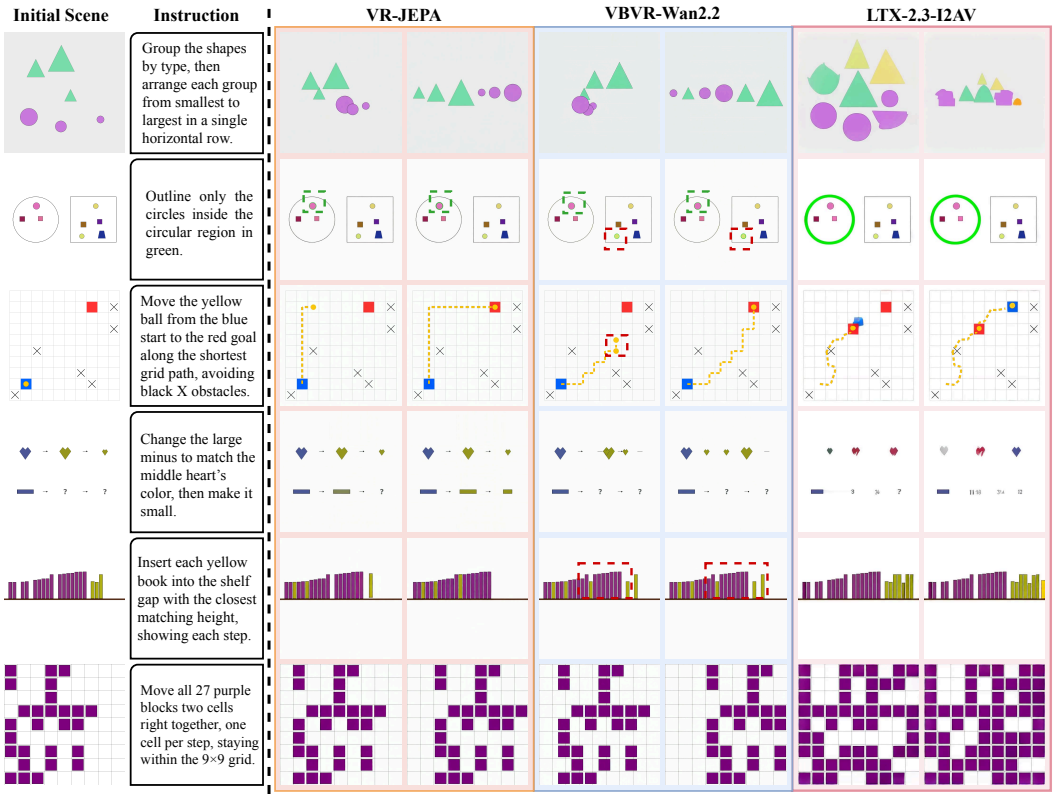

Figure 7: **Additional qualitative comparisons with open-source models on VBVR-Pro-Bench.** Given the initial state and instruction, we compare the video reasoning results of VR-JEPA with VBVR-Wan2.2 and LTX-2.3-I2AV. Green and red dashed boxes highlight correct results and reasoning errors, respectively.

C.2 Comparisons with Proprietary Video Models

Figure 8 compares VR-JEPA with Kling VIDEO 3.0 and Seedance 2.0. In the domino example, VR-JEPA completes the red branch while stopping the blue branch at the marked gap, whereas both baselines propagate beyond it. In the animal-matching example, VR-JEPA places each face in its corresponding outline, while both baselines mismatch the identities and destinations. These examples illustrate better adherence to interaction rules and object correspondence.

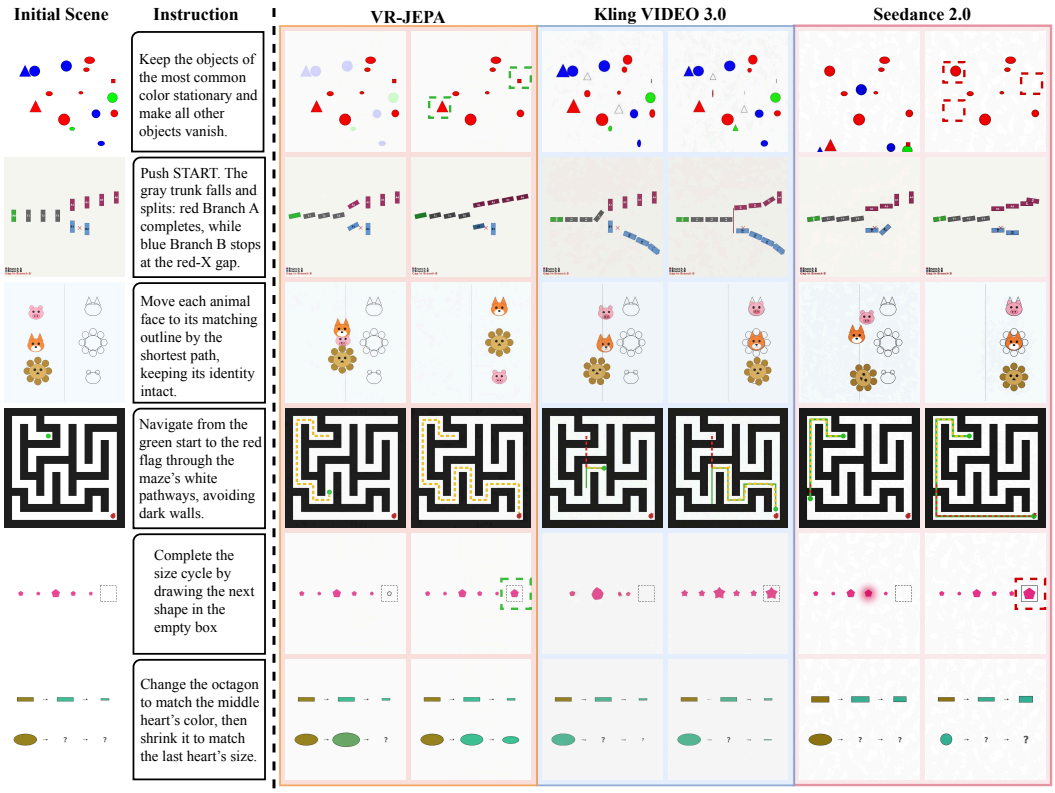

Figure 8: **Additional qualitative comparisons with proprietary video models on VBVR-Pro-Bench.** Given the initial state and instruction, we compare the video reasoning results of VR-JEPA with Kling VIDEO 3.0 and Seedance 2.0. Green and red dashed boxes highlight correct results and reasoning errors, respectively.

C.3 Visual Representation Space Ablation

Figure 9 complements Table 3 by comparing video reasoning results under the three visual representation configurations.

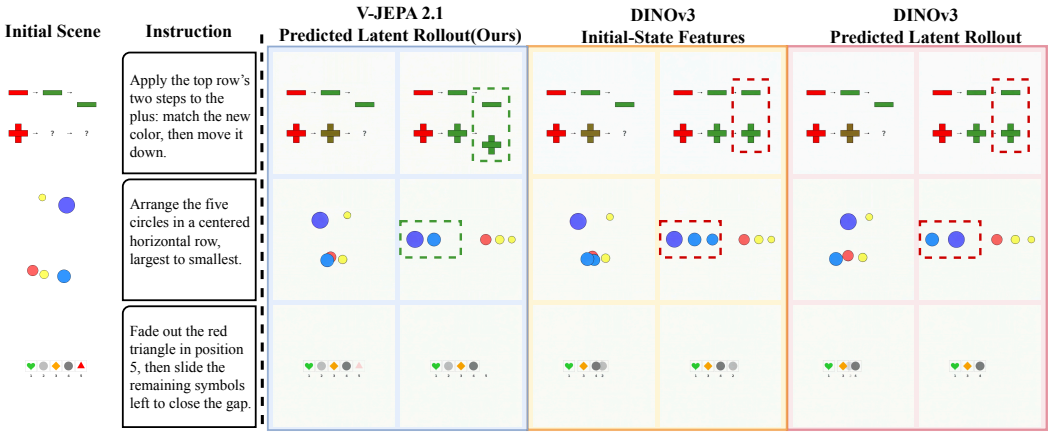

Figure 9: **Qualitative comparison of visual representations on VBVR-Pro-Bench.** Given the initial state and instruction, we compare video reasoning results guided by, from left to right: (1) predicted latent rollouts in V-JEPA 2.1 space (VR-JEPA), (2) initial-state DINOv3 features, and (3) predicted latent rollouts in DINOv3 space. Green and red dashed boxes highlight correct results and reasoning errors, respectively.

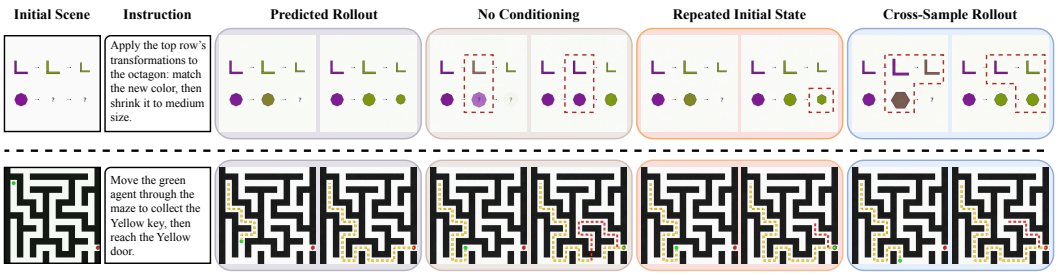

Figure 10: **Qualitative ablation of latent trajectory guidance.** We compare video reasoning results using the predicted rollout with three interventions: removing trajectory conditioning, repeating the initial state across rollout steps, and substituting a rollout from another sample. Red dashed boxes highlight reasoning errors.

C.4 Comparison of latent trajectory guidance

Figure 10 compares generations conditioned on the predicted rollout with three interventions: removing trajectory conditioning, repeating the initial state, and substituting another sample's rollout. In the upper example, predicted-rollout conditioning more closely follows the reference sequence of shape and attribute changes, whereas the interventions introduce mismatches in intermediate or final states. In the maze example, the predicted-rollout result more closely follows the reference path, while the intervened results deviate at the locations marked in red. Removing conditioning tests the presence of trajectory guidance; repeating the initial state tests whether its temporal evolution matters; and substituting another sample's rollout tests the importance of instance-specific content. These examples illustrate the different failure modes, while the quantitative intervention results are needed to assess how consistently they occur.

C.5 Task-Dependent Expert Routing

Figure 11 visualizes the mean routing weights for ten tasks, averaged across layer4 and layer5, the two SR-MoE blocks of the latent predictor. The task-dependent distributions are broadly consistent with the experts' designated skills. Maze navigation and LEGO assembly assign nearly all routing weight to SPA (0.993) and CMP (0.998), respectively, while gravity and bouncing favors DYN (0.688). Unique-shape identification assigns its largest weight to VIS, object-to-target matching to IDN, and next-color prediction and sequence completion to ABS.

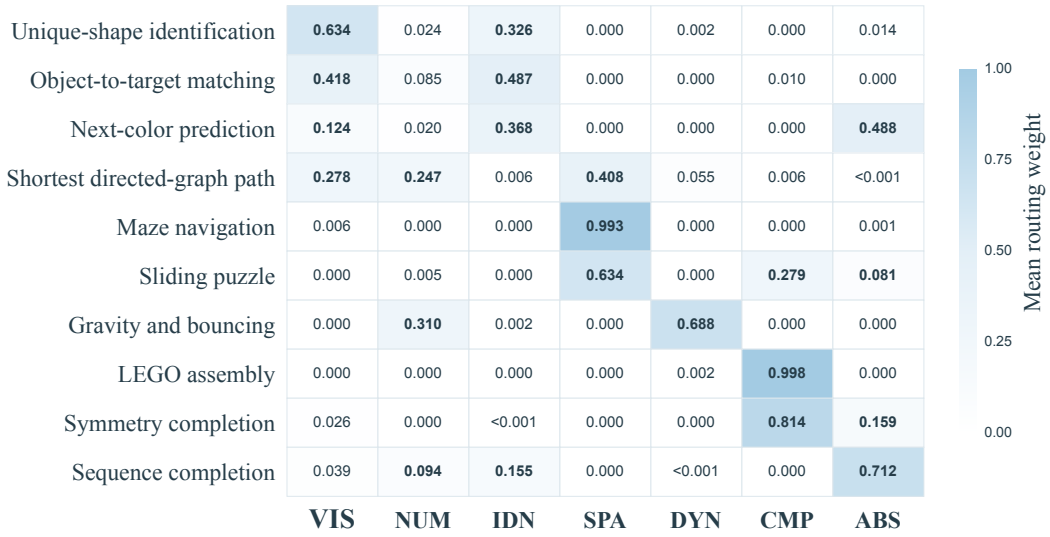

Figure 11: **Task-dependent expert routing.** Mean routing weights for ten tasks, averaged across the two SR-MoE predictor layers (layer4 and layer5). Rows denote tasks and columns denote the seven skill-specific experts; darker cells indicate larger weights. The distributions show both concentrated expert selection and allocations spread across multiple experts.

Other tasks distribute substantial average weight across several experts. Sliding puzzle allocates weight primarily to SPA (0.634) and CMP (0.279), while shortest directed-graph path draws on SPA (0.408), VIS (0.278), and NUM (0.247). These patterns are consistent with task-dependent expert composition rather than a fixed assignment of every task to a single expert. The averages summarize expert utilization rather than establish each expert's causal contribution; the removal analysis in Section C.6 provides complementary evidence by examining the effects of suppressing a selected expert.

C.6 SKILL-SPECIFIC EXPERT REMOVAL

Figure 12 compares learned routing with removal of the highest-weight expert in seven selected examples, ordered by the designated skills VIS, NUM, IDN, SPA, DYN, CMP, and ABS. The interventions produce several distinct errors. Border completion leaves a shape unoutlined; ordinal selection marks the largest circle instead of the second-largest; and shape sorting duplicates a small red square. The maze output includes an invalid path, and the domino output propagates beyond the marked gap. The final two examples select the wrong size in a shape-color sequence and the wrong color in a repeating cycle.

These errors are consistent with the reasoning operations associated with the removed experts and provide qualitative evidence of differentiated contributions. Because the experiment removes the highest-weight expert in selected examples, it does not establish an exclusive mapping between each expert and one skill or separate skill specificity from general sensitivity to removing a strongly weighted expert.

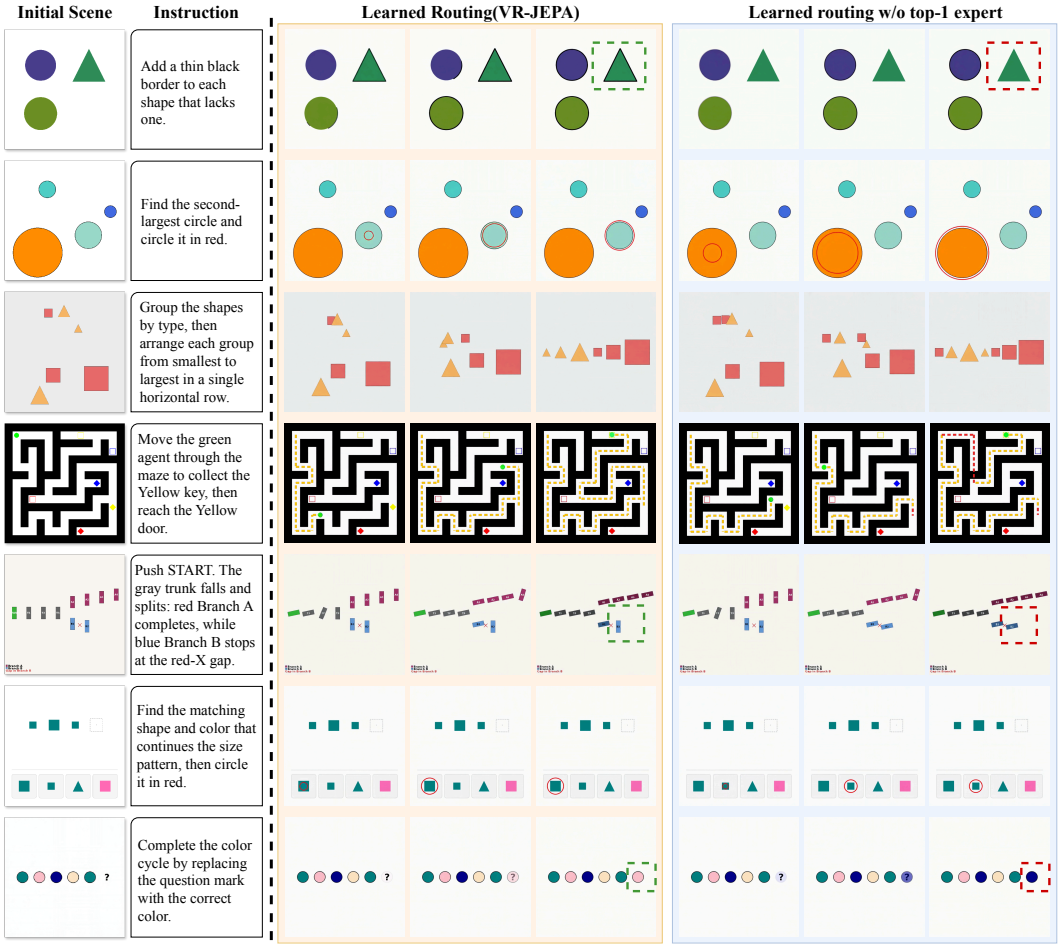

Figure 12: **Probing expert specialization through skill removal.** We remove the highest-weight expert for each of seven examples and examine whether the resulting errors reflect the loss of its associated reasoning skill. From top to bottom, the removed experts correspond to visual discrimination, numerical reasoning, object identity and persistence, spatial reasoning and planning, physical dynamics, compositional reasoning, and abstract rule inference.