CARVE: LEARNING LATENT ACTIONS UNDER CROSS-VIEW VISUAL HETEROGENEITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent action models learn action-centric dynamics from unlabeled robot videos, but existing methods predominantly operate on a single camera stream. Recent multi-view formulations instead emphasize view invariance, overlooking that synchronized cameras provide complementary but not interchangeable evidence about the same physical action. We introduce CARVE (Cross-view Action Representation with View-specific Evidence), a latent action framework that selectively shares transition information across synchronized views. For each target view, CARVE forms global components from synchronized multi-view context and retains target-view private components. A shared future-dynamics model routes global information across target views while restricting private information to its corresponding predictor. Because synchronized multi-view videos are substantially less abundant than front-view recordings, CARVE uses a progressive three-stage procedure that combines front-view pretraining, amplitude-guided multi-view adaptation, and global-private decomposition. CARVE consistently improves long-horizon video prediction and downstream robot control over strong latent action baselines. Representation analyses and ablations further show that selective cross-view sharing captures complementary action information while preserving target-specific transition cues.

1 INTRODUCTION

Robot videos provide abundant supervision for learning how the visual world changes, yet they are often collected without action annotations. Latent action models (Bruce et al., 2024; Schmidt & Jiang, 2024) address this challenge by inferring latent transition representations from observation pairs of adjacent frames and using them to model future dynamics. Some methods further use such representations for robot policy pretraining (Ye et al., 2025; Bu et al., 2025) and predictive world modeling (Gao et al., 2025b), enabling large-scale video data to support downstream robot learning or physical modeling without dense action labels.

Most latent action models, however, operate on a single camera stream at a time, potentially missing complementary action cues from different viewpoints. This limits their applicability to real-world embodied robotic systems, which commonly rely on multi-camera observations. Recent multi-view extensions exploit synchronized observations and typically encourage latent actions to remain invariant or consistent across viewpoints (Lee et al., 2026; Wan et al., 2026). While such consistency can reduce view-specific nuisance factors, synchronized views are complementary rather than interchangeable, as differences in visibility and projection can cause the same physical transition to reveal different action cues across viewpoints. Only enforcing cross-view consistency over the latent representations may therefore suppress informative view-specific evidence. Moreover, multi-view latent action learning faces two training challenges: (1) synchronized multi-view videos are substantially less abundant than single-view front-camera recordings, and (2) different camera streams typically exhibit different observed motion distributions, making a uniform temporal sampling strategy across views unsuitable for the balanced training. Together, these challenges call for a new framework that selectively shares transition information across views while progressively adapting from large-scale front-view training to multi-view learning in a balanced manner.

Based on the above discussions, we propose CARVE, a novel multi-view latent action framework that addresses both representation and adaptation challenges under cross-view visual heterogeneity.

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.

CARVE decomposes synchronized transitions into cross-view reusable global components and view-specific private components, and asymmetrically routes all global components together with only the matched private component to each target-view predictor. This design preserves complementary cross-view evidence without enforcing invariance over the entire transition representation. To address the limited availability and heterogeneous motion distributions of synchronized multi-view data, CARVE adopts a progressive training strategy that transfers action-centric knowledge learned from abundant front-view videos to multi-view observations. During adaptation, we further introduce an amplitude-matched sampling strategy that aligns transition scales across camera streams, thus enabling more balanced learning across different views. Together, these designs allow CARVE to effectively exploit complementary multi-view evidence while accommodating cross-view heterogeneity.

To validate CARVE, we conduct extensive experiments across multiple tasks, including multi-view video prediction, action predictability, robot policy learning, and visual planning. CARVE consistently improves long-horizon prediction and downstream robot control over strong latent action baselines such as MVP-LAM and DiLA. Our analysis further shows that combining synchronized global components yields more informative action representations than relying on a single view. Collectively, these results demonstrate the effectiveness of CARVE by exploiting complementary cross-view information, preserving view-specific transition cues, and achieving balanced multi-view adaptation.

Our main contributions are summarized as follows:

- We propose a global-private latent action structure that decomposes synchronized multi-view transitions into cross-view reusable global components and view-specific private components, together with asymmetric routing that selectively shares complementary information across views.
- We develop a progressive training strategy that transfers latent action representations learned from abundant front-view videos to limited and heterogeneous multi-view data via amplitude-guided cross-view adaptation, significantly enhancing the effectiveness of multi-view latent action learning.
- By integrating the proposed global-private structure with progressive multi-view training, we introduce CARVE, a novel view-aware latent action framework that consistently improves the performance of several tasks such as long-horizon prediction, action representation, and downstream robot control over strong latent action baselines.

2 Related Work

Latent action models. World models predict environment transitions conditioned on action labels (Ha & Schmidhuber, 2018; Guo et al., 2024; Zhu et al., 2025). Recent controllable generative world models extend this interface to multi-view, action-conditioned rollouts (Guo et al., 2026; Barcellona et al., 2025; Gao et al., 2025a). Latent action models instead infer latent actions directly from videos (Chen et al., 2025; Wang, 2026; Jeong et al., 2026; Yoshimoto et al., 2026; Bu et al., 2026). Early formulations learn discrete action codes for controllable video models (Bruce et al., 2024; Schmidt & Jiang, 2024); LAPA and UniVLA transfer video-derived codes to robot policies (Ye et al., 2025; Bu et al., 2025), while AdaWorld develops continuous latent actions for adaptable world modeling (Gao et al., 2025b). LAOM further shows that visual distractors can corrupt unsupervised latent actions (Nikulin et al., 2025). These approaches predominantly learn from one observation stream at a time. Our focus is the additional structure required when synchronized cameras provide different evidence about the same physical transition.

Multi-view representation and latent action learning. Multi-view learning commonly extracts information that is stable across synchronized observations. Time-Contrastive Networks align demonstrations in time (Sermanet et al., 2018), Contrastive Multi-view Coding maximizes shared information across views (Tian et al., 2020), and multi-view world models use masked prediction or contrastive objectives to improve robotic state representations and control (Seo et al., 2023; Kinose et al., 2023; Pang et al., 2025). Recent latent action methods bring this principle to transition representations. MVP-LAM uses cross-viewpoint reconstruction to discourage view-specific shortcuts (Lee et al., 2026), whereas MuCoLA aligns synchronized latent action distributions through cross-view

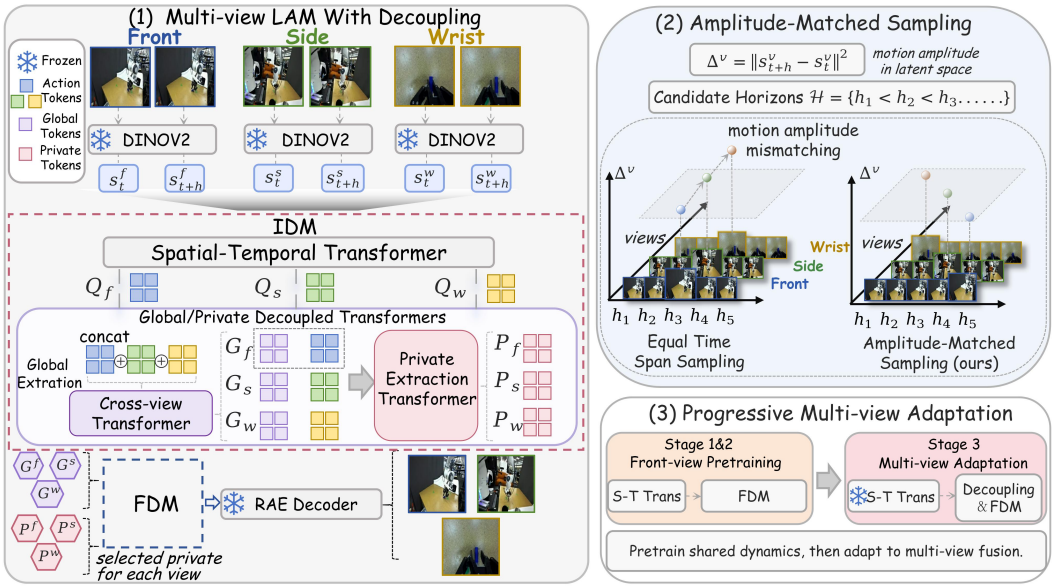


Figure 1: Overview of CARVE. (1) **Multi-view latent action decomposition** separates synchronized transitions into cross-view global and view-specific private components for selective fusion. (2) **Amplitude-matched sampling** selects temporal horizons according to motion amplitude to reduce cross-view amplitude scale mismatch. (3) **Progressive multi-view adaptation** transfers a single-view latent action backbone to synchronized multi-view decomposition and fusion.

self-distillation (Wan et al., 2026). Both motivate view-invariant or view-consistent action semantics. CARVE addresses a complementary limitation: synchronized views are not interchangeable, so enforcing consistency on the entire latent action representation can suppress evidence brought by different camera streams.

Shared–private and disentangled representations. Separating common and domain-specific factors is a longstanding strategy for preventing nuisance variation from contaminating shared representations (Bousmalis et al., 2016). Multi-view and temporal constraints have likewise been used to isolate geometry from appearance (Rhodin et al., 2018). Within latent action world models, DiLA separates motion-related structure from visual content to reconcile action abstraction with high-fidelity generation (Zhang et al., 2026), while FLAM factorizes the dynamics of independently acting scene entities (Wang et al., 2026). CARVE pursues a different decomposition axis by partitioning transition evidence by its predictive availability: global components may be routed across target views, whereas each private component is restricted to its matched target.

3 Problem Setup

We assume access to an abundant front-view video collection $\mathcal{D}_{\text{sv}}$ and a smaller synchronized multi-view collection $\mathcal{D}_{\text{mv}}$. A synchronized multi-view sequence in $\mathcal{D}_{\text{mv}}$ is denoted as:

$$X_{1:T} = \left\{ \{\mathbf{x}_t^v\}_{t=1}^{T} \right\}_{v=1}^{V}, \tag{1}$$

where T is the sequence length, V is the number of camera views, and $\mathbf{x}_t^v$ denotes the RGB observation from view v at time t. For a temporal horizon $h \in \mathcal{H}$, the pair $\tau_{t,h}^v = (\mathbf{x}_t^v, \mathbf{x}_{t+h}^v)$ describes the transition observed from view v. The synchronized set $\mathcal{T}_{t,h} = \{\tau_{t,h}^v\}_{v=1}^{V}$ contains observations of the same physical transition from all views. Given $\mathcal{T}_{t,h}$, our goal is to learn an action-centric representation of the underlying transition from heterogeneous synchronized observations.

4 METHOD

Existing latent action models primarily learn transition representations from a single camera stream. Recent multi-view extensions seek to leverage synchronized observations, but typically overemphasize cross-view consistency and train directly on limited and heterogeneous multi-view data, making it challenging to preserve view-specific cues and achieve effective training. To address these issues, we propose CARVE, a multi-view latent action framework comprising a global-private action architecture and a progressive three-stage training procedure.

The overall architecture of CARVE is illustrated in Figure 1. Given the current and future visual states, we employ a frozen DINOv2 encoder (Oquab et al., 2023; Darcet et al., 2024) to extract visual features, which are then fed into the inverse dynamics model (IDM). The IDM consists of a spatio-temporal transformer followed by the proposed global-private decoupling transformer. The spatio-temporal transformer first extracts intermediate action tokens that summarize the observed transition. These tokens are then processed by the global-private decoupling transformer to produce the corresponding global and private latent action components. The forward dynamics model (FDM) predicts the future latent state conditioned on the current state and the inferred global-private latent actions. An RAE decoder (Zheng et al., 2026) further reconstructs the predicted latent state into an RGB image for visualization. The architectures of the spatio-temporal transformer, FDM, and RAE decoder follow or slightly extend standard designs from prior work and are detailed in Appendix B.2. In the following Sections 4.1 and 4.2, we introduce our two main contributions: the global-private decoupling transformer and the progressive training strategy, respectively.

4.1 GLOBAL-PRIVATE DECOUPLING ARCHITECTURE

Synchronized cameras observe the same underlying physical transition, yet each view provides distinct evidence due to differences in projection, visibility, and occlusion. A fully shared latent action across views may suppress informative view-specific cues, whereas independent per-view latent actions fail to exploit complementary cross-view shared evidence. To balance cross-view sharing and view-specific preservation, CARVE introduces a novel global-private decoupling structure.

For each view v, the spatio-temporal transformer produces K action tokens $Q_v \in \mathbb{R}^{K\times D}$, where D denotes the token dimension. A global transformer T_G jointly contextualizes the resulting VK synchronized action tokens from V views and produces K view-indexed global tokens for each view. A private transformer T_P then derives the corresponding private components by removing the global information from Q_v. These processes are formulated as:

$$\begin{aligned} [G_1;\ldots;G_V] &= T_G\left([Q_1;\ldots;Q_V]\right), \qquad G_v \in \mathbb{R}^{K\times D}, \\ P_v &= T_P\left([Q_v;\mathrm{sg}(G_v)]\right), \qquad P_v \in \mathbb{R}^{K\times D}. \end{aligned} \tag{2}$$

Here, G_v and P_v represent global and private components derived from view v, respectively. Details of T_G and T_P are presented in Appendix B.2. The stop-gradient operator sg prevents the private branch from backpropagating gradients into the corresponding global representation. Let $\mathbf{G} = \{G_j\}_{j=1}^{V}$ denote the full set of global components. For each target view v, CARVE then applies an asymmetric routing strategy: for reconstruction, the future-dynamics model (FDM) receives all global components across views and the private components matched to the target view:

$$\widehat{s}^{\,v}_{t+h} = s^v_t + F\left(s^v_t, \mathbf{G}, P_v, h\right). \tag{3}$$

Here, $\hat{s}^{\,v}_{t+h}$ denotes the predicted future latent state of target view v at time $t+h$ and F refers to the FDM. This routing allows reusable transition evidence to propagate across views while restricting private components to only contain information from their corresponding target view.

Functional decomposition. An architectural split alone does not guarantee that the global and private components learn complementary functions. We therefore introduce three reconstruction objectives to encourage their intended roles. Specifically, we employ a joint action-space decoder D to reconstruct the target transition tokens from $\mathbf{G}$ together with the matched private component P_v for each view v, and a global-only decoder D_G to reconstruct the same targets using $\mathbf{G}$ alone. Both decoders are implemented as four-layer transformers. We further construct mismatched inputs by keeping $\mathbf{G}$ fixed while replacing P_v with the private component P_u from another synchronized view,

where $u \neq v$. The three reconstruction errors are formulated as follows:

$$\mathcal{E}^v_{\text{joint}} = \|D(\mathbf{G}, P_v) - Q_v\|_2^2, \quad \mathcal{E}^v_{\text{global}} = \|D_G(\mathbf{G}) - Q_v\|_2^2, \quad \mathcal{E}^{v\leftarrow u}_{\text{cross}} = \|D(\mathbf{G}, P_u) - Q_v\|_2^2,\ u \neq v. \tag{4}$$

Here, $\mathcal{E}^v_{\text{joint}}$, $\mathcal{E}^v_{\text{global}}$, and $\mathcal{E}^{v\leftarrow u}_{\text{cross}}$ denote the reconstruction errors using $\mathbf{G}$ with the matched private component, $\mathbf{G}$ alone, and $\mathbf{G}$ with a mismatched private component, respectively. Based on these reconstruction errors, we define three objectives:

$$\begin{aligned} \mathcal{L}_{\text{rec}} &= \frac{1}{V}\sum_{v=1}^{V}\left(\mathcal{E}^v_{\text{joint}} + \mathcal{E}^v_{\text{global}}\right), \quad \mathcal{L}_{\text{gain}} = \frac{1}{V}\sum_{v=1}^{V}\left[\mathcal{E}^v_{\text{joint}} - \mathcal{E}^v_{\text{global}} - \varepsilon\right]_+, \\ \mathcal{L}_{\text{cross}} &= \frac{1}{V(V-1)}\sum_{v=1}^{V}\sum_{u\neq v}\left[\mathcal{E}^v_{\text{global}} - \mathcal{E}^{v\leftarrow u}_{\text{cross}} - \varepsilon\right]_+, \end{aligned} \tag{5}$$

where $\mathcal{L}_{\text{rec}}$ constrains both the complete representation and the global components to retain essential information. $\mathcal{L}_{\text{gain}}$ encourages the matched private component to complement information not captured by the global components alone. $\mathcal{L}_{\text{cross}}$ prevents private components from other views from providing additional reconstruction benefits, thereby encouraging cross-view reusable information to reside exclusively in the global components. $\varepsilon > 0$ denotes the margin used in the two constraints. Together, these objectives encourage the global components to capture cross-view reusable information while preserving view-specific complementary information in the private components.

4.2 Progressive Three-Stage Training

Directly learning multi-view latent actions from synchronized videos is challenging due to the scarcity of such data. Moreover, heterogeneous camera streams exhibit different observed action-amplitude distributions, which can lead to imbalanced training across views. To address these challenges, we propose a strategy that first establishes a stable action-centric representation using abundant front-view data and then progressively adapt the model to the multi-view setting. Specifically, CARVE is trained in three stages as shown in Figure 1(b) and (c). Stage I learns a basic latent action backbone from only large-scale front-view data. Stage II adapts it to support input from other views using a balanced amplitude-guided balancing strategy. Finally, Stage III learns the multi-view fusion capability with the proposed global-private decomposition method and multi-view future dynamics construction.

Stage I: Front-View Latent Action Pretraining. We first pretrain the model on a single-view dataset captured by the front camera, which is substantially larger than the datasets available for other camera views. Given the current and future states s^v_t and s^v_{t+h} extracted by the DINO encoder, the spatio-temporal transformer captures the underlying physical transition using K learnable action queries, which aggregate the transition information into token-level latent actions:

$$Q^v_{t,h} = I(s^v_t, s^v_{t+h}) \in \mathbb{R}^{K\times D}. \tag{6}$$

The resulting $Q^v_{t,h}$ is fed into the FDM for reconstruction and computing the following loss:

$$\widehat{s}^v_{t+h} = s^v_t + F\left(s^v_t, Q^v_{t,h}, h\right), \qquad \mathcal{L}_{\text{I}} = \left\|\widehat{s}^v_{t+h} - s^v_{t+h}\right\|_2^2. \tag{7}$$

Here, F denotes the base FDM, h is the temporal horizon. $\mathcal{L}_{\text{I}}$ minimizes the discrepancy between the predicted and target future latent states, thereby grounding the inferred action tokens in the observed transition. With the visual encoder frozen, the spatio-temporal transformer and FDM are jointly optimized using $\mathcal{L}_{\text{I}}$; the resulting backbone initializes Stage II.

Stage II: View-Balanced Amplitude Curriculum. The Stage-I model supports only front-view inputs. Stage II further adapts the model to other camera views. A naive strategy is to train on frame pairs from different views using fixed or uniformly sampled temporal horizons. However, under cross-view visual heterogeneity, the same temporal horizon can correspond to substantially different observed motion amplitudes across cameras. Consequently, the shared backbone may be trained predominantly on small-motion transitions from some views and large-motion transitions from others, leading to imbalanced training across camera streams. We therefore propose a balanced training strategy that samples transitions for each view according to their observed motion amplitudes.

For a candidate transition starting at frame t in view v with horizon h, we define its normalized feature-space motion amplitude as:

$$a^v_{t,h} = \frac{2\,\mathrm{RMS}(s^v_{t+h} - s^v_t)}{\mathrm{RMS}(s^v_t) + \mathrm{RMS}(s^v_{t+h})}. \tag{8}$$

Here, $\mathrm{RMS}(\cdot)$ denotes the root-mean-square over all spatial and channel dimensions. Specifically, for a tensor z with N elements, $\mathrm{RMS}(z) = \sqrt{N^{-1}\sum_{i=1}^{N} z_i^2}$. During adaptation, we progressively organize training samples for each view from small- to large-amplitude transitions. Let $\lambda \in [0, 1]$ denote the normalized training steps. Before training, for each view, we compute the motion amplitudes of all M candidate transitions and sort them in ascending order. At training steps λ, we choose the amplitude $a^\star$ located at the corresponding position $\lambda M - th$ in the ordered list. Thus, the target amplitude moves progressively from small-motion transitions at the beginning of training to larger-motion transitions later on. For each sampled view and starting frame, we then select the candidate horizon $h^\star$ whose motion amplitude is closest to the current target $a^\star$:

$$h^\star = \arg\min_{h \in \mathcal{H}} \left| a^v_{t,h} - a^\star \right|. \tag{9}$$

The selected transition is used to optimize the spatio-temporal transformer and FDM with the Stage-I reconstruction objective (Equation 7). With normalization and amplitude-guided sampling, transitions selected at the same training stage exhibit similar relative motion amplitudes across different views. This reduces cross-view disparities in transition scale during training, thus providing a more balanced backbone for the subsequent Stage-III training.

Stage III: Global–Private Action Decomposition. Stage II adapts the latent action backbone to different camera views. Stage III further introduces synchronized multi-view training to learn multi-view fusion with the proposed global–private decoupling mechanism illustrated in Section 4.1. With the visual encoder and spatio-temporal transformer frozen, modules in the global-private decoupling architecture (Section 4.1) and FDM are optimized using:

$$\begin{aligned} \mathcal{L}_{\mathrm{fdm}} &= \frac{1}{V}\sum_{v=1}^{V} \|\widehat{s}^v_{t+h} - s^v_{t+h}\|_2^2, \\ \mathcal{L}_{\mathrm{III}} &= \mathcal{L}_{\mathrm{fdm}} + \mathcal{L}_{\mathrm{rec}} + \mathcal{L}_{\mathrm{gain}} + \mathcal{L}_{\mathrm{cross}}, \end{aligned} \tag{10}$$

where $\mathcal{L}_{\mathrm{rec}}$, $\mathcal{L}_{\mathrm{gain}}$ and $\mathcal{L}_{\mathrm{cross}}$ are defined in Equation 5, s^v_{t+h} and $\hat{s}^v_{t+h}$ denote the ground-truth and FDM-predicted future latent states, respectively. Equation 10 grounds the decomposition in future prediction: global evidence is shared across target branches, whereas private evidence is restricted to its corresponding view.

5 Experiments

We evaluate CARVE from three complementary perspectives: predictive and downstream task performance, latent representation quality, and controlled ablation of the proposed design choices. We use consistent evaluation protocols across methods whenever applicable, with detailed settings provided in the Appendix.

5.1 Experimental Setup

Datasets. CARVE is pretrained in Stage I on a mixture of single-view videos from Fractal, Bridge-Data V2, and Something-Something V2. Stages II–III use synchronized DROID (Khazatsky et al., 2024) trajectories with aligned front, side, and wrist views for multi-view adaptation and decomposition. We evaluate reconstruction on DROID Formal300, a held-out set of 300 synchronized clips sampled from 100 DROID episodes. Dataset composition and split statistics are provided in Appendix A.

Baselines and protocols. We compare CARVE with a broad set of strong latent action baselines, including recent multi-view methods such as MVP-LAM and MuCoLA, as well as representative single-view approaches including LAPA, AdaWorld, and DiLA. All methods use the same benchmark-specific splits, preprocessing, and downstream evaluation protocol whenever applicable. Robot action labels are introduced only for downstream probes and controllers.

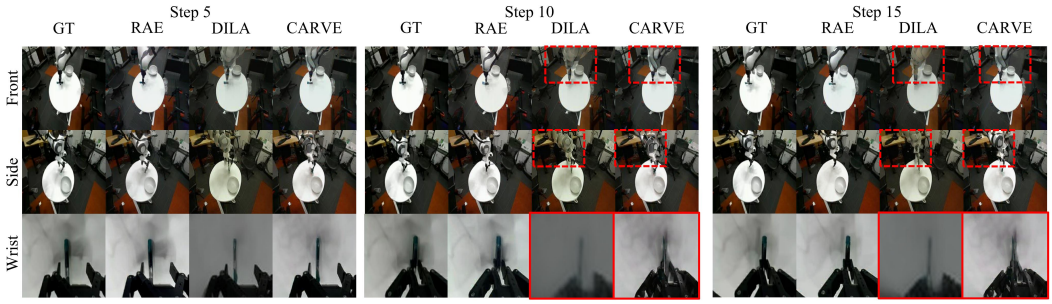


Figure 2: Qualitative comparison on long-horizon multi-view video reconstruction at Steps 5, 10, and 15. Rows show synchronized front, side, and wrist views, while columns compare ground truth, direct RAE reconstructions decoded from the ground-truth latent states, DiLA, and CARVE predictions. CARVE produces visibly more accurate and consistent predictions than DiLA, particularly at later rollout steps where accumulated prediction errors make long-horizon forecasting more challenging.

5.2 Main results

We evaluate CARVE on four different tasks: (1) long-horizon multi-view video reconstruction, predicting future frames through extracted latent actions; (2) action regression, predicting 7D robot actions from frozen latent representations; (3) LIBERO behavior cloning, learning manipulation policies from inferred latent action targets; and (4) VP2 visual planning, planning physical actions through latent action FDM rollouts.

Video Reconstruction. We first evaluate CARVE on long-horizon multi-view video reconstruction over synchronized robot videos. Following the evaluation protocol of DiLA (Zhang et al., 2026), we perform 15-step rollouts, where each step denotes recursive prediction from the current latent state to the next sampled future state, and report PSNR, SSIM, and LPIPS on the same held-out synchronized clips for all methods. As shown in Table 1, CARVE achieves the best result across all three reconstruction metrics. Compared with DiLA, CARVE improves PSNR from 16.372 to 17.009 and SSIM from 0.561 to 0.575, while reducing LPIPS from 0.334 to 0.269. Qualitative comparisons in Figure 2 further show that CARVE preserves scene structure and interaction details more faithfully than DiLA, especially at later rollout steps where prediction errors accumulate.

Table 1: 15-step video reconstruction on held-out synchronized multi-view robot videos.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
LAPA	16.239	0.542	0.461
AdaWorld	16.366	0.556	0.469
DiLA	16.372	0.561	0.334
CARVE	**17.009**	**0.575**	**0.269**

Action Regression. We further evaluate action regression by predicting 7-dimensional robot actions from frozen single- and multi-view latent representations. We freeze the learned latent representations and train the same linear probe to predict 7-dimensional robot actions on LIBERO Goal and DROID Formal300; details are provided in Appendix C.2. As shown in Table 3, combining synchronized information across views consistently improves action prediction. Front and wrist outperform either front alone or competing latent action representations on LIBERO Goal, while incorporating all three views yields the strongest performance on DROID Formal300. These results demonstrate that multi-view fusion yields more informative latent action representations for predicting robot actions.

Policy Learning on LIBERO. We evaluate downstream policy learning on the four LIBERO suites (Liu et al., 2023). For each method, a shared BCTransformer provided by official LIBERO benchmark predicts frozen latent-action targets from image observations and task instructions, followed by a lightweight latent-to-action decoder. All methods use the same demonstrations, policy architecture, and training protocol; details are provided in Appendix C.4. CARVE achieves the highest mean success rate of 71.50%, outperforming the strongest baseline, AdaWorld, by 9.50 percentage

Table 2: LIBERO behavior-cloning success rates (%).

Method	Long	Spatial	Object	Goal	Mean
LAPA	35.00	53.67	58.33	67.00	53.50
DiLA	57.00	50.33	63.67	61.33	58.08
AdaWorld	56.33	54.33	75.00	62.33	62.00
MVP-LAM	24.00	59.67	69.00	65.33	54.50
CARVE	**58.67**	**67.33**	**81.33**	**78.67**	**71.50**

Table 3: Linear action probing on LIBERO Goal and DROID Formal300. G_f, G_s, and G_w denote the front, side, and wrist global components, respectively. LIBERO provides only front and wrist views; therefore, the three-view CARVE variant $G_f + G_s + G_w$ cannot be evaluated on this dataset; lower RMSE and MAE are better.

	LIBERO Goal		*DROID Formal300*	
Representation	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓
DiLA	0.41301 ± 0.00114	0.22899 ± 0.00107	0.28598 ± 0.00151	0.20385 ± 0.00119
LAPA	0.38794 ± 0.00139	0.22324 ± 0.00028	0.26120 ± 0.00127	0.18969 ± 0.00087
AdaWorld	0.38505 ± 0.00016	0.22266 ± 0.00025	0.26973 ± 0.00010	0.18705 ± 0.00007
MVP-LAM	0.38452 ± 0.00007	0.21928 ± 0.00002	0.25743 ± 0.00011	0.18214 ± 0.00031
CARVE (G_f)	0.40814 ± 0.00008	0.23371 ± 0.00041	0.26325 ± 0.00032	0.18517 ± 0.00072
CARVE ($G_f + G_w$)	**0.38138 ± 0.00012**	**0.21762 ± 0.00019**	0.25609 ± 0.00023	0.17969 ± 0.00054
CARVE ($G_f + G_s + G_w$)	–	–	**0.23117 ± 0.00005**	**0.16356 ± 0.00004**

Table 4: VP2 Any-time success (%), RoboSuite results are averaged over four seeds, while RoboDesk results are averaged over three seeds. Baseline results are taken from the corresponding publications.

Method	RoboSuite Push	Open Slide	Red Button	Green Button	Flat Block	Upright Block
LAPA	52.10±7.23	0.00±0.00	17.22±8.77	33.59±4.89	0.00±0.00	10.45±7.10
AdaWorld	64.37±3.71	1.30±0.56	22.88±2.99	8.78±3.55	13.33±6.67	6.11±1.24
DiLA	68.00±1.41	15.00±5.00	20.83±5.95	35.83±4.93	10.00±5.77	3.33±2.72
MuCoLA	68.90±3.64	3.38±1.88	27.68±2.79	51.32±17.29	0.00±0.00	19.76±4.51
CARVE (ours)	**71.75±3.30**	**16.67±5.77**	**41.11±6.94**	**52.22±6.94**	**25.56±6.94**	**30.00±10.00**

points. CARVE achieves the best performance across all four LIBERO suites, demonstrating strong and consistent transfer of the learned latent actions to downstream manipulation.

Visual Planning with VP2. We evaluate model-predictive control on the VP2 benchmark (Tian et al., 2023). CARVE serves as the predictive world model within the official MPPI (Williams et al., 2017) planner, which samples physical-action sequences and rolls them out through the model. We train a lightweight action-to-latent MLP to map each sampled action sequence into the corresponding latent action space before rollout. Planner settings, task counts, and action-grounding details are provided in Appendix C.5. We include published VP2 results from LAPA, AdaWorld, DiLA, and MuCoLA as contextual references. As shown in Table 4, CARVE achieves the highest success rate among the compared methods on all tasks, demonstrating its high effectiveness for visual planning. Qualitative successful and failed planning trajectories are shown in Figure 5 in the Appendix.

5.3 Latent Action Representation Analysis

We next examine whether CARVE separates action-relevant information from view-specific variation. On synchronized transitions, we extract the global components G_v and private components P_v, visualize their geometry with UMAP (Healy & McInnes, 2024), and use frozen-feature linear probes to measure action- and view-related information.

As shown in Figure 3, the unfactorized latent action space already exhibits structure associated with translation magnitude and dominant motion direction. After global-private decomposition, translation magnitude is substantially more predictable from the global representation than from the private representation. The UMAP visualizations further show stronger cross-view mixing among global components, whereas private components retain clearer view-dependent structure. Linear probing provides consistent quantitative evidence: global features are more informative for action retrieval, while private features are substantially more predictive of view identity. Together, these results demonstrate the intended separation between cross-view reusable action information and view-specific transition cues.

5.4 Ablation Study

We further conduct experiments to validate three key designs in CARVE: (1) amplitude-guided cross-scale sampling, (2) the global-private decomposition, and (3) the decomposition objectives and asymmetric routing. Definitions of all variants are provided in Appendix D.1.

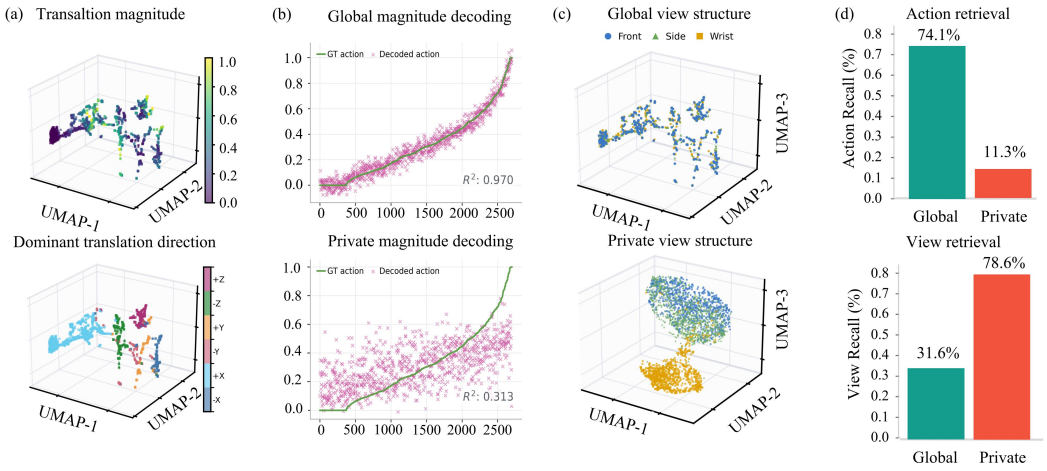


Figure 3: Latent action representation analysis on held-out synchronized transitions. (a) Translation magnitude and dominant translation direction. (b) Global and private magnitude decoding. (c) Global and private view structure. (d) Action and view retrieval.

Table 5: Controlled ablations on 15-step video reconstruction. Higher PSNR/SSIM and lower LPIPS are better.

Sampling				*Routing*			
Variant	PSNR ↑	SSIM ↑	LPIPS ↓	Variant	PSNR ↑	SSIM ↑	LPIPS ↓
Full CARVE	**17.009**	**0.575**	**0.269**	Full CARVE	**17.009**	**0.575**	**0.269**
Uniform	16.425	0.536	0.293	Matched $G_v + P_v$ only	16.574	0.561	0.275
No cross-scale	16.221	0.533	0.337	w/o stop-gradient	16.101	0.524	0.295
Objectives				*Representation*			
Variant	PSNR ↑	SSIM ↑	LPIPS ↓	Variant	PSNR ↑	SSIM ↑	LPIPS ↓
Full CARVE	**17.009**	**0.575**	**0.269**	Full CARVE	**17.009**	**0.575**	**0.269**
w/o $\mathcal{L}_{\text{rec}}$	16.198	0.511	0.277	Action fusion	15.764	0.530	0.301
w/o $\mathcal{L}_{\text{gain}}$	16.232	0.506	0.281	Global only	15.977	0.534	0.286
w/o $\mathcal{L}_{\text{cross}}$	16.411	0.532	0.291	Private only	16.381	0.530	0.278

As shown in Table 5, removing amplitude-guided sampling or cross-scale coverage consistently degrades reconstruction, while direct action fusion and either global or private components alone also underperform the full model. The objective and routing ablations further show that each decomposition loss contributes to performance, and that removing cross-view global routing or stop-gradient degrades reconstruction. Despite introducing the global-private decomposition, CARVE remains lightweight, adding only approximately 7.2% parameters relative to the full single-view backbone. Detailed parameter analysis is provided in Appendix B.5. These results validate the complementary roles of the proposed sampling, factorization, and routing designs.

6 Conclusion and Limitations

This paper introduces CARVE, a multi-view latent action framework designed to address cross-view visual heterogeneity. CARVE combines global-private decomposition and asymmetric routing for selective cross-view information sharing, along with a progressive training strategy that adapts latent action learning from single-view videos to heterogeneous multi-view observations. Across diverse tasks, including video prediction, action probing, behavior cloning, and visual planning, CARVE consistently outperforms strong latent action baselines. These results demonstrate that effective multi-view latent action learning benefits from jointly capturing cross-view reusable information and view-specific transition cues, rather than enforcing full view invariance.

CARVE currently assumes synchronized cameras and a fixed global-private factorization. Handling missing, asynchronous, or moving views is left for future work.

AI USE STATEMENT

Generative AI tools, including ChatGPT, were used during this work to assist with research brainstorming, feedback on methodological and experimental design, literature organization, translation, and editing the manuscript for clarity and readability. All model designs, experimental implementations, data processing, quantitative results, and scientific claims were developed, executed, verified, and reviewed by the authors. AI-generated suggestions were treated only as intermediate feedback and were independently evaluated before being incorporated into the work. The authors take full responsibility for the content and conclusions of this paper.

REPRODUCIBILITY STATEMENT

We will release the training configurations, model and evaluator code, checkpoint hashes, data-manifest construction scripts, and paired evaluation protocols needed to reproduce the reported results.

REFERENCES

Leonardo Barcellona, Andrii Zadaianchuk, Davide Allegro, Samuele Papa, Stefano Ghidoni, and Efstratios Gavves. Dream to manipulate: Compositional world models empowering robot imitation learning with imagination. In *International Conference on Learning Representations*, volume 2025, pp. 56729–56763, 2025.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *Advances in neural information processing systems*, 29, 2016.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first international conference on machine learning*, 2024.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Xizhou Bu, Qingda Hu, Lei Zhou, Lingfeng Zhang, Yingbo Tang, Zihao Liu, Xinyi Tao, Zhiqiang Ma, Qingqiu Huang, Chufeng Tang, et al. What matters for latent actions in robot learning. *arXiv preprint arXiv:2608.19613*, 2026.

Yi Chen, Yuying Ge, Weiliang Tang, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for learning robot manipulation from videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19752–19763. IEEE, 2025.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International conference on learning representations*, volume 2024, pp. 2632–2652, 2024.

Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Cai Zhehao, and Lin Shao. Flip: Flow-centric generative planning as general-purpose manipulation world model. In *International Conference on Learning Representations*, volume 2025, pp. 21927–21948, 2025a.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. Adaworld: Learning adaptable world models with latent actions. 2025b.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *2017 IEEE international conference on computer vision (ICCV)*, pp. 5843–5851. IEEE, 2017.

Yanjiang Guo, Yucheng Hu, Jianke Zhang, Yen-Jen Wang, Xiaoyu Chen, Chaochao Lu, and Jianyu Chen. Prediction with action: Visual policy learning via joint denoising process. *Advances in Neural Information Processing Systems*, 37:112386–112410, 2024.

Yanjiang Guo, Lucy Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 6121–6138, 2026.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

John Healy and Leland McInnes. Uniform manifold approximation and projection. *Nature Reviews Methods Primers*, 4(1):82, 2024.

Youngjoon Jeong, Jihwan Yu, Minsoo Jo, Junha Chun, and Taesup Kim. Polar: Factorizing extent and mode in latent actions for robot policy learning. *arXiv preprint arXiv:2606.21139*, 2026.

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

Akira Kinose, Masashi Okada, Ryo Okumura, and Tadahiro Taniguchi. Multi-view dreaming: Multi-view world model with contrastive learning. *Advanced Robotics*, 37(19):1212–1220, 2023.

Jung Min Lee, Dohyeok Lee, Seokhun Ju, Taehyun Cho, Jin Woo Koo, Li Zhao, Sangwoo Hong, and Jungwoo Lee. Mvp-lam: Learning action-centric latent action via cross-viewpoint reconstruction. *arXiv preprint arXiv:2602.03668*, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. *arXiv preprint arXiv:2502.00379*, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Jing-Cheng Pang, Nan Tang, Kaiyuan Li, Yuting Tang, Xin-Qiang Cai, Zhen-Yu Zhang, Gang Niu, Masashi Sugiyama, and Yang Yu. Learning view-invariant world models for visual robotic manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 54853–54876, 2025.

Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *European conference on computer vision*, pp. 765–782. Springer, 2018.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *International Conference on Learning Representations*, volume 2024, pp. 9379–9395, 2024.

Younggyo Seo, Junsu Kim, Stephen James, Kimin Lee, Jinwoo Shin, and Pieter Abbeel. Multi-view masked world models for visual robotic manipulation. In *International Conference on Machine Learning*, pp. 30613–30632. PMLR, 2023.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1134–1141. IEEE, 2018.

Stephen Tian, Chelsea Finn, and Jiajun Wu. A control-centric benchmark for video prediction. *arXiv preprint arXiv:2304.13723*, 2023.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pp. 776–794. Springer, 2020.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on robot learning*, pp. 1723–1736. PMLR, 2023.

Shenghua Wan, Xiaohai Hu, Xunlan Zhou, Lei Yuan, Le Gan, and De-Chuan Zhan. Multi-view consistent latent action learning for world modeling and control. In *Forty-third International Conference on Machine Learning*, 2026.

Junming Wang. Latentvla: Taming latent space for generalizable and long-horizon bimanual manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18593–18601, 2026.

Zizhao Wang, Chang Shi, Jiaheng Hu, Kevin Rohling, Roberto Martín-Martín, Amy Zhang, and Peter Stone. Factored latent action world models. *arXiv preprint arXiv:2602.16229*, 2026.

Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information theoretic mpc for model-based reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 1714–1721. IEEE, 2017.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *International Conference on Learning Representations*, volume 2025, pp. 28213–28239, 2025.

Takanori Yoshimoto, Yang Hu, Naruya Kondo, and Tatsuya Matsushima. Flexlam: Resolving the bottleneck trade-off in latent action learning. *arXiv preprint arXiv:2606.19408*, 2026.

Tianqiu Zhang, Muyang Lyu, Yufan Zhang, Fang Fang, and Si Wu. Dila: Disentangled latent action world models. *arXiv preprint arXiv:2605.15725*, 2026.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, volume 2026, pp. 35791–35820, 2026.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025.

SUPPLEMENTARY OVERVIEW

This supplementary material is organized as follows. Appendix A describes the datasets, splits, and input preprocessing. Appendix B provides the architecture and training details. Appendix C specifies the evaluation protocols for reconstruction, action prediction, representation analysis, behavior cloning, and visual planning. Appendix D collects the ablation definitions, model-capacity scaling study, additional results, and cross-view action analysis. Appendix E describes the qualitative rollouts and planning failure cases.

A DATASETS AND PREPROCESSING

We use data from human–object interaction, robot manipulation, synchronized multi-view observation, and downstream control benchmarks. Stage I is trained on a large single-view mixture to establish a general action-centric backbone, whereas Stages II–III use synchronized DROID observations to adapt the model to heterogeneous camera streams and learn the global-private decomposition. LIBERO and VP2 are used only for downstream evaluation.

A.1 SINGLE-VIEW PRETRAINING DATA

Fractal. Fractal (Brohan et al., 2022) is used as one source of robot-manipulation video in the Stage-I single-view mixture. We use its front-view stream only and treat the videos as observation-only data; action labels and synchronized multi-view observations are not required during latent action pretraining.

Bridge. BridgeData V2 (Walke et al., 2023) provides additional tabletop manipulation videos for Stage I. Its diverse object interactions complement the other robot-video sources and expose the single-view latent action backbone to a broader range of manipulation transitions. As with the other Stage-I sources, only visual observations are used for CARVE pretraining.

Something-Something V2. Something-Something V2 (Goyal et al., 2017) contains human–object interaction videos in which the temporal evolution of an interaction is often more informative than static appearance. We include it in Stage I to broaden the motion patterns observed by the latent action backbone beyond robot-only data.

A.2 SYNCHRONIZED DROID DATA

DROID (Khazatsky et al., 2024) is the primary robot-video source for multi-view learning. It provides aligned front, side, and wrist observations for Stages II–III. We use 19,001 trajectories for training and 150 trajectories for validation. Stage II samples the three camera streams independently for view-balanced adaptation, while Stage III consumes synchronized triplets with the same starting time and temporal horizon across all views.

A.3 DROID FORMAL300

For reconstruction and representation analysis, we construct Formal300 as an episode-disjoint held-out evaluation split. It contains 300 synchronized clips randomly sampled from 100 held-out DROID source episodes. Each clip contains 16 frames from the front, side, and wrist cameras. Source videos are recorded at 15 FPS and sampled with stride 3, giving a temporal span of 3.0 seconds per clip and 15 adjacent transition pairs per clip-view.

A.4 DOWNSTREAM BENCHMARKS AND DATASET SUMMARY

LIBERO. LIBERO (Liu et al., 2023) evaluates whether the learned latent actions are useful for downstream robot policy learning. It contains 40 manipulation tasks organized into the Long, Goal, Object, and Spatial suites, with 50 official demonstrations per task. We use LIBERO Goal for action predictability and all four suites for behavior cloning, training one BCTransformer policy per suite under the controlled protocol described in Appendix C.4.

Table 6: Detailed statistics of DROID Formal300.

Statistic	Value
Source episodes	100
Clips	300
Views per clip	3
Clip-view instances	900
Frames per clip-view	16
Source FPS / sampling stride	15 / 3
Temporal span per clip	3.0 s
Transition pairs per clip-view	15
Synchronized clip-level transition pairs	4,500
View-transition instances	13,500
Materialized RGB frames	14,400

Table 7: Dataset usage and split statistics. "Sync." indicates whether synchronized views are consumed jointly.

Dataset	Role	Views	Sync.	Size / split
Single-view mixture	Stage I	front	No	103,417 videos
DROID multi-view	Stages II–III	F/S/W	II: No; III: Yes	19,001 train; 150 val.
DROID Formal300	Recon. / probe	F/S/W	Yes	100 episodes; 300 clips
LIBERO	Probe / BC	F/W	–	40 tasks; 50 demos/task
VP2	Planning	F/W	–	benchmark task set

VP2. VP2 (Tian et al., 2023) is a control-centric benchmark for evaluating predictive models through visual model-based planning. We use its RoboSuite and RoboDesk manipulation tasks to test whether CARVE's learned dynamics can support closed-loop planning when physical actions are mapped into the latent action space. Detailed planning settings are provided in Appendix C.5.

A.5 Input Processing and Transition Construction

Each observation is converted to RGB and resized to 256×256. The frozen DINOv2-with-registers-base visual encoder internally receives a bicubically resized 224×224 image and produces a $16 \times 16 \times 768$ spatial feature grid after removing the special/register tokens. CARVE predicts future states in this latent space. A frozen ViT-XL RAE decoder maps the predicted latent state back to a 256×256 RGB image for visualization and image-space evaluation.

Training sequences contain $T = 16$ observations. For a valid starting index t, candidate transitions are constructed using $\mathcal{H} = \{1, 2, 4, 8\}$, with candidates removed whenever $t + h$ exceeds the sequence boundary. Stage II samples camera streams independently. Stage III requires the complete synchronized front/side/wrist triplet, uses the same starting index and selected horizon for all three views.

B Architecture and Training Details

B.1 Model Parameters

CARVE uses a frozen DINOv2 feature space and learns the transition model on top of the resulting spatial tokens. Table 8 summarizes the paper configuration. The IDM contains spatial, temporal, and pooling/cross- attention transformer blocks. It uses $K = 4$ learned action queries with latent code dimension $D = 4$, yielding a 16-dimensional flattened latent action representation per component. The transformer modules use an internal hidden width of 512, which is distinct from the latent code dimension D.

For target view v, the fusion FDM receives the current target latent state, all global components $\{G_u\}_{u=1}^{V}$, only the matched private component P_v, and the transition-horizon embedding. The base

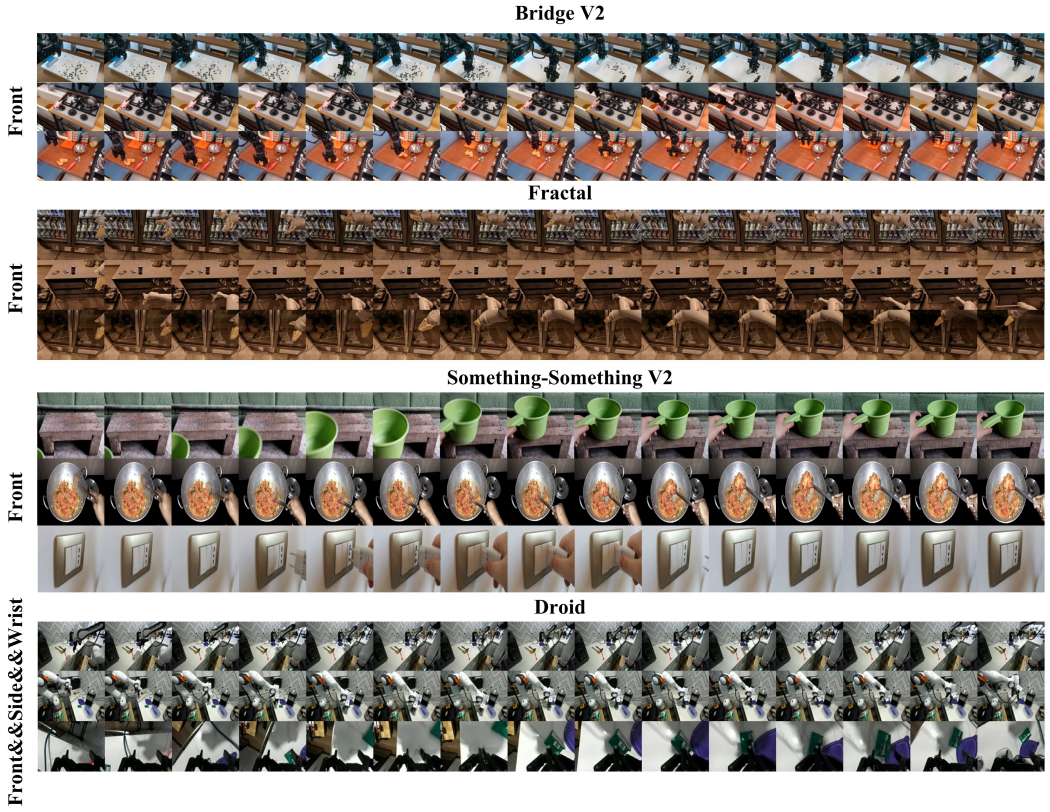


Figure 4: Representative observations from the datasets used by CARVE. The figure includes examples from the Stage-I single-view sources (Fractal, Bridge, Something-Something V2), synchronized front/side/wrist DROID observations used in Stages II–III, and downstream LIBERO and VP2 tasks.

single-view FDM used in Stages I–II follows the same four-layer latent-dynamics backbone without the multi-view routing inputs.

B.2 Detailed Module Architectures

This section details the modules summarized in Table 8.

Spatio-temporal Transformer. The inverse-dynamics backbone operates on the current and future DINO feature grids and contains spatial, temporal, and pooling/cross-attention transformer blocks with depths $2/2/2$. The internal hidden width is 512, with eight attention heads, head dimension 64, and dropout 0.1. Spatial blocks model within-frame token interactions, while temporal blocks aggregate transition information between the two observations. Finally, $K = 4$ learned action queries pool the transition into K latent-action tokens. Each query produces a code of dimension $D = 4$, giving a 16-dimensional flattened latent-action representation per component.

Global Transformer. For synchronized multi-view inputs, the global transformer T_G concatenates the K transition tokens from all V views and jointly contextualizes the resulting VK tokens. It uses hidden width 512, two transformer layers, eight attention heads, feed-forward ratio 4.0, and dropout 0.1. The output is partitioned back into V view-indexed groups $\{G_v\}_{v=1}^{V}$, each containing K global tokens. The paper configuration supports up to three synchronized views.

Private Transformer. The private transformer T_P is shared across views and uses the same width, depth, head count, feed-forward ratio, and dropout as T_G. For each view v, it receives the local transition tokens Q_v together with the corresponding global context $\mathrm{sg}(G_v)$ and produces K

Table 8: CARVE architecture specification.

Component	Parameter	Value
Input / visual state	RGB input / encoder input	256×256 / 224×224
	frozen visual encoder	DINOv2-with-registers-base
	latent grid	$16\times16\times768$
IDM	hidden width	512
	spatial / temporal / pooling depth	2 / 2 / 2
	attention heads / head dimension	8 / 64
	dropout	0.1
	learned action queries K / code dimension	4 / 4
global transformer T_G	width / depth / heads	512 / 2 / 8
	feed-forward ratio / dropout	4.0 / 0.1
	maximum views	3
private transformer T_P	width / depth / heads	512 / 2 / 8
	feed-forward ratio / dropout	4.0 / 0.1
	global context gradient	stop-gradient
Reconstruction D, D_G	architecture	four-layer transformers
	width / heads / FF ratio	512 / 8 / 4.0
Fusion FDM	state channels / hidden width	768 / 512
	depth / heads / head dimension	4 / 8 / 64
	maximum horizon embedding	8
	view conditioning	source/target embeddings enabled
Visualization decoder	architecture	frozen RAE ViT-XL decoder

private tokens P_v. The stop-gradient on G_v prevents the private branch from modifying the global representation through this pathway.

Forward Dynamics Model (FDM). CARVE uses a four-layer transformer-based latent-dynamics backbone in two conditioning forms. In Stages I–II, the single-view FDM conditions the current visual latent state on the latent-action tokens $Q^v_{t,h}$ together with the horizon embedding and predicts a residual update in the DINO latent space. In Stage III, the same backbone is extended to multi-view conditioning: for target view v, the fusion FDM receives all routed global components $\mathbf{G} = \{G_j\}_{j=1}^{V}$, the matched private component P_v, and the transition-horizon embedding, while unmatched private components are excluded. Source- and target-view embeddings are included to retain camera identity during fusion. The FDM uses state channels 768, hidden width 512, eight attention heads with head dimension 64, and a maximum horizon embedding of 8. In both forms, the predicted residual is added to the current latent state to obtain the future latent state.

Visual Encoder and RAE Decoder. We use the DINOv2-B encoder with Registers as the frozen visual encoder. The input image is resized to 224×224 and encoded with patch size 14, producing a 16×16 grid of 768-dimensional patch tokens after discarding the CLS and register tokens. For image reconstruction, we use the pretrained ViT-XL decoder released with RAE. The decoder follows an MAE-style transformer architecture with 28 layers, hidden width 1152, 16 attention heads, and an FFN width of 4096. Following the DINOv2 RAE configuration, the decoder uses patch size 16 to reconstruct 256×256 RGB images from the 16×16 latent grid. Both the visual encoder and decoder remain frozen throughout CARVE training.

B.3 Stage-Wise Training

CARVE is optimized progressively. Stage I learns the front-view latent action backbone. Stage II continues adapting the same IDM and base FDM to all camera streams with amplitude-guided transition sampling. Stage III freezes the Stage-II backbone and learns the global-private decomposition together with the synchronized fusion FDM.

Table 9: Stage-wise training schedule. "Independent" means that camera streams are sampled separately; "synchronized" means that the same starting index and horizon are used for all three views.

Stage	Data / views	Updates	Batch	Trainable modules	Horizons
I	mixed front-view data	12,000	4	IDM + base FDM	1
II	DROID F/S/W	12,000	4	IDM + base FDM	$\{1, 2, 4, 8\}$
III	DROID F/S/W	12,000	4	decomposition + fusion FDM	1

Table 10: Parameter overhead of the global-private decomposition.

Model	Parameters	Increase
Single-view baseline	541.7M	–
+ global-private decomposition	580.6M	+7.2%

Across stages, optimization uses AdamW with learning rates 10^{-4} for both the inverse-dynamics and world-model components, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 10^{-5}, no gradient accumulation, and gradient clipping at a maximum norm of 6.0. We use random seed 42, six data-loader workers in the final amplitude-training configurations, shuffled training data, and pinned host memory. Stages I–II use a CosineAnnealingLR scheduler with $T_{\max} = 100$ and $\eta_{\min} = 10^{-6}$ in the raw world-model path. The Stage-III decoupled path uses a constant LambdaLR multiplier of 1.0.

Stage-III freezing policy. The DINOv2 encoder, RAE modules, and Stage-II spatio-temporal latent action backbone are frozen. The global transformer, private transformer, action-space reconstruction modules, and synchronized fusion FDM remain trainable. Stage III uses three required views, batch size 4, 12,000 updates, and stride 3.

B.4 Training Objective

Stage III combines the future-state prediction loss $\mathcal{L}_{\mathrm{fdm}}$ with the decomposition losses $\mathcal{L}_{\mathrm{rec}}$, $\mathcal{L}_{\mathrm{gain}}$, and $\mathcal{L}_{\mathrm{cross}}$ defined in the main paper. Objective ablations are specified in Appendix D.1.

B.5 Parameter Counts

Table 10 reports the parameter counts for the paper configuration. The DINOv2 encoder and RAE decoder remain frozen throughout training.

B.6 Reproducibility and Artifacts

The released artifact will provide the stage-wise training configurations, model and evaluation code, dataset manifests and splits, metric implementations, and commands required to reproduce the reported results. The paper configuration uses four-layer transformer reconstruction decoders D and D_G. All three stages are trained in FP32 on four NVIDIA A100 PCIe GPUs with 40 GB of memory per GPU. We use a per-GPU batch size of 4, corresponding to a global batch size of 16, without mixed precision or gradient accumulation. The complete three-stage training pipeline takes approximately 16 hours.

C Evaluation Protocols

C.1 Video Reconstruction

All methods use the episode-disjoint Formal300 split defined in Appendix A.3. Starting from the observed context, each model recursively feeds its predicted latent state into the next prediction step for a 15-step rollout. PSNR, SSIM, and LPIPS are computed on decoded RGB predictions, and Table 1 reports aggregate results over the 300 clips. The RAE decoder is fixed across methods.

C.2 Action Predictability

The latent action model is frozen, and a linear regressor predicts the 7-dimensional robot action. Splits are episode-disjoint, and feature standardization is fitted on the training split. RMSE and MAE use the same action dimensions and normalization for all representations. Reported values are averaged over three probe seeds. For CARVE, G_f, G_s, and G_w denote the front, side, and wrist global components, respectively. We evaluate $G_f + G_w$ on LIBERO and both $G_f + G_w$ and $G_f + G_s + G_w$ on DROID, depending on the available synchronized views. These comparisons test whether incorporating additional synchronized views provides complementary action-relevant information beyond a single front view.

C.3 Latent Representation Analysis

We evaluate frozen CARVE representations on DROID Formal300 using an episode-disjoint 80/20 split over the 100 source episodes (split seed 3407). For each synchronized transition, CARVE produces a 16-dimensional global representation and a 16-dimensional private representation for each of the front, side, and wrist views. The corresponding physical-action target is the mean 7-dimensional DROID command over the stride-3 transition interval. Feature standardization is fitted on the training episodes only and applied unchanged to the held-out data.

For visualization, we apply UMAP with Euclidean distance, **n_neighbors**=30, **min_dist**=0.1, and **random_state**=42. We use three-dimensional projections for the visualizations in Figure 3. UMAP is fitted separately for each representation, and the resulting coordinates are kept fixed when recoloring the same representation by view identity, translation magnitude, or dominant translation direction.

For quantitative probing, we train a linear regressor from the frozen latent representation to the 7-dimensional physical action using MSE and SGD (learning rate 0.01, batch size 32). We use 800 training transitions and 200 held-out transitions and repeat each probe with three initialization seeds, reporting mean and standard deviation. All representation variants use the same split, preprocessing, and probe configuration.

C.4 LIBERO Behavior Cloning

Data and latent action targets. We evaluate on the 40 LIBERO tasks organized into the Long, Spatial, Object, and Goal suites, using 50 official demonstrations per task. Each suite contains ten tasks and is used to train an independent policy and action decoder. For each method, the frozen latent action model first labels the demonstration transitions. CARVE concatenates the front and wrist global and private components, $[G_f; G_w; P_f; P_w]$, yielding a 64-dimensional target. Latent actions, physical actions, and proprioceptive states are standardized using statistics computed from the training data only.

Policy architecture. All methods use the same BCTransformer backbone, conditioned on front and wrist RGB observations, 15-dimensional proprioceptive states, and a learned task embedding. Proprioception comprises joint positions, end-effector position and axis-angle orientation, and gripper positions. Two ImageNet-pretrained ResNet-18 encoders process the 128×128 images. A four-layer causal transformer with hidden width 64 and eight attention heads integrates these inputs over a ten-frame observation history and predicts a four-step latent action chunk. The output dimension is adjusted to the latent representation of each method.

Action decoder and policy training. We first train a two-layer action decoder, $\mathrm{Linear}(d_z, d_z) \to \mathrm{SiLU} \to \mathrm{Linear}(d_z, 7)$, where d_z is the latent action dimension. The decoder minimizes MSE between its predictions and standardized physical actions for 10,000 updates using SGD with learning rate 0.01 and momentum 0.9.

We then freeze the decoder and optimize the policy with

$$\mathcal{L}_{\mathrm{BC}} = \mathcal{L}_{\mathrm{latent}} + 0.5\mathcal{L}_{\mathrm{action}} + 0.1\mathcal{L}_{\mathrm{temporal}}. \tag{11}$$

The latent loss is a Smooth L1 loss against the standardized latent action targets. The action loss is the MSE between actions decoded from the policy predictions and standardized ground-truth actions. The temporal loss applies Smooth L1 supervision to adjacent action increments within each predicted

chunk. Gradients from the action losses propagate through the frozen decoder to the policy. Thus, physical action labels supervise both decoder fitting and downstream policy training, while the latent action teacher remains frozen.

Policies are trained for 50 epochs with eight sampled windows per trajectory per epoch and a global batch size of 28. We use AdamW with learning rate 10^{-4}, weight decay 10^{-4}, cosine learning-rate decay, and gradient clipping at norm 5.0. Training images undergo random cropping and mild brightness augmentation. All methods, including MVP-LAM, follow the same data selection, training schedule, loss weights, and evaluation procedure.

Closed-loop evaluation. At each control step, the policy predicts a four-step chunk, but only its first decoded action is executed before observing the environment again. The six continuous action dimensions are clipped to $[-1, 1]$, and the gripper command is binarized by its sign. Each episode begins with five zero-action settling steps and allows up to 600 subsequent control steps.

We evaluate the fixed final policies in three groups, using seeds 3407, 3408, and 3409 with disjoint official initial-state ranges 0–9, 10–19, and 20–29, respectively. Each group contains ten trials per task, giving 100 episodes per suite and 400 episodes across the four suites. Across all three groups, each method is evaluated on 1,200 episodes. Suite success rates are averaged across groups, and the overall score is the macro-average over the four suites.

C.5 VP2 Visual Planning

Benchmark and action grounding. We evaluate CARVE on RoboSuite Push and five RoboDesk tasks: Open Slide, Red Button, Green Button, Flat Block, and Upright Block. The planner samples physical actions, whereas CARVE predicts future latent states conditioned on latent actions. We bridge these interfaces with a two-layer SiLU MLP that maps physical actions to IDM transition tokens. The predicted tokens are then processed by the global-private decomposition and the multi-view FDM. For RoboDesk, the shared adapter additionally incorporates a learned task embedding, which remains fixed across all candidates within an episode.

Adaptation objectives. Adaptation proceeds through adapter initialization, joint adaptation, and dual-path calibration. First, the adapter is trained to match transition tokens extracted by the frozen video IDM:

$$\mathcal{L}_Q = \mathrm{MSE}(\widehat{Q}, \mathrm{sg}(Q)), \tag{12}$$

where $\widehat{Q}$ is predicted from physical actions and Q is inferred from the corresponding observation pairs. Joint adaptation combines token alignment with recursive latent-state prediction and teacher-forced state-increment prediction:

$$\mathcal{L}_{\mathrm{joint}} = \mathcal{L}_Q + \mathcal{L}_{\mathrm{rollout}}. \tag{13}$$

Here, $\mathcal{L}_{\mathrm{rollout}}$ is the MSE between recursively predicted and ground-truth latent states, starting from the first observed state, which is computed in the frozen visual encoder's feature space.

Finally, dual-path calibration jointly supervises the physical-action path and the video-IDM path:

$$\mathcal{L}_{\mathrm{cal}} = \mathcal{L}_{\mathrm{rollout}} + \mathcal{L}_{\mathrm{video}} + 0.1\mathcal{L}_{\mathrm{action}}. \tag{14}$$

The video-path loss $\mathcal{L}_{\mathrm{video}}$ is a teacher-forced one-step latent-state reconstruction loss using video-inferred transition tokens. An auxiliary action decoder predicts standardized physical actions from the global components of each path:

$$\mathcal{L}_{\mathrm{action}} = \frac{1}{2}\left[\mathrm{MSE}(\widehat{a}_{\mathrm{video}}, a_{\mathrm{norm}}) + \mathrm{MSE}(\widehat{a}_{\mathrm{action}}, a_{\mathrm{norm}})\right]. \tag{15}$$

Token alignment is disabled during calibration. The same loss weights are used for RoboSuite and RoboDesk. The visual encoder, RAE decoder, and video IDM remain frozen; the action adapter, decomposition modules, and FDM are adapted, together with the auxiliary action decoder during calibration.

Planning configuration. We use the VP2 MPPI planner with two context frames and a ten-step prediction horizon. Each episode contains 15 control actions, beginning with a zero action. Thereafter, the controller replans after every action, performs one MPPI distribution update, and executes the first

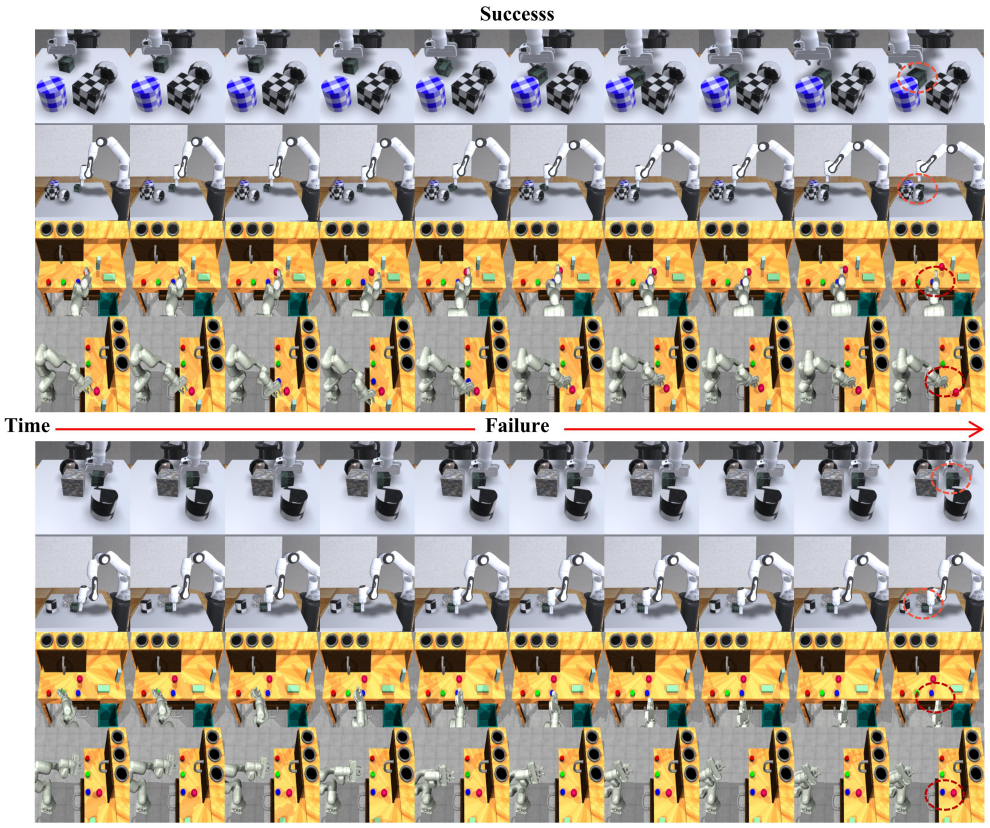


Figure 5: Representative VP2 planning traces. Each row shows the initial observation, target observation, executed trajectory, and final outcome.

action of the selected sequence. All tasks use $\beta = 0.7$, $\gamma = 0.05$, and an initial sampling standard-deviation multiplier of 1.25 relative to the benchmark defaults. Open Slide uses 800 candidate sequences per replan; all other tasks use 200. Sampled actions are clipped to $[-1, 1]$.

Evaluation and statistics. We report official Any-time success: an episode succeeds if the simulator-defined task condition is satisfied at any evaluated state during execution. This criterion is distinct from the classifier score used for planning. RoboSuite Push uses 100 goals per run and the official object-distance threshold of 0.05; each RoboDesk task uses 30 goals per run. RoboSuite Push results are averaged over four seeds, while RoboDesk results are averaged over three seeds. The CARVE entries report mean success rates and sample standard deviations across runs. Published baseline values are retained as reported in their respective sources.

D Additional Results and Ablations

D.1 Complete Ablation Study

The main ablation table changes one design choice at a time under the same 15-step reconstruction protocol. Table 11 clarifies the intervention performed by each reported variant. Wherever the representation permits, the visual encoder, data split, transition construction, and FDM capacity are kept matched.

Temporal sampling. The Uniform variant retains the candidate set $\mathcal{H} = \{1, 2, 4, 8\}$ but removes amplitude-based horizon selection. The No-cross-scale variant removes cross-horizon coverage, isolating whether the gain comes merely from seeing multiple temporal scales. Their lower recon-

Table 11: Definitions of the controlled ablations reported in Table 5.

Group	Variant	Intervention / question isolated
Sampling	Full CARVE	amplitude-guided cross-scale curriculum
	Uniform	sample candidate horizons uniformly from $\{1, 2, 4, 8\}$
	No cross-scale	remove cross-horizon coverage and use a single temporal scale
Representation	Full CARVE	global-private decomposition with asymmetric routing
	Action fusion	fuse unfactorized multi-view action information directly
	Global only	remove the matched private branch from prediction
	Private only	remove cross-view global evidence from prediction
Objectives	w/o $\mathcal{L}_{\text{rec}}$	remove complete-transition reconstruction constraint
	w/o $\mathcal{L}_{\text{gain}}$	remove matched-private complementarity constraint
	w/o $\mathcal{L}_{\text{cross}}$	remove mismatched-private target-specificity constraint
Routing	Matched $G_v + P_v$ only	remove global components from the other synchronized views
	w/o stop-gradient	replace $\text{sg}(G_v)$ with G_v in private extraction

Table 12: Effect of model capacity on 15-step video reconstruction.

Configuration	Params	PSNR ↑	SSIM ↑	LPIPS ↓
Enlarged single-view backbone ($d = 720$)	580.58M	15.751	0.525	0.295
global-private decomposition	580.61M	**17.009**	**0.575**	**0.269**

struction scores relative to Full CARVE indicate that both cross-scale coverage and amplitude-aware selection contribute to the final model.

Factorization and components. Action fusion removes the explicit global-private factorization and directly uses the multi-view action representation for future prediction. Global-only and Private-only retain one branch of the factorized representation while removing the other from the prediction input. These controls test whether the improvement can be explained by multi-view capacity alone or requires the intended division between reusable and target-specific evidence.

D.2 Cross-View Action Information

We evaluate cross-view complementarity through linear action prediction on two datasets with two and three camera views, respectively. As shown in Table 3, progressively concatenating global components from additional synchronized views improves prediction on both datasets under the same linear-probe protocol. These gains support the interpretation that additional views provide complementary action-relevant information beyond that available from a single view.

D.3 Model Capacity Scaling

To isolate the effect of architectural decomposition from differences in model capacity, we conducted a parameter-matched comparison between the proposed global/private model and a single-view backbone with increased hidden width. The enlarged backbone uses a hidden dimension of 720 and contains $580.58M$ parameters. Under the same counting convention, the decomposed model contains $580.61M$ parameters. Thus, the comparison controls total model capacity while changing the parameterization and information flow. These results indicate that the observed improvement is not explained by a larger parameter budget alone.

E Qualitative Planning Analysis

Figure 5 presents representative successful and failed VP2 planning trajectories. Comparing executed trajectories helps distinguish failures caused by world model prediction from those arising during planning or task evaluation.