ThinkJEPA: Empowering Latent World Models with Large Vision–Language Reasoning Models

Haichao Zhang[1] **Yijiang Li**[2] **Shwai He**[3] **Tushar Nagarajan**[4]
Mingfei Chen[5] **Jianglin Lu**[1] **Ang Li**[3] **Yun Fu**[1]

[1]Northeastern University [2]University of California San Diego [3]University of Maryland
[4]The University of Texas at Austin [5]University of Washington

zhang.haich@northeastern.edu yunfu@ece.neu.edu

Project Page: https://www.zhanghaichao.xyz/ThinkJEPA/
Code: https://github.com/Hai-chao-Zhang/ThinkJEPA

Abstract

Recent progress in latent world models (e.g., V-JEPA2) has shown promising capability in forecasting future world states from video observations. Nevertheless, dense prediction from a short observation window limits temporal context and can bias predictors toward local, low-level interpolation, making it difficult to capture long-horizon semantics and leading to low downstream utility. Vision–language models (VLMs), in contrast, provide strong semantic grounding and general knowledge by reasoning over uniformly sampled observation frames, but they are not ideal as standalone dense predictors due to compute-driven sparse sampling, a language-output bottleneck that compresses fine-grained interaction states into text-oriented representations, and a data-regime mismatch when adapting to small action-conditioned datasets. We propose a VLM-guided JEPA-style latent world modeling framework that combines dense-frame dynamics modeling with long-horizon semantic guidance via a dual pathway design: a dense JEPA branch for fine-grained motion and interaction cues, and a uniformly sampled observation VLM *thinker* branch with a larger temporal stride for knowledge-rich guidance. To transfer the VLM's progressive reasoning signals effectively, we introduce a hierarchical pyramid representation extraction module that aggregates multi-layer VLM representations into guidance features compatible with latent prediction. Across EgoDex, EgoExo4D, BAIR Robot Pushing, and Physion, ThinkJEPA outperforms diverse latent world model and trajectory prediction baselines across egocentric trajectory prediction, long-horizon rollout, robotic latent prediction, and physical-scene forecasting. Beyond forecasting, a closed-loop Reacher evaluation shows that ThinkJEPA increases planning success from 42% with the original V-JEPA predictor to 58%, demonstrating that improved latent predictions can benefit action selection. Together, these results establish the value of video-conditioned VLM guidance for latent forecasting and downstream planning, providing a foundation for future research in agentic robotics that connects contextual understanding with predictive dynamics.

1 Introduction

World models [46] aim to learn predictive abstractions of the environment that support forecasting, planning, and control. Among them, *latent* world models are particularly appealing: by predicting in representation space, they avoid generating photorealistic pixels or detailed 3D geometry, which can be computationally expensive and often unnecessary for downstream decision making. This

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

paradigm, exemplified by JEPA-style methods (e.g., V-JEPA2 [3]), promises improved efficiency and encourages the model to emphasize higher-level structure (e.g., dynamics and physical constraints) rather than overfitting to appearance.

Despite strong progress in V-JEPA2 [3] and its variants, existing JEPA-style latent world models still face two key limitations. **(1) Limited temporal perspective for prediction.** Most approaches rely on a short observation window consisting of densely sampled observation frames to predict future latents. While dense sampling captures fine-grained motion, it restricts temporal context and can bias the predictor toward local dynamics, missing longer-horizon semantics and event-level cues that are critical for robust forecasting. **(2) Weak semantic grounding and general knowledge alignment.** The latent space is typically learned via self-supervised visual representation learning (often related to masked reconstruction/prediction objectives), which yields motion-sensitive features but provides limited alignment to open-vocabulary concepts and compositional knowledge. As a result, the predictor may model *how* things move without understanding *what* the entities are and *which attributes or relations* matter, limiting generalization beyond a narrow domain (e.g., a single manipulation dataset).

A natural alternative is to leverage modern vision-language models (VLMs), which excel at high-level video understanding [41, 9] and reasoning due to large-scale pretraining and multimodal alignment [31, 32]. When applied to uniformly sampled observation frames with a larger temporal stride, VLMs can capture long-range context, recognize entities and their attributes, and draw upon general world knowledge [28, 48] that is often missing from purely visual latent predictors. This complementary capability motivates a promising direction: *using a VLM as a thinker to guide latent world modeling.* However, directly using VLMs as standalone dense predictors is often impractical and can be suboptimal in representation for fine-grained dynamics. **Compute-driven sparsity.** Video VLMs operate under quadratic attention cost and GPU memory constraints, and thus typically process only a small number of uniformly sampled observation frames. This design provides long-horizon context but makes it difficult to model high-FPS, fine-grained dynamics crucial for physical interaction and manipulation. **Language-output bottleneck.** [36] Most VLM pipelines ultimately produce *language* outputs (e.g., captions, rationales, or action descriptions). To generate text, visual information is progressively transformed through stacked transformer layers toward language-generation objectives and discrete token prediction. This induces an output bottleneck: fine-grained spatial details and continuous interaction states (e.g., contact, precise trajectories, fast motions) are compressed into a language-compatible representation, which is effective for semantic recognition but often inadequate for accurate physical forecasting. Consequently, language-based planning with VLM outputs can be coherent in text yet physically inconsistent. **Data regime mismatch.** [43] Moreover, deploying VLMs for domain-specific prediction or control often requires adaptation to relatively small, domain-specific datasets, where naïve fine-tuning can hurt general knowledge and semantic capabilities (e.g., catastrophic forgetting [44]).

These observations suggest that VLMs are best used as *semantic and knowledge-guidance providers*, rather than standalone dense predictors. We therefore propose to *integrate a VLM-thinker branch into a JEPA-style latent world model*, combining dense-frame dynamics modeling with long-horizon semantic guidance in a unified framework. Specifically, we retain the dense-frame observation pathway of V-JEPA-style models to preserve fine-grained motion and interaction cues, while introducing a second branch that feeds uniformly sampled observation frames to a VLM to obtain long-horizon, knowledge-rich guidance. These VLM signals are injected into the JEPA predictor to improve semantic grounding and enhance the generalization of future latent prediction.

A further challenge is *how* to extract useful guidance from a VLM. Using only the final-layer VLM features is often suboptimal: deeper layers are increasingly shaped toward language-generation objectives, while intermediate layers can contain richer visual reasoning signals with better spatial sensitivity. Motivated by this observation, we introduce a **hierarchical pyramid representation extraction** module that aggregates multi-depth VLM representations and distills them into guidance features compatible with the JEPA predictor, enabling the predictor to benefit from the VLM's progressive reasoning process rather than a single terminal representation.

Our contributions are summarized as follows:

- We propose a **VLM-guided JEPA-style latent world model** that integrates a VLM as a *thinker* to provide semantic grounding and general knowledge guidance for future latent prediction.

- We design a **dual-temporal pathway** for observation frames: (i) a dense-frame JEPA pathway for fine-grained dynamics modeling, and (ii) a uniformly sampled VLM pathway with a larger temporal stride to capture long-horizon context and high-level concepts.
- We introduce a **hierarchical pyramid representation extraction** module that aggregates multi-layer VLM features to preserve visual reasoning cues and inject them into the JEPA predictor.
- Extensive experiments across EgoDex, EgoExo4D, BAIR Robot Pushing, and Physion show that ThinkJEPA consistently improves over diverse latent world model and trajectory prediction baselines, demonstrating strong performance across egocentric trajectory prediction, long-horizon rollout, robotic latent prediction, and physical-scene forecasting.

2 Related Works

Latent World Models. Latent world models [12–14] aim to learn predictive abstractions of the environment that support forecasting, planning, and control. By modeling dynamics in a learned representation space, these approaches enable efficient prediction of future states without explicitly generating high-dimensional observations. Early works explore latent dynamics learning for video prediction [8] and model-based reinforcement learning [15], where a compact latent state captures the temporal evolution of observations. Recent advances in predictive representation learning further strengthen this paradigm. In particular, JEPA-style approaches [21, 2] learn representations through predictive objectives that encourage models to capture higher-level structure such as motion patterns and physical interactions. Recent systems such as V-JEPA2 demonstrate the scalability of this approach and show promising results for video understanding and world modeling tasks. Despite these advances, most latent world models are learned solely from visual signals and lack alignment with open-vocabulary semantics or external knowledge, which can limit their ability to incorporate higher-level cues for complex forecasting scenarios.

Vision-Language Models. Vision-language models (VLMs) have achieved remarkable progress in multimodal representation learning by aligning visual and textual modalities using large-scale image–text data [1, 37, 24, 23, 52, 51]. Early approaches focus on joint representation learning and multimodal understanding tasks such as image captioning and visual question answering. More recent multimodal large language models (MLLMs) extend pretrained language models to process visual tokens, enabling instruction following and multimodal reasoning capabilities [19, 53, 26, 27, 18]. Representative systems such as LLaVA series [30, 22] integrate vision encoders with large language models through projection layers or cross-attention mechanisms. While these models demonstrate strong semantic reasoning and multimodal understanding capabilities, they are primarily designed for perception and reasoning tasks, and are not optimized for modeling structured physical dynamics[25].

Multimodal Fusion. Language has increasingly been used as a high-level control signal for visual generation and decision-making systems. Text-conditioned generative models enable natural language prompts to guide image synthesis and editing, as demonstrated by diffusion-based approaches such as DALL·E, Imagen, and Diffusion Transformers (DiT) [38, 40, 34]. Language guidance has also been explored in embodied decision-making frameworks, where large language models provide high-level instructions or goals for perception and action [6]. These works highlight the potential of language as a flexible interface for controlling visual and embodied systems. However, leveraging language signals to guide structured physical forecasting remains relatively underexplored. DIVE-Bench [45] studies high-FPS video understanding, while VQToken [47] learns discrete representations for extreme token reduction. LinkedOut [48] transfers video LLM features to recommendation. LTrajDiff [49] and OOSTraj [50] fuse visual and multimodal signals for trajectory prediction under noisy, occluded observations. ThinkJEPA integrates VLM-derived semantic guidance into dense latent forecasting.

JEPA-style predictors with VLMs. Recent work has explored combining language models with JEPA-style representations, but largely in directions that differ from latent world modeling. For example, VL-JEPA [7] incorporates language signals into a joint-embedding predictive framework, and other approaches use V-JEPA representations as inputs to large language models for video understanding [3]. While effective for multimodal understanding, these designs often shift the primary output interface toward language generation or do not explicitly maintain a latent forecasting interface for downstream world-model tasks. In contrast, ThinkJEPA retains JEPA-style latent forecasting and leverages VLM semantics as *guidance* by injecting VLM-derived features into the JEPA predictor, preserving dense latent prediction while adding long-horizon semantic cues.

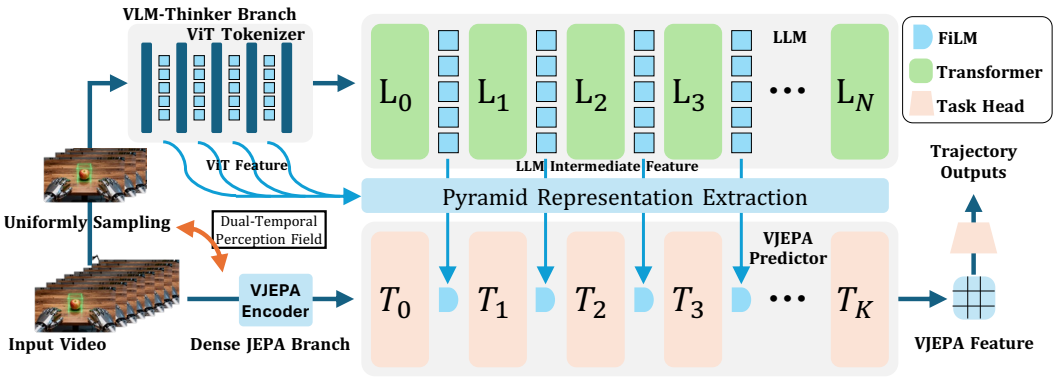

Figure 1: **Overall Architecture of ThinkJEPA.** ThinkJEPA couples a dense JEPA branch for fine-grained latent dynamics modeling with a uniformly sampled VLM-*thinker* branch that provides long-horizon semantic guidance. The VLM guidance—including visual tokens from the ViT visual tokenizer and intermediate hidden states from the language model—is distilled by a *pyramidal representation extraction* module and injected into the V-JEPA predictor via layer-wise modulation. Concretely, guidance derived from language-model layers $\{L_0, \ldots, L_N\}$ is mapped to modulation parameters for predictor layers $\{T_0, \ldots, T_K\}$. The predicted future latents are concatenated with past teacher latents to form the full latent sequence, which is then fed into a task head to produce downstream trajectory predictions.

3 Methodology

3.1 Problem Definition

Basic Settings. Given a video clip v with N frames, our goal is to forecast future latent representations that support downstream tasks; in this work, we focus on 3D hand trajectory prediction. We adopt a JEPA-style latent world modeling paradigm: a visual backbone encodes video frames into latent tokens, and a transformer predictor forecasts future latent tokens from past observations. To improve semantic grounding and long-horizon reasoning, we further condition the predictor on cached features from a video VLM *thinking* model (we use `Qwen3-VL (Thinking)` in our implementation), which serves as a *thinker* providing knowledge-rich guidance.

Long-Horizon Latent Forecasting via Recursion. For long videos where the forecasting horizon exceeds the clip length supported by a single forward pass, we adopt the standard recursive rollout strategy commonly used in JEPA-style predictors. Concretely, the predictor takes the latent tokens forecast in the previous step as input for the next step, enabling iterative rollout of future latents beyond the original window. Although recursion allows arbitrarily long-horizon forecasting, it is susceptible to error accumulation over time. Accordingly, we evaluate both one-shot forecasting and recursive rollouts in our experiments, and analyze robustness under long-horizon prediction.

3.2 Dual-Temporal Perception Field Sampling Architecture

A central challenge in combining VLM reasoning with latent world modeling is the mismatch between (i) the *dense temporal signal* required for accurate dynamics forecasting and (ii) the *long-horizon temporal context* required for semantic understanding and event-level reasoning. Dense sampling preserves high-frequency motion and interaction cues but typically covers only a short time span, whereas sparse uniform sampling covers a long time span but discards dense motion details. To reconcile this trade-off under practical compute and memory budgets, ThinkJEPA adopts a *dual-temporal perception-field* design that explicitly assigns these two roles to two complementary branches.

Given an input video clip $v = \{I_t\}_{t=1}^{N}$ with N frames as *observation*, we construct two temporally sampled inputs: (i) a uniformly sampled clip v_u for the VLM-*thinker* branch, providing a large temporal perception field for global context and semantics; and (ii) a densely sampled clip v_d for the JEPA branch, providing high-frequency temporal cues for fine-grained latent forecasting. The two branches are synchronized at the sample level (derived from the same v) and later fused through layer-wise guidance injection (Sec. 3).

Large temporal perception field sampling for the VLM thinker branch. Video VLMs are powerful for semantic grounding because they can identify entities, attributes, and event-level relationships by leveraging large-scale multimodal pretraining. However, applying transformer-based VLMs to long videos is constrained by quadratic attention cost and GPU memory usage, which typically limits the number of frames that can be processed in a single forward pass. As a result, VLMs commonly adopt *uniform temporal sampling*: a small set of frames is selected to span a long time horizon. Although this choice inevitably discards dense motion details, it maximizes temporal coverage and enables the VLM to reason over long-range context. In ThinkJEPA, we follow this practice and use the VLM branch specifically for long-horizon semantics and knowledge guidance (rather than dense dynamics prediction). We use `Qwen3-VL (Thinking)` as the VLM thinker and cache its intermediate representations for efficient conditioning of the latent predictor. Formally, we define the uniformly sampled clip

$$v_u \;=\; \{I_{s_i}\}_{i=1}^{N_u}, \qquad s_i \;=\; \left\lfloor 1 + (i-1)\cdot \frac{N-1}{N_u - 1} \right\rfloor, \tag{1}$$

where N_u is the number of sampled frames for the VLM thinker branch. This sampling spans the entire clip, providing a large temporal perception field under limited compute.

Dense frame sampling for the JEPA branch. In contrast, JEPA-style latent world modeling requires dense temporal observations to accurately forecast future latents. Fine-grained dynamics, contact changes, and subtle interactions are often expressed as high-frequency temporal signals that are poorly captured by sparse sampling. Therefore, ThinkJEPA uses a dense sampling strategy for the JEPA branch and restricts it to a shorter observation window, where all frames are retained. Formally, we define an observation window starting at frame index t_0 and construct the dense clip

$$v_d \;=\; \{I_t\}_{t=t_0}^{t_0+N_d-1}, \tag{2}$$

where N_d is the number of densely sampled frames. The V-JEPA backbone encodes v_d into per-frame patch tokens, producing past latent tokens F^{past}. A JEPA-style predictor then forecasts future latent tokens $\hat{F}^{\text{fut}}$ from F^{past}. These predicted latents serve as the target representation for downstream heads (e.g., trajectory regression), while the VLM branch provides complementary long-horizon semantic guidance to improve grounding and generalization.

Why dual-temporal sampling matters. The uniform VLM sampling and dense JEPA sampling are not redundant: they target different failure modes. Uniform sampling enables the VLM thinker to access long-range context and semantics that are difficult to infer from a short dense window, whereas dense sampling enables accurate modeling of high-frequency dynamics that sparse VLM inputs cannot represent reliably. By coupling these two perception fields and injecting VLM guidance into the JEPA predictor, ThinkJEPA benefits from both long-horizon semantic context and fine-grained dynamic cues in future latent forecasting.

3.3 JEPA-style latent tokenization and forecasting

The visual backbone encodes a densely sampled clip into per-frame spatial tokens $F \in \mathbb{R}^{B\times T\times P\times D}$, where B is the batch size, T is the number of frames in the observation window, P is the number of spatial tokens per frame, and D is the backbone latent dimension. We split the clip into past and future segments and use a masked-token transformer predictor to forecast future latent tokens from past tokens. The predictor operates in an internal dimension D_p and projects its outputs back to the backbone latent space of dimension D.

Rollout of the JEPA branch. Densely sampled inputs provide strong motion and interaction cues, but they also limit the temporal duration that can be processed in a single forward pass due to compute and memory constraints. For videos whose length exceeds the JEPA observation window, we therefore perform *recursive rollout* by repeatedly forecasting the next segment and feeding the predicted latents into the subsequent step.

Let W denote the number of frames per JEPA window (e.g., $W = T_p + T_f$), and let k index rollout steps. At step k, the predictor takes past latent tokens F_k^{past} and outputs future latent tokens $\hat{F}_k^{\text{fut}}$:

$$\hat{F}_k^{\text{fut}} \;=\; g\big(F_k^{\text{past}}\big)\,, \tag{3}$$

where $g(\cdot)$ is the JEPA-style predictor. For the next step, we set the past tokens to be the previously predicted future tokens (or a shifted window that includes them):

$$F_{k+1}^{\text{past}} \;\leftarrow\; \hat{F}_k^{\text{fut}}. \tag{4}$$

By iterating Eqs. (3), we can roll out arbitrarily long-horizon latent forecasts.

While rollout enables long-horizon prediction, it is susceptible to *error accumulation* and remains limited by the local temporal context within each window. This motivates incorporating VLM-thinker guidance, which provides complementary long-horizon semantic context to stabilize forecasting and improve generalization (Sec. 3.2).

3.4 VLM Thinker: Hierarchical Pyramid Representation Extraction

Complementarity via injecting VLM guidance into JEPA. Prior work has explored combining language and JEPA-style representations in different directions. For example, VL-JEPA [7] and approaches that feed V-JEPA features into LLMs for video understanding [3] primarily treat JEPA features as inputs to a language model. While effective for video-to-text understanding, this design shifts the output space toward language generation and does not directly preserve a latent world model interface for downstream prediction. In contrast, our goal is to retain JEPA-style *latent forecasting* while leveraging VLM semantics as *guidance*. This is non-trivial because the VLM must provide useful long-horizon semantic context without replacing the dense dynamics modeling of the JEPA predictor.

As discussed in Sec. 3.2, uniform sampling enables the VLM thinker to access long-range context and event-level semantics under limited compute, whereas dense sampling provides the JEPA branch with high-frequency temporal signals for fine-grained dynamics. We combine these two pathways by injecting VLM guidance into the JEPA predictor in a layer-wise manner. Concretely, given a uniformly sampled clip v_u and a densely sampled clip v_d, the predictor forecasts future latent tokens conditioned on both VLM guidance and an optional text prompt:

$$\hat{F}^{\text{fut}} \;=\; g\big(F^{\text{past}}(v_d)\,;\,\phi(v_u),\,p\big)\,, \tag{5}$$

where $F^{\text{past}}(v_d)$ are past latent tokens extracted by the V-JEPA backbone from the dense clip, $\phi(v_u)$ denotes VLM-derived guidance features from the uniform clip, p denotes the text prompt provided to the VLM thinker, and $g(\cdot;\cdot)$ is the V-JEPA predictor. In practice, the VLM thinker prompt p is generated from a general summarization request, which helps the thinker focus on relevant entities and events.

Hierarchical pyramid representation extraction. A key question is *which* VLM representations are most suitable for guiding latent forecasting. Using only the final-layer VLM features can be suboptimal, since deeper layers are increasingly shaped toward language-generation objectives, while intermediate layers often retain richer visual reasoning cues and better spatial sensitivity. This observation is supported by prior analyses showing that aggregating intermediate LLM representations can outperform using a single terminal layer for downstream tasks (e.g., [48]). Moreover, visual tokenizer outputs may lose fine-grained cues after passing through multimodal fusion and language decoding stages.

Motivated by these findings, we propose a *hierarchical pyramid representation extraction* module that aggregates multi-depth VLM signals. Specifically, we combine (i) visual tokens from the VLM visual encoder (ViT tokenizer) and (ii) intermediate hidden states from selected language-model layers, forming a depth-wise pyramid over the VLM. These multi-depth features are pooled and projected into the predictor space, yielding guidance features $\phi(v_u)$ that preserve both low-level visual cues and high-level semantic reasoning traces.

Layer-wise guidance. We inject the extracted thinker guidance into the JEPA predictor via feature-wise linear modulation (FiLM) [35]. For predictor block ℓ, the guidance produces modulation parameters $(\gamma_\ell, \beta_\ell)$, and we modulate the block input as

$$\mathrm{FiLM}(z;\gamma_\ell,\beta_\ell) \;=\; \gamma_\ell \odot z + \beta_\ell. \tag{6}$$

This yields layer-wise, sample-specific conditioning that injects semantic and knowledge cues into latent forecasting without requiring the VLM to act as a dense predictor.

Joint prediction for downstream regression. For the basic setting, we follow standard V-JEPA downstream protocols [3] by feeding the predicted latent tokens into a task head for trajectory regression. For long-horizon forecasting with recursive rollout (Sec. 3.3), we concatenate the past latents and the predicted future latents into a full-length token sequence, which is then fed to the temporal regression head to produce the target trajectories.

Table 1: **Quantitative comparison across datasets.** We report trajectory metrics (ADE/FDE/Acc) and latent forecasting metrics (FD/SL1/CD). FD/SL1/CD denote V-JEPA feature distance, latent SmoothL1, and latent cosine distance, respectively. All values are reported with three decimal places.

Dataset	Model	ADE↓	FDE↓	Acc↑	FD↓	SL1↓	CD↓
EgoDex	Qwen3-VL Thinking	0.142	0.144	0.084	99.538	1.656	0.615
	V-JEPA Predictor	0.071	0.066	0.471	74.223	1.252	0.317
	ThinkJEPA	**0.061**	**0.056**	**0.596**	**74.032**	**1.248**	**0.315**
EgoExo4D	Qwen3-VL Thinking	0.661	0.690	0.038	104.548	1.756	0.690
	V-JEPA Predictor	0.659	0.636	0.074	89.244	1.520	0.469
	ThinkJEPA	**0.622**	**0.597**	**0.171**	**79.654**	**1.364**	**0.359**

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate ThinkJEPA on four video forecasting benchmarks covering complementary regimes: egocentric trajectory forecasting (EgoDex [17] and EgoExo4D [11]), robotic latent rollout (BAIR Robot Pushing [10]), and action-free physical-scene forecasting (Physion [5]). EgoDex and EgoExo4D evaluate 3D human/hand trajectory prediction, BAIR evaluates action-conditioned rollout in frozen V-JEPA latent space, and Physion evaluates object-contact prediction from observed and predicted future latents. Detailed dataset protocols, splits, preprocessing, metric definitions, and implementation details are provided in Sec. A.

Observation-only VLM access. To avoid future-frame leakage, the VLM branch is restricted to observed frames in all forecasting experiments. For EgoDex, EgoExo4D, and Physion, uniformly sampled VLM frames are drawn only from the observation window; for BAIR, the VLM observes only the initial block $x_0{:}x_4$, which is reused for all rollout steps. No future labels or target-frame metadata are provided to the VLM branch for BAIR or Physion.

Baselines. We compare ThinkJEPA with Qwen3-VL [4] Thinking (VLM-only), V-JEPA Predictor [3] (JEPA-only), DINO-WM [54], persistence/copy-last for latent rollout, and task-specific EgoDex trajectory baselines including decoder-only and encoder-decoder predictors trained with Behavior Cloning [33], DDPM [16], and Flow Matching [29]. Additional baseline details and training settings are provided in Sec. A.3.

4.2 Experiment Results

Quantitative Comparison: Tab. 1 reports main comparison on EgoDex and EgoExo4D datasets. On these trajectory forecasting benchmarks, ThinkJEPA consistently outperforms both single-branch baselines in trajectory prediction, achieving substantially lower ADE/FDE and markedly higher Acc. Compared to the **V-JEPA Predictor**, injecting VLM-thinker guidance improves semantic grounding while preserving dense dynamics cues, leading to a large gain in downstream trajectory accuracy. Compared to **Qwen3-VL Thinking**, ThinkJEPA avoids relying on sparse, language-oriented representations as a standalone predictor and instead uses the VLM as guidance, yielding a more physically grounded forecast. In addition to trajectory metrics, ThinkJEPA also improves latent forecasting quality (lower FD/SL1/CD), indicating that guidance injection benefits representation prediction rather than only the downstream head. Overall, these results show that **ThinkJEPA can surpass both a strong VLM baseline and a strong latent world model baseline** by integrating long-horizon VLM reasoning with dense JEPA-style latent forecasting.

Task-specific trajectory Comparison: To compare against direct trajectory prediction methods, we include the EgoDex benchmark baselines [20], which combine decoder-only or encoder-decoder predictors with Behavior Cloning (BC), DDPM, and Flow Matching (FM) objectives. Detailed baseline descriptions are provided in Sec. B.5. As shown in Tab. 2, ThinkJEPA achieves the best ADE/FDE among all compared methods. Compared with the strongest task-specific baseline, Decoder-only + BC, ThinkJEPA reduces ADE from 0.0767 to 0.0610 and FDE from 0.0818 to 0.0560. These results show that VLM-guided latent forecasting provides a stronger trajectory representation than directly predicting trajectories with conventional trajectory prediction heads.

Table 2: **Comparison with EgoDex trajectory prediction baselines on EgoDex.** We compare ThinkJEPA against the trajectory prediction baselines reported in EgoDex, including decoder-only and encoder-decoder architectures with Behavior Cloning, DDPM, and Flow Matching. ThinkJEPA achieves the best ADE/FDE among all compared methods.

Group	Model	ADE↓	FDE↓
Trajectory Baselines	Decoder-only + Behavior Cloning	0.0767	0.0818
	Decoder-only + DDPM	0.1148	0.1238
	Decoder-only + Flow Matching	0.1527	0.1574
	Encoder-decoder + Behavior Cloning	0.0774	0.0924
	Encoder-decoder + DDPM	0.1272	0.1245
	Encoder-decoder + Flow Matching	0.1736	0.1557
Latent Forecasting	Qwen3-VL Thinking	0.1420	0.1440
	V-JEPA Predictor	0.0710	0.0660
	ThinkJEPA	**0.0610**	**0.0560**

Table 3: **BAIR Robot Pushing latent rollout.** We evaluate three-step autoregressive rollout in frozen V-JEPA latent space using 5-frame blocks. ThinkJEPA uses only the initial observed block $x_0{:}x_4$ for VLM conditioning.

Method	FD/L2↓	SL1↓	CD↓
Persistence	86.385	1.511	0.430
V-JEPA Predictor	68.711	1.117	0.301
DINO-WM	71.720	1.176	0.323
ThinkJEPA	**67.049**	**1.086**	**0.285**

Table 4: **Physion object-contact prediction.** All non-oracle methods use only the observation prefix. ThinkJEPA achieves the best Acc/AUC.

Method	Acc↑	AUC↑
V-JEPA observed-only	0.738	0.820
V-JEPA Predictor	0.741	0.821
DINO-WM	0.648	0.705
ThinkJEPA	**0.754**	**0.828**

BAIR Robot Pushing: Action-Conditioned Latent Rollout: Setup. We evaluate action-conditioned latent rollout on BAIR Robot Pushing [10], where the model observes the initial 5-frame block and autoregressively predicts the next three future latent blocks in frozen V-JEPA space; detailed protocol are provided in Sec. B.6. **Results and analysis.** Tab. 3 reports BAIR latent rollout results. ThinkJEPA improves over the V-JEPA Predictor on all latent metrics, reducing FD from 68.711 to 67.049, SL1 from 1.117 to 1.086, and CD from 0.301 to 0.285. It also outperforms DINO-WM and the persistence baseline, supporting the effectiveness of ThinkJEPA for action-conditioned latent rollout in robotic interaction.

Physion: Action-Free Physical Scene Forecasting: Setup. We evaluate action-free physical-scene forecasting on Physion [5], where models observe only the video prefix and predict future latent states for downstream object-contact prediction; detailed split, and observation protocol are provided in Sec. B.7. **Results and analysis.** Tab. 4 reports Physion object-contact prediction results. ThinkJEPA achieves the best Accuracy and AUC, improving over the V-JEPA Predictor from 0.741 to 0.754 in Acc and from 0.821 to 0.828 in AUC. Thus, Physion supports the benefit of ThinkJEPA for

Table 5: **Ablation studies.** We vary the VLM token sources and the thinker module. **Encoder** denotes encoding tokens, **AR** denotes autoregressive tokens; **No-dual** disables the dual-temporal VLM pathway, and **No-Th** removes the thinker module. We abbreviate latent metrics as **FD** (feature distance), **SL1** (SmoothL1), and **CD** (cosine distance).

Abl.	ADE↓	FDE↓	Acc↑	FD↓	SL1↓	CD↓
Encoder+V-JEPA predictor	0.128	0.129	0.100	78.869	1.340	0.360
Encoder-only	0.143	0.145	0.086	102.910	1.700	0.615
AR+V-JEPA predictor	0.128	0.130	0.098	78.514	1.333	0.356
AR-only	0.142	0.144	0.086	102.910	1.700	0.615
No-dual-temporal sampling	0.128	0.130	0.099	78.862	1.340	0.360
No-Th	0.071	0.066	0.471	74.223	1.252	0.317
ThinkJEPA	0.061	0.056	0.596	74.032	1.248	0.315

Table 6: **VLM layer selection on EgoDex.** We compare guidance derived from different VLM layer selections. **FD** denotes V-JEPA feature distance, **SL1** denotes latent SmoothL1, and **CD** denotes latent cosine distance.

Variant	ADE↓	FDE↓	Acc↑	FD↓	SL1↓	CD↓
Last-layer	0.128	0.130	0.099	78.858	1.340	0.360
Mid-layer	0.128	0.131	0.098	78.517	1.333	0.356
All layers (ThinkJEPA)	0.061	0.056	0.596	74.747	1.263	0.324

action-free physical-scene forecasting. DINO-WM performs worse under the same split, suggesting that V-JEPA-style latent prediction is a stronger representation backbone.

Ablation baselines: We further conduct controlled ablations to understand which VLM signals are necessary for effective guidance. We study two aspects: (i) the contribution of different VLM token sources, and (ii) the effect of VLM layer selection. Detailed ablation settings are provided in Sec. B.8.

Ablation on VLM token sources: Tab. 5 studies the contribution of encoding tokens and autoregressive (AR) tokens. Single-source VLM guidance is insufficient and can even degrade the JEPA-only predictor, indicating that naively injecting partial VLM signals may introduce incomplete or mismatched conditioning. Encoding tokens mainly preserve input-side visual-context summaries, whereas AR tokens capture generation-side hidden states. When either source is used in isolation, it lacks the complementary information needed to guide dense latent forecasting effectively. In contrast, ThinkJEPA combines both token sources through pyramid representation extraction and layer-wise modulation, enabling the JEPA predictor to receive both visual grounding and generation-side semantic guidance. This suggests that the gain is not due to adding an arbitrary VLM feature, but arises from the interaction between input-side visual grounding and generation-side VLM hidden states.

Ablation on VLM layer selection: Tab. 6 compares guidance derived from different VLM layer selections. We observe a trade-off: last-layer guidance slightly improves trajectory metrics, whereas mid-layer guidance yields better latent forecasting quality. This is consistent with the intuition that deeper layers are increasingly shaped toward language-generation objectives, while intermediate layers retain richer visual reasoning cues. The all-layer pyramid variant achieves the strongest overall performance, motivating our hierarchical multi-depth guidance design.

Long-horizon recursive rollout. To evaluate forecasting beyond a single prediction window, we perform recursive rollout and report horizon-specific trajectory errors in Tab. 7. Qwen3-VL Thinking degrades sharply at longer horizons, confirming that sparse VLM representations are not reliable standalone dense predictors. The V-JEPA Predictor remains stable but accumulates error as the rollout horizon increases. In contrast, ThinkJEPA achieves the best performance across all horizons, suggesting that VLM-guided latent forecasting helps stabilize iterative prediction while preserving dense dynamics modeling. Detailed rollout settings and additional latent-distance diagnostics are provided in Sec. B.9.

Qualitative Results. As shown in Fig. 2, we visualize predicted future hand trajectories by decoding the forecasted V-JEPA latents with the downstream task head and overlaying the resulting 3D joints on a reference frame. Overall, ThinkJEPA produces more plausible and accurate trajectories: the final endpoint (deep red) aligns more closely with the hand in the reference frame, and the temporal progression is smoother and more diverse over time. In contrast, as highlighted by the yellow circles,

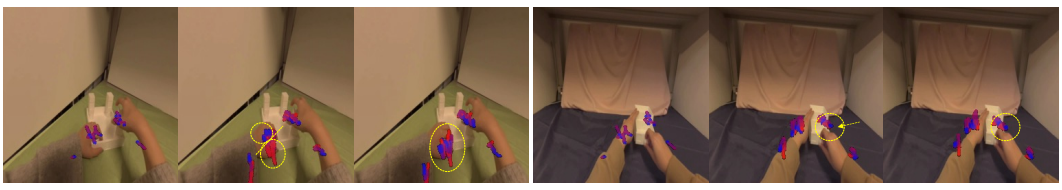

Figure 2: **Qualitative results.** Predicted future hand-manipulation trajectories visualized as heat maps overlaid on the reference frame. Colors indicate temporal progression from blue (earlier) to red (later). Ideally, trajectories transition smoothly from blue to red, indicating coherent motion over time. ThinkJEPA produces smoother trajectories with better temporal consistency and joint alignment.

Table 7: **Recursive rollout on EgoDex: trajectory error vs. horizon.** We perform autoregressive rollout for horizons $H \in \{4, 8, 16, 32\}$. A@H and F@H denote ADE@H and FDE@H, respectively; the lower, the better.

Model	A@4	A@8	A@16	A@32	F@4	F@8	F@16	F@32
Qwen3-VL Thinking	0.140	0.819	1.375	1.026	0.143	2.850	0.286	1.092
V-JEPA Predictor	0.121	0.126	0.134	0.142	0.124	0.136	0.149	0.153
ThinkJEPA	**0.071**	**0.078**	**0.092**	**0.111**	**0.073**	**0.090**	**0.118**	**0.136**

the V-JEPA baseline often exhibits temporally collapsed predictions, where blue points concentrate in a small region, indicating that multiple timesteps and joints are predicted to overlap. In the first example, the VLM-only baseline hallucinates a non-existent left hand, while the V-JEPA baseline yields less accurate joint localization and noisier motion compared to our method.

4.3 Closed-Loop Planning on Reacher

Table 8: **Closed-loop Reacher planning.**

Model	Success (%)↑	Goal error↓
Original V-JEPA	42.0	0.310
Dual-VJEPA	46.0	0.258
ThinkJEPA	**58.0**	**0.253**

We evaluate whether improved latent prediction benefits goal-directed control on MuJoCo Reacher [42], using model predictive control with the cross-entropy method [39]. All methods share 50 evaluation episodes, initial states, goals, the frozen V-JEPA encoder and goal representations, action candidates, planner randomness, planning horizon, execution budget, and success criterion. VLM guidance uses only the observed history.

As shown in Tab. 8, ThinkJEPA improves success over Original V-JEPA by 16 percentage points. Using 10,000 paired episode-level bootstrap resamples, the 95% confidence interval is $[+2, +32]$ percentage points, supporting improved closed-loop action selection. The 12-point advantage over Uniform Dual-VJEPA has a confidence interval of $[-4, +28]$ and remains statistically inconclusive. Final-goal-error differences are also inconclusive. These results establish planning utility relative to the original predictor, while motivating larger evaluations of the VLM-specific benefit.

5 Conclusion

We presented **ThinkJEPA**, a VLM-guided latent forecasting framework that combines long-horizon contextual guidance from a vision-language *thinker* with dense JEPA dynamics prediction. Its dual-temporal design uses uniformly sampled observations for the VLM branch and dense observations for the JEPA branch. Pyramid extraction aggregates VLM representations across multiple depths, while layer-wise modulation integrates this guidance into the JEPA predictor. This design preserves the latent forecasting interface and improves future-state prediction within a fixed V-JEPA representation space. Experiments on EgoDex, EgoExo4D, BAIR Robot Pushing, and Physion demonstrate the effectiveness of ThinkJEPA across egocentric trajectory prediction, long-horizon recursive rollout, action-conditioned robotic latent prediction, and physical-scene forecasting. The closed-loop Reacher evaluation further demonstrates that these prediction improvements can benefit action selection, increasing planning success from 42% with the original V-JEPA predictor to 58%. Together, these results establish the value of combining video-conditioned VLM guidance with dense latent dynamics for both future-state prediction and downstream planning. More broadly, ThinkJEPA provides a foundation for future research in agentic robotics by connecting contextual understanding with predictive dynamics. This framework offers a concrete approach to integrating broad visual knowledge and detailed motion cues into robotic agents that anticipate physical changes and plan their actions.

6 Acknowledgements

This work was conducted at Northeastern University and was supported by Northeastern University's Physical AI Research Initiative (PAIR).

We would like to extend our special thanks to Yuan Tian, Frances Lau, Eric Whitemore, Yue Liu, Ziyi Kou, and Stefan Scherer from Meta Reality Labs Research for their inspiring discussions, valuable technical feedback, and continued encouragement throughout this project.

References

[1] Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., Simonyan, K.: Flamingo: a visual language model for few-shot learning. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022)

[2] Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., Ballas, N.: Self-supervised learning from images with a joint-embedding predictive architecture (2023)

[3] Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Muckley, M., Rizvi, A., Roberts, C., Sinha, K., Zholus, A., et al.: V-jepa 2: Self-supervised video models enable understanding, prediction and planning. arXiv preprint arXiv:2506.09985 (2025)

[4] Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)

[5] Bear, D., Wang, E., Mrowca, D., Binder, F.J., Tung, H.Y., Pramod, R., Holdaway, C., Tao, S., Smith, K.A., Sun, F.Y., et al.: Physion: Evaluating physical prediction from vision in humans and machines. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1) (2021)

[6] Brohan, A., Chebotar, Y., Finn, C., Hausman, K., Herzog, A., Ho, D., Ibarz, J., Irpan, A., Jang, E., Julian, R., et al.: Do as i can, not as i say: Grounding language in robotic affordances. In: Conference on robot learning. pp. 287–318. PMLR (2023)

[7] Chen, D., Shukor, M., Moutakanni, T., Chung, W., Yu, J., Kasarla, T., Bolourchi, A., LeCun, Y., Fung, P.: Vl-jepa: Joint embedding predictive architecture for vision-language. arXiv preprint arXiv:2512.10942 (2025)

[8] Ebert, F., Finn, C., Dasari, S., Xie, A., Lee, A., Levine, S.: Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. arXiv preprint arXiv:1812.00568 (2018)

[9] Feng, Y., Li, Y., Zhang, W., Zheng, S., Luo, H., Yue, Z., Lu, Z.: Videoorion: Tokenizing object dynamics in videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 20401–20412 (October 2025)

[10] Finn, C., Goodfellow, I., Levine, S.: Unsupervised learning for physical interaction through video prediction. Advances in neural information processing systems **29** (2016)

[11] Grauman, K., Westbury, A., Torresani, L., Kitani, K., Malik, J., Afouras, T., Ashutosh, K., Baiyya, V., Bansal, S., Boote, B., et al.: Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19383–19400 (2024)

[12] Ha, D., Schmidhuber, J.: World models. arXiv preprint arXiv:1803.10122 (2018)

[13] Hafner, D., Lillicrap, T., Ba, J., Norouzi, M.: Dream to control: Learning behaviors by latent imagination (2020)

[14] Hafner, D., Lillicrap, T., Norouzi, M., Ba, J.: Mastering atari with discrete world models. arXiv preprint arXiv:2010.02193 (2020)

[15] Hafner, D., Pasukonis, J., Ba, J., Lillicrap, T.: Mastering diverse domains through world models. arXiv preprint arXiv:2301.04104 (2023)

[16] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)

[17] Hoque, R., Huang, P., Yoon, D.J., Sivapurapu, M., Zhang, J.: Egodex: Learning dexterous manipulation from large-scale egocentric video (2025)

[18] Huang, S., Dong, L., Wang, W., Hao, Y., Singhal, S., Ma, S., Lv, T., Cui, L., Mohammed, O.K., Patra, B., Liu, Q., Aggarwal, K., Chi, Z., Bjorck, J., Chaudhary, V., Som, S., Song, X., Wei, F.: Language is not all you need: Aligning perception with language models. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)

[19] Huang, W., Jia, B., Cao, S., Ye, Z., Xu, Z., Hu, Y., Lin, S., et al.: Vision-r1: Incentivizing reasoning capability in multimodal large language models. In: International Conference on Learning Representations. vol. 2026, pp. 63794–63812 (2026)

[20] Jia, X., Donat, A., Huang, X., Zhao, X., Blessing, D., Zhou, H., Wang, H.A., Zhang, H., Wang, Q., Lioutikov, R., et al.: X-il: Exploring the design space of imitation learning policies. arXiv preprint arXiv:2502.12330 (2025)

[21] LeCun, Y., et al.: A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. Open Review **62**(1), 1–62 (2022)

[22] Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Li, Y., Liu, Z., Li, C.: Llava-onevision: Easy visual task transfer. ArXiv **abs/2408.03326** (2024)

[23] Li, J., Li, D., Savarese, S., Hoi, S.C.H.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: International Conference on Machine Learning (2023)

[24] Li, J., Li, D., Xiong, C., Hoi, S.C.H.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International Conference on Machine Learning (2022)

[25] Li, Y., Gao, Q., Zhao, T., Wang, B., Sun, H., Lyu, H., Hawkins, R.D., Vasconcelos, N., Golan, T., Luo, D., et al.: Core knowledge deficits in multi-modal language models. arXiv preprint arXiv:2410.10855 (2024)

[26] Li, Y., Liang, Y., Tian, Y., Wang, B., Zhang, K., Yin, Z., Fu, D., Torr, P., Vasconcelos, N.: Self-supervised visual on-policy distillation. arXiv preprint arXiv:2608.14144 (2026)

[27] Li, Y., Wang, B., Liang, Y., Tian, Y., Fu, D., Vasconcelos, N.: On-policy self-distillation without any supervision. arXiv preprint arXiv:2608.06296 (2026)

[28] Li, Y., Wang, B., Vasconcelos, N.: Learning to think and explore in 3d world. arXiv preprint arXiv:2610.21783 (2026)

[29] Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)

[30] Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. ArXiv **abs/2304.08485** (2023)

[31] Lu, J., Wang, H., Xu, Y., Wang, Y., Yang, K., Fu, Y.: Representation potentials of foundation models for multimodal alignment: A survey. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. pp. 16669–16684 (2025)

[32] Lu, J., Wang, H., Yang, K., Zhang, Y., Jenni, S., Fu, Y.: The indra representation hypothesis for multimodal alignment. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)

[33] Nasiriany, S., Maddukuri, A., Zhang, L., Parikh, A., Lo, A., Joshi, A., Mandlekar, A., Zhu, Y.: Robocasa: Large-scale simulation of everyday tasks for generalist robots. arXiv preprint arXiv:2406.02523 (2024)

[34] Peebles, W.S., Xie, S.: Scalable diffusion models with transformers. 2023 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 4172–4182 (2022)

[35] Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: Film: Visual reasoning with a general conditioning layer. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)

[36] Pikabea, I., Lacunza, I., Velasco, O.P., Escolano, C., Gonzalez-Agirre, A., Hernando, J., Villegas, M.: Breaking language barriers in visual language models via multilingual textual regularization. In: Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics. pp. 299–337 (2025)

[37] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (2021)

[38] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. ArXiv **abs/2204.06125** (2022)

[39] Rubinstein, R.: The cross-entropy method for combinatorial and continuous optimization. Methodology and computing in applied probability **1**(2), 127–190 (1999)

[40] Saharia, C., Chan, W., Saxena, S., Lit, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Gontijo-Lopes, R., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M.: Photorealistic text-to-image diffusion models with deep language understanding. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. NIPS '22, Curran Associates Inc., Red Hook, NY, USA (2022)

[41] Tang, Y., Bi, J., Xu, S., Song, L., Liang, S., Wang, T., Zhang, D., An, J., Lin, J., Zhu, R., et al.: Video understanding with large language models: A survey. IEEE Transactions on Circuits and Systems for Video Technology (2025)

[42] Todorov, E., Erez, T., Tassa, Y.: Mujoco: A physics engine for model-based control. In: 2012 IEEE/RSJ international conference on intelligent robots and systems. pp. 5026–5033. IEEE (2012)

[43] Xiao, J., Huang, N., Qin, H., Li, D., Li, Y., Zhu, F., Tao, Z., Yu, J., Lin, L., Chua, T.S., et al.: Videoqa in the era of llms: An empirical study. International Journal of Computer Vision **133**(7), 3970–3993 (2025)

[44] Zhai, Y., Tong, S., Li, X., Cai, M., Qu, Q., Lee, Y.J., Ma, Y.: Investigating the catastrophic forgetting in multimodal large language models. In: NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following (2023)

[45] Zhang, H., Chai, W., He, S., Li, A., Fu, Y.: Dive-bench: Codec-style high-fps video understanding. In: European Conference on Computer Vision. pp. 418–435. Springer (2026)

[46] Zhang, H., Chen, M., He, S., Xu, Z., Shen, Y., Huang, Y., Lu, J., Li, Y., She, Y., Fu, Y.: A survey of physical ai: A history from chatgpt to world models and embodied agents. Preprints (June 2026). https://doi.org/10.20944/preprints202606.0173.v1, `https://doi.org/10.20944/preprints202606.0173.v1`

[47] Zhang, H., Fu, Y.: Vqtoken: Neural discrete token representation learning for extreme token reduction in video large language models. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)

[48] Zhang, H., Lu, Y., Wang, L., Li, Y., Chen, D., Xu, Y., Fu, Y.: Linkedout: Linking world knowledge representation out of video llm for next-generation video recommendation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7111–7121 (2026)

[49] Zhang, H., Xu, Y., Lu, H., Shimizu, T., Fu, Y.: Layout sequence prediction from noisy mobile modality. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 3965–3974 (2023)

[50] Zhang, H., Xu, Y., Lu, H., Shimizu, T., Fu, Y.: Oostraj: Out-of-sight trajectory prediction with vision-positioning denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14802–14811 (2024)

[51] Zhang, W., Feng, Y., Luo, H., Li, Y., Yue, Z., Zheng, S., Lu, Z.: Unified multimodal understanding via byte-pair visual encoding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12976–12986 (2025)

[52] Zhang, W., Xie, Z., Feng, Y., Li, Y., Xing, X., Zheng, S., Lu, Z.: From pixels to tokens: Byte-pair encoding on quantized visual modalities. arXiv preprint arXiv:2410.02155 (2024)

[53] Zheng, Z., Yang, M., Hong, J., Zhao, C., Xu, G., Yang, L., Shen, C.: Deepeyes: Incentivizing" thinking with images" via reinforcement learning. In: International Conference on Learning Representations. vol. 2026, pp. 126775–126798 (2026)

[54] Zhou, G., Pan, H., Lecun, Y., Pinto, L.: Dino-wm: World models on pre-trained visual features enable zero-shot planning. In: International Conference on Machine Learning. pp. 79115–79135. PMLR (2025)

A Additional Experimental Details

A.1 Datasets and Protocols

A.1.1 EgoDex.

EgoDex [17] is a large-scale benchmark for egocentric dexterous manipulation. It provides egocentric videos paired with 3D hand and finger pose annotations, making it a natural testbed for latent video forecasting and downstream trajectory regression. We use the same cached train/test split for all EgoDex methods to ensure fair comparison across VLM-only, V-JEPA-only, ThinkJEPA, and ablation variants.

A.1.2 EgoExo4D.

EgoExo4D [11] is a large-scale multimodal, multiview dataset of skilled human activities. It contains synchronized egocentric and exocentric videos, with annotations including 3D body pose, 3D hand pose, and gaze. We use EgoExo4D to evaluate whether ThinkJEPA transfers beyond the EgoDex setting to broader human-motion forecasting from egocentric observations.

A.1.3 BAIR Robot Pushing.

BAIR Robot Pushing [10] is used to evaluate action-conditioned latent video rollout in a robotic interaction setting. We use the `bair_robot_pushing_small` dataset with 64×64 videos. The full TFDS split contains 43,264 training videos and 256 official test videos; in our controlled run, we use 2,000 training videos, 500 validation videos, and the official 256-video test split. Each sequence uses the first 20 frames and is divided into four 5-frame latent blocks. The model observes the first block $x_0{:}x_4$ and autoregressively predicts the next three future latent blocks $x_5{:}x_9$, $x_{10}{:}x_{14}$, and $x_{15}{:}x_{19}$ in frozen V-JEPA latent space. Action-conditioned variants additionally receive robot actions, while visual-only baselines use only observed latent blocks.

A.1.4 Physion.

Physion [5] is used to evaluate action-free physical scene forecasting. We use a controlled split with 2,000 training videos, 400 validation videos, and 1,200 test videos across eight scenarios: collide, contain, dominoes, drape, drop, link, roll, and support. The observation ratio is 0.5: non-oracle methods only observe the prefix of each video and must predict future latent states from the observed context. Future frames are used only as prediction targets.

A.1.5 Observation-only VLM access.

For all datasets, ThinkJEPA restricts VLM access to observed frames only. For EgoDex, EgoExo4D, and Physion, VLM frames are sampled from the observation window. For BAIR, the VLM observation is fixed to the initial observed block $x_0{:}x_4$ and reused for all three rollout steps. Thus, the VLM branch never observes target future frames during forecasting or rollout. For BAIR and Physion, no future labels or target-frame metadata are provided to the VLM branch.

A.2 Evaluation Metrics

A.2.1 Trajectory metrics.

Let $\hat{Y} \in \mathbb{R}^{B\times T_f\times J\times 3}$ and $Y \in \mathbb{R}^{B\times T_f\times J\times 3}$ denote predicted and ground-truth 3D trajectories over T_f future frames and J joints. We report ADE, the mean Euclidean distance over all future frames and joints; FDE, the Euclidean error at the final future frame averaged over joints and batch; and threshold Accuracy, the fraction of predicted joint positions whose Euclidean error is below 0.05 m.

A.2.2 Latent forecasting metrics.

For latent prediction, we compare predicted and target V-JEPA latents using FD (feature ℓ_2 distance), SL1 (SmoothL1 distance), and CD (cosine distance, defined as $1-\cos(\cdot)$). These metrics measure representation-level forecasting quality independent of the downstream trajectory head.

A.2.3 Rollout metrics.

For recursive rollout evaluation, A@H and F@H denote ADE@H and FDE@H at rollout horizon $H \in \{4, 8, 16, 32\}$. For BAIR latent rollout, we report FD/L2, SL1, and CD over autoregressively predicted latent blocks.

A.2.4 Physion contact metrics.

For Physion, we evaluate object-contact prediction (OCP) using binary classification Accuracy and ROC-AUC. These metrics are distinct from the 0.05 m trajectory Accuracy used for EgoDex and EgoExo4D.

A.3 Baselines and Variants

A.3.1 ThinkJEPA.

ThinkJEPA uses dense-frame V-JEPA tokens for latent forecasting and injects VLM-thinker guidance derived from both *encoding tokens* and *autoregressive (AR) tokens*. The VLM guidance is obtained from multi-depth VLM representations and injected into the V-JEPA predictor via layer-wise modulation.

A.3.2 Qwen3-VL Thinking (VLM-only).

To isolate the VLM branch, we disable the dense V-JEPA latent input while keeping the VLM branch unchanged. We then train the same downstream head on VLM-derived representations. This baseline tests whether long-horizon VLM representations alone can support dense trajectory forecasting under matched task supervision.

A.3.3 V-JEPA Predictor (JEPA-only).

We train a V-JEPA predictor and the same downstream head without VLM conditioning. This baseline represents dense latent forecasting without semantic VLM guidance.

A.3.4 Persistence / copy-last.

For latent rollout experiments such as BAIR Robot Pushing, we include a persistence baseline that repeats the last observed latent block for all future rollout steps. This baseline does not model dynamics and serves as a sanity-check lower bound.

A.3.5 DINO-WM.

We compare against DINO-WM [54], a latent predictor based on DINOv2 visual latents with a transformer future predictor. This baseline tests whether the observed gains are specific to the V-JEPA latent space or also hold against another strong self-supervised visual representation.

A.3.6 Task-specific trajectory baselines.

For EgoDex trajectory prediction, we additionally compare against decoder-only and encoder-decoder trajectory predictors combined with Behavior Cloning [33], DDPM [16], and Flow Matching [29], following the EgoDex benchmark protocol.

A.3.7 Controlled ablations.

We conduct additional ablations on VLM token sources, VLM layer selections, conditioning mechanisms, and prompt-only VLM inference. For token-source ablations, **Encoder** denotes encoding tokens and **AR** denotes autoregressive tokens.

B Model Implementation Details

B.1 Backbone.

We use a V-JEPA-L backbone (`vit_large_rope`) to extract per-frame patch tokens with latent dimension D=1024.

B.2 VLM-injected V-JEPA predictor.

We implement a V-JEPA predictor operating in an internal dimension D_p=384 and inject VLM-thinker guidance into the predictor via layer-wise FiLM modulation. We condition each predictor block using $(\gamma_\ell, \beta_\ell)$ derived from cached `Qwen3-VL (Thinking)` representations. The cache provides *encoding tokens* and *autoregressive (AR) tokens*, which are projected to D_p, pooled, and mapped to per-layer FiLM parameters using lightweight MLP adapters. For hierarchical pyramid extraction, we cache intermediate hidden states from VLM layers $\mathcal{L} = \{0, 4, 8, 12, 16, 20, 24, 27\}$.

B.3 Trajectory head.

We use a lightweight temporal trajectory regression head for downstream prediction. The head first aggregates spatial tokens within each frame via attention pooling with a learnable query, producing a per-frame representation. It then applies temporal MLP blocks to model cross-frame dependencies, followed by stride-2 temporal downsampling to align the temporal resolution with the prediction horizon. Finally, a linear projection regresses 3D trajectories with output shape $32 \times 52 \times 3$.

B.4 Shared hyperparameters.

Unless otherwise specified, all experiments share a 64-frame input clip at resolution 256×256, a past/future split of 32/32 frames, a V-JEPA-L backbone with depth/dim 24/1024, a VLM-injected V-JEPA predictor with dimension $D_p = 384$, depth 12, and 6 heads, and a cached `Qwen3-VL (Thinking)` VLM thinker with token dimension $D_c = 2048$.

B.5 EgoDex Trajectory Prediction Baselines

Following the EgoDex benchmark protocol [20], we compare against six task-specific trajectory prediction baselines. These baselines are formed by combining two Transformer architectures with three policy representations.

B.5.1 Architectures.

The **decoder-only** baselines directly predict future trajectories from past observations using a single Transformer-style predictor. The **encoder-decoder** baselines first encode the observed context and then decode future trajectories conditioned on the encoded representation.

B.5.2 Policy representations.

We evaluate three trajectory modeling objectives: **Behavior Cloning (BC)**, which directly regresses future trajectories under supervised imitation; **Denoising Diffusion Probabilistic Models (DDPM)** [16], which model future trajectory generation through a denoising process; and **Flow Matching (FM)** [29], which learns a continuous transformation from noise to trajectory samples. All baselines are trained under the EgoDex 2-second trajectory prediction setting and serve as task-specific references for egocentric hand trajectory forecasting.

B.6 BAIR Robot Pushing Protocol

We evaluate action-conditioned latent video forecasting on BAIR Robot Pushing [10] using `bair_robot_pushing_small`. The full TFDS split contains 43,264 training videos and 256 official test videos; in our controlled run, we use 2,000 training videos, 500 validation videos, and the official 256-video TFDS test split. Each video is resized to 64×64, and we use the first 20 frames of each sequence. We divide each sequence into four 5-frame latent blocks. The model observes the first

block x_0:x_4 and performs three-step autoregressive rollout to predict x_5:x_9, x_{10}:x_{14}, and x_{15}:x_{19} in frozen V-JEPA latent space. Action-conditioned variants additionally receive robot actions, while visual-only baselines use only the observed latent blocks.

B.7 Physion Protocol

We evaluate action-free physical-scene forecasting on Physion [5]. The benchmark contains eight physical-interaction scenarios: collide, contain, dominoes, drape, drop, link, roll, and support. We use a controlled split with 2,000 training videos, 400 validation videos, and 1,200 test videos, balanced across scenarios. The observation ratio is 0.5: non-oracle methods observe only the prefix of each video and must predict future latent states from the observed context.

B.7.1 Evaluation.

We evaluate object-contact prediction (OCP) using a binary readout on top of observed tokens and predicted future latent tokens. We report Accuracy, Balanced Accuracy, and ROC-AUC. These metrics are distinct from the 0.05 m trajectory Accuracy used for EgoDex and EgoExo4D.

B.8 Ablation Settings

B.8.1 Token-source ablations.

To assess which VLM token sources contribute to guidance, we evaluate variants that selectively enable different components: (i) encoding tokens + V-JEPA predictor, (ii) encoding tokens only, (iii) AR tokens + V-JEPA predictor, (iv) AR tokens only, (v) no dual-temporal VLM pathway, and (vi) no thinker module. Here, **Encoder** denotes encoding tokens extracted from the VLM input-side representations, and **AR** denotes autoregressive tokens extracted from generation-side hidden states. The no-thinker variant removes VLM guidance and reduces to the V-JEPA predictor baseline.

B.8.2 Layer-selection ablations.

To study the role of hierarchical pyramid extraction, we compare guidance derived from different VLM layer selections, including last-layer guidance, mid-layer guidance, and the full all-layer pyramid guidance used in ThinkJEPA. All variants follow the same training and evaluation protocol unless otherwise specified.

B.9 Long-Horizon Rollout Evaluation

To evaluate long-horizon forecasting behavior beyond a single prediction window, we perform recursive rollout. We use a short-window predictor configuration with T_p=4 and T_f=4 for each step and recursively roll out to horizons $H \in \{4, 8, 16, 32\}$. At each horizon, we report trajectory errors using ADE@H and FDE@H, denoted as A@H and F@H in the main paper. We also compute Accuracy@H after autoregressive rollout and aggregate latent-distance diagnostics, including L2 distance, SmoothL1 distance, and cosine distance, over the rollout trajectory.

C Prompt + Video to VLM-Conditioned Features

C.1 Experimental setting.

This study evaluates a prompt-conditioned VLM feature design, where the predictor takes cached visual features as the primary input and uses language-modulated VLM features as external conditioning. The training setup follows the same backbone and downstream trajectory head as the main paper, while changing only the conditioning path.

C.2 Experimental details.

The input video is first represented by cached ViT-style spatiotemporal features, which serve as the main predictive substrate. In parallel, video frames together with a text prompt are passed through `Qwen3-VL (Thinking)` to extract VLM conditioning features. The conditioning features

Variant	ADE↓	FDE↓	Acc↑	FD↓	SL1↓	CD↓
Prompt + video → VLM condition	0.069	0.062	0.495	**74.007**	**1.248**	**0.315**
ThinkJEPA	**0.061**	**0.056**	**0.596**	74.747	1.263	0.324

Table 9: **Prompt + video to VLM-conditioned features.** The predictor uses cached visual features as the main trajectory backbone and VLM-derived features as external conditioning.

Stride	ADE↓	FDE↓	Acc↑	FD↓	SL1↓	CD↓
Temporal stride 1	0.071	0.064	0.471	73.920	1.246	0.314
Temporal stride 2	0.073	0.071	0.458	74.266	1.247	0.317

Table 10: **Temporal stride ablation.** Denser temporal sampling improves trajectory prediction and latent forecasting quality.

include two complementary streams: encoder-side representations and autoregressive generation-side representations. These VLM features are injected into the predictor rather than replacing the visual backbone.

C.3 Analysis.

As shown in Tab. 9, prompt-conditioned VLM features provide a competitive design choice. Compared with the full ThinkJEPA model, this variant achieves slightly weaker trajectory prediction (ADE/FDE/Acc: 0.069/0.062/0.495 vs. 0.061/0.056/0.596), but slightly better latent forecasting metrics (FD/SL1/CD: 74.007/1.248/0.315 vs. 74.747/1.263/0.324). These results suggest that prompt-conditioned VLM features are effective for representation guidance, while the full ThinkJEPA design yields a stronger overall downstream trade-off.

D Temporal Stride Ablation

D.1 Experimental setting.

This study examines the role of temporal sampling granularity in the dual-temporal design. We compare two temporal strides while keeping the predictor architecture, training budget, and conditioning mechanism fixed.

D.2 Experimental details.

EgoDex trajectories are first represented as 64 uniformly sampled temporal points over each episode. The prediction protocol uses 32 past points and 32 future points. For **stride 1**, no temporal decimation is applied, so the predictor observes all 64 sampled points. For **stride 2**, the temporal sequence is subsampled before the past/future split, resulting in a coarser temporal representation. Since the original sequence is already uniformly sampled to 64 points, stride 1 corresponds to denser temporal coverage, while stride 2 reduces temporal resolution.

D.3 Analysis.

The results in Tab. 10 show that denser temporal sampling improves both trajectory prediction and latent forecasting quality. Stride 1 outperforms stride 2 on all reported metrics. Compared with both stride variants, the full ThinkJEPA model further improves downstream trajectory performance, achieving the best ADE/FDE/Acc overall on this split.

Conditioning	ADE↓	FDE↓	Acc↑	FD↓	SL1↓	CD↓
FiLM	**0.0706**	**0.064**	0.471	**73.878**	**1.245**	**0.314**
Cross-attn	0.0707	0.066	**0.475**	73.965	1.247	0.315
AdaLN	0.0708	0.065	0.474	74.280	1.253	0.317

Table 11: **Conditioning mechanism ablation.** We compare FiLM, cross-attention, and AdaLN under the same training setup.

E Conditioning Mechanism Ablation

E.1 Experimental setting.

This study compares three conditioning operators under the same backbone, data split, and training budget. Only the conditioning mechanism is varied.

E.2 Experimental details.

We compare three ways of injecting VLM guidance into the predictor: (i) FiLM-style affine modulation, (ii) cross-attention conditioning, and (iii) AdaLN-style adaptive normalization. All variants consume the same cached VLM features and the same base visual representation stream, so differences can be attributed to the conditioning operator itself.

E.3 Analysis.

Tab. 11 shows that all three conditioning mechanisms are competitive. FiLM provides the strongest latent forecasting quality among the three variants, while cross-attention and AdaLN remain close alternatives. Compared with these controlled variants, the full ThinkJEPA model achieves substantially better trajectory prediction (ADE/FDE/Acc), indicating that the final design used in the paper offers the strongest downstream performance under the current setting.

F Direct Visual Conditioning and Deepstack-Token Removal

F.1 Experimental setting.

This study compares two variants: (i) removing the VLM branch entirely and conditioning only on direct visual features, and (ii) keeping the VLM branch but removing the deepstack/thinking-token contribution. We further compare both variants against the full ThinkJEPA model.

F.2 Experimental details.

For **direct visual conditioning**, the predictor removes all VLM conditioning and operates only on visual backbone features. This serves as a controlled visual-only baseline within the same predictor family. For **deepstack-token removal**, the VLM branch is preserved, but the generation-side thinking/deepstack token contribution is explicitly dropped before conditioning is consumed by the predictor. This removal is implemented using token filtering and hard zeroing, ensuring that the removed tokens do not leak through the conditioning path.

F.3 Analysis.

The results in Tab. 12 show that both ablations remain competitive. Dropping deepstack tokens yields slightly stronger latent forecasting quality than direct visual conditioning alone, suggesting that the full VLM branch contributes non-trivial information. However, both variants are weaker than the full ThinkJEPA model in downstream trajectory performance, and ThinkJEPA achieves the best ADE/FDE/Acc overall. This indicates that the complete VLM guidance pathway is most effective when used as part of the full model design.

Variant	ADE↓	FDE↓	Acc↑
Direct visual conditioning	0.071	0.066	0.475
Drop deepstack tokens	0.072	0.066	0.464
ThinkJEPA	**0.061**	**0.056**	**0.596**

Table 12: **Direct visual conditioning vs. deepstack-token removal.** Both ablations remain competitive, while ThinkJEPA achieves the strongest downstream trajectory performance.

F.4 Why FiLM as the default conditioning operator.

Although we compare multiple conditioning operators in the supplementary experiments, we choose FiLM as the default design in ThinkJEPA because our primary goal is to improve *latent feature prediction*, rather than only optimizing the downstream regression head. FiLM performs feature-wise modulation directly in the predictor latent space, allowing the VLM thinker to refine the predicted representation while preserving the JEPA-style latent forecasting interface. Compared with cross-attention, FiLM is lighter-weight and introduces less structural change to the predictor, making it easier to attribute gains to guidance rather than additional token interactions. Compared with normalization-based conditioning such as AdaLN, FiLM provides a more direct channel-wise control over the latent features themselves, which is particularly aligned with our objective of improving representation-level prediction quality. For this reason, we adopt FiLM as the main conditioning operator in the paper, while including other variants as complementary ablations.

G Pure Prompt-Only VLM Baseline

G.1 Experimental setting.

This study evaluates a pure VLM baseline without any task-specific prediction head. Unlike the VLM-only baseline in the main paper, which uses a trained downstream head on top of VLM-derived features, this study directly prompts the VLM with video and text and asks it to output future 3D trajectories in structured form. Its purpose is to provide a zero-shot reference point for direct prompting without task-specific adaptation.

G.2 Experimental details.

We use `Qwen3-VL (Thinking)` as a prompt-only baseline. The model observes only the past segment of the video and is instructed to predict future hand trajectories in world coordinates. It outputs a small set of future waypoints in JSON format, which are then interpolated to the full prediction horizon for evaluation. No trajectory head is trained, making this a zero-shot or prompt-only baseline.

G.3 Analysis.

As shown in Tab. 13, the pure prompt-only baseline performs dramatically worse than ThinkJEPA, with ADE/FDE of 10.855/10.927 compared to 0.061/0.056 for our method. This large gap confirms that direct prompting of a general-purpose VLM is insufficient for fine-grained metric-space trajectory prediction. In addition, parsing success is poor in this setting, indicating that structured trajectory generation itself is unstable under pure prompting. We therefore regard this baseline as an intentionally weak but informative reference point, rather than a competitive predictor for this benchmark.

G.4 Implication for the main-paper VLM-only baseline.

The result in Tab. 13 also clarifies why the *VLM-only* baseline in the main paper is implemented with a trained task head rather than direct prompting. A general-purpose VLM that has not been fine-tuned for future trajectory prediction performs very poorly in this setting, even though it possesses strong general semantic reasoning ability. This indicates that zero-shot prompting alone is insufficient for fine-grained metric-space forecasting of hand motion. Therefore, the main-paper *VLM-only* baseline is intentionally designed as a fairer and stronger comparison: it uses the same task-specific training

Baseline	ADE↓	FDE↓	Acc↑
Qwen3-VL prompt-only	10.855	10.927	0.000
ThinkJEPA	**0.061**	**0.056**	**0.596**

Table 13: **Pure prompt-only VLM baseline.** We directly prompt `Qwen3-VL (Thinking)` to predict future trajectories from video and text, without any task-specific fine-tuning or trained prediction head. The large performance gap to ThinkJEPA indicates that zero-shot prompting is not sufficient for fine-grained metric-space trajectory forecasting. This study is included as a weak reference point only; the *VLM-only* baseline in the main paper is a substantially fairer comparison because it is trained with the same task-specific supervision and downstream head.

protocol and downstream prediction head, while removing the JEPA latent forecasting pathway. In this way, the comparison in the main paper isolates the benefit of JEPA-style latent prediction versus VLM-based features under matched supervision and optimization, rather than comparing against an intentionally weak zero-shot prompt baseline.

Hyperparameter	Value
Input frames (T)	64
Past/Future split (T_p/T_f)	32/32
Input resolution	256×256
Backbone	V-JEPA-L (`vit_large_rope`)
Backbone depth / dim	24 / 1024
Patch embedding	`Conv3d` kernel/stride $(2, 16, 16)$
Predictor	VLM-injected V-JEPA predictor
Predictor dim (D_p)	384
Predictor depth / heads	12 / 6
RoPE / mask tokens	enabled / 2
VLM thinker	`Qwen3-VL (Thinking)` (cached)
VLM token dim (D_c)	2048
Cache clips (N_c)	8
Encoder token length (L_{enc})	480
AR token length (L_{ar})	15
Pyramid layers ($\mathcal{L}$)	$\{0, 4, 8, 12, 16, 20, 24, 27\}$
Guidance injection	layer-wise FiLM
Temporal downsampling	AvgPool stride 2 (64→32)
Output shape	$32 \times 52 \times 3$

Table 14: Key architectural hyperparameters and tensor dimensions.

H Implementation Details

H.1 Shared implementation setting.

Tab. 14 summarizes the key architectural hyperparameters and tensor dimensions used throughout the supplementary experiments. Unless otherwise specified, all experiments share the same base configuration: a 64-frame input clip at resolution 256×256, a V-JEPA-L backbone for latent token extraction, and a VLM-injected V-JEPA predictor operating in a latent dimension of $D_p = 384$. The VLM thinker is instantiated with cached `Qwen3-VL (Thinking)` features, including both encoder tokens and autoregressive tokens, and multi-depth VLM representations are extracted from the pyramid layer set $\mathcal{L} = \{0, 4, 8, 12, 16, 20, 24, 27\}$. Guidance is injected into the predictor via layer-wise FiLM modulation, and the final latent sequence is decoded through temporal downsampling to produce $32 \times 52 \times 3$ trajectory outputs. This table is provided to clarify the common experimental backbone shared by the controlled ablations in the supplementary material.

I Limitations and Future Directions

ThinkJEPA marks a step toward unifying a cortex-like VLM for high-level semantic understanding with a cerebellum-like JEPA branch for low-level latent dynamics. By connecting JEPA-style latent

prediction with VLM-based semantic guidance, ThinkJEPA opens several directions that are not fully explored in this work. First, extending ThinkJEPA to agentic robotics requires integrating latent forecasting with hierarchical planning and feedback-driven interaction. Future systems could use the VLM thinker to decompose tasks into semantic subgoals and monitor progress, while a planner evaluates candidate actions through JEPA-based latent rollouts. Key challenges include grounding semantic subgoals in latent objectives, recovering from execution failures, and coordinating reasoning and control under real-time constraints. Second, generalization beyond the evaluated scenarios remains an open challenge. Stronger VLM thinkers with broader world knowledge may improve generalization, but their robustness under distribution shift requires further evaluation. Finally, scaling ThinkJEPA to longer videos and more diverse embodied environments may require more efficient VLM feature caching and more adaptive temporal guidance mechanisms. More broadly, we hope this work inspires further research on bridging semantic reasoning and latent world modeling with larger and more diverse embodied video datasets.

J Broader Impacts

ThinkJEPA studies VLM-guided latent forecasting for video-based world modeling. Potential positive impacts include improving video understanding, embodied AI, robot learning, and physical-scene forecasting systems that require compact predictive representations without expensive pixel-level generation. A key motivation of this work is that existing latent world models may lack high-level semantic grounding, common-sense knowledge, and language-guided context when forecasting future states from video alone. By introducing a VLM thinker as a source of broad visual–semantic guidance, ThinkJEPA may help reduce forecasting errors caused by purely local visual extrapolation, especially in settings that require long-horizon context, object-level semantics, or physical commonsense.

The main societal risks arise when predictive world models are deployed in downstream decision-making or robotics systems without sufficient validation. Incorrect forecasts, biased data, or distribution shift may lead to unsafe or unfair decisions, particularly in human-centered environments. In agentic robotics, errors in semantic subgoals and latent forecasts may compound over successive actions, motivating runtime monitoring and mechanisms for human intervention. Egocentric and embodied video data may also raise privacy concerns if collected or used without appropriate consent and safeguards. Potential mitigations include careful dataset governance, privacy-preserving data handling, uncertainty estimation, human oversight, and restricting deployment in safety-critical settings until the forecasting model is validated under diverse real-world conditions. Overall, we view ThinkJEPA as a predictive component for semantically grounded embodied systems; reliable autonomous deployment requires additional planning, control, and safety validation.