Supplementary Material for “Simoun: Synergizing Interactive Motion-appearance Understanding for Vision-based Reinforcement Learning”

Yangru Huang1, Peixi Peng2,4 *, Yifan Zhao3, Yunpeng Zhai3, Haoran Xu3,4, Yonghong Tian1,2,4 *
1 School of Computer Science, Peking University
2 School of Electronic and Computer Engineering, Peking University
3 School of Intelligent Systems Engineering, Sun Yat-sen University
4 Peng Cheng Laboratory
yruhuang@stu.pku.edu.cn, {ppeng, zhaoyf, ypzhai, yhtian}@pku.edu.cn, xuhr@mail.sysu.edu.cn

The contents of the supplementary material are organized as follows:

- Sec. 1 provides additional implementation details and experiment setup, including network architecture, RL environment description, and hyperparameter settings.
- Sec. 2 demonstrates the effectiveness of Simoun in more complex and challenging environments.
- Sec. 3 conducts various further ablations, including the effects of different losses (Sec. 3.1), the influence of input frame stack size for motion modeling (Sec. 3.2) and the parameter β (Sec. 3.3), performance with alternative fusion schemes such as bilinear fusion (Sec. 3.4), impacts of different augmentation methods for the appearance path in Simoun (Sec. 3.5), comparison between the proposed structural interactive module with vanilla cross-attention mechanism (Sec. 3.6), comparison with other intrinsic rewards (Sec. 3.7) and other “dual-path” networks in video recognition (Sec. 3.8), results on more environments (Sec. 3.9) including DrawerWorld benchmark and Atari-100k tasks, and the analyse of motion-appearance embedding space (Sec. 3.10).
- Sec. 4 analyses the limitation of Simoun in its current form and suggests possible directions for future studies.

1. Additional Implementation Details

Network Architecture Fig. A1 gives the detailed network architecture used in our approach. For the motion/appearance paths, the observation encoder is implemented with four convolutional (Conv) layers. Each Conv layer has 3 × 3 kernel size with 32 output channels, followed by a ReLU activation. For DMControl, all layers have a stride of 1 except for the first Conv layer, which has a stride of 2. However, for CARLA, the stride of all layers is set to 2. After the last Conv layer, a fully connected (FC) layer with layer normalization (LN) is used to reduce the dimension of features to 50. The weight matrices of the Conv and FC layers are initialized with orthogonal initialization [9] and the biases are set to zero. For DMControl, we further apply a Tanh layer after the LN layer following previous methods [16]. In addition, the input sizes of the two environments are also different. For DMControl, the input is rendered at 100 × 100 and cropped to 84 × 84. For Carla, the input is the horizontally concatenated images from three cameras on the roof of vehicles, which has a size of 84 × 252.

Both the action-conditioned transition model and the reward prediction model have a sequence of [FC, LN, ReLU, FC] layers. Differently, the former outputs the predicted latent vector of the next observation, and the latter outputs a single value of the predicted reward of the next observation. For the critic network, clipped double Q-learning [12, 3] is adopted for stability. Each critic network includes a 3-layer MLP with ReLU activations. The target critic network is updated by the critic network using momentum updating. For the actor network, a 3-layer MLP is employed similar to the critic network but outputs mean and covariance of the diagonal Gaussian representing the policy. We set the hidden dimension of the actor and critic MLPs to 1024.

Environment Details We use two common RL environments (DMControl and CARLA) in our experiments. Deepmind control (DMControl) suite is a physics-based simulation of Reinforcement Learning environments, based on MuJoCo physics [11]. Our setup on DMControl is similar to [16]. Six tasks are used for evaluation as shown in Fig. A2, including cartpole-swingup, reachesay, cheetah-run, walker-walk, finger-spin, and ball-in-cup-catch. CARLA is a simulator for autonomous driving
research. Following DBC [17], we adopt CARLA with version 0.9.6. The goal of the agent in CARLA is to drive on Highway 8 located in Town 4 and cover the maximum possible distance in 1000 time steps while avoiding collisions with 20 other moving vehicles. This will be accomplished under clear noon weather conditions. As shown in Fig. A3 (b), the RGB observation with $84 \times 252$ is obtained from three attached cameras. CARLA’s output action is a 2D vector that includes both thrust (where braking is represented as negative thrust) and steering values. The reward function is designed following [17]:

$$r_t = v_{ego} \hat{u}_{highway} \cdot \Delta t - \lambda_c \cdot collision - \lambda_s \cdot |steer|$$

(1)

where $v_{ego}$ is the velocity vector of the ego vehicle, $\hat{u}_{highway}$ is the highway’s unit vector, and $\Delta t$ is the simulation time discretization. The first term aims to encourage progression along the highway as long as possible. Meanwhile, the last two terms penalize collision and excessive steering. $\lambda_c = 10^{-4}$ and $\lambda_s = 1$ are trade-off coefficients. The action repeat is set to 4 for all agents.

**Hyperparameter Settings** At the beginning of the training process, the agent acquires 1000 transitions for DMControl and 100 transitions for CARLA while following a random policy. Afterward, it gathers additional transitions.
Based on the learned policy. These collected transitions are saved in a replay buffer, and the training batch size is configured to 128. Other detailed hyperparameters such as learning rate and update frequency are listed in Table A2. During the evaluation, the agents utilize the mean policy action obtained from the trained actor network. Each agent is evaluated once every 20,000 training environment steps, and the evaluation process is repeated for 20 episodes.

The components of prior state-of-the-art algorithms

All methods in Table 1&2 in the main paper use SAC as the base. The components are summarized in below table. We clarify that ①-④ aren't merely auxiliary but strategically selected as a part of Simou to make the best of Motion-Appareance modeling for better performance. We have also ablated ⑤-⑥ in Fig. A5.

<table>
<thead>
<tr>
<th>Component</th>
<th>Simou</th>
<th>CARLA</th>
<th>Cheetah-run Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAC</td>
<td>Max</td>
<td>Max</td>
<td>Max</td>
</tr>
<tr>
<td>Dreamer</td>
<td>Max</td>
<td>Max</td>
<td>Max</td>
</tr>
<tr>
<td>CURL</td>
<td>Max</td>
<td>Max</td>
<td>Max</td>
</tr>
<tr>
<td>DPQ</td>
<td>Max</td>
<td>Max</td>
<td>Max</td>
</tr>
<tr>
<td>SVQA</td>
<td>Max</td>
<td>Max</td>
<td>Max</td>
</tr>
<tr>
<td>Play-Virtual</td>
<td>Max</td>
<td>Max</td>
<td>Max</td>
</tr>
</tbody>
</table>

Table A2. Hyperparameters used in DMControl and CARLA.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>DMControl</th>
<th>CARLA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init steps</td>
<td>h1000</td>
<td>h100</td>
</tr>
<tr>
<td>Input dimension</td>
<td>3 × 84 × 84</td>
<td>3 × 84 × 252</td>
</tr>
<tr>
<td>Action repeat</td>
<td>2(finger)</td>
<td>4</td>
</tr>
<tr>
<td>Discount factor λ</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>Number of training steps</td>
<td>500,000</td>
<td>100,000</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
<td>Adam</td>
</tr>
<tr>
<td>Actor learning rate</td>
<td>1e-3</td>
<td>1e-3</td>
</tr>
<tr>
<td>Critic learning rate</td>
<td>1e-3</td>
<td>1e-3</td>
</tr>
<tr>
<td>Gradual learning rate in SAC</td>
<td>1e-4</td>
<td>1e-3</td>
</tr>
<tr>
<td>Batch size</td>
<td>128</td>
<td>128</td>
</tr>
<tr>
<td>Actor update frequency</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Critic target update frequent</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Data augmentation</td>
<td>Random Conv</td>
<td>Random Conv</td>
</tr>
<tr>
<td>Intrinsic reward temperature</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>Intrinsic reward decay weight λ</td>
<td>2e-5</td>
<td>2e-5</td>
</tr>
</tbody>
</table>

Table A3. Performance on more complex environments. \( \Delta \) means the degree of performance degradation, and \( \downarrow \) denotes that smaller values are better results. The results of cheetah-run task in DMControl benchmark are reported.

<table>
<thead>
<tr>
<th>Task</th>
<th>Original</th>
<th>Video Hard</th>
<th>( \Delta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simou</td>
<td>693 ±53</td>
<td>422 ±46</td>
<td>271</td>
</tr>
<tr>
<td>Latent flow</td>
<td>610 ±38</td>
<td>232 ±35</td>
<td>378</td>
</tr>
<tr>
<td>CARLA</td>
<td>Clear sky</td>
<td>Variable weather</td>
<td>( \Delta )</td>
</tr>
<tr>
<td>Simou</td>
<td>281 ±30</td>
<td>213 ±24</td>
<td>68</td>
</tr>
<tr>
<td>Latent flow</td>
<td>182 ±11</td>
<td>89 ±13</td>
<td>93</td>
</tr>
</tbody>
</table>

2. Performance in Challenging Environments

Due to the complex nature of visual observations, it is crucial for vision-based RL agents to fit diverse complicated scenes. To verify such an ability of Simou, we compare it with the latent flow model under two complex training environment settings on DMControl and CARLA. For DMControl, as shown in Fig. A4 (a), the original environment background is static and clean. To increase the task difficulty, we choose the video-hard mode demonstrated in Fig. A4 (b) to train the agents, in which the background is replaced with dynamic natural videos. Similarly, we use a miscellaneous weather setting on CARLA to train the agents, as shown in Fig. A4 (d). The CARLA environment has been modified to include various types of distractors such as sun, rain, shadows, clouds, and more, which are representative of different weather conditions. These weather conditions are dynamically assigned for each episode and may even change during the course of the same episode. Table A3 shows the performance under these challenging environments. It is evident that the performance of Simou drops less compared with the latent flow model. Although latent flow explicitly models motion information by computing the difference of latent spatial vectors, exposure to complex environments with drastically changed appearance results in severely decreased performance. In contrast, Simou is less affected. The results indicate that Simou can handle complex environments due to the complementarity of the motion and appearance paths, which helps to ignore unwanted distracting elements.

3. Further Ablations

3.1. Effectiveness of Different Losses

As introduced in the main manuscript, several targeted loss terms are deliberately designed in Simou as:

$$L = L_{\text{tran}} + L_{\text{con}} + L_{\text{app}} + L_{\text{st}} + L_{\text{r}},$$

(2)

where the first three terms are responsible for learning motion, appearance, and reward-related features, respectively. In order to verify the effectiveness of these three loss terms, we conduct experiments on CARLA by removing one loss term and keep the other two untouched. Results are shown in Fig. A5. Several observations can be made: 1) The model without \( L_{\text{tran}} \) gets obvious performance degradation. This indicates the importance of the underlying environment dynamics for motion modeling. Without the explicit transition loss, the motion-related information cannot be learned.
2) The contrast loss $L_{con}$ is constructed to force different perspectives of the same observation to approach each other in latent space so as to learn good appearance representation. The model without $L_{con}$ also bears performance degradation. However, the performance drop is not as severe as removing $L_{tran}$. Such a phenomenon indicates that motion modeling constraint is more essential than appearance modeling constraint, while both can improve performance if introduced. 3) $L_{re}$ contributes positively to the performance. The reason is that $L_{re}$ forces the model to pay more attention to reward-related information. Conversely, removing $L_{re}$ while keeping the rest loss terms would make the model excessively describe the observation with detrimental features that are not relevant to the reward.

### 3.3. Influence of $\beta$

We report results on CARLA using fixed $\beta$ and give the learned $\beta$ values along training in the Table A4. It is evident from the results that adaptive learning of $\beta$ yields the maximum accumulated reward. As the number of training
3.4. Performance with Alternative Fusion Schemes

After acquiring the structure-enhanced motion and appearance features $f^m_t$ and $f^a_t$, they are fused by a fusion function. To test the effect of different fusion schemes, we ablate two alternatives:

**Concatenation fusion**: This is the scheme used in the main manuscript for model simplicity. Concatenation fusion stacks $f^m_t$ and $f^a_t$ across the feature channels. The process of concatenation does not establish a clear correspondence between $f^m_t$ and $f^a_t$, as it delegates the task of modeling their relationship to the subsequent layers.

**Bilinear fusion**: Bilinear fusion computes a matrix outer product of the two features. Specifically, given $f^m_t$ and $f^a_t$, bilinear fusion is calculated as:

$$f_t = f^m_t W f^a_t + b,$$

where $b$ is an additive bias and $W$ is a learnable weight matrix. By employing this fusion scheme, the model becomes capable of capturing intricate relationships between the two features, leading to the creation of a more powerful feature.

Experiment results of the two fusion schemes on CARLA are shown in Fig. A7. As we can see, both schemes are able to achieve comparable performance. However, bilinear fusion behaves more stable with less performance variation. Meanwhile, the episode returns at the last steps are slightly better than concatenation. The results indicate that Simoun can further benefit from complex fusion schemes of motion and appearance features. This fact provides Simoun with extra design space for deciding proper fusion methods. However, choosing the best fusion method is beyond the scope of this manuscript, and therefore we only explore the selected two schemes described above.

3.5. Impact of Different Augmentation Methods

In Simoun we adopt an unsupervised contrastive loss for appearance modeling. The contrastive loss works by constructing positive sample pairs using data augmentations. Here we investigate the effectiveness of various data augmentation methods, including a weak augmentation random shift [16], a strong augmentation random convolution [5], and a mixed augmentation. Random shift pads on each side of the input image and randomly crop the image to its original size. Random convolution augments the image color by passing the input observation through a random convolutional layer. Mixed augmentation adopts both random shift and random convolution by selecting only one of them at each time. In previous studies SECANT [3] and SVEA [4], the researchers show that strong augmentation is inherently non-deterministic, resulting in lower performance and training divergence. In contrast, as shown in Fig. A8, Simoun with strong augmentation can still achieve improved performance. The results indicate that the dual-path architecture of Simoun is able to alleviate the non-deterministic problem, where the motion path is not affected by augmentation and therefore can stabilize the state representation.

3.6. Comparison with Other Interactive Modules

The structural interactive module of Simoun leverages the complementarity between the two network paths by extracting latent motion-appearance structures. The key elements of the structural interactive module are the inter-frame attention and spatial gating operation for motion and appearance structure extraction, respectively. To verify the
effectiveness of these two elements, we compare the structural interactive module with a vanilla cross-attention module commonly used for cross-path interaction.

Specifically, as shown in Fig. A9 (b), the inputs of the vanilla cross-attention module are motion feature map $F^m_\ell$ and appearance feature map $F^a_\ell$. A cross-path attention map $X$ can be obtained as:

$$X = \sigma(Q^a_\ell^T K^m_\ell),$$

where $Q^a_\ell$ denotes a new feature map generated by feeding $F^a_\ell$ to a convolution layer and $K^m_\ell$ is obtained from $F^m_\ell$ similarly. $\sigma$ denotes the Softmax function. Then $X$ is used to modulate and update features from both paths:

$$F^m_\ell = F^m_\ell + V^m_\ell \cdot X, \quad F^a_\ell = F^a_\ell + V^a_\ell \cdot X,$$

where $V^m_\ell$ and $V^a_\ell$ are features obtained by passing $F^m_\ell$ and $F^a_\ell$ to another two convolutional layers, respectively.

Experiment results of the two modules on CARLA are shown in Fig. A10. As can be seen, the performance of cross-attention is rather poor, which indicates that the motion features and appearance features learned by the two paths should not be aligned directly. In contrast, the two kinds of features in the structural interactive module are excavated in a complementary manner by focusing on their own characteristics, which greatly improves feature quality.

3.7. Comparison with Other Intrinsic Rewards

Intrinsic reward mechanisms play a pivotal role in enhancing the learning capabilities of agents, especially in challenging environments. When compared with previous intrinsic rewards methods ICM [8] and FICM [15] in a direct and fair manner, our approach has obvious advantages as shown in Fig. A11(a). This indicates that our curiosity module is able to foster a more effective learning process by providing incentives for the agent to investigate states characterized by discrepancies between motion and appearance.

3.8. Compare to “Dual-path” Networks

Here we also compare our method with previous “Dual-path” Networks in video recognition with two-stream fu-
3.9. Results on More Environments

In our study, outcomes from both the DrawerWorld benchmark [13] and Atari-100k tasks [7] are presented in Table A5 and Table A6, respectively. The DrawerWorld benchmark, adapted from Meta World, focuses on manipulation tasks in grid settings and assesses performance in textured environments. Here we specifically utilize fabric textures for testing. Our chosen metric for evaluation is the success rate, quantified as the ratio of successful endeavors to open or close a drawer over a total of 100 attempts. As evident from Table A5, our methodology, Simoun, consistently outperforms the SPR baseline in the DrawerWorld benchmark. Regarding the Atari framework, it is characterized by its pixelated 2D visuals and a set of discrete player actions. The results in Table A6 underscore the efficacy of our approach in the Breakout and Hero tasks.

### Table A6. The results on Atari-100k tasks.

<table>
<thead>
<tr>
<th>Atari100k</th>
<th>SPR</th>
<th>Simoun</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breakout</td>
<td>17.1</td>
<td>20.7</td>
</tr>
<tr>
<td>Hero</td>
<td>7019.2</td>
<td>6973.4</td>
</tr>
</tbody>
</table>

### Table A7. The discrepancy of $|f'^n_t|$ and $|f^{n-1}_{t-1} - f^{n-2}_{t-2}| + |f^n_t - f^{n-1}_t|$ embedding space.

<table>
<thead>
<tr>
<th>Measurements</th>
<th>Initial</th>
<th>Trained</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutual Information</td>
<td>0.0059</td>
<td>0.1700</td>
</tr>
<tr>
<td>Pearson Correlation Coefficient</td>
<td>0.0081</td>
<td>0.0558</td>
</tr>
</tbody>
</table>

3.10. The Analyse of Embedding Space

To assess the alterations in the embedding space of motion and appearance paths, we employ two prevalent metrics: mutual information and pearson correlation coefficient. Firstly, we find that motion-appearance paths are interactive and they do have a closer (without explicitly constraining) embedding space with increased correlation after training, see Table A7. Secondly, the notion that the difference in image space closely aligns with the difference in feature space is an interesting constraint that leads to a robust flat embedding space which approximately ensures Lipschitz continuity [6]. Simoun happens to implicitly run towards this, which provides another view of why it works.

### 4. Limitation and Future Direction

Simoun is a promising framework for vision-based reinforcement learning. However, one limitation of Simoun is that the motion and appearance features are both learned from the environment itself. In some extreme cases, learning solely from the environment might be difficult due to significantly varying motion and appearance patterns. In this case, leveraging the rich knowledge in other data sources becomes critical. Therefore, a possible future direction is to incorporate Simoun with external high-capacity vision models pre-trained on large datasets. By leveraging the abundant motion and appearance features extracted from high-performance models (e.g., CNNs and vision transformers for large-scale video understanding), Simoun has the potential to achieve even better performance.

### References


