PROJECTOR-JEPA: A GATED DELTANET TWO-EXPERT RESIDUAL PROJECTOR BRIDGING VIDEO-LLMS TO WORLD-DYNAMICS UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent Video-LLMs have made rapid progress by combining powerful vision encoders with large language models. However, most models rely on a projector that simply maps frame-level visual features into the LLM input space, without explicitly modeling local changes between frames or long-range state transitions. This limitation can become a bottleneck in video understanding tasks that require temporal structure, such as motion reasoning, event ordering, and state change recognition. We propose Projector-JEPA, a two-expert residual projector that improves world-dynamics understanding in Video-LLMs by explicitly modeling local motion and global state transitions, while keeping the vision encoder, base projector, and LLM frozen. Projector-JEPA augments the output of the base projector with two Gated DeltaNet-based residual experts. The Local Expert processes a temporal sequence at each spatial position, while the State Expert processes spatially pooled frame summaries; both retain recurrent history. To supervise these complementary spatial granularities, we introduce JEPA-inspired auxiliary objectives that combine a local correspondence loss with a masked state reconstruction loss. We evaluate Projector-JEPA on seven Video-LLMs from the PLM, VideoLLaMA3, and InternVL3 families across five video understanding benchmarks. Projector-JEPA consistently improves performance across all models and benchmarks, with larger gains on tasks involving state changes, memory, and motion direction. These results suggest that the projector in Video-LLMs should be viewed not merely as a modality alignment module, but as a key component for conveying dynamic visual information to the LLM.

1 INTRODUCTION

Video-LLMs have become a central framework for multimodal video understanding, yet they remain limited in tracking physical interactions, temporal transitions, and action consequences. Here, *world-dynamics understanding* refers to inter-frame motion and direction, temporal order, state changes, object permanence, and physical or causal relations as measured by the evaluated video benchmarks. Our evaluation concerns these visual capabilities rather than general-purpose world modeling. Research on stronger video representations, including Joint Embedding Predictive Architecture (JEPA) approaches (Assran et al., 2023; 2025; Mur-Labadia et al., 2026), motivates learning predictive structure in the embedding space.

Although vision encoders have been actively improved, better visual representations do not necessarily translate into stronger video understanding in Video-LLMs. A projector connects the vision encoder (VE) to the LLM, commonly through a frame-wise MLP that maps visual features into the language embedding space. Such projectors do not explicitly model temporal flow or state transitions, leaving the LLM to infer relationships from separate frame representations. The LLM backbone's difficulty with temporal concepts can itself limit video temporal reasoning (Li et al., 2024b). This motivates constructing visual tokens that encode temporal changes before they reach the LLM, complementing projector designs focused on compression or cross-modal alignment.

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.

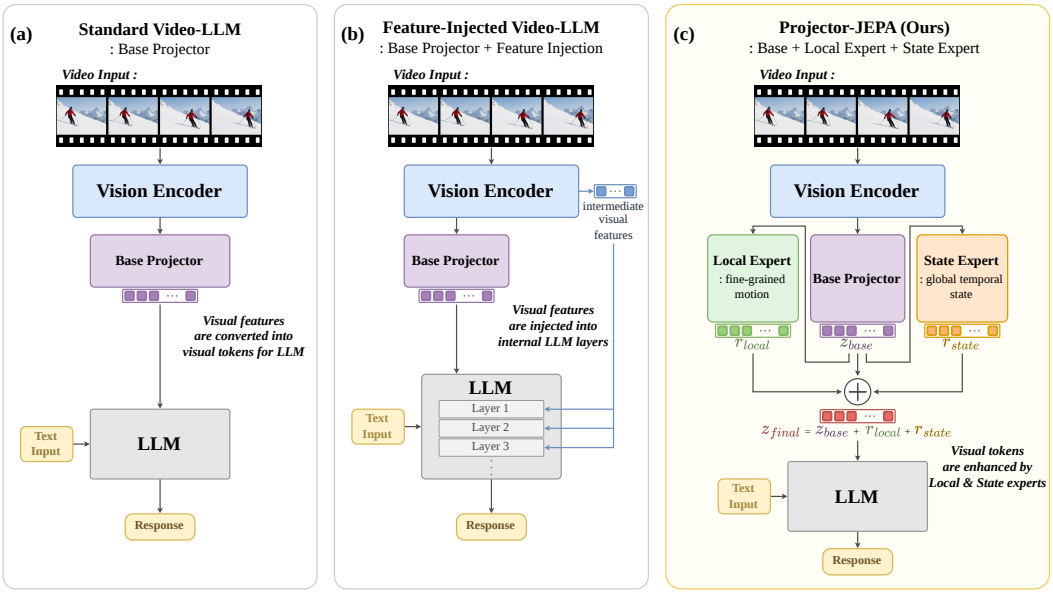


Figure 1: Overview of Video-LLM architectures. (a) Standard projector connection. (b) Injection of vision features into LLM layers. (c) Component-level overview of Projector-JEPA. Both residual branches are always active; computational placement is specified in Section 3.1 and Appendix C.

We propose Projector-JEPA, a two-expert residual projector built on the base projector of pretrained Video-LLMs. A Local Expert and a State Expert use Gated DeltaNet (Yang et al., 2024c) for spatially resolved and spatially pooled temporal processing, respectively. Both branches remain active and add their residuals to the base output. JEPA-style correspondence and masked-reconstruction objectives supervise the branches using the base projector's representations, while the VE, base projector, and LLM remain frozen.

We evaluate seven models from three families—Perception Language Model (PLM) (Cho et al., 2025), VideoLLaMA3 (Zhang et al., 2025), and InternVL3 (Zhu et al., 2025)—on MVBench (Li et al., 2024a), TempCompass (Liu et al., 2024), NEXT QA (Xiao et al., 2021), WorldBench (Upadhyay et al., 2025), and Perception Test (Patraucean et al., 2023). Training on an 80K-sample subset improves all 35 model–benchmark combinations, with a maximum gain of 6.08 percentage points. The t-SNE visualizations additionally show clearer frame-wise separation of projected tokens; we treat this as qualitative evidence of temporal structure rather than a frame-classification result. Our contributions are:

- A two-expert residual projector with Gated DeltaNet branches and JEPA-style auxiliary objectives for local motion and accumulated state, applicable to pretrained Video-LLMs.
- Evaluation across seven models and five benchmarks, supported by controlled branch/loss ablations and representation analyses that examine the combined design.

2 Related Work

2.1 Video-LLM

Video-LLMs and related multimodal LLMs, such as Video-ChatGPT (Maaz et al., 2024), Video-LLaMA (Zhang et al., 2023; 2025), Qwen-VL (Bai et al., 2023), InternVL (Chen et al., 2024b;a; Zhu et al., 2025), and PLM (Cho et al., 2025), have evolved into models that generate responses from video and text inputs by combining visual pretraining approaches such as CLIP (Radford et al., 2021), SigLIP (Zhai et al., 2023), BLIP (Li et al., 2022), and Perception Encoder (Bolya et al., 2025) with LLMs, such as LLaMA (Touvron et al., 2023; Grattafiori et al., 2024), Vicuna (Chiang et al., 2023), Mistral (Jiang et al., 2023), and Qwen (Yang et al., 2024a;b; 2025). Video-LLMs use image-

or video-to-text alignment training to make visual representations compatible with the language model. To this end, a projector, also referred to as an abstractor or connector, is inserted between the VE and the LLM and trained for cross-modal alignment. The role of the projector is to map visual information into the textual embedding space, thereby transforming the output of the VE into a representation that can be accepted by the LLM. LLM then receives the projected visual information together with text as multimodal input and generates a response.

2.2 Temporal Video Understanding

Prior work improves temporal understanding through video instruction tuning and language-side reasoning. VideoChat2 uses progressive multimodal training with diverse instruction data and is evaluated on the temporally focused MVBench tasks (Li et al., 2024a). T3 transfers temporal reasoning from synthesized text tasks to video understanding by adapting the language model (Li et al., 2024b). Projector-JEPA complements these directions by learning temporal residuals in the projector while keeping the vision encoder and LLM frozen.

2.3 Projector

A projector is a module that converts visual representations extracted by a VE into the embedding space of an LLM, thereby connecting different modalities. LLaVA demonstrated effective visual instruction tuning by connecting a pretrained vision encoder to an LLM (Liu et al., 2023). Since then, many VLMs and Video-LLMs have adopted this design as a standard architecture. Meanwhile, approaches such as the Q-Former in BLIP-2 (Li et al., 2023) and the Perceiver Resampler in Flamingo (Alayrac et al., 2022) have proposed using learnable queries to selectively extract and compress visual features. More recently, the role of the projector has expanded beyond simple feature mapping to include visual token compression, local context preservation, and multi-encoder fusion. Honeybee (Cha et al., 2024) proposed a projector structure that preserves local visual context, while Cambrian-1 (Tong et al., 2024) introduced a spatially aware connector to effectively integrate outputs from multiple VEs. In Video-LLMs, the projector must transform a large number of frame-level visual tokens into the limited input space of the LLM. Therefore, balancing visual information preservation and token efficiency has become an important design consideration.

3 Projector-JEPA

3.1 Architecture

A common pipeline for video VLMs maps visual content to language space via a VE $\mathcal{E}_v$ followed by a base projector $\mathcal{P}_b$, whose output $\mathbf{z}_b \in \mathbb{R}^{T\times S\times D}$ (T frames, S spatial tokens per frame, D the LLM input dimension) is fed to a frozen language model. Because $\mathcal{P}_b$ acts on each frame independently, any temporal evidence the LLM consumes must be reconstructed from per-frame embeddings alone. This motivates temporal processing of both spatially resolved tokens and pooled frame summaries before the LLM.

We introduce Projector-JEPA, a two-expert residual projector $\mathcal{M} = (g_\ell, g_s)$ inserted between $\mathcal{P}_b$ and the LLM. Both branches are always active, with no routing or sparse expert selection. They operate on disjoint parameter sets and produce additive residuals at the same shape as $\mathbf{z}_b$:

$$\mathbf{z}_{\text{final}} \;=\; \mathbf{z}_b \;+\; \mathbf{r}_\ell \;+\; \mathbf{r}_s, \qquad \mathbf{r}_\ell, \mathbf{r}_s \in \mathbb{R}^{T\times S\times D},$$

where $\mathbf{r}_\ell = g_\ell(\mathbf{z}_b)$ and $\mathbf{r}_s = g_s(\mathbf{z}_b)$. Because $\mathbf{z}_{\text{final}}$ inherits the layout and dimensionality of $\mathbf{z}_b$, no modification of the downstream LLM is required. Branch g_ℓ runs independent recurrences per spatial position, while g_s runs a recurrence over spatial-mean frame summaries. The two branches differ in spatial granularity and auxiliary supervision, rather than a hard separation of temporal receptive fields; both retain recurrent history.

Both experts share a "bottleneck $\to$ TempConv$_3$ $\to$ GDN $\to$ up-projection" backbone, where $\mathrm{Bottleneck}$ is a linear down-projection $\mathbb{R}^D \to \mathbb{R}^B$ ($B \ll D$), $\mathrm{TempConv}_3$ is a depthwise temporal convolution with kernel size 3, and the up-projection returns to $\mathbb{R}^D$. The Gated DeltaNet (GDN)

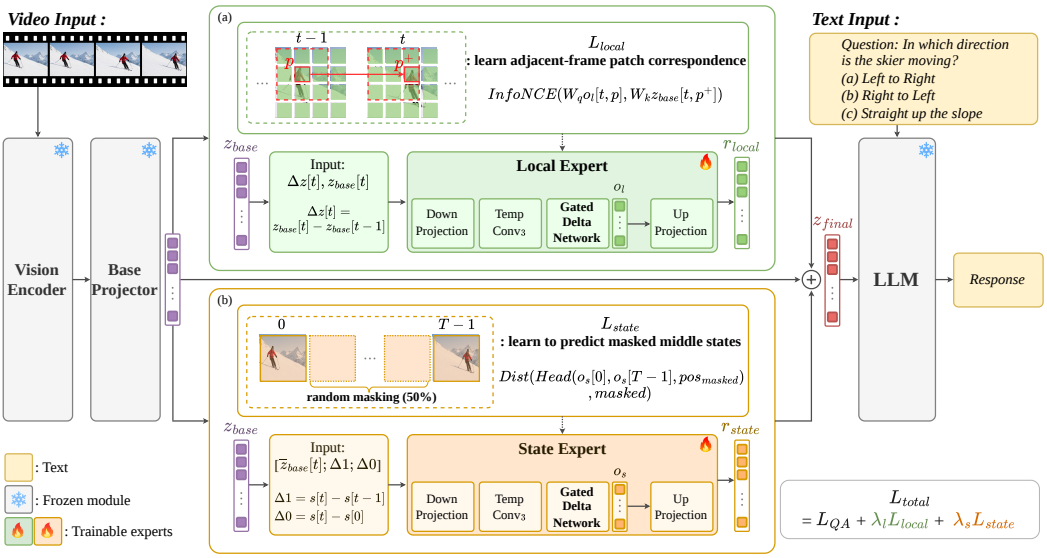


Figure 2: Overview of Projector-JEPA. For every backbone, the frozen base projector supplies both GDN branches, whose residuals are added before the LLM. Masked state reconstruction is an auxiliary training objective.

cell maintains a recurrent state $\mathbf{S}_t \in \mathbb{R}^{B\times B}$. Using the standard scalar-gate notation of Yang et al. (2024c),

$$\mathbf{S}_t = \alpha_t \mathbf{S}_{t-1}\big(\mathbf{I} - \beta_t \mathbf{k}_t \mathbf{k}_t^\top\big) + \beta_t \mathbf{v}_t \mathbf{k}_t^\top, \qquad \mathbf{o}_t = \mathbf{S}_t \mathbf{q}_t.$$

Here $\mathbf{q}_t, \mathbf{k}_t, \mathbf{v}_t \in \mathbb{R}^B$ are input-dependent projections and $\alpha_t, \beta_t \in (0,1)$ are scalar retention and update gates. Equivalently, the correction is $\beta_t(\mathbf{v}_t - \alpha_t \mathbf{S}_{t-1}\mathbf{k}_t)\mathbf{k}_t^\top$, added to $\alpha_t \mathbf{S}_{t-1}$. The delta rule reads the stored value for the current key and updates it through the gated correction. The retention-gate bias is initialized to 2 (Appendix C). Table 7 in Appendix F motivates our choice of GDN, which combines input-dependent memory updates with linear scaling in T.

3.1.1 Local Expert

The local expert g_ℓ targets per-token, frame-to-frame change. We construct its input from the per-token frame difference $\boldsymbol{\delta}_t = \mathbf{z}_{b,t} - \mathbf{z}_{b,t-1} \in \mathbb{R}^{S\times D}$ together with the current frame $\mathbf{z}_{b,t}$:

$$\widetilde{\mathbf{h}}_{\ell,t} \;=\; \mathrm{TempConv}_3\Big(W_\delta^\ell\ \mathrm{LN}(\boldsymbol{\delta}_t) \;+\; W_b^\ell\, \mathbf{z}_{b,t}\Big),$$

where $W_\delta^\ell, W_b^\ell \in \mathbb{R}^{B\times D}$ are the bottleneck projections of the difference and base streams respectively. The GDN cell is then run independently for every spatial position $p \in [S]$, yielding $\mathbf{o}_{\ell,t,p} \in \mathbb{R}^B$. The local residual is the up-projected GDN output

$$\mathbf{r}_{\ell,t,p} \;=\; \mathbf{U}^\ell\, \mathbf{o}_{\ell,t,p},$$

added back to $\mathbf{z}_{b,t,p}$. Each spatial position retains its own recurrent trajectory, preserving spatial granularity without imposing a short temporal window.

3.1.2 State Expert

The state expert g_s processes spatially aggregated temporal context. Its input is built from the spatial-mean frame summary $\bar{\mathbf{z}}_{b,t} = \frac{1}{S}\sum_p \mathbf{z}_{b,t,p} \in \mathbb{R}^D$ together with two displacement features—the instantaneous one-step displacement and the cumulative displacement from the first frame:

$$\widetilde{\mathbf{h}}_{s,t} \;=\; \mathrm{TempConv}_3\Big(W^s\ \mathrm{LN}\big[\,\bar{\mathbf{z}}_{b,t} \,\|\, \bar{\mathbf{z}}_{b,t} - \bar{\mathbf{z}}_{b,t-1} \,\|\, \bar{\mathbf{z}}_{b,t} - \bar{\mathbf{z}}_{b,0}\,\big]\Big),$$

with $W^s \in \mathbb{R}^{B\times 3D}$. The GDN now runs *once per frame* (one position per timestep), so its hidden state $\mathbf{S}_{s,t} \in \mathbb{R}^{B\times B}$ aggregates global temporal evidence. The state residual is the up-projected GDN output

$$\mathbf{r}_{s,t} \;=\; \mathbf{U}^s\,\mathbf{o}_{s,t} \;\in\; \mathbb{R}^D,$$

which is broadcast to all spatial positions before being added to $\mathbf{z}_{b,t}$. At inference, this branch uses the complete frame sequence and the displacement features above; the training-only masking procedure is defined in Section 3.2.2.

3.2 Training Objective

We train Projector-JEPA in a single phase with the VE, base projector, and LLM all kept frozen; only the parameters of g_ℓ, g_s, and two small auxiliary heads are updated. The total loss combines the standard next-token QA cross-entropy on $\mathbf{z}_{\text{final}}$ with two auxiliary losses for feature-based correspondence and masked latent reconstruction:

$$\mathcal{L} \;=\; \mathcal{L}_{\text{QA}}\big(\mathbf{z}_{\text{final}}\big) \;+\; \lambda_\ell\,\mathcal{L}_{\text{local}} \;+\; \lambda_s\,\mathcal{L}_{\text{state}}.$$

The QA cross-entropy difference $\mathcal{L}_{\text{QA}}(\mathbf{z}_b) - \mathcal{L}_{\text{QA}}(\mathbf{z}_{\text{final}})$ measures the change in QA prediction loss (Figure 4). The two auxiliary losses are detailed below.

3.2.1 Local Correspondence Loss

To bias g_ℓ toward token-level frame-to-frame motion, we apply an InfoNCE-style contrastive objective with a feature-based correspondence target. For each query position $p \in S$ at frame t, let $\mathcal{N}(p) \subset S$ be the 3×3 spatial neighborhood of p in frame t. The *target* is the neighbor whose base feature is closest to the previous-frame anchor at the same position:

$$p^+_{t,p} \;=\; \arg\max_{p'\in\mathcal{N}(p)} \cos\big(\mathbf{z}_{b,t-1,p},\; \mathbf{z}_{b,t,p'}\big).$$

The student predicts this target from the local GDN hidden via a contrastive logit, with asymmetric query/key projections—query from the local hidden, key from base features—$\mathbf{q}_{t,p} = W_q\,\mathbf{o}_{\ell,t,p}$ and $\mathbf{k}_{t,p} = W_k\,\mathbf{z}_{b,t,p}$:

$$\mathcal{L}_{\text{local}} \;=\; -\sum_{t=1}^{T-1}\sum_{p\in S} \log \frac{\exp\big(\mathbf{q}_{t,p}\cdot \mathbf{k}_{t,\,p^+_{t,p}} / \tau\big)}{\sum_{p'\in\mathcal{N}(p)} \exp\big(\mathbf{q}_{t,p}\cdot \mathbf{k}_{t,p'} / \tau\big)},$$

with contrastive temperature $\tau = 0.07$ and the sum convention in Appendix C. The target $p^+_{t,p}$ is a frozen-feature similarity match, not a ground-truth object displacement. The student predicts this feature-based correspondence from its recurrent representation, which incorporates the current base feature and frame-to-frame differences.

3.2.2 Masked State Reconstruction Loss

To train g_s to predict hidden middle-frame features, sample $\mathcal{M} \subset \{1,\dots,T-2\}$ with $|\mathcal{M}| \approx 0.5(T-2)$. Masking is applied to the student input for this auxiliary training objective (Appendix C); the frozen, unmasked features define the reconstruction targets. Let $\mathbf{o}^{\mathcal{M}}_{s,t}$ denote outputs of this masked stream. For each $m \in \mathcal{M}$,

$$\hat{\mathbf{s}}_m \;=\; \text{StateHead}\big(\,\mathbf{o}^{\mathcal{M}}_{s,0} \,\|\, \mathbf{o}^{\mathcal{M}}_{s,T-1} \,\|\, \tfrac{m}{T-1}\,\big) \;\in\; \mathbb{R}^D,$$

where $\text{StateHead} : \mathbb{R}^{2B+1} \to \mathbb{R}^D$ concatenates the two endpoint GDN outputs with the masked frame's relative position. The loss is the one-minus-cosine to the spatial-mean base feature at frame m:

$$\mathcal{L}_{\text{state}} \;=\; \mathbb{E}_{\mathcal{M}}\,\frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}} \Big(1 \;-\; \cos\big(\hat{\mathbf{s}}_m,\; \bar{\mathbf{z}}_{b,m}\big)\Big).$$

This JEPA-inspired objective reconstructs frozen latent targets from visible temporal context, including the final endpoint; it is masked reconstruction rather than future prediction from past observations alone. The trainable components in this phase are therefore $\{g_\ell, g_s, \text{StateHead}, W_q, W_k\}$; optimization details are summarized in Appendix C.

4 EXPERIMENTAL SETUP

Baseline Models We adopted three families of Video-LLMs that use different VE as our baselines. Specifically, we conducted extensive experiments using a total of seven Video-LLMs: Perception Language Model (PLM) (Cho et al., 2025) with a Perception Encoder at 1B, 3B, and 8B scales; VideoLLaMA3 (VL3) (Zhang et al., 2025) with SigLIP at 2B and 7B scales; and InternVL3 (IV3) (Zhu et al., 2025) with InternViT at 2B and 8B scales. Depending on the model size, PLM uses Llama-3.2-1B, Llama-3.2-3B, and Llama-3.1-8B (Grattafiori et al., 2024) as its LLM backbone. VL3 uses Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct (Yang et al., 2024b), while IV3 uses Qwen2.5-1.5B and Qwen2.5-7B (Yang et al., 2024b). All baseline models use a two-layer MLP projector, which is the most widely adopted connection module between the VE and the LLM.

Benchmarks To evaluate the video understanding capabilities of Video-LLMs from multiple perspectives, we use five benchmarks: MVBench (Li et al., 2024a), TempCompass (Liu et al., 2024), WorldBench (text-based) (Upadhyay et al., 2025), NEXT QA (Xiao et al., 2021), and Perception Test (Patraucean et al., 2023). These benchmarks cover temporal reasoning abilities, including event order, motion direction, speed, and state changes, as well as object interactions, physical scene changes, causal reasoning, and perceptual reasoning. Thus, our evaluation is designed to assess not only standard VideoQA performance, but also temporal change understanding and real-world physical understanding in videos.

Train Datasets We assemble a pool of approximately 240K samples from nine video instruction and video QA datasets and train on an 80K-sample subset. The main datasets are PE-Video with human annotation (Bolya et al., 2025) and LLaVA-Video instruction tuning data (Zhang et al., 2024), which are used for Perception Encoder-based video training and improving general video instruction-following capability, respectively. We further incorporate ActivityNet-QA (Yu et al., 2019), YouCook2 (Zhou et al., 2017), NEXT QA (Xiao et al., 2021), Charades (Sigurdsson et al., 2016), EgoSchema (Mangalam et al., 2023), and other datasets to strengthen everyday activity QA, procedural understanding, grounded QA, action understanding, and long-video comprehension. As a result, our training data covers not only general video instruction following, but also temporal reasoning and diverse video understanding scenarios.

Implementation Details For evaluation, all models use greedy decoding with a temperature of 0. For each model, we conduct three independent training runs with different random seeds and report the average performance of the validation-selected checkpoint from each run. We use an 80K-sample subset, approximately one-third of the 240K-sample pool, for both projector and LoRA training. LoRA training follows the same three-run reporting protocol. Training details are provided in Appendix C; checkpoint selection and evaluation splits are specified in Appendix C.1.

5 RESULTS

As shown in Table 1, Projector-JEPA consistently improves performance across all five large-scale benchmarks, which together include 41 subtasks, on seven Video-LLMs, despite keeping both the VE and LLM frozen during training. This suggests that the local experts and state experts provide the LLM with richer visual tokens by capturing inter-frame relationships that the base projector fails to reflect. Importantly, the newly trained projector operates between the VE output and the LLM input, improving performance without updating the pretrained alignment parameters. Both auxiliary targets are derived from the frozen base projector; no external teacher model or EMA target network is used. Training uses no explicit alignment loss.

We additionally evaluate LLM adaptation with LoRA (Table 1). LoRA-Projector-JEPA starts from a trained expert checkpoint and jointly updates the experts and LLM adapters in a further adaptation phase; LoRA-base adapts the LLM while using the frozen base projector. The former achieves higher scores than LoRA-base across the evaluated model–benchmark combinations. However, the settings differ in initialization and optimization history. Reusing the same 80K samples does not equalize their training histories, so this comparison does not isolate the effect of LoRA alone or establish that representation quality is the sole cause of the gains. Appendix D details the adaptation protocol.

Table 8 summarizes performance by overlapping diagnostic task groups, and Table 9 gives qualitative examples. Across all seven models, the largest gains occur in State Change / Memory, followed by

Table 1: Benchmark accuracy (%). Adapted models report mean ± standard deviation over three seeds; Base scores have no training-seed variation. Bold indicates the better result in the base setting, and underline indicates the better result in the LoRA setting. Proj.-JEPA denotes Projector-JEPA. LoRA-Proj.-JEPA further adapts pretrained experts jointly with LLM adapters (Appendix D).

Model	Method	MVBench	TempC.	WorldB.	NEXT QA	Perc. Test
PLM-1B	base	66.48	61.65	49.21	77.36	64.12
	Proj.-JEPA	**68.11** ±0.4	**63.03** ±0.3	**54.14** ±0.6	**78.90** ±0.5	**64.84** ±0.4
	LoRA-base	66.55 ±0.3	60.48 ±0.5	49.72 ±0.4	77.21 ±0.3	63.98 ±0.4
	LoRA-Proj.-JEPA	68.08 ±0.5	63.34 ±0.4	54.14 ±0.6	79.12 ±0.4	65.68 ±0.5
PLM-3B	base	73.00	66.27	60.77	83.99	77.38
	Proj.-JEPA	**74.23** ±0.3	**70.73** ±0.6	**66.85** ±0.7	**84.55** ±0.4	**78.52** ±0.5
	LoRA-base	72.87 ±0.4	66.35 ±0.5	60.92 ±0.4	83.58 ±0.3	77.45 ±0.4
	LoRA-Proj.-JEPA	74.55 ±0.4	70.88 ±0.5	66.30 ±0.6	84.33 ±0.4	79.26 ±0.5
PLM-8B	base	74.98	72.15	62.98	85.80	82.25
	Proj.-JEPA	**76.26** ±0.4	**73.45** ±0.5	**65.19** ±0.6	**86.25** ±0.3	**84.31** ±0.5
	LoRA-base	74.45 ±0.4	72.21 ±0.5	63.30 ±0.5	85.91 ±0.3	81.19 ±0.6
	LoRA-Proj.-JEPA	76.65 ±0.5	74.48 ±0.4	66.85 ±0.7	86.59 ±0.4	84.57 ±0.5
VL3-2B	base	62.70	64.11	52.49	79.04	61.52
	Proj.-JEPA	**63.22** ±0.4	**65.24** ±0.5	**55.80** ±0.6	**79.99** ±0.4	**62.45** ±0.5
	LoRA-base	62.48 ±0.3	64.22 ±0.4	52.85 ±0.5	79.15 ±0.4	61.64 ±0.4
	LoRA-Proj.-JEPA	63.42 ±0.4	65.38 ±0.5	56.91 ±0.7	80.65 ±0.5	63.33 ±0.5
VL3-7B	base	66.60	69.62	64.09	83.05	67.87
	Proj.-JEPA	**68.58** ±0.5	**73.41** ±0.6	**67.40** ±0.6	**84.02** ±0.4	**68.88** ±0.5
	LoRA-base	66.51 ±0.4	69.65 ±0.5	64.52 ±0.4	83.12 ±0.4	68.11 ±0.5
	LoRA-Proj.-JEPA	68.87 ±0.5	73.66 ±0.6	67.96 ±0.6	84.53 ±0.5	69.45 ±0.5
IV3-2B	base	68.58	61.91	49.17	75.89	60.34
	Proj.-JEPA	**69.75** ±0.4	**63.22** ±0.5	**53.04** ±0.6	**77.03** ±0.4	**62.26** ±0.5
	LoRA-base	67.79 ±0.4	62.11 ±0.4	49.21 ±0.5	75.54 ±0.4	60.11 ±0.4
	LoRA-Proj.-JEPA	69.98 ±0.5	63.95 ±0.5	54.14 ±0.6	77.76 ±0.5	63.62 ±0.5
IV3-8B	base	74.05	71.20	67.40	84.66	74.62
	Proj.-JEPA	**75.21** ±0.4	**72.28** ±0.5	**67.96** ±0.5	**85.82** ±0.4	**76.56** ±0.5
	LoRA-base	73.92 ±0.4	71.34 ±0.4	66.60 ±0.6	84.78 ±0.4	74.55 ±0.5
	LoRA-Proj.-JEPA	75.68 ±0.5	72.59 ±0.5	69.61 ±0.7	85.75 ±0.4	77.72 ±0.6

Motion / Direction. Gains in Spatial / Object Existence and Counting are smaller. These differences describe the observed improvements; they do not establish differences in task difficulty or performance ceilings. Fine-grained / Egocentric gains are also modest in the tested configurations.

6 Analysis

6.1 Controlled Ablations

Table 2 ablates both branches and auxiliary losses across three backbones. All trainable variants use QA supervision. Full exceeds both QA-only variants in all six model–benchmark settings. Removing either auxiliary loss lowers both MVBench and TempCompass (TC) accuracy for every model, supporting the contribution of each objective. Full also exceeds Local-only in all six settings and State-only in five; State-only ties Full on VideoLLaMA3-2B MVBench (63.22), but scores lower on TC (65.12 versus 65.24). Both branches and both auxiliary objectives thus contribute to the combined design, which achieves the highest or joint-highest score in every setting.

Table 2: Component ablations across three backbones. Scores are accuracy (%); bold marks the highest score in each model–benchmark column, including ties. Disabling a loss retains both branches; single-branch variants remove the other branch.

Configuration	λ_ℓ	λ_s	Branches	PLM-1B		VideoLLaMA3-2B		InternVL3-2B	
				MVBench	TC	MVBench	TC	MVBench	TC
Base	–	–	–	66.48	61.65	62.70	64.11	68.58	61.91
Single GDN, QA only	0	0	1	66.92	62.72	62.77	64.55	68.58	62.21
Two branches, QA only	0	0	2	67.03	62.28	62.93	64.22	68.78	62.53
Without local loss	0	0.03	2	67.45	62.09	63.02	64.46	68.82	62.21
Without state loss	0.01	0	2	66.95	62.16	63.13	64.54	68.90	63.02
Local-only	0.01	–	1	67.15	62.47	62.88	64.56	69.12	62.45
State-only	–	0.03	1	67.98	61.96	**63.22**	65.12	68.84	62.89
Full	0.01	0.03	2	**68.11**	**63.03**	**63.22**	**65.24**	**69.75**	**63.22**

Table 3: Controls across three model families (MVBench / TC). Full exceeds both controls in six model–benchmark settings, yielding twelve pairwise comparisons.

Model	Base	QA-only branches	Base-MLP tuning	Full
PLM-1B	66.48 / 61.65	67.03 / 62.28	65.58 / 60.89	**68.11 / 63.03**
VideoLLaMA3-2B	62.70 / 64.11	62.93 / 64.22	62.67 / 64.37	**63.22 / 65.24**
InternVL3-2B	68.58 / 61.91	68.78 / 62.53	68.65 / 62.28	**69.75 / 63.22**

Table 3 evaluates six model–benchmark settings, yielding twelve pairwise comparisons against the two controls. All use the same data and 800-step schedule as Full. QA-only retains both branches without auxiliary losses; Base-MLP fine-tunes the pretrained frame-wise projector. Full wins all twelve comparisons, supporting the tested combination. Protocol details and branch-level accuracy gains are in Appendices H.1, H.2, and H.5.

6.2 Representation Diagnostics

The Local and State experts restructure temporal representations from the base-feature sequence. With trained parameters fixed, their residuals are functions of that sequence, not additional video observations. Table 4a reports cosine similarity to the base output. Low cosine similarity describes geometric differences; rotations or unrelated components can also reduce it, so orthogonality alone does not demonstrate useful temporal semantics.

We fit a ridge-regression probe ($\alpha = 1.0$) from base representations to branch outputs and evaluate test-split R^2. Table 4a reports unexplained variance under this fitted probe as $100(1 - R^2)$. For the State branch, the values are 75.90%, 69.40%, and 83.30% on PLM-1B, InternVL3-2B, and VideoLLaMA3-2B, respectively; Local values range from 45.20% to 59.70%. A separate nonlinear probe leaves 66.9% of Local and 44.7% of State variance unexplained (Appendix H.4).

These diagnostics characterize representational differences under the specified probes, rather than information gain over the complete base-feature sequence. Low reconstruction scores indicate that the fitted token-wise mappings do not recover the expert outputs well; they do not establish semantic novelty or uniquely explain the downstream improvements. In particular, the additional probe sees a same-frame Base token, whereas each expert processes temporal context. Low R^2 is therefore compatible with this context difference and does not exclude reconstruction by a more expressive or sequence-level probe.

Figure 3 visualizes how the two-expert projector restructures the base sequence. Base representations form compact groups with frame colors intermingled, whereas Projector-JEPA produces extended trajectories with clearer progression from early to late frames. This pattern illustrates temporal organization in the projected features, complementing the branch/loss controls in Section 6.1. The visualization describes representation geometry, not frame-index classification accuracy or an independent measure of temporal reasoning.

Table 4: Residual diagnostics before addition to Base. Cos. is $\cos(\mathbf{z}_b, \mathbf{r})$; Norm Ratio is $\|\mathbf{r}_s\|/\|\mathbf{z}_b\|$. Unexplained variance (%) is $100(1 - R^2)$ under the fitted ridge probe. Panel (b) compares State residuals across branch architectures; these are geometric statistics, not quality rankings.

(a) Branch representation diagnostics.

Model	Branch	Cos.	Unexplained (%)
PLM-1B	local	0.001	59.70
	state	0.028	75.90
InternVL3-2B	local	-0.013	47.20
	state	0.046	69.40
VideoLLaMA3-2B	local	0.021	45.20
	state	0.037	83.30

(b) State residuals on PLM-1B.

Metric	GDN	MLP	Cross-Attn
Cos.	0.028	0.214	0.137
Norm Ratio	0.745	0.631	0.684
R^2	0.241	0.842	0.763
Unexplained (%)	75.90	15.80	23.70

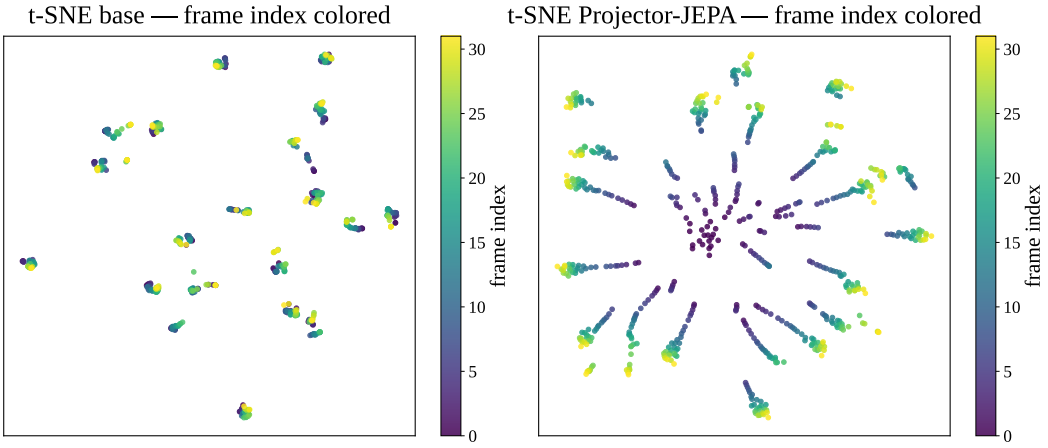


Figure 3: Temporal organization of projected representations on MVBench: frozen base projector (left) and Projector-JEPA (right). Colors denote frame index (0–31). We sample one 32-frame video from each of the 20 MVBench subtasks and visualize InternVL3-8B representations with t-SNE. Projector-JEPA displays clearer within-video progression across frame indices in these examples.

6.3 Comparison of Projector Architectures

Table 4b compares State residuals before addition to Base. All retain Base plus Local and State residuals; MLP and cross-attention replace the complete branches. Each MLP branch uses Base's two-layer architecture. Appendix H.3 distinguishes newly initialized MLP branches (65.08/61.27 on MVBench/TC) from fine-tuning the pretrained Base MLP (65.58/60.89), versus Full (68.11/63.03). Figure 5 shows the same Base, Local, and State components.

7 Conclusion

We show that temporal residual adaptation at the projector interface improves video understanding across seven pretrained Video-LLMs while keeping their vision encoders, base projectors, and language models frozen. Projector-JEPA combines Local and State residual branches with Gated DeltaNet and auxiliary targets derived from the frozen base projector. Across five benchmarks, the gains and controlled ablations support the tested combination of temporal branches and training objectives. These results identify the projector interface as a useful adaptation point for temporal video understanding, without establishing it as the sole bottleneck in Video-LLMs. Our findings concern benchmarked motion, temporal order, and state-change capabilities under the evaluated 32-frame protocols, rather than general-purpose world modeling.

AI USE STATEMENT

We used generative AI tools for translation and formatting of the manuscript.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a Visual Language Model for Few-Shot Learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15619–15629, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning. *arXiv preprint arXiv:2506.09985*, 2025.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv preprint arXiv:2308.12966*, 2023.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception Encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced Projector for Multimodal LLM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13817–13827, 2024.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to GPT-4V? Closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67 (12):220101, 2024a.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024b.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

Jang Hyun Cho, Andrea Madotto, Effrosyni Mavroudi, Triantafyllos Afouras, Tushar Nagarajan, Muhammad Maaz, Yale Song, Tengyu Ma, Shuming Hu, Suyog Jain, et al. PerceptionLM: Open-Access Data and Models for Detailed Visual Understanding. *arXiv preprint arXiv:2504.13180*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024.

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, 2021. URL `https://arxiv.org/abs/2106.09685`.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B, 2023. URL `https://arxiv.org/abs/2310.06825`.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. MVBench: A Comprehensive Multi-modal Video Understanding Benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22195–22206, 2024a.

Lei Li, Yuanxin Liu, Linli Yao, Peiyuan Zhang, Chenxin An, Lean Wang, Xu Sun, Lingpeng Kong, and Qi Liu. Temporal Reasoning Transfer from Text to Video. *arXiv preprint arXiv:2410.06166*, 2024b.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. TempCompass: Do Video LLMs Really Understand Videos? In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8731–8772, 2024.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12585–12602, 2024.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A Diagnostic Benchmark for Very Long-form Video Language Understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

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.

Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adria Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, et al. Perception Test: A Diagnostic Benchmark for Multimodal Video Models. *Advances in Neural Information Processing Systems*, 36:42748–42761, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models From Natural Language Supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *European conference on computer vision*, pp. 510–526. Springer, 2016.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A Fully Open, Vision-Centric Exploration of Multimodal LLMs. *Advances in Neural Information Processing Systems*, 37: 87310–87356, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*, 2023.

Rishi Upadhyay, Howard Zhang, Zhirong Lu, Lakshman Sundaram, Ayush Agrawal, Yilin Wu, Yunhao Ba, Alex Wong, Celso M. de Melo, and Achuta Kadambi. How Close are World Models to the Physical World? WorldBench project and benchmark release, 2025. URL `https://world-bench.github.io/`.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. NExT-QA: Next Phase of Question-Answering to Explaining Temporal Actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9777–9786, 2021.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 Technical Report, 2024a. URL https://arxiv.org/abs/2407.10671.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 Technical Report, 2024b. URL https://arxiv.org/abs/2412.15115.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated Delta Networks: Improving Mamba2 with Delta Rule. *arXiv preprint arXiv:2412.06464*, 2024c.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. ActivityNet-QA: A Dataset for Understanding Complex Web Videos via Question Answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 9127–9134, 2019. doi: 10.1609/aaai.v33i01.33019127.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. VideoLLaMA 3: Frontier Multimodal Foundation Models for Image and Video Understanding. *arXiv preprint arXiv:2501.13106*, 2025.

Hang Zhang, Xin Li, and Lidong Bing. Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding. In *Proceedings of the 2023 conference on empirical methods in natural language processing: system demonstrations*, pp. 543–553, 2023.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video Instruction Tuning With Synthetic Data. *arXiv preprint arXiv:2410.02713*, 2024.

Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards Automatic Learning of Procedures from Web Instructional Videos, 2017. URL https://arxiv.org/abs/1703.09788.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. InternVL3: Exploring Advanced Training and Test-Time Recipes for Open-Source Multimodal Models. *arXiv preprint arXiv:2504.10479*, 2025.

A LIMITATIONS

Our findings concern post-hoc adaptation of pretrained Video-LLMs under the evaluated 32-frame protocols. They do not establish general-purpose world modeling or causal specialization of the two branches. The LoRA comparisons involve different optimization histories, and the projector-architecture comparisons do not isolate the effect of the GDN block alone. Representation probes and visualizations characterize feature geometry but do not independently establish the mechanism underlying the downstream gains.

B BROADER IMPACTS

Projector-JEPA aims to improve temporal and state-aware video understanding in multimodal models, which can benefit applications such as video search, educational assistance, accessibility tools, robotics, and long-form video analysis. At the same time, stronger video understanding systems may also amplify risks associated with surveillance, privacy invasion, and automated interpretation of human behavior. Since our method builds on pretrained VLMs and video instruction data, it may inherit biases, annotation artifacts, or failure modes present in the underlying models and datasets. We therefore view Projector-JEPA as a step toward more capable video-language systems, but not as a substitute for careful evaluation, transparency about training data, and safeguards when deployed in sensitive or human-facing settings.

C TRAINING SETUP

We freeze the vision encoder, base projector, and LLM throughout training and update only the expert and auxiliary heads. Optimization uses AdamW with weight decay 0.01, $\beta = (0.9, 0.999)$, and a learning-rate schedule that linearly warms up over the first 10% of steps before decaying with a cosine. Each video is sampled at $T = 32$ uniformly spaced frames. We train on a heterogeneous GPU setup consisting of two NVIDIA A100 GPUs with 40GB memory and eight NVIDIA RTX A5000 GPUs with 24GB memory. Depending on the backbone, we use an effective batch size of either 64 or 96. The source pool contains approximately 240K samples from nine video instruction and video QA datasets; projector and LoRA training use the same 80K-sample subset. It is centered on PE-Video and LLaVA-Video instruction tuning data, and is supplemented with datasets such as ActivityNet-QA, YouCook2, NEXT QA, Charades, and EgoSchema to broaden coverage over general video instruction following, everyday activity QA, procedural understanding, grounded QA, temporal reasoning, and long-video comprehension.

Table 5: Per-model training hyperparameters. D denotes the base-projector output (LLM input) width and B the bottleneck width. Training time is approximate.

Model	Base VLM	D	B	Expert LR	Steps	Time
PLM-1B	`facebook/Perception-LM-1B`	2048	128	2e-4	800	$\sim$12h
PLM-3B	`facebook/Perception-LM-3B`	3072	128	1e-3	800	$\sim$18h
PLM-8B	`facebook/Perception-LM-8B`	4096	128	5e-4	1200	$\sim$36h
InternVL3-2B	`OpenGVLab/InternVL3-2B`	1536	128	2e-4	800	$\sim$12h
InternVL3-8B	`OpenGVLab/InternVL3-8B`	3584	128	1e-4	1200	$\sim$36h
VideoLLaMA3-2B	`DAMO-NLP-SG/VideoLLaMA3-2B`	1536	128	3e-4	800	$\sim$12h
VideoLLaMA3-7B	`DAMO-NLP-SG/VideoLLaMA3-7B`	3584	128	1e-3	1200	$\sim$36h

Across backbones, the temporal kernel size is 3, retention-gate initial bias is 2.0, bottleneck width is $B = 128$, and pooling ratio is 1, preserving the base projector's spatial granularity. Auxiliary weights are fixed at $\lambda_s = 0.03$ and $\lambda_\ell = 0.01$, and gradients are clipped at norm 1.0. We use single-precision training with Liger patches for RoPE, RMSNorm, and SwiGLU on Llama- or Qwen-based LLMs.

Attachment point and dimensions. For all backbones, both branches take the frozen base-projector output and add their residuals in LLM-input space. InternVL3-2B and InternVL3-8B use $D = 1536$ and $D = 3584$, respectively; neither uses the 1024-dimensional vision-encoder output as its branch input. Figure 2 specifies this post-projector connection; Figure 1(c) provides a component-level overview.

Training masks and inference inputs. For each training video, we uniformly sample $\lfloor 0.5(T-2) \rfloor$ distinct middle-frame indices, leaving both endpoints visible. The auxiliary state pass uses the masked summaries

$$\widetilde{\mathbf{z}}_t = \begin{cases} \mathbf{0}, & t \in \mathcal{M}, \\ \bar{\mathbf{z}}_{b,t}, & t \notin \mathcal{M}. \end{cases}$$

We apply this mask before constructing displacement features, normalization, down-projection, and temporal convolution. The entire student input is rebuilt from the masked sequence:

$$[\, \widetilde{\mathbf{z}}_t \,\|\, \widetilde{\mathbf{z}}_t - \widetilde{\mathbf{z}}_{t-1} \,\|\, \widetilde{\mathbf{z}}_t - \widetilde{\mathbf{z}}_0 \,], \qquad \widetilde{\mathbf{z}}_{-1} = \widetilde{\mathbf{z}}_0.$$

Thus the input at $m+1$ does not reuse an unmasked difference containing the hidden summary at m. The masked pass shares state-expert weights with the QA path. Its endpoint outputs $\mathbf{o}^{\mathcal{M}}_{s,0}$ and $\mathbf{o}^{\mathcal{M}}_{s,T-1}$ feed the reconstruction head, while the unmasked frozen summaries provide targets. The QA and local-correspondence paths use the unmasked sequence. At inference, the auxiliary masked pass and reconstruction head are disabled; the state branch uses the full frame summaries and their one-step and first-frame displacement features.

Boundary and loss conventions. We index frames from 0 to $T-1$ and set the first one-step difference to zero in both branches. Recurrent states are initialized to zero. The kernel-3 temporal convolutions use two zero-padded positions on the left and no right padding, preserving sequence length without accessing future inputs. The contrastive temperature is $\tau = 0.07$, distinct from the decoding temperature of zero. The local loss sums over valid pairs $t = 1, \ldots, T-1$ and spatial positions, without dividing by $(T-1)S$. The weight $\lambda_\ell = 0.01$ multiplies this sum, so its effective contribution depends on the number of valid pairs. The first frame has no correspondence-loss term. The state loss averages over masked frames.

C.1 Checkpoint Selection and Evaluation Protocol

Checkpoint selection. We reserve 4K validation examples from the 240K candidate pool, separately from the 80K training subset. Sampling is stratified by source dataset, and examples from the same source video are assigned to a single partition. The validation set is fixed across models and training seeds. Among checkpoints saved every 200 steps, we select the checkpoint with the lowest validation QA cross-entropy, averaged over answer tokens; ties select the earlier checkpoint. Auxiliary losses are not included in the selection metric. The same rule is used for projector training and LoRA adaptation. We report the mean performance of the selected checkpoints from three independently seeded runs.

Controlled evaluation. We uniformly sample 32 frames per video and use greedy decoding for all evaluated models. We do not tune prompts, frame counts, sampling strategies, or decoding parameters separately for individual benchmark results. Each Base model and its adapted variants are evaluated under the same protocol to measure their relative improvements.

Evaluation subsets. We use the released 20-task MVBench evaluation annotations, the TempCompass multiple-choice question set, the NEXT QA multiple-choice test split, and WorldBench's released text-question subset (the four category files under `textual_questions`). All three model families use the Perception Test multiple-choice validation split, `perceptiontest_val_mc`. This is distinct from official Perception Test test-server evaluation. Every Base and adapted variant uses the same questions and scoring convention within each benchmark.

Training/evaluation separation. Benchmark evaluation questions and video IDs are excluded when constructing the training and checkpoint-validation partitions. NEXT QA training examples are drawn from its training split; its test split is used only for final evaluation. Checkpoint selection uses the held-out instruction/QA validation set above, not any reported benchmark score.

D LoRA Training Setup

For LoRA training, we attach low-rank adapters (LoRA; Hu et al., 2021) to the language model of each backbone while keeping the vision encoder, base projector, and the non-adapter LLM parameters frozen. For LoRA-Projector-JEPA, the expert parameters are initialized from the trained expert checkpoint and continue to be updated with their own learning rates. LoRA-base adapts the LLM with the frozen base projector. The two settings therefore differ in initialization and optimization history, even though they reuse the same 80K-sample subset. Only the LoRA-specific hyperparameters are reported here. We use the standard PEFT `LoraConfig` with $\alpha = 2r$, dropout 0.05, no bias terms, and `task_type=CAUSAL_LM`. Adapters are inserted only into the attention Q and V projections of every transformer layer in the LLM.

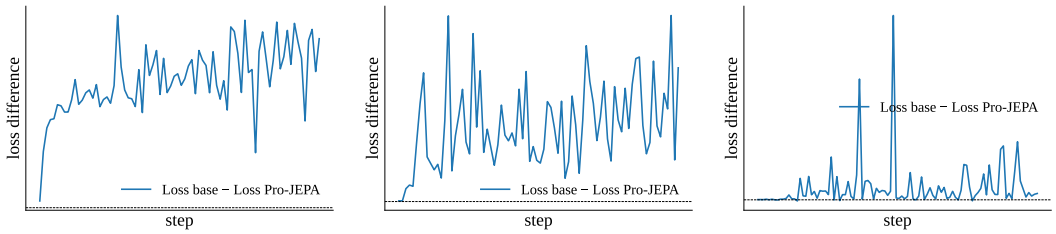
Table 6: LoRA training hyperparameters across backbones. Batch size denotes the effective batch size. We set $\alpha = 2r$ and dropout $= 0.05$ for all configurations, and adapters target $\{q_proj, v_proj\}$ in every LLM layer.

Model	Steps	Batch size	Rank r	LoRA LR
PLM-1B	800	96	16	1e-4
PLM-3B	800	96	16	5e-5
PLM-8B	1200	64	32	5e-5
InternVL3-2B	800	96	16	1e-4
InternVL3-8B	1200	64	32	2e-4
VideoLLaMA3-2B	800	96	16	1e-4
VideoLLaMA3-7B	1200	64	32	5e-5

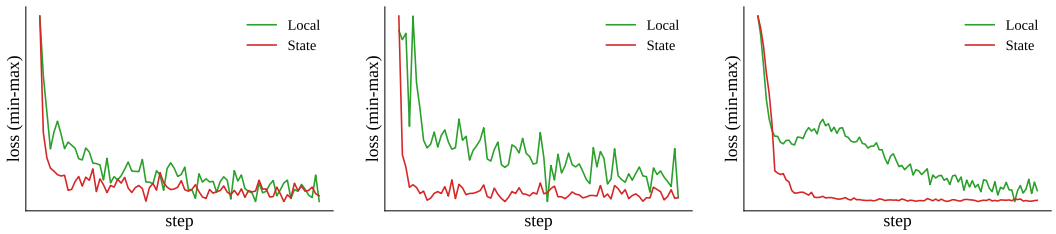
All backbones are trained with $T = 32$ frames per video, gradient clipping at norm 1.0, AdamW with weight decay 0.01 and $\beta = (0.9, 0.999)$, and a cosine learning-rate schedule with 50 warmup steps. The image resolution follows each backbone's native projector: 448 for PLM and InternVL3, and 384 for VideoLLaMA3. PLM-8B and VideoLLaMA3-7B use $r = 32$, $\alpha = 64$, and lr $= 5 \times 10^{-5}$. PLM-1B, PLM-3B, InternVL3-2B, and VideoLLaMA3-2B use $r = 16$ and $\alpha = 32$, with the model-specific learning rates in Table 6. InternVL3-8B uses $r = 32$, $\alpha = 64$, and lr $= 2 \times 10^{-4}$.

E TRAINING CURVES

Figures 4a, 4b, and 4c plot the QA cross-entropy difference $\mathcal{L}_{\mathrm{QA}}(\mathbf{z}_b) - \mathcal{L}_{\mathrm{QA}}(\mathbf{z}_{\mathrm{final}})$ during training. Positive values indicate lower QA cross-entropy with Projector-JEPA on the evaluated training batches. This is a prediction-loss difference, not the total objective including auxiliary losses or a measure of added information. Figures 4d, 4e, and 4f separately show the Local and State auxiliary losses. These curves describe optimization behavior; downstream performance is evaluated separately.



(a) PLM-8B QA cross-entropy difference (b) VL3-7B QA cross-entropy difference (c) IV3-8B QA cross-entropy difference



(d) PLM-8B auxiliary loss (e) VL3-7B auxiliary loss (f) IV3-8B auxiliary loss

Figure 4: Training curves for PLM-8B, VL3-7B, and IV3-8B. Panels (a)–(c) show the QA cross-entropy difference (Base minus Projector-JEPA); panels (d)–(f) show the Local and State auxiliary losses.

F PROJECTOR-JEPA ARCHITECTURE SELECTION

Table 7: Operator-level properties and core parameter counts. This design comparison is distinct from the complete projector alternatives evaluated in Section 6.3 and Appendix H.3.

Property	MLP	Conv1D (dense)	Cross-Attn	**Gated DeltaNet**
Params (mixer only)	131 K	49 K	65 K	82 K
Cross-frame interaction	none	local-k	full	recurrent history
Long-range memory	✗	✗	✓	✓ (selective)
Non-linear cross-frame coupling	✗	weak	✓	✓
Input-dependent temporal mixing	✗	✗	attention	state gates
Cost scaling in T	$\mathcal{O}(T)$	$\mathcal{O}(T)$	$\mathcal{O}(T^2)$	$\mathcal{O}(T)$

Table 7 reports core-operator properties and parameter counts, separately from the complete projector comparisons. Parameter counts exclude the down- and up-projections and auxiliary heads; they are not the total trainable parameter counts of Projector-JEPA. The Conv1D comparison uses the dense-kernel counting convention: $B^2k = 128^2 \times 3 = 49{,}152$ weights (approximately 49K). This differs from the shared depthwise $\mathrm{TempConv}_3$ preprocessing layer in Section 3.1, whose kernel has $Bk = 384$ weights, excluding bias.

Table 7 describes operator-level properties of MLP, dense temporal Conv1D, cross-attention, and Gated DeltaNet. A frame-wise MLP has no cross-frame interaction, Conv1D uses a fixed local kernel, cross-attention permits input-dependent interactions across frames with $\mathcal{O}(T^2)$ time scaling, and GDN carries recurrent history with linear scaling. These operator properties motivate the design; they do not define the experimental controls as interchangeable blocks in an otherwise identical architecture.

All three residual-projector configurations retain the additive layout $\mathbf{z}_{\text{final}} = \mathbf{z}_b + \mathbf{r}_\ell + \mathbf{r}_s$ and the frozen Base projector. The MLP configuration replaces each complete Local/State branch with a newly initialized two-layer MLP having the same architecture as the Base projector. The cross-attention configuration analogously uses cross-attention residual branches. Thus the comparisons change the whole branch architecture, rather than only the GDN cell in an otherwise fixed temporal-preprocessing path. Table 4b compares the State residual $\mathbf{r}_s$ in all three cases, before addition to Base. Fine-tuning the pretrained Base MLP, without the added residual branches, is a separate control.

Gated DeltaNet uses the scalar retention gate α_t to decay memory and the delta correction $\beta_t(\mathbf{v}_t - \alpha_t \mathbf{S}_{t-1}\mathbf{k}_t)\mathbf{k}_t^\top$ to update the association for the current key. This combines global memory decay with a key-specific update in linear time; it is not equivalent to full softmax attention. In our two branches, independent token trajectories preserve spatial detail, while a spatially pooled trajectory aggregates frame-level context. Both can use recurrent history. The downstream controls in Appendix H.3 support the complete proposed architecture over the tested MLP-projector alternatives; they do not isolate a causal effect of the GDN operator.

G Task-Level and Qualitative Results

G.1 Performance by Task Group

Table 8 organizes the evaluated subtasks into non-exclusive diagnostic groups and reports group-level gains over Base. For example, `moving_count` appears under both Motion / Direction and Counting. The listed subtasks illustrate each category; the reported group summaries should not be read as a recomputation over only those displayed examples or as a disjoint partition of benchmark questions. The largest gains are in State Change / Memory, followed by Motion / Direction, across the seven backbones. For each diagnostic group, we compute the unweighted mean of its constituent subtasks' accuracy gains over Base, after averaging each subtask score across the three seeds. A subtask assigned to more than one group contributes separately to each group's mean; group means are not pooled into an overall score.

Table 8: Performance differences from Base for non-exclusive diagnostic groups. All values are absolute percentage-point gains over the base setting (Δ vs. Base, %p). For each model column, the largest gain is shown in bold, and the second-largest gain is underlined. Representative subtasks are illustrative, not a complete membership list; groups overlap and their gains must not be pooled into an overall score. WB, TC, MVB, NQ, and PT denote WorldBench, TempCompass, MVBench, NEXT QA, and Perception Test, respectively.

Diagnostic group	PLM-1B	PLM-3B	PLM-8B	VL3-2B	VL3-7B	IV3-2B	IV3-8B
Motion / Direction	<u>+3.1</u>	<u>+3.5</u>	<u>+2.9</u>	<u>+2.4</u>	<u>+2.1</u>	<u>+3.4</u>	<u>+2.7</u>
Temporal Sequence / Order	+2.4	+2.3	+2.7	+2.1	+2.0	+2.5	+2.1
Counting	+1.2	+1.3	+0.5	+0.3	+0.4	+1.1	+0.7
Spatial / Object Existence	+0.5	+0.7	+0.2	+1.1	+0.7	+0.6	+0.5
State Change / Memory	**+3.8**	**+3.6**	**+3.5**	**+4.2**	**+4.9**	**+3.9**	**+3.6**
Causal Reasoning	+1.2	+0.9	+0.3	+1.8	+1.2	+1.3	+1.2
Fine-grained / Egocentric	+0.6	+0.6	+1.4	+1.6	+1.3	+1.4	+1.6

Diagnostic group	Representative subtasks
Motion / Direction	WB: motion_physics; TC: direction, speed; MVB: moving_direction, moving_count, etc.; PT: physics
Temporal Sequence / Order	TC: action, order; MVB: action_sequence, action_prediction, etc.; NQ: TN, TC, TP
Counting	MVB: action_count, moving_count; NQ: DC
Spatial / Object Existence	WB: obj_perm, scale_perspective, etc.; MVB: object_existence, object_interaction; NQ: DL, DO; PT: semantics
State Change / Memory	MVB: state_change, unexpected_action, etc.; PT: memory
Causal Reasoning	MVB: counterfactual_inference; TC: attribute_change; NQ: CW, CH
Fine-grained / Egocentric	MVB: fine_grained_action, fine_grained_pose; MVB: object_shuffle, egocentric_navigation

G.2 Qualitative Examples

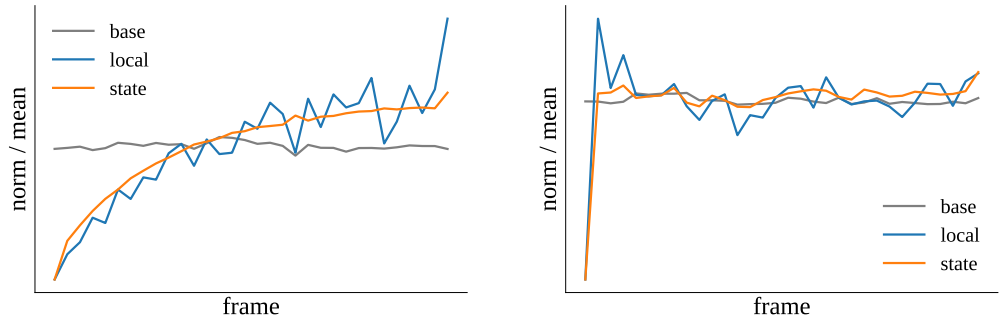
Table 9 presents MVBench cases where Projector-JEPA corrects an error made by the vanilla PLM-1B model. The groups organize examples by task characteristics, without assigning individual corrections to a branch.

Table 9: Representative qualitative examples on MVBench. We compare the vanilla PLM-1B backbone with Projector-JEPA on cases where the base model fails but the proposed projector produces the correct answer. Examples are grouped by task characteristics; these full-model predictions do not attribute individual corrections to a specific branch.

Subtask	**Question**	**GT**	**Vanilla PLM-1B**	**+ Projector-JEPA**
Short-horizon motion and event-ordering examples				
action antonym	What activity does the video depict? (A) Twisting (B) Untwisting (C) Not sure	A	(B) Untwisting ✗	(A) Twisting ✓
moving direction	What direction is the green cylinder moving in the video? (B) Up & right; (C) Down & right; (D) Up & left	C	(D) Up and to the left ✗	(C) Down and to the right ✓
moving attribute	What is the material of the *last* object that enters the scene? (A) rubber (B) metal	B	(A) rubber ✗	(B) metal ✓
action sequence	What happened *before* the person held the food? (A) Put down clothes; (B) Opened the laptop; (C) Put down food; (D) Took the dish	B	(C) Put down the food ✗	(B) Opened the laptop ✓
Persistent-state and long-range-context examples				
state change	If the person tried to turn on the device and it was not working, what was the reason?	A	(B) The device worked ✗	(A) Already *unplugged at the beginning* of the video ✓
scene transition	Which choice matches the scene changes?	A	(D) Street → Office ✗	(A) Alley → Room ✓
counter-factual	What will happen if the sphere is removed?	A	(D) Yellow ↔ blue cube ✗	(A) Blue ↔ brown cube ✓
ego. navigation	"Walk straight, turn left between couch and counter, enter next room, stop at first white couch." Next action?	B	(C) Turn right + forward ✗	(B) Turn left + forward ✓

G.3 Frame-wise Activation Norms

Figure 5 complements the main-text representation visualization (Figure 3) by comparing activation magnitudes across frames. Both panels retain $\mathbf{z}_b + \mathbf{r}_\ell + \mathbf{r}_s$: `base` denotes the frozen base-projector output, and `local`/`state` denote the corresponding learned residuals before addition. Panel (a) uses the proposed GDN branches; panel (b) replaces each full branch with an MLP having the Base projector architecture. Each curve is divided by its own temporal mean. In this PLM-3B example, the GDN residual norms vary more across frames than the MLP residual norms. This describes activation geometry, while Appendix H.3 reports downstream performance.



(a) GDN residual branches

(b) MLP residual branches

Figure 5: Frame-wise norms on PLM-3B with GDN branches (a) and MLP branches (b). In both panels, `base` is the frozen Base output; `local` and `state` are residual-branch outputs before addition. Panel (b) retains the two-branch additive layout and replaces each complete branch with a Base-architecture MLP. Each curve is normalized by its temporal mean.

H Additional Controlled Experiments

We provide protocol details for the controlled ablations in Section 6.1 and additional downstream and representation analyses. MVBench and TempCompass (TC) scores are reported in percent, with higher values indicating better performance. These experiments complement the main benchmark results and the representation analyses in Section 6.

H.1 Branch and auxiliary-loss ablations

Table 2 evaluates the same ablations on PLM-1B, VideoLLaMA3-2B, and InternVL3-2B: one or two QA-only branches, each auxiliary loss disabled with both branches retained, and each branch alone with its corresponding auxiliary objective. All trainable variants use QA supervision. The weights λ_ℓ and λ_s control the local and state losses; a dash in a weight column indicates an absent branch.

Both single-branch variants improve over Base in all six settings. Full exceeds Local-only throughout and State-only in five settings, tying it on VideoLLaMA3-2B MVBench (63.22). Removing the local loss reduces MVBench accuracy by 0.20–0.93 points and TC by 0.78–1.01; removing the state loss reduces MVBench accuracy by 0.09–1.16 points and TC by 0.20–0.87 points. Both auxiliary losses therefore improve both benchmarks for all three backbones when added to the otherwise complete configuration. Full also exceeds both QA-only configurations in every setting. These comparisons support each branch and auxiliary objective as contributors to the combined model's performance, with Full achieving the highest or joint-highest score throughout.

H.2 Controls across model families

We repeat two controls on PLM-1B, VideoLLaMA3-2B, and InternVL3-2B using the same training data and an 800-step training schedule as the corresponding full configuration. The architecture-only control retains both residual branches and disables both auxiliary objectives ($\lambda_\ell = \lambda_s = 0$). The Base-MLP control fine-tunes the pretrained frame-wise projector using the QA loss, without adding the proposed branches. The vision encoder and LLM remain frozen in these controls.

Table 3 covers six model–benchmark settings, yielding twelve pairwise comparisons against the two controls; Full wins all twelve. The controls can improve individual scores, but neither matches Full. These matched projector comparisons support the combined design. The LoRA results in Table 1 use a different adaptation protocol and optimization history.

H.3 Downstream comparison of projector architectures

Table 10 evaluates projector alternatives on downstream tasks, complementing the representation statistics in Table 4b. The two MLP controls differ in initialization and placement. *New MLP branches* keeps the pretrained Base projector frozen and trains newly initialized Local and State residual branches, each using the Base projector's two-layer MLP architecture. Their outputs are added to Base. *Fine-tuned Base MLP* instead updates the pretrained projector without adding branches. The cross-attention configuration in Table 4b retains the same two-residual layout and uses cross-attention branches. These are complete-branch architecture comparisons, not GDN-cell-only substitutions.

Table 10: Downstream projector controls on PLM-1B. New MLP branches are initialized separately and added to Base; Base-MLP fine-tuning updates the pretrained projector without residual branches. Full uses the proposed GDN branches.

Configuration	Trainable module	MVBench	TC
Base	None	66.48	61.65
New MLP branches	New Local/State MLPs	65.08	61.27
Fine-tuned Base MLP	Pretrained Base projector	65.58	60.89
Full GDN projector	Recurrent residual branches	**68.11**	**63.03**

Full outperforms both MLP controls, which remain below Base on both benchmarks. This supports the complete residual design rather than isolating the GDN cell.

H.4 Linear and Nonlinear Reconstruction of Expert Representations

Linear reconstruction probe. We use the following protocol for the representation diagnostics in Table 4a. Beyond cosine similarity, we further evaluate how well the outputs of the Local and State experts can be reconstructed from the base projector output. To this end, we formulate a linear regression problem that predicts the target branch representation $\mathbf{Y}_b$ from the base projector representation $\mathbf{Z}_{\text{base}}$, where $b \in \{\text{local}, \text{state}\}$. The goal is to measure whether each branch can be explained by a simple linear transformation of the base representation:

$$\hat{\mathbf{Y}}_b = \mathbf{Z}_{\text{base}}\mathbf{W} + \mathbf{c}.$$

Here, $\mathbf{c}$ is broadcast over samples. We fit $\mathbf{W}$ and $\mathbf{c}$ using ridge regression with the following objective:

$$\mathcal{L}(\mathbf{W}, \mathbf{c}) = \|\mathbf{Y}_b - \mathbf{Z}_{\text{base}}\mathbf{W} - \mathbf{c}\|_F^2 + \alpha \|\mathbf{W}\|_F^2 ,$$

where $\alpha = 1.0$. We then evaluate the fitted linear predictor on the test split using the coefficient of determination, R^2:

$$R^2 = 1 - \frac{\left\|\mathbf{Y}_b^{\text{te}} - \hat{\mathbf{Y}}_b^{\text{te}}\right\|_F^2}{\left\|\mathbf{Y}_b^{\text{te}} - \bar{\mathbf{Y}}_b^{\text{te}}\right\|_F^2}.$$

A high R^2 indicates good reconstruction under the fitted ridge-regression probe. A low R^2 indicates that the branch representation is not well reconstructed from the base output using the fitted ridge-regression probe; it does not rule out all linear mappings.

Nonlinear reconstruction probe. Linear reconstruction alone cannot exclude a nonlinear transformation of the base representation. We therefore compare Ridge regression with an MLP regression probe, reconstructing each expert output from the corresponding same-frame Base token. Table 11 reports per-token R^2 on the top 64 principal components. This is a separate probe analysis from the representation statistics in Table 4a.

Table 11: Linear and nonlinear token-wise reconstruction probes on the top 64 PCA components. The last column is the variance unexplained by the tested nonlinear probe.

Branch	Linear R^2	Nonlinear R^2	$100(1 - R^2_{\text{nonlinear}})$
Local	0.297	0.331	66.9%
State	0.423	0.553	44.7%

The MLP increases R^2 from 0.297 to 0.331 for Local and from 0.423 to 0.553 for State, leaving 66.9% and 44.7% of the respective target variance unexplained. The expert outputs are therefore not adequately reconstructed by either the tested linear or nonlinear token-wise mappings. The probe observes a same-frame Base token, whereas the experts process temporal context, so this input-context difference is also compatible with low reconstruction scores. These results neither rule out a more expressive or sequence-level probe nor establish information gain over the complete video-level Base sequence.

H.5 Accuracy Gains from Single and Combined Branches

Table 12 summarizes the MVBench accuracy gains of the single-branch and full PLM-1B configurations. Accuracy is the three-seed mean reported in Table 2; gains are absolute percentage-point differences from Base.

Table 12: MVBench accuracy and gains over Base for PLM-1B. Gain is the variant's mean accuracy minus Base's 66.48%.

Variant	MVBench (%)	Gain vs. Base (pp)
Base	66.48	–
Local-only	67.15	$+0.67$
State-only	67.98	$+1.50$
Full	**68.11**	$\mathbf{+1.63}$

Local-only and State-only improve over Base by 0.67 and 1.50 points, respectively. Full improves by 1.63 points, exceeding Local-only by 0.96 points and State-only by 0.13 points. Both branches are useful individually, while their joint configuration achieves the highest MVBench accuracy. The corresponding loss ablations in Table 2 further support the contribution of both auxiliary objectives.