Local Motion Matters: A Deconstruct–Recompose Paradigm for Reinforcement Learning Pre-training from Videos

Supplementary Material

7. Details of Pre-training and Fine-tuning

This section provides the detailed optimization objectives used in both the Pre-training and Fine-tuning stages. These objectives correspond directly to the DRP training schedule described in Section 4.4 of our main manuscript. The complete pre-training and fine-tuning algorithms are provided in Algorithm 1 and Algorithm 2, respectively.

7.1. Pre-training Objectives

During pre-training, DRP optimizes two objectives: a Masked AutoEncoder (MAE) reconstruction objective for deconstructing global motion into local motion representations, and a latent dynamics model objective for recomposing local motion representations. The full pre-training stage is summarized in Algorithm 1.

MAE Reconstruction Objective. To learn local motion representations that capture the spatiotemporal relationships among Atomic Actions, we train the Dual-Attention Encoder (DAE) using a Masked Autoencoder (MAE) reconstruction objective. During pre-training, we randomly mask a subset of Atomic Action tokens and require the decoder to reconstruct only the corresponding masked Atomic Actions from the visible tokens:

$$\mathcal{L}_{\text{MAE}} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \|\hat{u}_i - u_i\|_2^2 , \tag{5}$$

where $\mathcal{M}$ denotes the set of masked Atomic Actions, u_i represents the original Atomic Action patch, and $\hat{u}_i$ is the reconstructed patch. This objective encourages the DAE to model the spatiotemporal relationships necessary to infer the masked Atomic Actions.

Latent Dynamics Objective. The latent dynamics model learns to recompose local motion representations through the aggregated action representation a_t^{agg}. Following Dreamer, the latent dynamic model is trained with an image reconstruction term and a KL regularization term:

$$\mathcal{L}_{\text{dyn}} = \mathbb{E}_{q_\theta}\left[\sum_{t=1}^{T}\Big(-\ln p_\theta(o_t \mid z_t) + \beta_z \mathcal{L}_z\Big)\right],$$
$$\mathcal{L}_z = \text{KL}\Big[q_\theta(z_t \mid z_{t-1}, a_{t-1}^{\text{agg}}, o_t) \,\Big\|\, p_\theta(\hat{z}_t \mid z_{t-1}, a_{t-1}^{\text{agg}})\Big]. \tag{6}$$

This objective ensures that the recomposed motion representation acquires dynamic semantics.

Algorithm 1 DRP Pre-training

Initialize parameters ζ of DAE, θ of latent dynamics model, image encoder, and decoder randomly
Load unlabeled video dataset $\mathcal{D}$
for every iteration **do**
  Randomly sample videos $\{o_{1:T}\} \sim \mathcal{D}$
  # Atomic Action Extraction
  Compute optical flow $\{F_{1:T-1}\}$ using Sea-RAFT
  Segment foreground mask M_1 via Grounded SAM
  Track K keypoints $\{p_{1:T}^k\}$ using Co-tracker
  Extract Atomic Action patches $\{u_t^k\}$
  # Local Motion Representation Learning
  Tokenize u_t^k into τ_t^k and prepend '[MAT]'
  Encode tokens via DAE to obtain local motion representations
  # Recompose via Latent Dynamics Model
  Obtain aggregated representation a_t^{agg} from '[MAT]'
  Infer the latent state z_t:

$$z_t \sim q_\theta(z_t \mid z_{t-1}, a_{t-1}^{agg}, o_t).$$

  Update latent dynamic model by minimizing $\mathcal{L}_{dyn}$ in Eq. (6).
end for

7.2. Fine-tuning Objectives

The Fine-tuning stage jointly fine-tunes the DAE and the latent dynamics model, and maps the recomposed motion representations into the downstream agent's action space to accelerate policy learning. The overall objective consists of three components: (1) MAE fine-tuning of the DAE, (2) fine-tuning of the latent dynamics model, and (3) the Action-Specific Dynamics Model objective, as detailed in Algorithm 2.

MAE Fine-tuning for DAE. The DAE is slowly fine-tuned using the same MAE objective as in pre-training:

$$\mathcal{L}_{\text{MAE}}^{\text{ft}} = \mathcal{L}_{\text{MAE}}. \tag{7}$$

This retains the pre-trained transferable local motion representations while allowing the model to gradually incorporate new ones specific to the downstream agent (e.g., rigid-joint motions).

Latent Dynamics Fine-tuning. The pre-trained latent dynamics model is slowly fine-tuned so that the Motion Aggregation Token '[MAT]' can selectively recompose the local motion representations to match the downstream agent:

$$\mathcal{L}_{\text{dyn}}^{\text{ft}} = \mathcal{L}_{\text{dyn}}, \tag{8}$$

with the same form as in Eq. (6).

Action-Specific Dynamics Model Objective. We further introduce an Adapter together with an Action-Specific Dynamics Model to map the recomposed local motion representations into the downstream agent's action space. The Adapter bridges the pre-trained latent dynamics model and the Action-Specific Dynamics Model.

The optimization objective of the Action-Specific Dynamics Model is:

$$\mathcal{L}_{\text{action}} = \mathbb{E}_{q_\phi, q_\theta}\Big[\sum_{t=1}^{T}\Big(-\ln p_\theta(o_t|s_t,c) - \beta_r \ln p_\varphi(r_t|s_t) + \beta_s \mathrm{KL}\left[q_\phi(s_t|s_{t-1},a_{t-1},z_t) \parallel p_\phi(\hat{s}_t|s_{t-1},a_{t-1})\right]\Big)\Big], \tag{9}$$

where s_t is the agent-specific state, z_t is the latent state of the pre-trained latent dynamics model. c is a context variable adopted from IPV [38], computed by encoding randomly sampled frames from the video. This context variable c captures static context information (e.g., backgrounds and textured appearance).

8. Policy Learning

During the fine-tuning stage of DRP, policy learning is performed on top of the Action-Specific Dynamics Model. Following Dreamer, both the actor and critic are optimized entirely using imagined trajectories generated within the latent space of the Action-Specific Dynamics Model. This design enables highly sample-efficient reinforcement learning and allows the agent to fully leverage the transferable motion knowledge acquired during pre-training.

8.1. Latent Imagination Rollouts

Given the current agent-specific latent state s_t and policy π_ψ, the dynamics model recursively produces an imagined trajectory:

$$\{\hat{s}_\tau, \hat{a}_\tau, \hat{r}_\tau\}_{\tau=t}^{t+H},$$

where H denotes the imagination horizon, and τ represents the various time steps in the imagined trajectory. Each imagined transition is generated by:

- sampling an action $\hat{a}_\tau \sim \pi_\psi(\cdot \mid \hat{s}_\tau)$,
- predicting the next latent state via the transition model $p_\phi(\hat{s}_{\tau+1} \mid \hat{s}_\tau, \hat{a}_\tau)$ of Action-Specific Dynamics Model,
- predicting the reward $\hat{r}_\tau$ via $p_\varphi(\hat{r}_\tau \mid \hat{s}_\tau)$.

Algorithm 2 DRP Fine-tuning

Load pre-trained parameters ζ of DAE, θ of latent dynamics model, image encoder and decoder
Initialize parameters ϕ of Action-Specific Dynamics, ω of Adapter, and φ of reward predictors randomly
Initialize parameters ψ of actor $\pi_\psi(a|s)$ and ξ of critic $v_\xi(s)$
Initialize replay buffer $\mathcal{B}$
for every iteration **do**
    # Collect Transitions
    Get state $z_t \sim q_\theta(z_t \mid z_{t-1}, a_{t-1}^{\text{agg}}, o_t)$, $s_t \sim q_\phi\big(s_t \mid s_{t-1}, a_{t-1}, z_t\big)$
    Get action $a_t \sim \pi_\psi(a_t \mid s_t)$
    Add transition $\{o_t, a_t, r_t\}$ to replay buffer $\mathcal{B}$
    # Deconstruct Fine-tuning
    Extract Atomic Actions $\{u_t^k\}$
    Fine-tune the DAE by minimizing $\mathcal{L}_{\text{MAE}}^{\text{ft}}$ in Eq. (7)
    # Recompose Fine-tuning
    Compute aggregated representation a_t^{agg}
    Fine-tune latent dynamics model by minimizing $\mathcal{L}_{\text{dyn}}^{\text{ft}}$ in Eq. (8)
    # Action-Specific Dynamics Model
    Compute latent state z_t of latent dynamics model
    Learn Action-Specific Dynamics Model by minimizing $\mathcal{L}_{\text{action}}$ in Eq. (9)
    # Policy Learning
    Imagine future latent rollouts $\{\hat{s}_\tau, \hat{a}_\tau, \hat{r}_\tau\}_{\tau=t}^{t+H}$ using the Action-Specific Dynamics Model and actor
    Compute λ-return V_τ^λ in Eq. (10)
    Update critic by minimizing $\mathcal{L}_{critic}$ in Eq. (11)
    Update actor by minimizing $\mathcal{L}_{actor}$ in Eq. (12)
end for

These imagined trajectories serve as the sole training data for the actor and critic, reducing reliance on real environment interaction.

8.2. Critic Learning

The critic v_ξ estimates the value of latent states. We use the λ-return to construct multi-step bootstrapped targets:

$$V_\tau^\lambda \doteq \hat{r}_\tau + \gamma \begin{cases} (1-\lambda)\, v_\xi(\hat{s}_{\tau+1}) + \lambda\, V_{\tau+1}^\lambda, & \tau < t+H, \\ v_\xi(\hat{s}_{\tau+1}), & \tau = t+H. \end{cases} \tag{10}$$

The critic is trained to regress the λ-return using a squared loss:

$$\mathcal{L}_{critic}(\xi) = \mathbb{E}\left[\sum_{\tau=t}^{t+H} \frac{1}{2}\left(v_\xi(\hat{s}_\tau) - \mathrm{sg}(V_\tau^\lambda)\right)^2\right], \tag{11}$$

where $\mathrm{sg}(\cdot)$ is the stop-gradient operator.

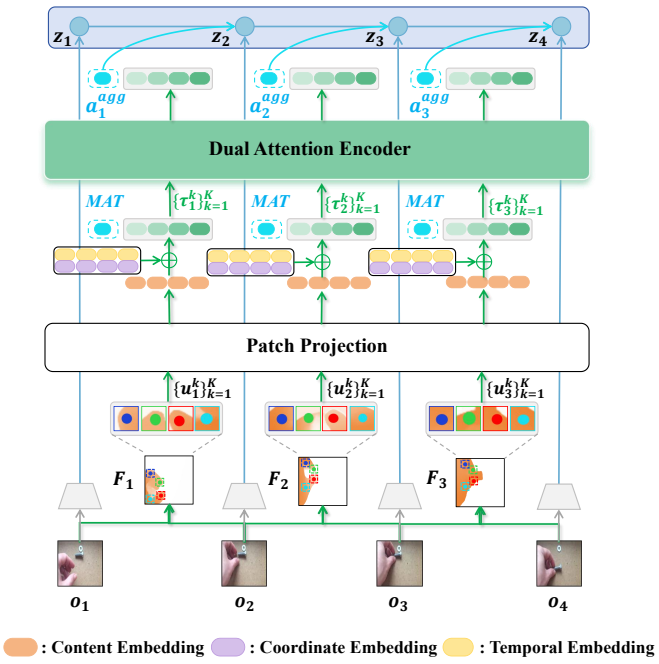

Figure 9. **Illustration of DAE Processing.** For each extracted Atomic Action, we generate a token by combining its Content Embedding with Coordinate and Temporal Embeddings. These tokens, prepended with a learnable '[MAT]' token, are processed by the DAE to produce the aggregated motion representation a^{agg}.

8.3. Actor Learning

The actor is optimized to select actions that maximize the critic's λ-return on imagined trajectories. An entropy regularization term encourages sufficient exploration:

$$\mathcal{L}_{actor}(\psi) = \mathbb{E}\left[\sum_{\tau=t}^{t+H}\left(-V_\tau^\lambda - \eta\,\mathcal{H}[\pi_\psi(\hat{a}_\tau \mid \hat{s}_\tau)]\right)\right], \quad (12)$$

where η controls the entropy weight. A detailed algorithmic description of the policy learning process is provided in the *Policy Learning* block of Algorithm 2.

9. Dual-Attention Formulation of the DAE

This section provides the mathematical formulation of the Dual-Attention Encoder (DAE). The DAE consists of two complementary attention branches: Intra-Frame Attention and Inter-Frame Attention, which models the spatial and temporal relationships among Atomic Actions, respectively.

DAE Token Construction. As shown in Figure 9, for each timestep t, the input to the DAE is constructed from the extracted Atomic Actions. Given a tracked keypoint p_t^k, a local optical flow patch u_t^k is first cropped and passed through a Patch Projection module to obtain a content embedding. This embedding is then combined with a Coordinate Embedding and a Temporal Embedding to yield the final token embedding. To enable aggregation during the Recompose phase, a learnable Motion Aggregation Token '[MAT]' is prepended to each frame's token sequence. Thus, the complete token set at timestep t is: $\mathcal{T}_t = \{[\text{MAT}],\ \tau_t^1,\ \tau_t^2,\ \ldots,\ \tau_t^K\}$. Stacking all tokens yields the matrix: $X_t \in \mathbb{R}^{(K+1)\times d}$, which serves as the input to the Dual-Attention Blocks. The updated '[MAT]' token after Dual-Attention produces the aggregated motion representation a_t^{agg} used in the latent dynamics model.

Intra-Frame Attention. This branch models spatial relationships within each frame. Given X_t, the query/key/value projections are:

$$Q_t = X_t W_Q^{\text{intra}}, \quad K_t = X_t W_K^{\text{intra}}, \quad V_t = X_t W_V^{\text{intra}}, \quad (13)$$

with $W_Q^{\text{intra}}, W_K^{\text{intra}}, W_V^{\text{intra}} \in \mathbb{R}^{d\times d}$. The Intra-Frame Attention output is:

$$\text{IntraAttn}(X_t) = \text{Softmax}\left(\frac{Q_t K_t^\top}{\sqrt{d}}\right) V_t. \quad (14)$$

This allows the model to capture how local motion components interact spatially within the same frame.

Inter-Frame Attention. To capture temporal relationships, Inter-Frame Attention applies causal self-attention across timesteps for each local part k. Given the sequence of features for the k-th token across time:

$$\mathcal{S}^k = \{\tau_1^k, \tau_2^k, \ldots, \tau_T^k\} \in \mathbb{R}^{T\times d},$$

we compute:

$$Q^k = \mathcal{S}^k W_Q^{\text{inter}}, \quad K^k = \mathcal{S}^k W_K^{\text{inter}}, \quad V^k = \mathcal{S}^k W_V^{\text{inter}}, \quad (15)$$

where $W_Q^{\text{inter}}, W_K^{\text{inter}}, W_V^{\text{inter}} \in \mathbb{R}^{d\times d}$. Causal masking is applied using $M_{\text{causal}} \in \mathbb{R}^{T\times T}$ to prevent attending to future timesteps. The Inter-Frame Attention output is:

$$\text{InterAttn}(\mathcal{S}^k) = \text{Softmax}\left(\frac{Q^k (K^k)^\top}{\sqrt{d}} + M_{\text{causal}}\right) V^k. \quad (16)$$

Dual-Attention Block. Each DAE block integrates the two attention mechanisms sequentially with residual connections and Layer Normalization. Let Z denote the input tensor to the block. The update process is formulated as:

$$Z' = Z + \text{IntraAttn}(Z) \quad (17)$$

$$Z'' = Z' + \text{InterAttn}(Z') \quad (18)$$

$$Z_{\text{out}} = Z'' + \text{MLP}(\text{LN}(Z'')), \quad (19)$$

Figure 10. **Examples of DMControl Remastered.**

where Eq. (17) applies Intra-Frame Attention for each frame, and Eq. (18) applies Inter-Frame Attention for each token sequence across time. This sequential design ensures that the model captures both spatial consistency and temporal dynamics effectively.

Discussion. Intra-Frame Attention captures spatial relationships among Atomic Actions within each frame, while Inter-Frame Attention captures the temporal relationships of each local motion trajectory. Together, these two attention mechanisms enable the DAE to learn transferable local motion representations.

10. Experimental Details

10.1. Pre-training Dataset

Following IPV, we adopt the Something-Something-V2 (SSV2) dataset for unsupervised pre-training. SSV2 contains more than 220K videos of humans interacting with everyday objects, covering a wide range of motion patterns and object manipulations. After filtering out videos with fewer than 25 frames, we retain 162K videos for pre-training. This large-scale and diverse dataset provides a rich data foundation for learning transferable local motion representations.

10.2. Benchmark Environments

10.2.1. DMControl Remastered

DMC Remastered (DMCR) is a variant of the DeepMind Control Suite featuring randomly generated graphics, emphasizing visual diversity. In each episode, seven factors affecting visual conditions are randomly sampled, including background, floor texture, robot body color, target color, reflectance, camera position, and lighting. Following IPV, we evaluate the same three locomotion tasks: "Walker Run", "Hopper Stand", and "Cheetah Run", as shown in Figure 10. Each episode lasts 1000 steps with an action repeat of 2, and rewards range from 0 to 1. All methods are trained for 1.02M environment steps to ensure fair comparison.

10.2.2. Meta-World

Meta-World consists of 50 diverse robotic manipulation tasks. Following IPV and PreLAR, we evaluate six same tasks: "Dial Turn", "Drawer Open", "Lever Pull", "Button Press Topdown Wall", "Reach", and "Door Lock", as shown in Figure 11. Each episode contains 500 steps with no action repetition, and the action space is 4-dimensional. Rewards range from 0 to 10. All methods are trained for 255K environment steps.

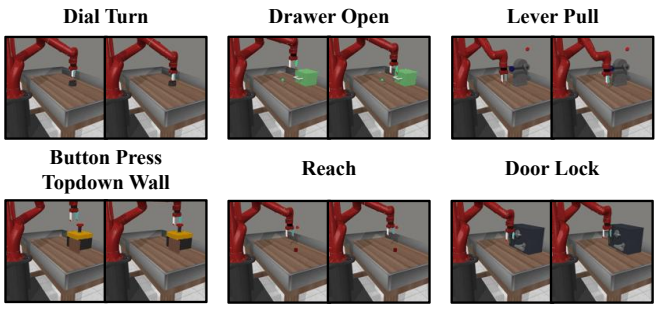

Figure 11. **Examples of Meta-World.**

11. Additional Ablation Studies

We further evaluate the key design choices of DRP, including the two forms of positional embeddings (Coordinate Embedding and Temporal Embedding) used in the Dual-Attention Encoder (DAE) and the Adapter architecture.

11.1. Ablation on Positional Embeddings in DAE

When encoding Atomic Actions, the DAE augments the content embedding with two forms of positional information: (1) a *coordinate embedding* that specifies where the flow patch is located, and (2) a *temporal embedding* that indicates the timestep at which it occurs. These two positional cues are critical for enabling the DAE to capture the spatial and temporal relationships among Atomic Actions.

To evaluate the importance of these two positional encodings, we individually remove one type of positional encoding from the DRP. As shown in Figure 12, "DRP w/o Coord. Emb." denotes the variant without the Coordinate Embedding, and "DRP w/o Temp. Emb." denotes the variant without the Temporal Embedding. The experimental results indicate that removing either positional embedding leads to significant performance degradation. This suggests that spatial position and temporal order information are critical for modeling the spatiotemporal relationships among atomic actions, and are therefore essential for learning transferable local motion representations.

11.2. Ablation on the Adapter Design

During the fine-tuning stage, the Adapter serves as a key bridge connecting the pre-trained latent dynamics model and the Action-Specific Dynamics Model. To evaluate the necessity of the Adapter, we replace it with a simple Multi-Layer Perceptron (MLP) projection layer.

As shown in the Figure 13, the variant where the Adapter is replaced by an MLP projection layer, denoted as "DRP

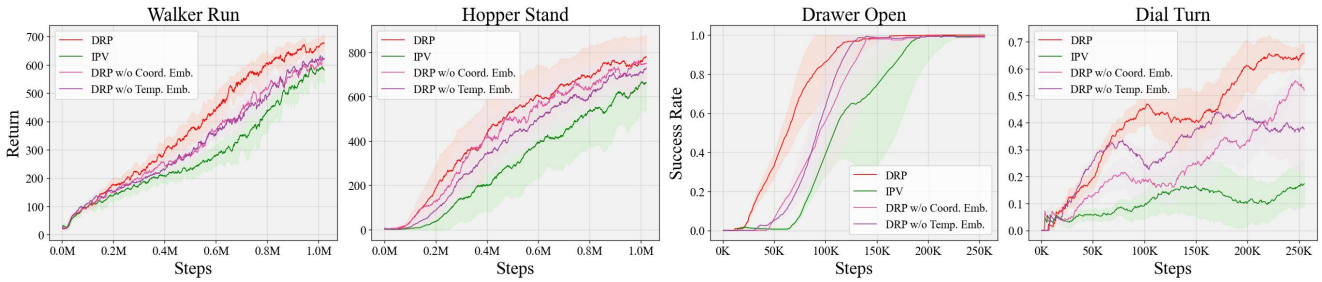

Figure 12. **Ablation study on Positional Embeddings in DAE.** We analyze the necessity of the two positional encodings within the DAE module. The results compare our full model DRP against variants where either the Coordinate Embedding (w/o Coord. Emb.) or the Temporal Embedding (w/o Temp. Emb.) is removed on the "Walker Run" and "Hopper Stand" tasks from DMCR, and the "Drawer Open" and "Dial Turn" tasks from Meta-World.

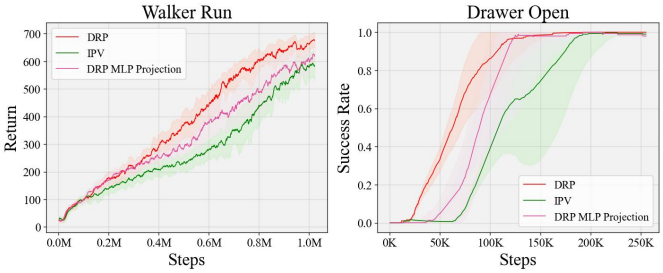

Figure 13. **Ablation study on the Adapter design.** We compare DRP with a variant where the Adapter is replaced by a simple Multi-Layer Perceptron (MLP) projection layer (denoted as "DRP MLP Projection") on the "Walker Run" task from DMCR and the "Drawer Open" task from Meta-World.

MLP Projection," results in a significant performance drop. This indicates that the MLP's limited expressive capacity prevents it from effectively mapping the local motion representations to the agent-specific action space. This result highlights the critical role of the Adapter in effectively aligning the transferable local motion representation space with the agent-specific action space.

12. Additional Visualization and Analysis

In this section, we provide additional qualitative analyses to further validate the effectiveness of our proposed local motion representations. We present visualizations spanning three aspects: source-domain video prediction, image reconstruction, and t-SNE analysis of local motion representations.

12.1. Source-Domain Video Prediction

In addition to Figure 8(a) in the main manuscript, we provide more open-loop video prediction results on the source domain SSV2 test set to further evaluate the effectiveness of the learned local motion representations. Given initial video frames, the latent dynamics model performs open-loop predictions conditioned on the local motion representations.

As shown in Figure 15, our model generates more accurate future frame predictions compared to the baselines. Specifically: (1) **Hand Motion:** DRP accurately captures the hand's movement and its subsequent exit from the frame, whereas the predictions from IPV and PreLAR fail to exhibit this motion. (2) **Relative Positional Relationships:** In the process of pushing an object toward a wall corner, DRP accurately predicts the relative positional relationship as the object approaches the wall corner. In contrast, the subsequent predictions of IPV and PreLAR fail to exhibit this dynamic change in relative position. Furthermore, the wall corner disappears in IPV's later predictions, while PreLAR fails to capture the corner throughout the entire sequence. (3) **Object Removal:** DRP accurately predicts the process of the hand and cup being removed from the view. Conversely, IPV and PreLAR incorrectly predict the cup as still visible. (Note: Since the object under the cup is occluded in the initial frames, DRP is unable to predict this purple object after the cup is removed, which is a reasonable and expected behavior). (4) **Complex Manipulation:** DRP successfully captures the complex forward and backward motion in the plugging task. Both IPV and PreLAR fail to predict the backward movement of the arm. Furthermore, arm deformation is observed in PreLAR's predictions.

These extended results further confirm that our learned local motion representations effectively capture robust and transferable motion patterns.

12.2. Image Reconstruction Analysis

To further investigate the advantages of our proposed local motion representations, we visualize the image reconstruction results generated by the pre-trained latent dynamics model. As shown in Figure 16, compared to the baseline model, our model's latent state captures finer visual details. Specifically: **(1) Visual accuracy:** DRP accurately preserves the specific shape of the hand, whereas

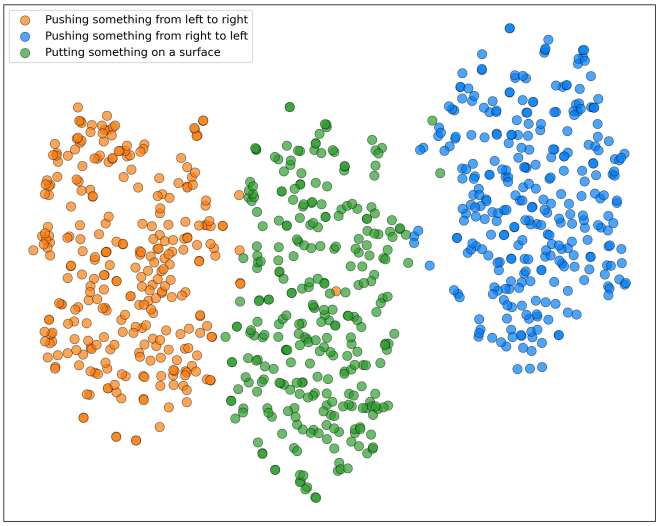

Figure 14. **t-SNE Visualization of Composed Local Motion Representations.** We perform t-SNE visualization on the composed local motion representations corresponding to three manipulation behaviors that share overall visual similarity ("Pushing something from left to right", "Pushing something from right to left", and "Putting something on a surface").

IPV generates a blurry reconstruction of hands. Similarly, DRP clearly captures the shape of the manipulated object, while PreLAR produces a blurry appearance. **(2) Geometric accuracy:** DRP correctly maintains the object's length, whereas IPV shortens it in the reconstruction. Moreover, DRP successfully captures the object's orientation, whereas PreLAR fails to reproduce the correct orientation.

These results demonstrate that the latent dynamics model based on local motion representations is better at capturing motion-relevant visual features.

12.3. t-SNE Analysis of Local Motion Representations

To further investigate the effectiveness of our proposed local motion representations, we perform t-SNE visualization on the recomposed local motion representations during pre-training, as shown in Figure 14.

Specifically, we sample video clips of length $T = 25$ from the SSV2 dataset, which are respectively annotated with three different labels: "Pushing something from left to right", "Pushing something from right to left", and "Putting something on a surface". We then visualize the recomposed local motion representations of these clips using t-SNE, as shown in Figure 14. Notably, no video labels are utilized during the pre-training stage. Although these three manipulation behaviors exhibit overall visual similarity (especially the two pushing actions in opposite directions), their corresponding recomposed local motion representations form clearly distinct clusters in the latent space.

Table 2. **Hyperparameter analysis on the Keypoints number K.**

Keypoints number K	$K = 8$	$K = 16$	$K = 32$
Walker Run	646 ± 45	$\mathbf{681 \pm 39}$	667 ± 43
Hopper Stand	775 ± 123	796 ± 114	$\mathbf{799 \pm 105}$

Table 3. **Hyperparameter analysis on the MAE mask ratio ρ.**

Mak Ratio ρ	$\rho = 0.3$	$\rho = 0.5$	$\rho = 0.7$
Walker Run	648 ± 42	$\mathbf{681 \pm 39}$	660 ± 47

This visualization result strongly demonstrates that our learned local motion representations are capable of effectively capturing and distinguishing the agent's different motion patterns.

13. Hyperparameter Analysis

Apart from the newly introduced hyperparameters: the number of tracked keypoints K, the MAE mask ratio ρ, the local optical flow patch size P, and the number of Dual-Attention Blocks L, we adopt the same hyperparameter settings as IPV and PreLAR to ensure a fair comparison. The complete set of hyperparameters is provided in Table 5.

13.1. Number of Tracked Motion Keypoints

This section analyzes the selection of the hyperparameter K, the number of motion keypoints tracked in our Atomic Action Extraction module. We investigate the influence of K by varying its value across $K \in \{8, 16, 32\}$. These hyperparameter experiments are conducted on the DMCR tasks "Walker Run" and "Hopper Stand", with the results summarized in Table 2.

As shown in the results, selecting an insufficient number of keypoints ($K = 8$) leads to suboptimal performance. This is likely because a small K fails to adequately capture all meaningful local motion components that constitute the global motion. Conversely, increasing the number of tracked keypoints from $K = 16$ to $K = 32$ yields minimal performance improvement, and even results in a slight decrease on the "Walker Run" task. Crucially, this increase significantly improves the overall computational cost.

Therefore, we select $K = 16$ as the default number of tracked motion keypoints in this work, as it strikes the balance between performance and computational cost.

13.2. MAE Mask Ratio

We investigate the influence of the MAE masking ratio ρ by evaluating $\rho \in \{0.3, 0.5, 0.7\}$. These hyperparameter experiments are conducted on the DMCR tasks "Walker Run", with the results summarized in Table 3.

Table 4. **Hyperparameter analysis on the local optical flow patch size P.**

Patch Size P	$P = 12$	$P = 16$	$P = 24$
Walker Run	649 ± 52	$\mathbf{681 \pm 39}$	652 ± 32

A low masking ratio ($\rho = 0.3$) reduces the reconstruction difficulty, which may fail to learn robust spatiotemporal relationships. Conversely, an excessively high masking ratio ($\rho = 0.7$) leads to excessive information loss, making effective reconstruction impossible. As shown in Table 3, a ratio of 0.5 achieves the optimal performance.

13.3. Local Patch Size

We evaluate the influence of the local optical flow patch size P by varying its value across $P \in \{12, 16, 24\}$. These hyperparameter experiments are conducted on the DMCR task "Walker Run", with the results summarized in Table 4.

A smaller patch size ($P = 12$) results in an excessively localized receptive field. Such localized features tend to exhibit high variance when capturing motion patterns and are highly sensitive to small fluctuations, leading to representations that lack robustness. Conversely, an excessively large patch size ($P = 24$) implies insufficient deconstruction, where a single patch may include multiple distinct local motion patterns, thereby hindering effective cross-domain transfer. The experimental results demonstrate that $P = 16$ balances precise local motion capture and representation robustness.

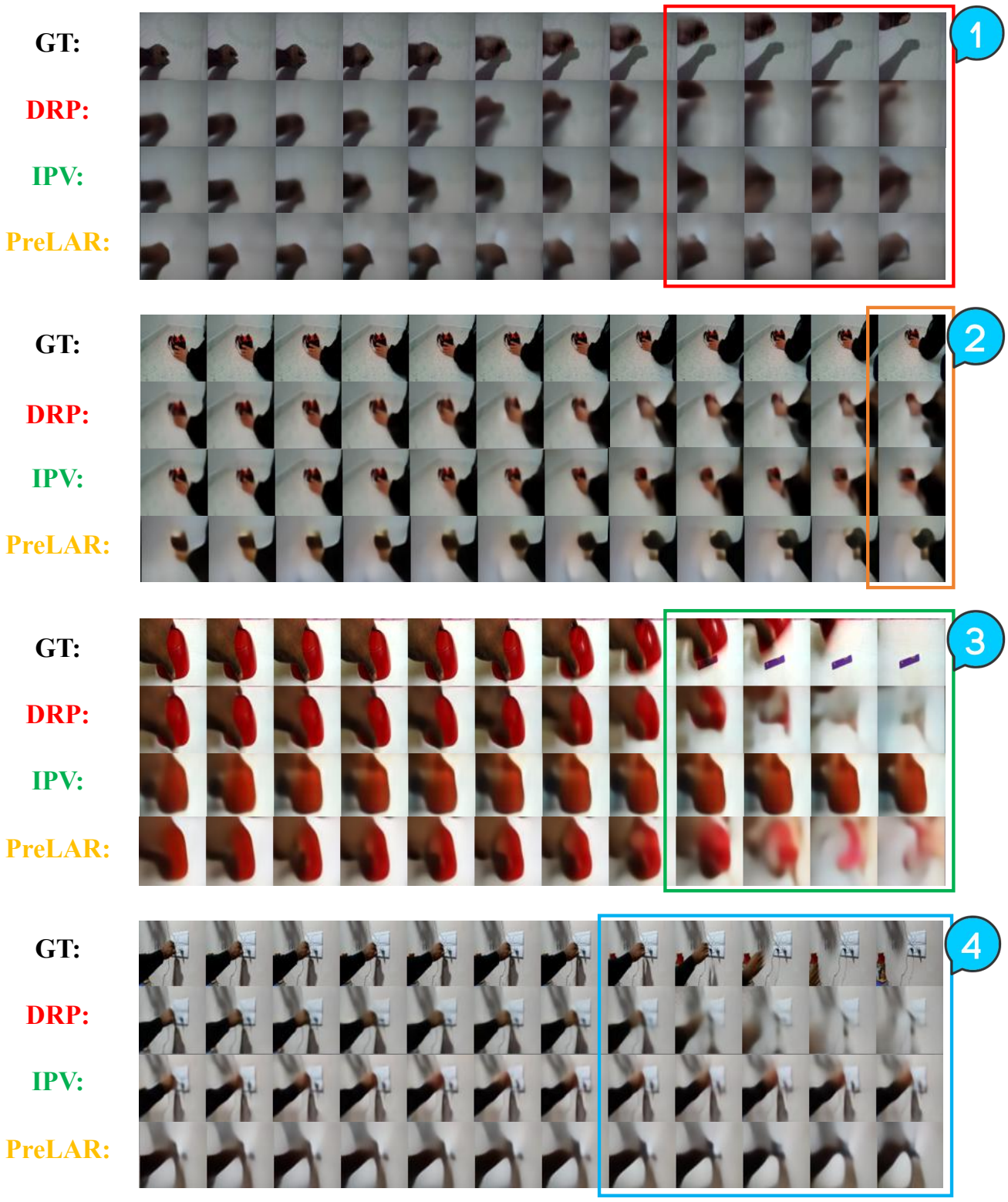

Figure 15. **Open-loop video prediction on the source domain (SSV2) test set.** We compare the prediction results of our method (DRP) with IPV and PreLAR, where "GT" denotes Ground Truth.

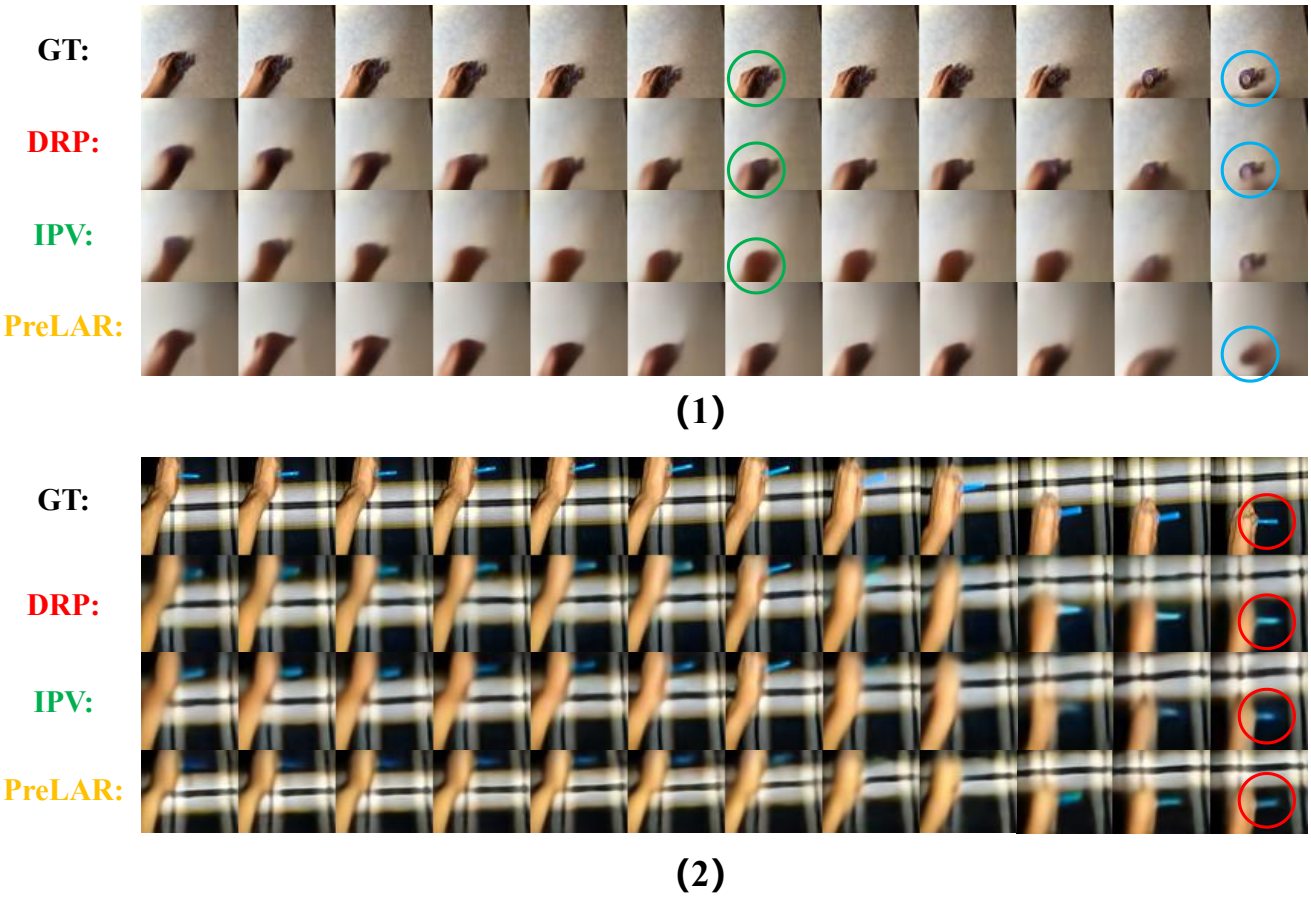

Figure 16. **Image reconstruction visualization from the latent dynamics model.** We compare the reconstruction results of our method (DRP) with IPV and PreLAR, where "GT" denotes Ground Truth.

Table 5. **Hyperparameters in our experiments.** We use the same hyperparameter setting as IPV.

	Hyperparameter	Value
Pre-training from Videos	Image size	$64 \times 64 \times 3$
	Image preprocess	Linearly rescale from [0, 255] to [-0.5, 0.5]
	Video segment length T	25
	KL weight β_z	1.0
	Optimizer	Adam
	Learning rate	3×10^{-4}
	Batch size	16
	Training iterations	6×10^5
	Number of tracked keypoints K	16
	Local optical flow patch size P	16
	MAE mask ratio ρ	0.5
	Number of Dual Attention Blocks L	6
Fine-tuning with MBRL	Observation size	$64 \times 64 \times 3$
	Observation preprocess	Linearly rescale from [0, 255] to [-0.5, 0.5]
	Trajectory segment length T	50
	Random exploration	5000 environment steps for Meta-world 1000 environment steps for DMCR
	Replay buffer capacity	10^6
	Training frequency	Every 5 environment steps
	Action-conditional KL weight β_s	1.0
	Representative reward predictor weight β_r	1.0
	Intrinsic reward weight λ	1.0 for Meta-World 0.1 for DMCR
	Imagination horizon H	15
	Discount γ	0.99
	λ-target discount	0.95
	Entropy regularization η	1×10^{-4}
	Batch size	50 for Meta-World 16 for DMCR
	World model optimizer	Adam
	World model learning rate	3×10^{-4}
	Actor optimizer	Adam
	Actor learning rate	8×10^{-5}
	Critic optimizer	Adam
	Critic learning rate	8×10^{-5}
	Evaluation episodes	10
	Number of tracked keypoints K	16
	Local optical flow patch size P	16
	MAE mask ratio ρ	0.5
	Number of Dual Attention Blocks L	6