Policy-shaped prediction: improving world modeling through interpretability

Anonymous Author(s)
Affiliation
Address
email

Abstract

Model-based reinforcement learning (MBRL) offers sample-efficient policy optimization but is susceptible to distractions. We address this by developing Policy-Shaped Prediction (PSP), a method that empowers agents to interpret their own policies and shape their world models accordingly. By combining gradient-based interpretability, pretrained segmentation models, and adversarial learning, PSP outperforms existing distractor-reduction approaches. This work represents an interpretability-driven advance towards robust MBRL.

1 Introduction

Model-based reinforcement learning (MBRL) offers a promising path to data-efficient policy learning, demonstrating impressive performance with high-dimensional sensory data [Hafner et al., 2023]. However, MBRL world models are particularly susceptible to distracting stimuli, a challenge that persists despite numerous attempts to address it [Deng et al., 2022, Fu et al., 2021, Wang et al., 2022, Seo et al., 2022, Wu et al., 2023, Schrittwieser et al., 2020].

We introduce Policy-Shaped Prediction (PSP), a novel method that uses gradient-based interpretability to identify and focus on important parts of an image-based environment. PSP interprets its own policy to prioritize relevant information, synergizing task-informed gradient-based loss weighting with a pre-trained segmentation model [Kirillov et al., 2023]. This approach creates a distraction-suppressing agent that outperforms leading image-based MBRL agents, particularly excelling against challenging and intricate, yet learnable, distractors. Our key contributions include:

- The development of PSP, combining gradient-based interpretability with pretrained segmentation to focus learning on important environment features.
- A challenging new benchmark for testing robustness to learnable distractions.
- Demonstration of PSP's 2x improvement in robustness against challenging distractions while maintaining good performance in non-distracting settings.

2 Method

We introduce PSP, a method to reduce an agent's sensitivity to useless distractions by focusing on sensory stimuli that are most relevant to its policy, rather than seeking to model everything in the environment. Our guiding intuition is that gradient-based interpretability techniques, traditionally used for post-hoc analysis, can be leveraged during training to highlight pixels in the environment that are important to the agent's policy. Additionally, using image segmentation we aggregate these pixelwise salience signals to identify important objects.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

PSP employs (1) gradients of the policy with respect to image inputs to identify task-relevant elements of the image, (2) a segmentation model to aggregate gradients within each object in the image, and (3) an adversarial objective to the image encoder of the world model that discourages encoding of duplicate information about the previous action. Figure 1 illustrates the training modifications made by this method to the underlying DreamerV3 [Hafner et al., 2023] architecture. Notably, since these modifications only affect the training stage of the world model, the DreamerV3 agent remains unaltered during inference. Below, we describe each of the three key components in detail.

2.1 Task-informed image reconstruction through interpretability-based weighting

Our approach builds upon the core idea that signals most important to the actor and/or critic should be given special importance in the world model. We extend the concepts of Value-Gradient weighted Model loss (VaGraM) [Voelcker et al., 2022] to high dimensional image inputs, which the previous work did not demonstrate. This extension to the image domain is inspired by gradient-based interpretability methods such as saliency maps [Simonyan et al., 2013, Shrikumar et al., 2017, Ancona et al., 2019]. Doing so requires novel work mitigating the problems of using gradient signals for high dimensional image input rather than low dimensional proprioceptive input. Additionally, we test with complex signals that are present in the same image inputs that contain useful information, whereas VaGraM tests on simple additional appended "distractor dimensions", which are independent of the state space and reward function.

While VaGraM focused solely on the using a gradient signal from the value function, we hypothesize that the gradient of the policy may provide an even more informative signal. To compute the policy-gradient weighting, we first sum across the dimensions of the action vector $\mathbf{a} = \mathbb{E}(\pi(\mathbf{s}))$, where $\mathbf{s}$ is the latent state of the world model, to produce a scalar $a = \sum_j a_j$, and then take the gradient with respect to the pixels of the input image x. To apply this weighting in the context of DreamerV3 [Hafner et al., 2023], we scale the image reconstruction loss term at each pixel i, for reconstructed image $\hat{x}_i$.

$$\mathcal{L}_{\text{image}}(\phi) = \sum_{x_i} \frac{\partial a}{\partial x_i} (\hat{x}_i - x_i)^2 \tag{1}$$

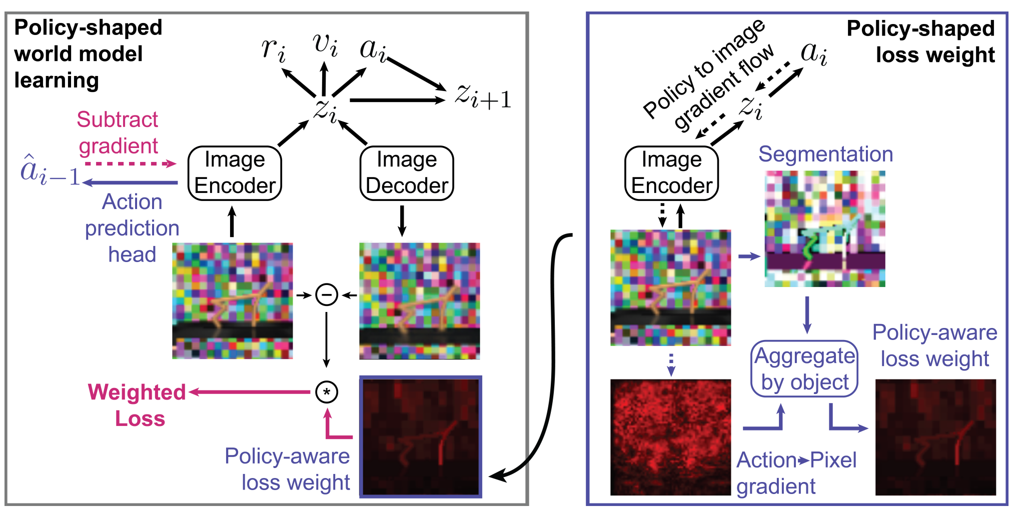

Figure 1: Policy-Shaped Prediction in an environment with challenging distractions. (left) Training of an otherwise-unaltered DreamerV3 agent is modified in two ways: 1) A head is added to predict the previous action based on the image encoding, and the gradient of the head is subtracted from the gradient of the image encoder, and 2) the loss is scaled pixelwise by a policy-shaped loss weight. (right) The loss weight uses the gradient of the policy to the input pixels. The image is segmented, and the pixel weights are averaged within each segmented object. Dashed lines signify gradient flow.

2.2 Improving saliency maps with object-based aggregation of gradient weights

Gradient-based weighting of the world model's reconstruction faces challenges due to noisiness from small-scale fluctuations [Smilkov et al., 2017]. While more computationally demanding approaches exist [Sundararajan et al., 2017, Smilkov et al., 2017], we introduce a novel, efficient solution: object-based aggregation of explainability signals using the objects detected by any high-quality segmentation model (SEG). We used the Segment Anything Model (SAM) [Kirillov et al., 2023], but other models of sufficient quality may be utilized.

During data collection, we segment each image into object masks, including a mask for unassigned pixels. The weight of a pixel x_i in segment $\text{SEG}(x_i)$ is:

$$W_i = \frac{1}{||\text{SEG}(x_i)||} \sum_{j \in \text{SEG}(x_i)} |\partial a / \partial x_j| \tag{2}$$

We clip the raw salience map to the 99th percentile before aggregation. If all gradients are zero, we set $W_i = 1$ for all i. We also linearly interpolate between the salience weighting and a uniform weighting: $W_i'' = \alpha W_i' + (1 - \alpha)$ where $W_i' = \text{width} \cdot \text{height} \cdot W_i / \sum_i W_i$ and $\alpha = 0.9$. This allows the world model to maintain reasonable reconstruction of less-salient aspects of the environment.

2.3 Adversarial action prediction head

The DreamerV3 world model consists of three main components: a convolutional neural network (CNN) image encoder $z_t \sim q_\phi(z_t | h_t, e_t)$ with $e_t = \text{CNN}_\rho(x_t)$, which processes the input image, serves as a prior during training, and encodes the environment state during inference; a recurrent state space machine (RSSM) consisting of $h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1})$ and $\hat{z}_t \sim p_\phi(\hat{z}_t | h_t)$ that is trained to simulate the progression of latent states given actions; and an image decoder, $\hat{x}_t \sim p_\phi(\hat{x}_t | h_t, z_t)$ which reconstructs the image from the latent state. Problematically, the encoder can capture information about previous actions from the image, despite this information already being provided directly to the RSSM through the action input. In other words, z_t may source information about a_{t-1} directly through x_t, despite a_{t-1} being an argument to f_ϕ during the computation of h_t. Unfortunately, our reconstruction loss weighting may not solve this problem, since during backpropagation from the actor-critic functions, we do not distinguish information about previous actions that comes from the image versus the action input to the RSSM.

To prevent the encoder from redundantly capturing previous action information already provided to the RSSM, we introduce an adversarial MLP head:

$$\hat{a}_{t-1} = \text{MLP}_\omega(\text{CNN}_\rho(\text{sg}(x_t))) \tag{3}$$

$$\mathcal{L}_{\text{AdvHead}}(\hat{a}_{t-1}, a_{t-1}) = (\hat{a}_{t-1} - a_{t-1})^2 \tag{4}$$

During world model training, we subtract the scaled gradient $\epsilon \cdot \nabla_\theta \mathcal{L}(\hat{a}_{t-1}, a_{t-1})$ from the overall gradient ($\epsilon = 1e3$), ensuring action information comes solely from the provided action vector.

Our training procedure for a DreamerV3 agent is shown in Algorithm 1. We note that it should be possible to apply these concepts of gradient-based weighting, segmentation-based aggregation, and adversarial action prediction to world models other than our chosen DreamerV3 architecture.

3 Experiments

To evaluate the model's performance we design our experiments around the following questions:

- Q1. Is our agent robust against distractors which are learnable by the world model, but of no utility for the actor-critic?
- Q2. What aspects of the environment are assigned importance by our method?
- Q3. Is our agent robust against distractors that are unrelated to the agent's actions?
- Q4. Does our agent maintain performance in standard, lower-distraction environments?
- Q5. What are the contributions of each component of our method?

Algorithm 1 Policy-Shaped Prediction training (for DreamerV3)

Input: World model parameterized by ϕ, policy π paramatarized by θ, image encoder parametrized by ρ, replay buffer with image transitions $(x_{t-1}, a_{t-1}, x_t, r_t, c_t)$, SEG segmentation model (SAM, in our application), action prediction MLP parameterized by ω
for training iteration 1, 2, … **do**
  Sample batch of transition sequences
  $G = \nabla_x \pi_\theta$ # Gradient of policy with respect to input image pixels
  $S = \text{SEG}(x)$ # Segmentation of input image
  $W = \text{agg}(G, S)$ # Aggregate gradient using segmentation
  $W'_i = W_i / \sum_i W_i$ # Normalize weighting
  $W'' = \alpha W' + (1 - \alpha)\mathbf{1}_{\text{shape}}(W')$ # Linearly interpolate with uniform weighting
  $\mathcal{L}_{\text{pred}}(\phi) = -\ln p_\phi(x_t \mid z_t, h_t) \odot W'' - \ln p_\phi(r_t \mid z_t, h_t) - \ln p_\phi(c_t \mid z_t, h_t)$ # Weighted DreamerV3 prediction loss
  $\mathcal{L}(\phi) = \mathbb{E}_{q_\phi}\left[\sum_{t=1}^{T} \left(\beta_{\text{pred}}\mathcal{L}_{\text{pred}}(\phi) + \beta_{\text{dyn}}\mathcal{L}_{\text{dyn}}(\phi) + \beta_{\text{rep}}\mathcal{L}_{\text{rep}}(\phi)\right)\right]$ # DreamerV3 model loss
  $\hat{a}_{t-1} = \text{MLP}_\omega(\text{stop\_gradient}(\text{CNN}_\rho(x_t)))$ # Adversarial action prediction head
  $\phi \leftarrow \text{Adam}(\nabla\mathcal{L} - \epsilon * \partial\mathcal{L}(\hat{a}_{t-1}, a_{t-1})/\partial\rho, \phi)$
  $\mathcal{L}_{\text{AdvHead}}(\hat{a}_{t-1}, a_{t-1}) = (\hat{a}_{t-1} - a_{t-1})^2$
  $\omega \leftarrow \text{Adam}(\nabla\mathcal{L}_{\text{AdvHead}}, \omega)$
end for

3.1 Experimental details

Baselines We test four Model-Based RL approaches as baselines: DreamerV3 [Hafner et al., 2023], and three methods specifically designed to handle distractions – Task Informed Abstractions [Fu et al., 2021], Denoised MDP (method in Figure 2b) [Wang et al., 2022], and DreamerPro [Deng et al., 2022]. Additionally, we choose DrQv2 [Yarats et al., 2021a] as a representative baseline Model-Free approach. For all agents, we use 3 random seeds per task, and default hyperparameters.

Environment details Visual observations are 64 × 64 × 3 pixel renderings. We test performance in three environments: DeepMind Control Suite (DMC) [Tassa et al., 2018], Reafferent DMC (described below), and Distracting Control Suite [Stone et al., 2021] (background video initialized to a random frame each episode, 2,000 grayscale frames from the "driving car" Kinetics dataset [Kay et al., 2017]). For each environment, we test two tasks: Cheetah Run and Hopper Stand. We selected these tasks because they present different levels of difficulty, allowing us to assess how distraction-sensitivity depends on the inherent difficulty of the task. For ablation experiments, we test on Cheetah Run.

3.2 Reafferent Deepmind Control Suite

To test our hypothesis, we devised the Reafferent Deepmind Control environment, inspired by [Stone et al., 2021]. This environment features distracting backgrounds that depend deterministically on the agent's previous action and elapsed time, mimicking complex self-generated distractors in the natural world. The background consists of 2,500 16x16 color grids, mapped to 625 time values and 4 discretized values of the first action dimension.

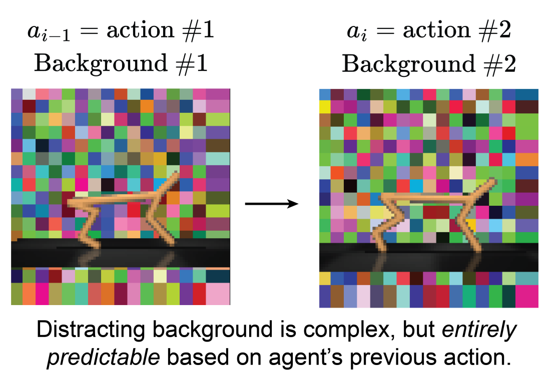

Figure 2: Schematic of the Reafferent Deepmind Control environment. The distracting background is entirely predictable based on the agent's previous action and the elapsed time in the episode.

Many methods encode assumptions about the forms distractors will take (usually uncorrelated to agent actions, reward, or both), rather than a means of generally identifying and ignoring distractors. We hypothesize that a learning-based approach, in which we avoid distraction by learning what is actually important for the agent to get things done, has the potential to overcome even learnable-but-not-useful distractions.

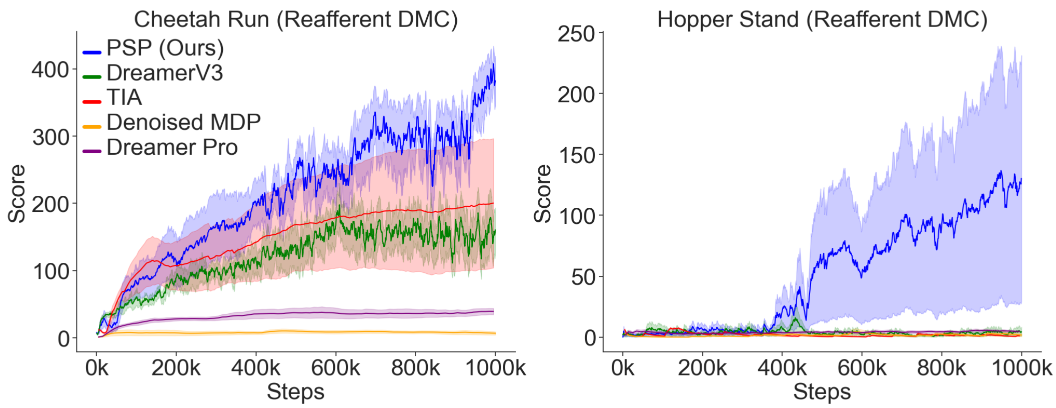

Figure 3: Training curve comparisons on Reafferent Deepmind Control. Mean $\pm$ std. err.

We find that baseline MBRL methods perform poorly in this environment (Table 1, Figure 3), often reproducing the distracting background at the expense of accurately modeling the agent (Figure 4). In contrast, our method demonstrates substantial improvement over existing baselines, achieving scores beyond their reach despite some variance in performance and **affirmatively answering Q1**.

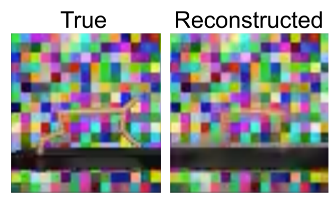

Figure 4: Denoised MDP reconstructs the background with a high degree of fidelity, but does not clearly render the Cheetah agent.

The model-free DrQv2 agent demonstrates robust performance, as expected since its CNN encoder is learned as part of the policy. In contrast, model-based methods face challenges when the world model's learning objective differs from the policy's. Our method bridges this gap, achieving superior performance while retaining the advantages of model-based RL.

PSP demonstrates a substantial improvement over the baselines (Table 1, Figure 3). Although it shows a higher than desired level of variance between runs, especially on the more challenging Hopper Stand task, it nevertheless achieves scores beyond the reach of any of the baselines. We note that none of the 12 runs across the 4 baseline methods demonstrate a score substantially above 0.

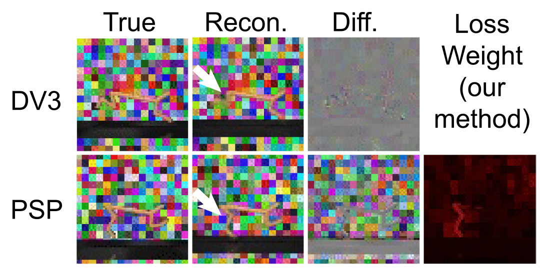

Figure 5: PSP vs. DreamerV3 on Reafferent Cheetah Run. From left: true, reconstructed, difference (true - recon.), loss weighting of PSP. DreamerV3 reproduces the background but not the back leg (see white arrow), and PCP renders the leg while not bothering to accurately model the background, **answering the question posed in Q2**.

3.3 Performance on unaltered DMC and Distracting Control Suite

Importantly, PSP performs comparably to other methods (including DreamerV3) on the unaltered Deepmind Control Suite, demonstrating

Table 1: Performance comparison across environments. DrQv2 is model-free, all others are model-based. TIA is task-informed abstraction, dMDP is denoised MDP, mean $\pm$ standard deviation.

Task	DrQv2	DreamerV3	DreamerPro	TIA	dMDP	PSP
Reafferent Control						
Cheetah Run	565.1 $\pm$ 35.5	158.4 $\pm$ 45.7	39.7 $\pm$ 9.0	200.4 $\pm$ 203.9	6.7 $\pm$ 4.3	**383.1 $\pm$ 23.8**
Hopper Stand	210.3 $\pm$ 353.8	4.6 $\pm$ 3.9	3.8 $\pm$ 1.0	0.9 $\pm$ 0.3	1.7 $\pm$ 2.5	**128.5 $\pm$ 215.7**
Unmodified Deepmind Control						
Cheetah Run	736.0 $\pm$ 17.0	521.1 $\pm$ 136.3	908.4 $\pm$ 1.6	773.7 $\pm$ 22.7	763.0 $\pm$ 62.8	712.3 $\pm$ 32.3
Hopper Stand	752.9 $\pm$ 206.8	867.4 $\pm$ 15.9	890 $\pm$ 11.2	298.4 $\pm$ 512	897.9 $\pm$ 14.2	865.6 $\pm$ 53.6
Distracting Deepmind Control						
Cheetah Run	364.4 $\pm$ 60.7	243.8 $\pm$ 81.2	179.1 $\pm$ 24	548.5 $\pm$ 238.9	397.4 $\pm$ 111.8	408.6 $\pm$ 125.1
Hopper Stand	781.1 $\pm$ 110.3	173.7 $\pm$ 160.9	561.8 $\pm$ 103.1	200.5 $\pm$ 171.7	13.2 $\pm$ 16.5	417.7 $\pm$ 118.9

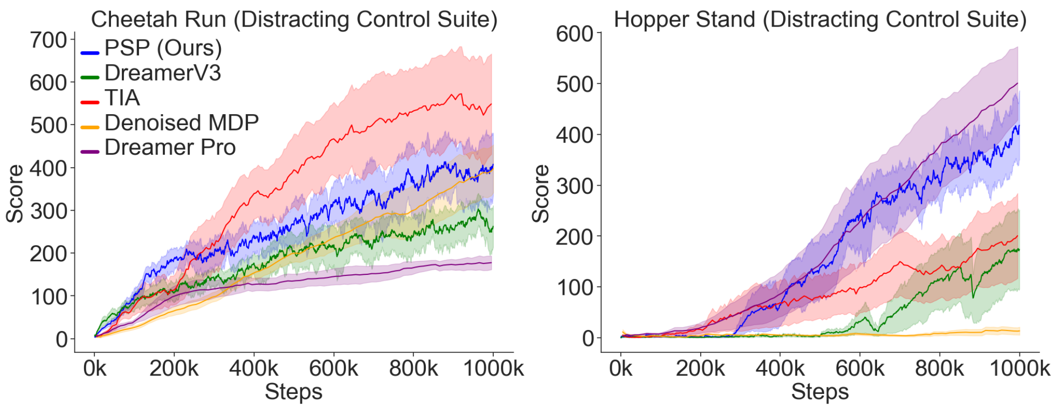

Figure 6: Training curve comparison on Distracting Control. Mean ± std. err.

Table 2: Performance of ablated versions of PSP (for reafferent and unaltered Cheetah Run).

Gradient weighting	Gradient weighting with segmentation	Unaltered	Reafferent
Policy	✓	712.3 ± 32.3	383.1 ± 23.8
Policy	✗	742.1 ± 79.7	188.4 ± 9.4
None	✗	521.1 ± 136.3	158.4 ± 45.7

that we have not introduced a tradeoff between performance on distracting and non-distracting environments and **addressing Q4**. (Table 1, Figure A1).

On Distracting Control tasks, in which the background distractor is uncoupled from the agent's actions, PSP produced consistently improved performance relative to baseline DreamerV3, in contrast to the more variable performance of DreamerPro, TIA, and Denoised MDP (Table 1, Figure 6), **addressing Q3**.

In sum, PSP exhibits similar performance to baseline methods in commonly used tests of distractor-suppression and in non-distracting environments, while also demonstrating unmatched performance on particularly challenging distractors that are complex but learnable. Given the success of MBRL in non-adversarial environments, even when compared with leading Model Free Reinforcement Learning techniques [Hafner et al., 2023] , this work points to ways of matching these gains in an adversarial setting.

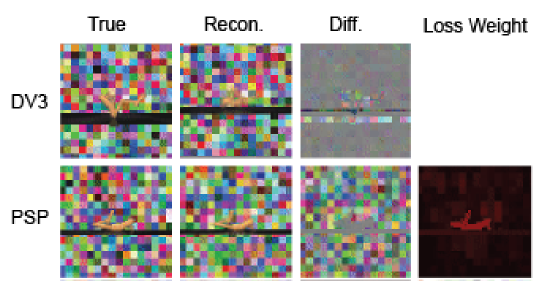

Figure 7: PSP vs. DreamerV3 on Reafferent Hopper Stand. From left: true, reconstructed, difference (true - recon.), loss weighting of PSP. DreamerV3 reproduces the background but not the agent, and PCP renders the agent while not bothering to accurately model the background, **answering the question posed in Q2**.

3.4 Ablation study

To understand the contributions of each subcomponent of the method (**Q5**), we conduct ablations on the reafferent and unaltered Cheetah Run (Table 2). We find that while some ablations trade off performance between the environments, our complete model has good performance on both. In particular, segmentation-based aggregation is critical to improving our model's performance amid distractors, while also maintaining its performance in the non-adversarial baseline. Overall, the results of the ablations confirm that combining segmentation, policy gradient sensory weighting, and adversarial action prediction results in the best scores across the unaltered and reafferent environments.

4 Related Work

Related Work Recent advances in Model Based RL (MBRL) like World Models [Ha and Schmidhuber, 2018], SimPLe [Kaiser et al., 2019], MuZero [Schrittwieser et al., 2020], EfficientZero [Ye et al., 2021], and DreamerV3 [Hafner et al., 2023] have shown impressive performance but remain susceptible to distractions [Lambert et al., 2020]. Various approaches have been proposed to address this, including (1) structural regularizations (DreamerPro [Deng et al., 2022], Agent Control-Endogenous State Discovery [Lamb et al., 2022], Task Informed Abstractions [Fu et al., 2021], Denoised MDPs [Wang et al., 2022]), ensemble methods [Clavera et al., 2018], and (2) learning-based approaches that use actor-critic functions to guide world modeling (VaGraM [Voelcker et al., 2022], Mismatched No More Eysenbach et al. [2022], Goal-Aware Prediction [Nair et al., 2020], Masked world models for visual control [Seo et al., 2023], The value equivalence principle for model-based reinforcement learning [Grimm et al., 2020], MuZero [Schrittwieser et al., 2020], Value Prediction Networks [Oh et al., 2017]). Parallel work in Model Free RL (MFRL) has also tackled distraction sensitivity, with methods like DrQv2 [Yarats et al., 2021a] and approaches using attention mechanisms [Mott et al., 2019], prototypes [Yarats et al., 2021b], and dynamic sparse training [Grooten et al., 2023a,b].

5 Discussion

PSP introduces a novel approach to model-based reinforcement learning that leverages interpretability techniques not just for analysis, but as an integral part of the learning process. By allowing the agent to interpret its own policy, PSP focuses the world model's capacity on aspects of the environment most relevant for decision making. This self-interpretation process comprises three key components: 1) We use gradient-based interpretability methods, analogous to saliency maps [Simonyan et al., 2013], to identify important pixels in the input image; 2) We aggregate pixel importance by object using a pre-trained segmentation model, providing a higher-level interpretation of the environment; 3) We employ an adversarial prediction head to prevent wasteful encoding of known information.

Our work opens avenues for future research in interpretable RL, such as using more advanced explainability gradient-based attribution methods like Integrated Gradients [Sundararajan et al., 2017, Ancona et al., 2019]. While PSP demonstrates promising results, it has limitations, including its object-centric assumptions and the computational requirements of the segmentation model.

Outlook In conclusion, PSP represents a significant step towards robust model-based RL via the direct integration of of model interpretability techniques. The findings here open other lines of inquiry such as using more explainable architectures, utilizing faster segmentation models and utilizing segmentation models designed for videos in order to do temporal aggregation.

References

M. Ancona, E. Ceolini, C. Öztireli, and M. Gross. Gradient-based attribution methods. *Explainable AI: Interpreting, explaining and visualizing deep learning*, pages 169–191, 2019.

I. Clavera, J. Rothfuss, J. Schulman, Y. Fujita, T. Asfour, and P. Abbeel. Model-based reinforcement learning via meta-policy optimization, 2018.

F. Deng, I. Jang, and S. Ahn. Dreamerpro: Reconstruction-free model-based reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pages 4956–4975. PMLR, 2022.

B. Eysenbach, A. Khazatsky, S. Levine, and R. R. Salakhutdinov. Mismatched no more: Joint model-policy optimization for model-based rl. *Advances in Neural Information Processing Systems*, 35: 23230–23243, 2022.

X. Fu, G. Yang, P. Agrawal, and T. Jaakkola. Learning task informed abstractions. In *International Conference on Machine Learning*, pages 3480–3491. PMLR, 2021.

C. Grimm, A. Barreto, S. Singh, and D. Silver. The value equivalence principle for model-based reinforcement learning. *Advances in Neural Information Processing Systems*, 33:5541–5552, 2020.

B. Grooten, G. Sokar, S. Dohare, E. Mocanu, M. E. Taylor, M. Pechenizkiy, and D. C. Mocanu. Automatic noise filtering with dynamic sparse training in deep reinforcement learning. *arXiv preprint arXiv:2302.06548*, 2023a.

B. Grooten, T. Tomilin, G. Vasan, M. E. Taylor, A. R. Mahmood, M. Fang, M. Pechenizkiy, and D. C. Mocanu. Madi: Learning to mask distractions for generalization in visual deep reinforcement learning. *arXiv preprint arXiv:2312.15339*, 2023b.

D. Ha and J. Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017. URL http://arxiv.org/abs/1705.06950.

A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

A. Lamb, R. Islam, Y. Efroni, A. Didolkar, D. Misra, D. Foster, L. Molu, R. Chari, A. Krishnamurthy, and J. Langford. Guaranteed discovery of control-endogenous latent states with multi-step inverse models. *arXiv preprint arXiv:2207.08229*, 2022.

N. Lambert, B. Amos, O. Yadan, and R. Calandra. Objective mismatch in model-based reinforcement learning. *arXiv preprint arXiv:2002.04523*, 2020.

A. Mott, D. Zoran, M. Chrzanowski, D. Wierstra, and D. Jimenez Rezende. Towards interpretable reinforcement learning using attention augmented agents. *Advances in neural information processing systems*, 32, 2019.

S. Nair, S. Savarese, and C. Finn. Goal-aware prediction: Learning to model what matters. In *International Conference on Machine Learning*, pages 7207–7219. PMLR, 2020.

J. Oh, S. Singh, and H. Lee. Value prediction network. *Advances in neural information processing systems*, 30, 2017.

J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Y. Seo, K. Lee, S. James, and P. Abbeel. Reinforcement learning with action-free pre-training from videos, 2022.

Y. Seo, D. Hafner, H. Liu, F. Liu, S. James, K. Lee, and P. Abbeel. Masked world models for visual control. In *Conference on Robot Learning*, pages 1332–1344. PMLR, 2023.

A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017.

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017. URL http://arxiv.org/abs/1706.03825.

A. Stone, O. Ramirez, K. Konolige, and R. Jonschkowski. The distracting control suite–a challenging benchmark for reinforcement learning from pixels. *arXiv preprint arXiv:2101.02722*, 2021.

M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. d. L. Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

C. Voelcker, V. Liao, A. Garg, and A.-m. Farahmand. Value gradient weighted model-based reinforcement learning. *arXiv preprint arXiv:2204.01464*, 2022.

T. Wang, S. S. Du, A. Torralba, P. Isola, A. Zhang, and Y. Tian. Denoised mdps: Learning world models better than the world itself. *arXiv preprint arXiv:2206.15477*, 2022.

J. Wu, H. Ma, C. Deng, and M. Long. Pre-training contextualized world models with in-the-wild videos for reinforcement learning, 2023.

D. Yarats, R. Fergus, A. Lazaric, and L. Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021a.

D. Yarats, R. Fergus, A. Lazaric, and L. Pinto. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pages 11920–11931. PMLR, 2021b.

W. Ye, S. Liu, T. Kurutach, P. Abbeel, and Y. Gao. Mastering atari games with limited data. *Advances in neural information processing systems*, 34:25476–25488, 2021.

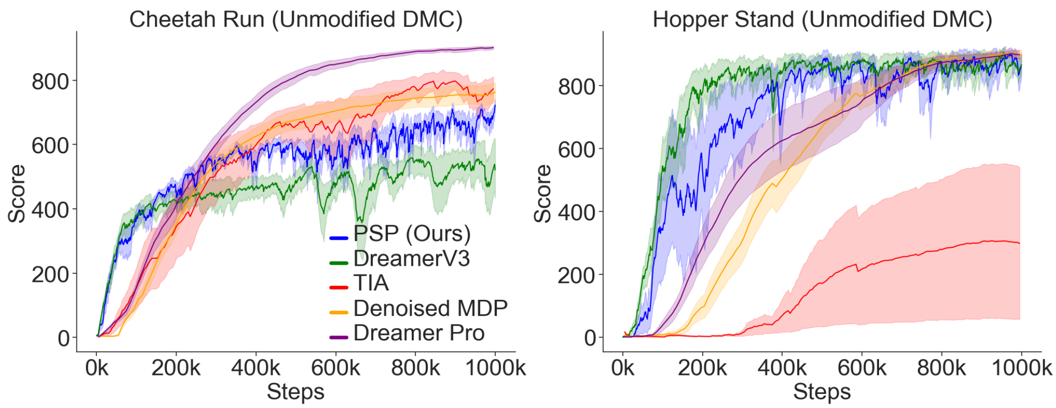

Figure A1: Training curve comparison on unmodified Deepmind Control. Mean $\pm$ std. err.

A Broader Impacts

At the current stage, this work remains reasonably far from any large societal impacts, as it is limited to agents interacting with small, simulated environments. Over the long term, however, if model-based RL algorithms are used to control robots or internet-connected agents (such as large language model agents), the potential for both large positive and negative societal impacts becomes relevant. On the positive side, intelligent agents that are capable of modeling the world and avoiding distractors have the potential to aid humans in a wide variety of scenarios, from housework, to medical applications, to exploration, to internet research. On the negative side, agents without proper safeguards have the potential to inflict harm on humans and the environment, whether through negligence or malfeasance. Ultimately, our work is targeted at producing the positive impacts, while still allowing for mitigation of the negative impacts.

B Experiments Compute Resources

Each trial of the PSP method used 4 Nvidia A40 GPUs to train the modified DreamerV3 model, and 4 A40 GPUs to run the segment anything model in parallel. Given an estimated 17 unique experiments for the final paper, 3 trials per experiment with our method, and about 1.5 days per training run, we used about 17 * 3 * 1.5 * 8 GPUs = 612 GPU days on A40 accelerators. Early experiments with this methodology likely used an additional 300. Baseline trials could be run on only a single A40 GPU or a desktop NVIDIA 2070 SUPER, usually in less than a day, and accounted for a comparably negligible level of resources.

We believe this level of resource consumption could be easily reduced. The modifications to the DreamerV3 model do not attempt to benchmark the most costly components. We suspect our method of parallelizing the new backpropagation from the policy to the image could be optimized further from its naive Jax implementation. Additionally, SAM could be supplanted by a more efficient segmentation model. We focused on establishing the basic technique with SAM, but replacing it should be the subject of future work.

C Code

An anonymized version of the code with instructions for reproducing these experiments will be available for reviewers at this anonymous GitHub Repository.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our contributions and claims, and match the details in the results section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, the paper clearly points out what we believe to be core limitations and assumptions of our work, as well as present limitations that we do not believe are inherent to the method.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The claims of this paper are tested empirically.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper itself should include enough detail to reproduce our results with the open source SAM and Dreamerv3 models. Because the implementation is not trivial, we will also release the GitHub repository in the camera ready version, which includes an implementation of the core algorithm for DreamerV3 and a shared implementation of the test environments for DreamerV3 and every baseline. We have not included it in the review version as the GitHub will identify the authors.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
    (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
    (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
    (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
    (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: An anonymized version of the code will be available at the linked GitHub Repository for reviewers.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We include all relevant details of our benchmark and (where we diverge from the default) models.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All experiments are performed with three trials and std. dev is reported.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided a transparent and reasonable estimate of compute requirements in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The only new dataset involved in this work was generated by a small Python procedure and has no privacy risks or ethical concerns. The implemented model modifies an open source repo and has had no interaction with human subjects.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification:This is foundational work for improving MBRL and any societal impacts are at least one order removed, but we have outlined the possible societal impacts of improved MBRL in general.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper describes a foundational change to MBRL and introduces no new datasets that pose a risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All previous work is cited and no proprietary code has been used beyond what is allowed by its license.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Source code will be provided before publication if paper is accepted.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.