Spatiotemporal Predictive Pre-training for Robotic Motor Control

Anonymous Author(s)
Affiliation
Address
email

Abstract

Robotic motor control necessitates the ability to predict the dynamics of environments and interaction objects. However, advanced self-supervised pre-trained visual representations (PVRs) in robotic motor control, leveraging large-scale egocentric videos, often focus solely on learning the static content features of sampled image frames. This neglects the crucial temporal motion clues in human video data, which implicitly contain key knowledge about sequential interacting and manipulating with the environments and objects. In this paper, we present a simple yet effective robotic motor control visual pre-training framework that jointly performs spatiotemporal prediction with dual decoders, utilizing large-scale video data, termed as **STP**. STP adheres to two key designs in a multi-task learning manner. First, we perform spatial prediction on the masked current frame for learning content features. Second, we utilize the future frame with an extremely high masking ratio as a condition, based on the masked current frame, to conduct temporal prediction of future frame for capturing motion features. This asymmetric masking and decoder architecture design is very efficient, ensuring that our representation focusing on motion information while capturing spatial details. We carry out the largest-scale BC evaluation of PVRs for robotic motor control to date, which encompasses 21 tasks within a real-world Franka robot arm and 5 simulated environments. Extensive experiments demonstrate the effectiveness of STP as well as unleash its generality and data efficiency by further post-pre-training and hybrid pre-training. Our code and weights will be released for further applications.

1 Introduction

In NLP and CV, adapting pre-trained foundation models from large-scale data to various downstream tasks has seen great success. For example, pre-trained visual representations using self-supervised [38, 15, 67, 2, 93] or weakly-supervised [71, 25, 55] methods exhibit strong generalization ability for visual understanding. However, in robot learning, due to data scarcity and homogeneity, some groundbreaking methods [53, 1] resort to training from scratch only using domain-specific data. Recently, inspired by the success of transfer learning in CV, many works [69, 73, 65, 58, 59, 19] have explored developing a pre-trained visual representation (PVR) using large-scale out-of-domain data for various robotic motor control tasks. Currently, one successful paradigm [73, 99, 59, 19] is to use large-scale egocentric video datasets [29] and train vanilla vision transformers (ViT) [22] based on MAE [38], which exhibits excellent learning efficiency and generalization ability for learning policy from raw pixel. Among them, the Ego4D [29] dataset offers numerous first-person human-object interaction scenes and good motion clues. We argue that although learning static spatial structure priors from task-relevant pre-training data sources is crucial, designing a more relevant self-supervised proxy task for motor control should not be overlooked. Therefore, in this paper, we aim to develop a more relevant self-supervised proxy task for robotic motor control representation learning.

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

Robotic motor control typically requires fine-grained spatial localization and relatively dense semantics. With its ability to effectively capture low-level geometry and space structure, MAE [38] pre-training excels at this task. However, is dense spatial content sufficient for robotic motor control? Some neuroscientific studies [50, 21, 88] suggest the brain's different areas or cells show specialization. Some are dedicated to processing the information of temporal object motion, while others focus on static spatial details. Their combination results in subjective pattern perception. Inspired by this finding, we hypothesize that an effective robotic motor control pre-training proxy task should require joint learning of spatial content features and temporal motion features. However, current methods [73, 59, 19] use MAE pre-training with image frames from human videos, capturing only static content features. They overlook the temporal motion clues in human videos, which implicitly contain key knowledge about sequential interaction with environment and manipulation of objects. Therefore, we aim to bridge this gap by incorporating these motion clues into our proxy task.

Based on the analysis above, the most critical challenge is the absence of action annotations in human video data for modeling object motion. To model interaction and manipulation actions from actionless video data, we propose to implicitly capture them by predicting future frame pixels based on current frame. However, predicting the future frame without any conditions could contain high uncertainty and be extremely difficult. Therefore, we propose to use the future frame with an extremely high masking ratio as a prompt condition, specifically 95%, which serves to reveal some behavior and dynamic priors, i.e. what to do and how to do it. In the experiments section, we will further explore different condition alternatives, including language narration and their combination. Additionally, directly and simply executing temporal prediction could lead the model to overlook static spatial details, and it is also not efficient enough. Therefore, another technical contribution of STP is to jointly perform spatial prediction by masking the current frame with 75% masking ratio. In summary, we present **STP**, a multi-task self-supervised pre-training framework through spatiotemporal predictive learning. Our STP asymmetrically mask the current frame and future frame from a video clip, using a spatial decoder to conduct spatial prediction for content learning and a temporal decoder to conduct temporal prediction for motion learning. This asymmetric masking and decoder architecture design ensures that our pre-trained encoder focusing on motion information while capturing spatial details.

Subsequently, we establish our evaluation scheme. Currently, how to adapt pre-trained visual representations for robotic motor control still remain an open question. Considering the expensive cost of robot data collection or exploration, we employ a data-efficient paradigm of few-shot behavior cloning by learning from demonstrations (Lfd). To demonstrate the generalization ability of visual representation, our primary evaluation scheme involves freezing the visual encoder during policy training. Additionally, considering that fine-tuning ViT with few demonstrations might lead to overfitting and masked modeling exhibits excellent data efficiency [86, 102, 52] in domain-in data, we further follow the post-pre-training [7, 93, 59] paradigm to perform STP pre-training with task-specific data to achieve better results. It is noteworthy that different tasks do not share representation in this setting. Finally, we conduct the largest-scale BC evaluation of PVRs for robotic motor control to date to demonstrate the effectiveness of STP, which encompasses 21 tasks (2 real-world tasks and 19 simulation tasks across 5 environments). These simulation tasks are derived from the union of manipulation and locomotion tasks from prior works [65, 59].

We make the following **four contributions**: **(1)** We present STP, a *self-supervised* visual pre-training framework for robotic motor control, which jointly conducts spatiotemporal prediction with *asymmetric masking and decoder architecture design* for content and motion features learning. **(2)** We further expand STP by performing hybrid pre-training with ImageNet-MAE and post-pre-training with task-specific data, unleashing its *generality* and *data efficiency*. **(3)** To our best knowledge, we conduct the *largest-scale BC evaluation* of PVRs for robotic motor control to date to demonstrate the effectiveness of STP. **(4)** Our experiments yield some insightful observations. In temporal prediction, language does not significantly enhance performance. Instead, *single-modality self-supervised paradigm* achieves the best results. This finding is highly encouraging for self-supervised robotic motor control representation learning. Moreover, in the few-shot BC setting, naively scaling up model size does not necessarily lead to improved outcomes. Finally, incorporating *more diverse* data and *domain-in* data into the pre-training can further enhance performance.

2 Related Work

Pre-trained Visual Representation Learning. Large-scale visual representation pre-training are continually empowering computer vision. The primary supervised learning methods include learning image recognition [40, 87] from ImageNet [20] and learning multi-modal alignment [71] from image-text pairs. Currently, self-supervised learning methods are enjoying significant popularity, primarily falling into two main categories. The first category utilizes contrastive learning [39, 15, 14] technique or joint-embedding architecture [13] to learn view-invariance. The second category performs masked modeling [7, 38, 100, 95, 4, 2] and predict the pixel or representation of invisible parts in space. In addition, some methods [106, 67, 8] have also proposed to combine different optimization objectives in a multi-task learning manner. Recently pre-trained visual representation learning for robotic motor control have bee rapidly developing [69, 65, 73, 99, 58, 57, 46, 59, 19]. These methods cover different backbones (ResNet [40], ViT [22]), different policy learning methods (reinforcement learning [99], behavior cloning [69, 65, 59], reward function [58] and task specification [42]), different adaptation schemes (linear probing [69, 65, 46, 59], fine-tuning [19] and designing adapters [78, 56]), and different evaluation environments (diverse simulation benchmarks). At present, it is still unclear how these factors collectively influence the performance. In this paper, we choose scalable vanilla vision transformer [22] as our backbone and data-efficient few-shot behavior cloning paradigm to conduct policy learning, while ensuring the backbone is frozen during policy training.

Temporal Predictive Learning. Early works once explored representation learning through future prediction, encompassing image [61], video [35, 80] and audio [66]. VideoMAE [86, 93] extend MAE [38] to 3D video architecture. Recently TrackMAE [17] and SiamMAE [33] predict the masked future frame based on unmasked current frame, leading to a better capture of temporal correspondence and achieving outstanding performance in object tracking and segmentation tasks. In robot learning, predicting future visual states primarily serves as a transition dynamic model such as World Models [62, 77] and Dreamer [76]. [85, 9] predict the future visual states using goal image in robot data. GR-1 [97] conducts language-conditioned video prediction for policy model pre-training in a frozen visual representation space. [96] proposed dynamics-aware representation learning, and [82, 72] employed forward dynamics for self-supervised pre-training. Some works explored to train video prediction models and utilize visual foresight [32], inverse dynamics models [18], goal-conditioned policy learning [23], and geometry estimation [51] methods for motor control, respectively. [92] fine-tuned pre-trained representations into dynamic and functional distance modules for manipulation tasks. Unlike these works, we utilize the public large-scale egocentric video data and employ masked spatiotemporal predictive learning as a *self-supervised proxy task (without any language or action annotations) for robotic motor control representation learning*, instead of designing elaborate architectures or methods for specific predictive tasks [28, 37].

Vision-based Robot Learning. Vision-based robot learning plays a crucial role in robotics community. Recently some related works focus on studying model architectures [44, 12, 47], observation spaces [107], downstream policy learning methods [41], sim-to-real transfer [79], designing adapters [78, 56], learning-from-scratch baseline [36], and affordance model [6, 105, 45, 60], in visuo-motor representation learning. Other related works [70, 5, 91, 101, 48] attempt to learn manipulation skills from small-scale and in-domain human videos. In addition, language-conditioned vision robot learning has received significant attention. Some works scale multimodal robotic data [42, 11, 34, 90, 24, 68, 84] or introduce Internet data and knowledge [81, 103, 10, 54, 43, 94, 64] for end-to-end robot learning. In our study, we pre-train a off-the-shelf visual representation from large-scale egocentric video datasets for robotic motor control tasks. Our method is more simple and general for different downstream tasks of motor control.

3 Method

In this section, we describe our method in details. First, we give an overview of our spatiotemporal predictive pretraining (STP) framework. Then, we give a technical description on our core components during pre-training: the masked image encoder and dual decoders scheme. Finally, we describe how to adapt our pre-trained encoder to downstream robotic motor control tasks.

3.1 Overview of STP

As illustrated in Figure 1, our STP aims to pre-train an image encoder for robotic motor control from video datasets. This pre-trained image encoder is subsequently frozen and directly transferred to solve motor control tasks. Specifically, given a video dataset $\mathcal{D}$, our goal is to learn an image encoder Φ_{enc}, that maps images to the visual representations. During pre-training and post-pre-training, $\mathcal{D}$ represents large-scale out-of-domain videos and task-specific demonstration videos, respectively. After pre-training, we reuse Φ_{enc} for downstream motor control policy learning. Specifically, the downstream

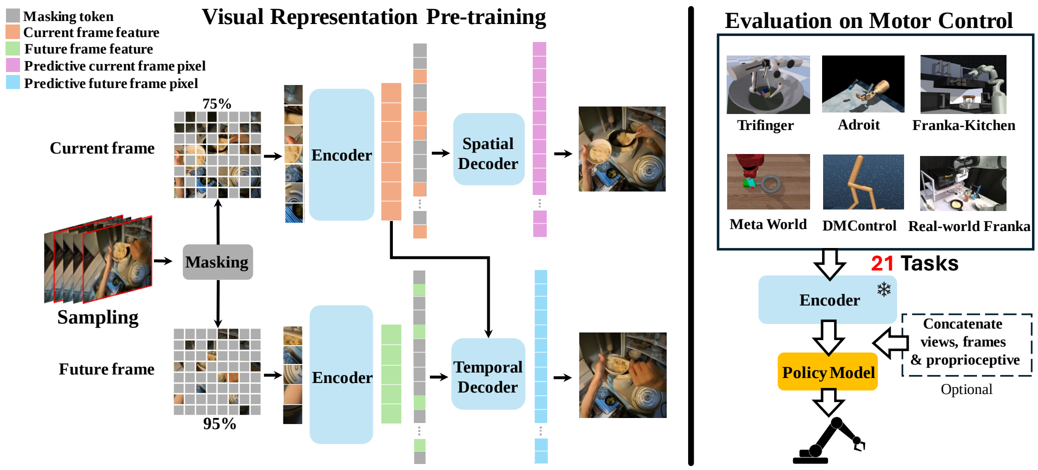

Figure 1: **STP framework**. **Left:** During pre-training, we sample the current frame and the future frame from the video clip, and carry out spatiotemporal predictive pre-training. **Right:** During motor control tasks evaluation, we freeze the pre-trained encoder to extract visual state representations and discard the decoders.

task will require an agent to make sequential action decisions based on visual observations $\mathcal{O}$. Instead of using the raw observation images as direct input like end-to-end policy learning from pixel, the agent will employ the pre-trained Φ_{enc} to extract its visual state representation $\Phi_{enc}(\mathcal{O})$ for the subsequent policy learning module.

3.2 Masked Image Encoder

We first introduce the pipeline of our image encoder. Our image encoder processes image frames using a vanilla vision transformer [22]. Given a image $\mathbf{I} \in \mathbb{R}^{C\times H\times W}$, we initially process it by the patch embedding layer to obtain its token sequences $\mathbf{T}$, where $\mathbf{T} = \{P_i\}_{i=1}^{N}$ and N is the the total token number, (e.g., N = 196 for a 224 × 224 image with a patch size of 16 × 16). Then we add the fixed 2D sine-cosine positional embeddings for all tokens. Following this, we mask and remove a part of tokens, according to a randomly generated masking map $\mathbb{M}(\rho)$, where ρ is the masking ratio. The encoder applies several transformer blocks (consisting of a global self-attention layer and a FFN layer) on all unmasked tokens: $\mathbf{Z} = \Phi_{enc}(\mathbf{T}^u)$, where $\mathbf{T}^u = \{T_i\}_{i\in(1-\mathbb{M}(\rho))}$. During this process, a [CLS] token is added at the beginning.

Then we describe our encoding process during pre-training. We randomly sample two frames from a video clip based on an interval: the current frame $\mathbf{I_c}$ and the future frame $\mathbf{I}_f$. Following the above pipeline, we randomly generate two asymmetric masking maps for the current frame and the future frame, denoted as $\mathbb{M}_c = \mathcal{M}_c(\rho^c)$ and $\mathbb{M}_f = \mathcal{M}_f(\rho^f)$, respectively. Each of these maps has a different masking ratio. We then use these maps to separately process the two frames and obtain their features, $\mathbf{Z}_c$ and $\mathbf{Z}_f$. As analyzed above, our STP aims to jointly learn content and motion features by spatiotemporal predictive learning. For content feature learning, we follow MAE [38], masking a portion of the current frame based on $\mathbb{M}_c$, with $\rho^c = 75\%$, and predict the masked parts during the decoding process. This encourages the model to learn spatial and geometric structure priors from the current frame data through spatial reasoning. For motion feature learning, we establish an objective to predict the future frame based on the masked current frame. However, predicting the future frame without any conditions could be meaningless and extremely challenging. Therefore, we use the future frame with an extremely high masking ratio as a condition, specifically $\rho^f = 95\%$, which reveals some behavior and dynamic priors. In the experiments section, we will further discuss different condition schemes, including language narration and the combination between them. In summary, our encoding process during pre-training can be formally described as follows:

$$\begin{cases} \mathbf{Z}_c = \Phi_{enc}(\mathbf{I}_c, \mathbb{M}_c), \\ \mathbf{Z}_f = \Phi_{enc}(\mathbf{I}_f, \mathbb{M}_f). \end{cases} \tag{1}$$

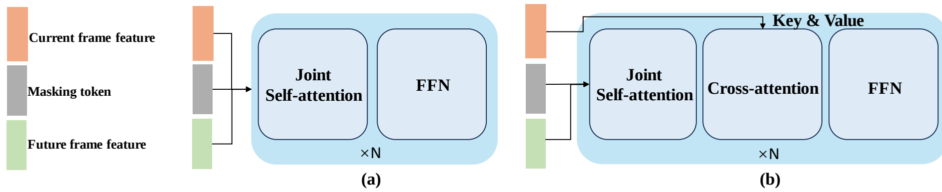

Figure 2: Temporal decoder design. **(a)** Standard joint-self architecture. **(b)** Our self-cross architecture.

3.3 Dual Decoders

To jointly capture static content and object motion features for better spatiotemporal understanding, our STP present a dual decoders scheme to predict both the pixel of current and future frame simultaneously in a multi-task learning manner. As shown in Figure 1, our dual decoder scheme includes a spatial decoder Φ_{dec_s} for spatial prediction and a temporal decoder Φ_{dec_t} for temporal prediction. We firstly give a technical description on them, respectively. Then we describe how we combine them into our final method.

Spatial Decoder. To capture static content features, our spatial decoder is solely utilized for processing the current frame visual feature. Specifically, after obtaining the masked current frame visual feature $\mathbf{Z}_c$, we concatenate it with some learnable masking tokens, leading to the formation of $\mathbf{Z}_c^d = \mathbf{Z}_c \cup \{\mathbf{M}_i\}_{i \in \mathbb{M}_c}$, where $\mathbb{M}_c$ is the current frame masking map. Then, each of these tokens further adds a corresponding positional embedding. Subsequently, $\mathbf{Z}_c^d$ undergoes decoding in the decoder and is continuously updated. The architecture of the spatial decoder block aligns with the standard transformer encoder block, comprised of a global self-attention layer and a FFN layer. Finally, with the deocoded token sequence $\mathbf{Z}_c^d$, our spatial decoder predicts the invisible tokens of the current frame $\hat{\mathbf{I}}_c^d$, operating under the current frame masking map $\mathbb{M}_c$.

Temporal Decoder. To capture motion features, our temporal decoder jointly processes the current frame and the future frame which serves as the temporal prediction condition. To elaborate, we firstly obtain the masked current frame feature $\mathbf{Z}_c$ and the masked future frame feature $\mathbf{Z}_f$. We then concatenate $\mathbf{Z}_f$ with the masking tokens that have the positional embedding added, resulting in $\mathbf{Z}_f^d$. Following this, $\mathbf{Z}_f^d$ and $\mathbf{Z}_c$ interact within the temporal decoder for decoding. The architecture of our temporal decoder block is in alignment with the standard transformer decoder block [89], consisting of a self-attention layer, a cross-attention layer, and a FFN layer, as shown in Figure 2 (b). During decoding, the self-attention layer and FFN are solely used to process $\mathbf{Z}_f^d$. For the cross-attention layer, $\mathbf{Z}_f^d$ is continuously updated as the query, while $\mathbf{Z}_c$, acting as the key and value, is kept constant. Compared to standard architecture, it ensures that the past frame representation space will not be updated in the temporal decoder and are specifically used for temporal correlation and prediction. This asymmetric interact architecture not only achieves more efficient training but also produces better results. Finally, with the decoded token sequence $\mathbf{Z}_f^d$, our temporal decoder predicts the invisible tokens of the future frame $\hat{\mathbf{I}}_f^d$, operating under the future frame masking map $\mathbb{M}_f$.

Multi-task Predictive Learning. As mentioned above, our STP jointly conducts spatiotemporal prediction by asymmetric masking ratio and dual decoders scheme, the whole decoding pipeline can be formally described as follows:

$$\begin{cases} \hat{\mathbf{I}}_c^d = \Phi_{dec\_s}(\mathbf{Z}_c^d), \\ \hat{\mathbf{I}}_f^d = \Phi_{dec\_t}(\mathbf{Z}_c, \mathbf{Z}_f^d). \end{cases} \tag{2}$$

Our loss function is the mean squared error (MSE) loss between the normalized masked pixels and the predicted pixels. So our loss function ℓ is as follows:

$$\ell = \mathrm{MSE}(\hat{\mathbf{I}}_c, \mathbf{I}_c) + \mathrm{MSE}(\hat{\mathbf{I}}_f, \mathbf{I}_f). \tag{3}$$

3.4 Downstream Policy Learning

To enable data and computation efficiency during the policy learning process, we adopt the paradigm of few-shot behavior cloning by learning from demonstrations (Lfd), and we keep the image encoder

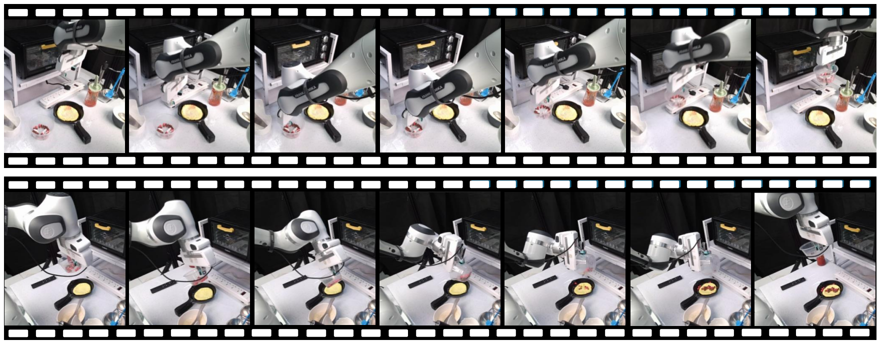

Figure 3: **The evaluation demonstrations of our real-world tasks.** For picking, the robot arm needs to pick up the bowl on the desktop. For pouring, the robot arm needs to pour the ingredients from the bowl into the pot.

frozen. Concretely, for each task, we are given offline expert demonstrations $\mathcal{S} = \{\tau_1, ..., \tau_n\}$, where each τ_i is a trajectory of robot observations and actions, denoted as $\tau_i = [(o_0, a_0), \ldots, (o_T, a_T)]$. Based on the $\mathcal{S}$, we train a policy mdoel, $\pi_\theta(a|\mathcal{C}(\Phi_{enc}(o)))$, parameterized by θ, which maps from robot's state representations to actions. Here, $\mathcal{C}$ represents an optional concatenation operation that effectively fuses multi-view and multi-frame visual features, along with the robot's proprioceptive state in the channel dimension. We optimize the π_θ through a standard behavior cloning MSE loss:

$$\min_{\theta} \sum\nolimits_{\boldsymbol{(o,a)\sim\mathcal{S}}} \mathrm{MSE}(a, \pi_\theta(\mathcal{C}(\Phi_{enc}(o)))). \tag{4}$$

4 Experiments

4.1 Implementation on Pre-training

We execute pre-training with data from EgoVLP [55] for comprehensive ablation and fair comparison. It processes untrimmed videos of Ego4D and filters out that miss language narrations and belong to validation or test sets, resulting in a total of 3.8 million clips, called as Egoclip. In pre-training, we sample a frame pair from each clip for training. As for all experiments, we employ ViT [22] as backbone. Additionally, we maintain consistency with prior works [73, 59], directly using the [CLS] token as the global representation. The pre-training hyperparameters can be found in section A.3.

4.2 Implementation on Downstream Policy

Evaluation Scheme. Following popular settings on PVRs for robotic motor control [65, 46, 59], for each task, we learn a single policy π which is structured as a MLPs network. The policy models utilize both the history of visual observation embeddings and optional robot proprioceptive as inputs, subsequently generating executable actions as outputs.

Simulation Tasks. We select the union of manipulation and locomotion tasks from prior works [65, 59] for evaluation, encompassing 19 tasks across 5 simulated environments. These inclue Meta-World [104] (Assembly, Bin-Picking, Button-Press, Drawer-Open, and Hammer), Franka-Kitchen [31] (Sliding Door, Turning Light On, Opening Door, Turning Knob, and Opening Microwave), Adroit [74] (Relocate and Reorient-Pen), DMControl [83] (Finger-Spin, Reacher-Hard, Cheetah-Run, Walker-Stand, and Walker-Walk), and Trifinger [98] (Reach-Cube and Push-Cube). More detailed simulation evaluation details can be found in section A.4.

Real-World Tasks. In our real-world experiments, we evaluate contact-rich picking and pouring tasks using a Franka Emika Research 3 robot arm in a tabletop environment, ensuring no duplication with simulation Franka-Kitchen [31]. For each task, we collect 100 noise demonstrations for training, and we conduct 20 trials per task during evaluation phase. The robotic arm and objects have different initial pose between training and testing. The evaluation demonstrations of our real-world tasks is shown in Figure 3. Please see section A.5 for more real-world setup details.

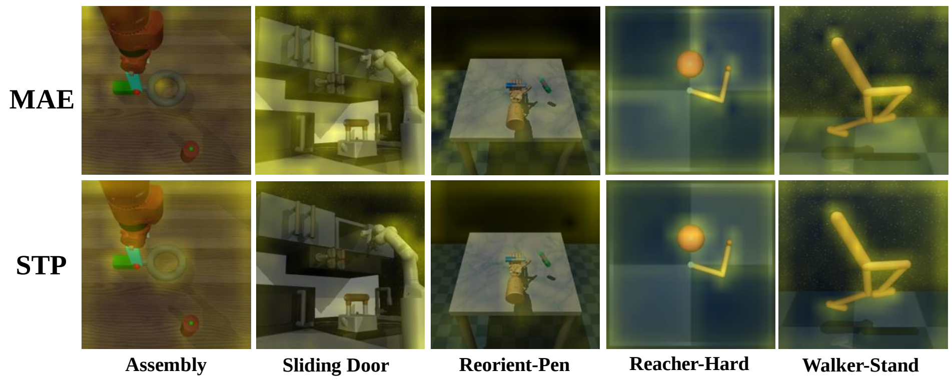

Figure 4: **Attention Visualization.** We use the [CLS] token as query, average the attention of all heads at the last layer of the frozen ViT encoder, and perform min-max normalization. We then upsample the attention map and overlay it on the original image, where the size of the attention value is directly proportional to the intensity of the yellow light. **Top:** MAE pre-training. **Bottom:** STP pre-training.

4.3 Performance on Downstream Simulation Tasks

In this section, we mainly analyze the performance of some pre-trained image representations on reproducible simulation tasks. Specifically, we first evaluate the following models: (1) public DINOv2 [67] that combines masked image modeling with self-distillation on large-scale image datasets; (2) public CLIP [71] that conducts contrastive learning on large-scale image-text pairs; (3) R3M trained based on Egoclip [55]; (4) public VC-1 [59]; (5) MAE trained based on Egoclip; (6) STP trained based on Egoclip. (7) STP that conducts hybrid pre-training with initialization using ImageNet-MAE [59]. Among them, (1) and (2) achieve excellent performance on core visual understanding tasks using zero-shot or linear probing evaluation settings. (3) and (4) utilize egocentric videos for robotic motor control. (5), (6) and (7) are used for fair comparison and exploring the potential benefits of STP from more diverse image data, respectively. The experimental results are presented in Table 1. Consistent with prior findings [41, 59], there is not a universal foundation model that performs optimally across all benchmarks. However, on the whole, the MAE method is superior due to its effective modeling of low-level geometry and spatial structure, especially for the MetaWorld tasks that demand fine-grained control. Another intriguing observation is that MAE underperforms in the Franka-Kitchen and Adroit tasks. We believe that this could be due to its relatively weaker semantic representation. Under a fair comparison, our STP outperforms MAE by 4.1 (59.6 $\rightarrow$ 63.7), and additionally benefits from a more diverse image data, improving by 0.5 (63.7 $\rightarrow$ 64.2). This is attributed to that our STP not only captures static content features but also effectively models motion information by extracting temporal clues from videos of interactions and manipulations with the environment and objects. Additionally, we provide the visualization of the attention maps (model (5) and (6)) of several specific tasks in Figure 4. The results indicate that, on top of effectively capturing spatial information, our method further encourages the model to focus on motion areas or objects, thereby providing a more *sparse and compact* representation for downstream low-data BC paradigm.

Next, we also evaluate and compare the adaptation results of our representations to downstream motor control tasks. Specifically, we evaluate following settings: (a) The MAE pre-trained representation undergoes further MAE post-pre-training with task-specific data, and is frozen during policy training; (b) The STP pre-trained representation undergoes further STP post-pre-training with task-specific data, and is frozen during policy training; (c) The STP pre-trained representation undergoes end-to-end fine-tuning with task-specific data; (d) STP pre-training is performed directly using task-specific data and the resulting representation is frozen during policy training. The results show that end-to-end fine-tuning fails to yield the best results, suggesting that naively fine-tuning VIT-base could still lead to overfitting under few-shot behavior cloning scheme. Conversely, (a) and (b) achieve competitive results, with our STP achieving a 3.9 (72.5 $\rightarrow$ 76.4) improvement on the weight average success rate than MAE, further demonstrating the effectiveness and data efficiency of our STP for in-domain data. In addition, the comparison between (a) and (d) also proves the effectiveness of pre-training with out-of-domain data. Finally, we also scale up both MAE and our STP to ViT-L/16, and the results still demonstrate the superiority of STP. Among them, compared to ViT-B/16, ViT-L/16 brings a smaller performance improvement, which may be due to the task's performance saturation. However, the ViT-L/16 of STP does not show improvement in Meta-World and Trifinger, indicating that simply

Table 1: Performance comparations of visual representations on simulation benchmarks. We report the average score across all tasks for each simulation environment. DINOv2 uses **ViT-B/14**, CLIP uses **ViT-B/32**, and unless otherwise specified, others use **ViT-B/16**. Mt-Wd, Fr-Ki, DMC, Adro, Tr-fi, and WA respectively represent MetaWorld, Franka-Kitchen, DMControl, Adroit, Trifinger, and weight average. * denotes that public VC-1 samples image frmaes form full Ego4D dataset.

	Pre-training Data	Mt-Wd	Fr-Ki	DMC	Adro	Tr-fi	WA
DINOv2 [67]	LVD-142M	77.9	41.2	59.4	50.7	69.0	59.6
CLIP [71]	Image-text pairs	75.5	39.8	52.2	**51.3**	57.7	55.6
R3M [65]	Ego	81.3	30.6	52.2	46.7	64.7	54.9
VC-1 [59]	Ego*+MNI	88.8	38.4	60.9	46.0	70.5	61.8
MAE [38]	Ego	85.1	36.7	59.2	43.4	**70.6**	59.6
STP	Ego	92.0	40.9	**62.1**	48.0	69.3	63.7
STP	Ego+I	**94.1**	**42.5**	61.6	47.3	66.7	**64.2**
MAE (Post PT)	Ego+Demo	93.6	46.9	81.1	58.0	76.8	72.5
STP (Post PT)	Ego+Demo	**97.3**	**53.6**	**82.8**	**63.3**	**78.0**	**76.4**
STP (E2E FT)	Ego	87.2	52.4	55.2	40.0	70.4	62.9
STP	Demo	70.3	30.4	52.5	38.0	70.8	51.8
MAE-L/16 (Post PT)	Ego+Demo	95.7	54.7	83.5	66.0	**77.6**	76.7
STP-L/16 (Post PT)	Ego+Demo	**97.3**	**57.4**	**85.0**	**70.0**	75.4	**78.4**

Table 2: The ablation experiment results. Me, Fra, DMC, Adr, Tri, and WA respectively represent MetaWorld, Franka-Kitchen, DMControl, Adroit, Trifinger, and weight average. All models use **ViT-B/16**.

(a) Current Frame Masking and Spatial Prediction.

ρ^c	Predict	Me	Fra	DMC	Adr	Tri	WA
75%	✓	**92.0**	**40.9**	**62.1**	**48.0**	**69.3**	**63.7**
75%		84.5	34.7	55.4	43.3	65.3	57.4
50%	✓	82.1	36.0	60.3	**48.0**	66.8	59.0
0%		79.2	39.7	54.8	44.0	63.1	57.0

(b) Temporal Prediction Condition Design.

Condition	Me	Fra	DMC	Adr	Tri	WA
L-E	82.1	30.7	55.5	42.0	63.8	55.4
95%	**92.0**	40.9	62.1	**48.0**	69.3	**63.7**
90%	91.2	**42.5**	62.8	44.7	65.9	63.4
L-E + 95%	91.0	37.7	**64.1**	46.7	**70.8**	63.1
L-D + 95%	88.0	34.3	62.6	46.7	69.3	60.9

(c) Temporal Decoder Architecture Design.

Decoder	Me	Fra	DMC	Adr	Tri	WA
8 joint-self	87.7	36.9	55.7	46.0	71.3	59.8
12 joint-self	88.5	35.0	55.7	46.0	67.0	59.1
8 self-cross	**92.0**	**40.9**	**62.1**	**48.0**	69.3	**63.7**

(d) Frame Sampling Strategy.

Frame interval	Me	Fra	DMC	Adr	Tri	WA
8	89.6	39.9	58.4	46.0	67.0	61.3
16	92.0	40.9	**62.1**	**48.0**	**69.3**	**63.7**
24	89.1	**41.1**	61.5	46.0	68.1	62.5
8, 24	**92.3**	37.1	57.3	42.0	68.4	60.8

scaling up model capacity does not necessarily lead to performance gains. In the few-shot BC setting, there is a risk of overfitting in both policy and backbone training.

4.4 Ablation on Downstream Simulation Tasks

In this section, we perform extensive ablation studies to further demonstrate the effectiveness of our joint spatial and temporal prediction, as well as temporal prediction condition design. In addition, we also study the influence of temporal decoder architecture design and future frame sampling strategy.

Current frame masking. The design of the current frame masking is crucial. On one hand, similar to MAE [38], masking some patches and predicting the missing parts can effectively promote the learning of image content features. On the other hand, the visible patches of the current frame need to interact with the condition to predict the future frame. Specifically, we mask the current frame at masking rates of 75%, 50%, and 0%, respectively, and optionally predict the missing parts through the spatial decoder. The results are shown in Table 2 (a). From results, we see that the masking ratio of 75% and performing spatial prediction still lead to the best performance. This demonstrates the importance of retaining MAE [38] for content features learning, especially for low-level manipulation in Meta-World, while a current frame with a high masking ratio (75%) is sufficient to interact with other conditions to predict the future frame.

Temporal prediction condition design. Subsequently, we discuss the influence of temporal prediction condition design. We implicitly model motion in actionless video data by predicting the pixels of the future frame. A direct and simple idea is to use language narration as a condition. The text tokens can be flexibly utilized as inputs to ViT [22], forming a multimodal encoder. Language narration provides a high-level behavior description, but lacks low-level visual dynamic priors for pixel-level prediction. However, leaking part of the future frame can effectively provide these priors. In order to explore how to construct a more meaningful temporal prediction proxy task, we compare the following schemes: (1) only language narration, (2) masking 95% of the future frame, (3) masking 90% of the future frame, (4) masking 95% of the future frame and language narration, and (5) masking 95% of the future frame and language narration, but the language is added in the temporal decoder, instead of being fused with the visible image patches in the multimodal encoder. We tokenize all language narration by pre-trained DistilBERT [75]. The results are shown in Table 2 (b). From results, we see that using only language as a prediction condition leads to a significant decline in performance, while leaking a small amount of future frame (masking 95%) in the temporal decoder can achieve competitive results. As for joint conditions of language and future frame with 95% masking ratio, adding language in the encoder is better than in the decoder. Additionally, adding language performs better on DMControl (64.1 vs. 62.1) and Trifinger (70.8 vs. 69.3), while not adding language performs better on Meta-World (92.0 vs. 91.0), Franka-Kitchen (40.9 vs. 37.7) and Adroit (48.0 vs. 46.7). We speculate the reasons for language hurts performance are as follows: (i) The input gap (multi-modal and single-modal) between upstream and downstream; (ii) Extra language in ViT may result in the loss of some fine-grained information capture. Furthermore, the latter does not require language supervision, and can provide a more scalable self-supervised solution.

Temporal decoder design. We also investigate the impact of the temporal decoder design. Specifically, we consider two types of decoder blocks. One is the joint-self architecture, as shown in Figure 2 (a), and similar joint architecture are adopted in [26, 102]. The other is the self-cross architecture, as shown in Figure 2 (b), and similar cross architecture are adopted in [3, 33]. We consider the following settings: (1) 8 joint-self decoder blocks, (2) 12 joint-self decoder blocks, (3) 8 self-cross decoder blocks. Among them, setting (2) and (3) have similar amounts of parameters for a fairer comparison. The results are shown in Table 2 (c). The results demonstrate the importance of maintaining a fixed representation space of the past frame during temporal prediction.

Frame sampling strategy. Finally, we investigate the impact of the sampling strategy between the current frame and future frame. The difficulty of temporal prediction is directly proportional to the frame interval values. We establish four settings where we fix the sampling intervals at 8, 16, and 24 respectively, and for the fourth setting, we randomly select an interval within the range of [8, 24]. The results are shown in Table 2 (d). The results show that an interval of 16 achieves the best balance for building temporal prediction proxy task.

4.5 Performance on Downstream Real-world Tasks

In this section, we report our experiment results on real-world picking and pouring tasks. We report the average success rate for each task. Specifically, we compare STP with the baseline MAE, both of which are trained on out-of-domain videos and kept frozen during policy training. The results are shown in Table 3. From the results, it can be seen that STP has achieved significant advantages in the pouring task. It can more accurately align with the moving bowl and the pot. In addition, although MAE and STP have a same success rate in picking tasks, STP tends to execute grasping in a better position. This indicates that the trend and conclusion of our STP are consistent in both simulation and the real-world, which also aligns with the findings of [79].

Table 3: Performance comparations on real-world tasks.

Method	Picking	Pouring	Average
MAE	65.0	45.0	55.0
STP	65.0	65.0	**65.0**

5 Conclusion

In this work, we have proposed the STP, a simple, efficient and effective self-supervised visual representation pre-training framework for robotic motor control. Our STP jointly performs spatiotemporal predictive learning on large-scale videos within a multi-task learning manner. Our STP captures content features by predicting the invisible areas within the masked current frame, and simultaneously captures motion features by using a future frame with an extremely high masking ratio as a condition to predict the invisible areas within that future frame. We carry out the largest-scale BC evaluation of PVRs for robotic motor control to date to demonstrate the effectiveness of STP. Furthermore, as for pre-training data, we also prove that extending STP to hybrid pre-training and post-pre-training could further unleash its generality and data efficiency.

References

[1] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *IJRR*, 39(1):3–20, 2020.

[2] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, pages 15619–15629, 2023.

[3] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. In *ECCV*, pages 348–367. Springer, 2022.

[4] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *ICML*, pages 1298–1312. PMLR, 2022.

[5] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. In Kris Hauser, Dylan A. Shell, and Shoudong Huang, editors, *RSS*, 2022.

[6] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *CVPR*, pages 13778–13790, 2023.

[7] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2021.

[8] Adrien Bardes, Jean Ponce, and Yann LeCun. Mc-jepa: A joint-embedding predictive architecture for self-supervised learning of motion and content features. *arXiv preprint arXiv:2307.12698*, 2023.

[9] Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, Alex X Lee, Maria Bauza, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, et al. Robocat: A self-improving foundation agent for robotic manipulation. *arXiv preprint arXiv:2306.11706*, 2023.

[10] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, 2023.

[11] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong T. Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: robotics transformer for real-world control at scale. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu, editors, *RSS*, 2023.

[12] Shaofei Cai, Zihao Wang, Xiaojian Ma, Anji Liu, and Yitao Liang. Open-world multi-task control through goal-aware representation learning and adaptive horizon prediction. In *CVPR*, pages 13734–13744, 2023.

[13] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021.

[14] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020.

[15] X Chen, S Xie, and K He. An empirical study of training self-supervised vision transformers. in 2021 ieee. In *ICCV*, pages 9620–9629.

[16] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu, editors, *RSS*, 2023.

[17] Yutao Cui, Cheng Jiang, Gangshan Wu, and Limin Wang. Mixformer: End-to-end tracking with iterative mixed attention. *arXiv preprint arXiv:2302.02814*, 2023.

[18] Yilun Dai, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *arXiv preprint arXiv:2302.00111*, 2023.

[19] Sudeep Dasari, Mohan Kumar Srirama, Unnat Jain, and Abhinav Gupta. An unbiased look at datasets for visuo-motor pre-training. In *CoRL*, 2023.

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.

[21] AM Derrington and P Lennie. Spatial and temporal contrast sensitivities of neurones in lateral geniculate nucleus of macaque. *The Journal of physiology*, 357(1):219–240, 1984.

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

[23] Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, et al. Video language planning. *arXiv preprint arXiv:2310.10625*, 2023.

[24] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 2023.

[25] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *CVPR*, pages 19358–19369, 2023.

[26] Xinyang Geng, Hao Liu, Lisa Lee, Dale Schuurams, Sergey Levine, and Pieter Abbeel. Multimodal masked autoencoders learn transferable representations. *arXiv preprint arXiv:2205.14204*, 2022.

[27] Abraham George, Alison Bartsch, and Amir Barati Farimani. Openvr: Teleoperation for manipulation. *arXiv preprint arXiv:2305.09765*, 2023.

[28] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *ICCV*, pages 13505–13515, 2021.

[29] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, pages 18995–19012, 2022.

[30] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. *arXiv preprint arXiv:2311.18259*, 2023.

[31] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *CoRL*, pages 1025–1037. PMLR, 2020.

[32] Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. Maskvit: Masked visual pre-training for video prediction. In *ICLR*, 2022.

[33] Agrim Gupta, Jiajun Wu, Jia Deng, and Li Fei-Fei. Siamese masked autoencoders. 2023.

[34] Huy Ha, Pete Florence, and Shuran Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *Conference on Robot Learning*, pages 3766–3777. PMLR, 2023.

[35] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCV Workshops*, pages 0–0, 2019.

[36] Nicklas Hansen, Zhecheng Yuan, Yanjie Ze, Tongzhou Mu, Aravind Rajeswaran, Hao Su, Huazhe Xu, and Xiaolong Wang. On pre-training for visuo-motor control: Revisiting a learning-from-scratch baseline. *arXiv preprint arXiv:2212.05749*, 2022.

[37] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. *NeurIPS*, 35:27953–27965, 2022.

[38] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022.

[39] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020.

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[41] Yingdong Hu, Renhao Wang, Li Erran Li, and Yang Gao. For pre-trained vision models in motor control, not all policy learning methods are created equal. *arXiv preprint arXiv:2304.04591*, 2023.

[42] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-Z: zero-shot task generalization with robotic imitation learning. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *CoRL*, volume 164 of *Proceedings of Machine Learning Research*, pages 991–1002. PMLR, 2021.

[43] Chuhao Jin, Wenhui Tan, Jiange Yang, Bei Liu, Ruihua Song, Limin Wang, and Jianlong Fu. Alphablock: Embodied finetuning for vision-language reasoning in robot manipulation. *arXiv preprint arXiv:2305.18898*, 2023.

[44] Ya Jing, Xuelin Zhu, Xingbin Liu, Qie Sima, Taozheng Yang, Yunhai Feng, and Tao Kong. Exploring visual pre-training for robot manipulation: Datasets, models and methods. *arXiv preprint arXiv:2308.03620*, 2023.

[45] Yuanchen Ju, Kaizhe Hu, Guowei Zhang, Gu Zhang, Mingrun Jiang, and Huazhe Xu. Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation. *arXiv preprint arXiv:2401.07487*, 2024.

[46] Siddharth Karamcheti, Suraj Nair, Annie S. Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu, editors, *RSS*, 2023.

[47] Heecheol Kim, Yoshiyuki Ohmura, and Yasuo Kuniyoshi. Multi-task robot data for dual-arm fine manipulation. *arXiv preprint arXiv:2401.07603*, 2024.

[48] Moo Jin Kim, Jiajun Wu, and Chelsea Finn. Giving robots a hand: Learning generalizable manipulation with eye-in-hand human video demonstrations. *arXiv preprint arXiv:2307.05959*, 2023.

[49] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[50] Andreas Kleinschmidt, Kai V Thilo, Christian Büchel, Michael A Gresty, Adolfo M Bronstein, and Richard SJ Frackowiak. Neural correlates of visual-motion perception as object-or self-motion. *Neuroimage*, 16(4):873–882, 2002.

[51] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. *arXiv preprint arXiv:2310.08576*, 2023.

[52] Xiangwen Kong and Xiangyu Zhang. Understanding masked image modeling via learning occlusion invariant feature. In *CVPR*, pages 6241–6251, 2023.

[53] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *JMLR*, 17(1):1334–1373, 2016.

[54] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023.

[55] Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Z XU, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, Weijie Kong, et al. Egocentric video-language pretraining. *NeurIPS*, 35:7575–7586, 2022.

[56] Xingyu Lin, John So, Sashwat Mahalingam, Fangchen Liu, and Pieter Abbeel. Spawnnet: Learning generalizable visuomotor skills from pre-trained networks. *arXiv preprint arXiv:2307.03567*, 2023.

[57] Yecheng Jason Ma, William Liang, Vaidehi Som, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. *arXiv preprint arXiv:2306.00958*, 2023.

[58] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. In *ICLR*, 2023.

[59] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Pieter Abbeel, Jitendra Malik, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? 2023.

[60] Pierre Marza, Laetitia Matignon, Olivier Simonin, and Christian Wolf. Task-conditioned adaptation of visual features in multi-task policy learning. *arXiv preprint arXiv:2402.07739*, 2024.

[61] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[62] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu, editors, *RSS*, 2023.

[63] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *CVPR*, pages 2630–2640, 2019.

[64] Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. EmbodiedGPT: Vision-language pre-training via embodied chain of thought. In *NeurIPS*, 2023.

[65] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *CoRL*, 2022.

[66] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

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

[68] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[69] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. In *ICML*, pages 17359–17371. PMLR, 2022.

[70] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *ECCV*, pages 570–587. Springer, 2022.

[71] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICLR*, pages 8748–8763. PMLR, 2021.

[72] Ilija Radosavovic, Baifeng Shi, Letian Fu, Ken Goldberg, Trevor Darrell, and Jitendra Malik. Robot learning with sensorimotor pre-training. 2023.

[73] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *CoRL*, pages 416–426. PMLR, 2023.

[74] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *RSS*, 2018.

[75] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[76] Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *CoRL*, pages 1332–1344. PMLR, 2023.

[77] Younggyo Seo, Kimin Lee, Stephen L James, and Pieter Abbeel. Reinforcement learning with action-free pre-training from videos. In *ICML*, pages 19561–19579. PMLR, 2022.

[78] Mohit Sharma, Claudio Fantacci, Yuxiang Zhou, Skanda Koppula, Nicolas Heess, Jon Scholz, and Yusuf Aytar. Lossless adaptation of pretrained vision models for robotic manipulation. In *ICLR*, 2023.

[79] Sneha Silwal, Karmesh Yadav, Tingfan Wu, Jay Vakil, Arjun Majumdar, Sergio Arnaud, Claire Chen, Vincent-Pierre Berges, Dhruv Batra, Aravind Rajeswaran, et al. What do we learn from a large-scale study of pre-trained visual representations in sim and real environments? *arXiv preprint arXiv:2310.02219*, 2023.

[80] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, pages 843–852. PMLR, 2015.

[81] Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul Wohlhart, Brianna Zitkovich, Fei Xia, Chelsea Finn, et al. Open-world object manipulation using pre-trained vision-language models. In *CoRL*, 2023.

[82] Yanchao Sun, Shuang Ma, Ratnesh Madaan, Rogerio Bonatti, Furong Huang, and Ashish Kapoor. Smart: Self-supervised multi-task pretraining with control transformers. In *ICLR*, 2023.

[83] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[84] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, et al. Octo: An open-source generalist robot policy, 2023.

[85] Garrett Thomas, Ching-An Cheng, Ricky Loynd, Felipe Vieira Frujeri, Vibhav Vineet, Mihai Jalobeanu, and Andrey Kolobov. Plex: Making the most of the available data for robotic manipulation pretraining. In *CoRL*, pages 2624–2641. PMLR, 2023.

[86] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *NeurIPS*, 35:10078–10093, 2022.

[87] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICLR*, pages 10347–10357. PMLR, 2021.

[88] David C Van Essen and Jack L Gallant. Neural mechanisms of form and motion processing in the primate visual system. *Neuron*, 13(1):1–10, 1994.

[89] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

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

[91] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.

[92] Jianren Wang, Sudeep Dasari, Mohan Kumar Srirama, Shubham Tulsiani, and Abhinav Gupta. Manipulate by seeing: Creating manipulation controllers from pre-trained representations. In *ICCV*, pages 3859–3868, 2023.

[93] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *CVPR*, pages 14549–14560, 2023.

[94] Lirui Wang, Jialiang Zhao, Yilun Du, Edward H Adelson, and Russ Tedrake. Poco: Policy composition from and for heterogeneous robot learning. *arXiv preprint arXiv:2402.02511*, 2024.

[95] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *CVPR*, pages 14668–14678, 2022.

[96] William F Whitney, Rajat Agarwal, Kyunghyun Cho, and Abhinav Gupta. Dynamics-aware embeddings. In *ICLR*, 2020.

[97] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023.

[98] Manuel Wuthrich, Felix Widmaier, Felix Grimminger, Shruti Joshi, Vaibhav Agrawal, Bilal Hammoud, Majid Khadiv, Miroslav Bogdanovic, Vincent Berenz, Julian Viereck, et al. Trifinger: An open-source robot for learning dexterity. In *CoRL*, pages 1871–1882. PMLR, 2021.

[99] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

[100] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *CVPR*, pages 9653–9663, 2022.

[101] Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. Xskill: Cross embodiment skill discovery. In *CoRL*, 2023.

[102] Jiange Yang, Sheng Guo, Gangshan Wu, and Limin Wang. Comae: Single model hybrid pre-training on small-scale rgb-d datasets. *arXiv preprint arXiv:2302.06148*, 2023.

[103] Jiange Yang, Wenhui Tan, Chuhao Jin, Bei Liu, Jianlong Fu, Ruihua Song, and Limin Wang. Pave the way to grasp anything: Transferring foundation models for universal pick-place robots. *arXiv preprint arXiv:2306.05716*, 2023.

[104] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, pages 1094–1100. PMLR, 2020.

[105] Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. *arXiv preprint arXiv:2401.11439*, 2024.

[106] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

[107] Haoyi Zhu, Yating Wang, Di Huang, Weicai Ye, Wanli Ouyang, and Tong He. Point cloud matters: Rethinking the impact of different observation spaces on robot learning. *arXiv preprint arXiv:2402.02500*, 2024.

A Appendix

A.1 Limitations and Discussion

Although STP has demonstrated superior performance in extensive experiments, there remain some challenges and future works. From the perspective of pre-training data, Ego4D provides numerous human-object interaction scenes and good motion clues. Building larger-scale and more diverse potential datasets such as [63, 30] to scale up STP is worth exploring. Regarding pre-training methods, exploring predictive targets outside of pixel space and more effective sampling and masking strategies present intriguing research directions. From an evaluation standpoint, we utilize a frozen ViT to extract agent state representations and adopt the paradigm of few-shot behavior cloning, other policy learning methods (reinforcement learning, visual reward function, visual task specification), have not been explored. In conclusion, as the first method of performing temporal prediction on large-scale videos for self-supervised visual representation learning intended for robotic motor control tasks, we hope STP can be taken as a strong baseline and facilitate further research along this direction.

A.2 The influence of the loss weight ratio between temporal prediction and spatial prediction

In this section, we further explore the influence of the loss weight ratio between temporal prediction and spatial prediction. Specifically, taking five tasks from Franka-Kitchen as examples, we load the pre-trained STP and perform post-pre-training with three different loss weight ratios (temporal to spatial). The results, as shown in Figure 5, are 54.7, 55.2, and 57.4 for the average results of the ratios 3:1, 1:3, and 1:1, respectively. The results indicate that due to the different attributes of the tasks, the trends are not consistent. However, overall, the 1:1 ratio achieves the best balance and results. We chose it as a universal setting.

A.3 Pre-training Details

In this section, we describe the details of our STP pre-training. Specifically, we list some key training and architectural hyperparameters of STP in Table 4. In addition, as for our MAE [38] baseline, we mainly follow the publicly available code of MAE[1]. Additionally, we train MAE and STP using the same data and number of epochs to ensure that the comparison between them is **completely fair**. Finally, we also provide some STP prediction results in Figure 6.

A.4 Simulation Environments Details

In this section, we first present further details of the STP post-pre-training on downstream simulation environments. Subsequently, we delineate the specific hyperparameters used in the behavior cloning policy training within these simulation environments. Finally, we provide the comprehensive evaluation scheme for each simulation environment.

[1] `https://github.com/facebookresearch/mae`

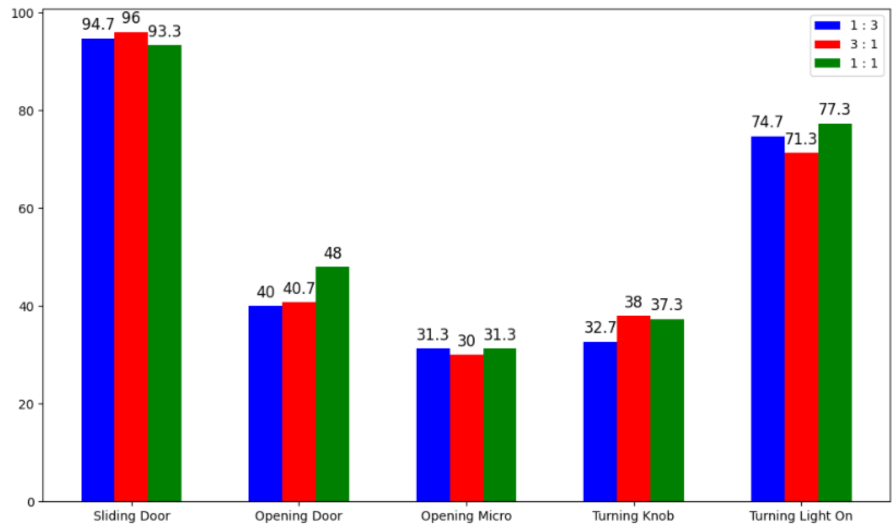

Figure 5: The results of different loss weight ratios between temporal prediction and spatial prediction.

Table 4: Training and architectural hyperparameters for STP pre-training.

Hyperparameter	**Value**
STP Pre-training	
optimizer	AdamW [49]
base learning rate	0.00015
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
effective batch size	4096
learning rate schedule	cosine decay
total epochs	50
warmup epochs	5
augmentation	RandomResizedCrop (0.8, 1)
Encoder ViT-base Architecture	
patch size	16
#layers	12
#MHSA heads	12
hidden dim	768
positional embedding	sin-cos initialization and fix
Dual Decoder ViT-base Architecture	
#layers	8
#MHSA heads	16
hidden dim	512
positional embedding	sin-cos initialization and fix

In regards to the STP post-pre-training, we utilize data that aligns with the policy training, and the specific architecture hyperparameters correspond to those listed in Table 4. Depending on the specific demonstration data, we adjust the values of total epochs, warmup epochs, effective batch size, and the frame interval, as shown in Table 5.

As for policy training and evaluation schemes, we primarily refer to the publicly available code[2] and training data of VC-1 [59] for Metaworld [104], DMControl [83], Adroit [74] and Trifinger [98]. Similarly, for Franka-Kitchen [31], we follow the public code[3] and training data of R3M [65]. Specifically, the policy training hyperparameters and evaluation schemes are shown in Table 6 and Table 7, respectively. About policy training, we completely follow the setting of prior works [65, 59] when freezing the encoder; when performing end-to-end fine-tuning, we make appropriate adjustments to the batch size and learning rate. About evaluation details, similar to prior works[65, 59], we establish all evaluation details such as the number of expert demonstrations and test trajectories, environmental viewpoints, optimization hyperparameters, base seeds, history windows size, and the use of robot proprioceptive. In Table 7, the term ‘prop.’ stands for whether proprioceptive information is used or not, and ‘history window size’ signifies the number of frames received by the policy model at each step, with features between frames being fused through concatenation. ‘Number of trajectories’ represents the quantity of trajectories evaluated. For tasks in Meta-World, Franka-Kitchen, Adroit, and Trifinger, we report the maximum success rate, whereas for tasks in DMControl, we report the maximum reward score, rescaling to be in the range of [0, 100] by dividing by 10. We report the average metric across tasks for each environment. In addition, it is worth noting that the metrics we report are the **average value across all base seeds and camera viewpoints**. Finally, we also report the results of our post-pre-training STP (ViT-B/16) on each task in Table 8.

In addition, we emphasize that different random seeds primarily affect the rendering of the initial frame in the sampled trajectories, as shown in Figure 7. During evaluation, the seed value we provide serves as the base seed, and the trajectory sampling process is depicted in Algorithm 1. **Therefore, the actual number of trajectories we evaluate is the number of trajectories multiplied by the number of base seeds.** For instance, for MetaWorld, we evaluate $25 \times 3 = 75$ trajectories, with random seeds for rendering being 100-124, 200-224, and 300-324.

Finally, for Franka-Kitchen, we utilize MuJoCo210, while all other simulation environments are based on MuJoCo200. Our policy training and evaluation environments are conducted on Cuda 11.3, NVIDIA TITAN Xp GPUs, and OpenGL 3.1.

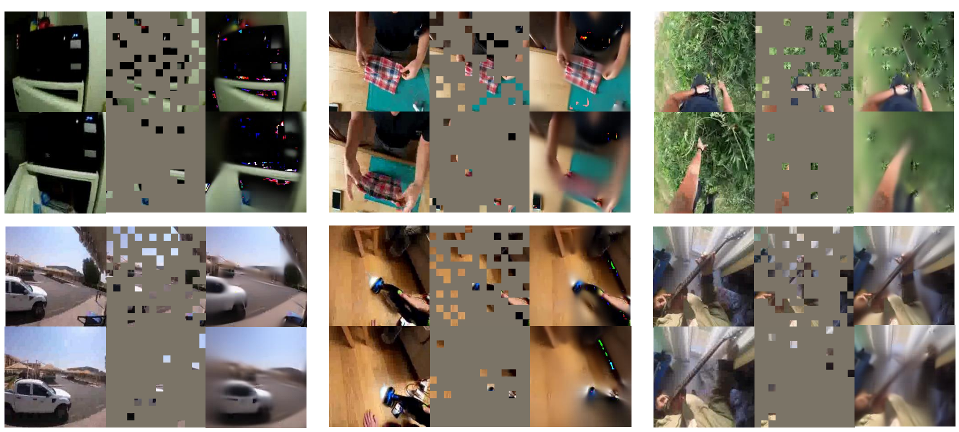

Figure 6: Some examples of our STP prediction result on Ego4D videos. For each six tuple, we show the ground-truth (left), masked frames (middle), STP prediciton results (right), current frames (top), and future frames (bottom). We simply overlay the output with the visible patches to improve visual quality.

[2]https://github.com/facebookresearch/eai-vc/tree/main/cortexbench

[3]https://github.com/facebookresearch/r3m/tree/eval/evaluation

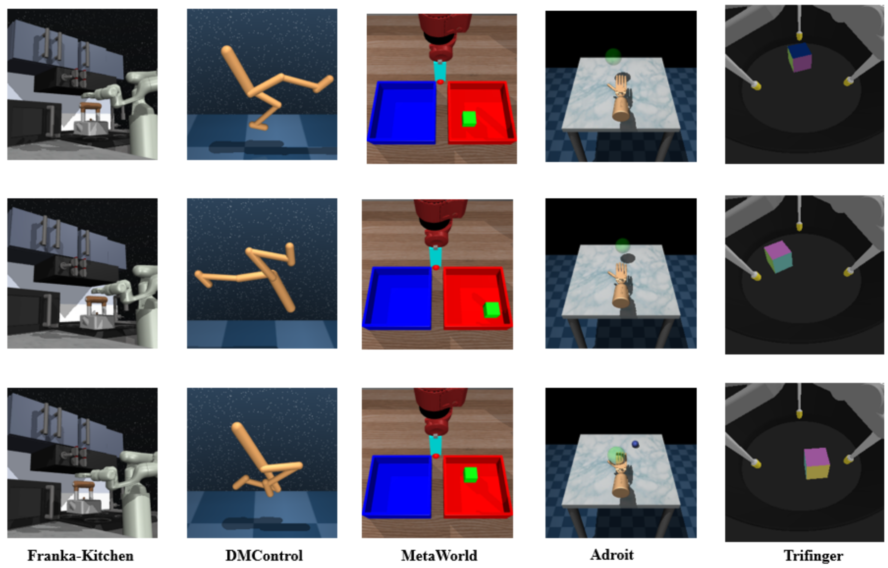

Figure 7: The visualization of initial frame rendering under different random seeds.

Algorithm 1 Trajectories Sampling Pseudocode

```
# num_traj: the number of evaluation trajectories
# base_seed: base seed for rollouts

# rollout to sample trajectories
    for ep in range(num_traj):
        seed = base_seed + ep
        env.set_seed(seed)
        o = env.reset()
```

A.5 Real-World Environments Details

In this section, we outline the details of our real-world setup and evaluation scheme. As depicted in Figure 8, our real-world scenario includes four camera viewpoints: top, left, right, and wrist. It includes two Kinect DK and two RealSense cameras. An example of four views is shown in Figure 9. Specifically, we utilize four different camera views and resize their resolution uniformly to 224×224. To effectively model the complex and multimodal action distribution in our real-world tasks, we select diffusion policy [16] as our policy model. In accordance with this approach, we concatenate the visual embeddings of all views from two sequential frames. Following the approach in [27], we collect robot data using a VR tele-operation setup. In this way, we collect 100 continuous trajectories for each task. It is worth noting that the quality of these demonstrations leaves room for improvement and contains a lot of noise. During the evaluation phase, we primarily evaluate two contact-rich tasks that have not appeared in Franka-Kitchen [31] benchmark: (1) Picking. It requires the robot arm to pick up the transparent bowel off the table; (2) Pouring. It requires the robot arm to pour at least three-quarter of the ingredients from the transparent bowl into the black pot. For each task, we change the initial pose of the robot arm and objects within a certain range as well as conduct 20 trials. In addition, there are different distractors on the desktop during training and testing, which also evaluates the robustness of the model to distractors. Throughout the process, we use ROS and MoveIt for hardware communication and motion planning.

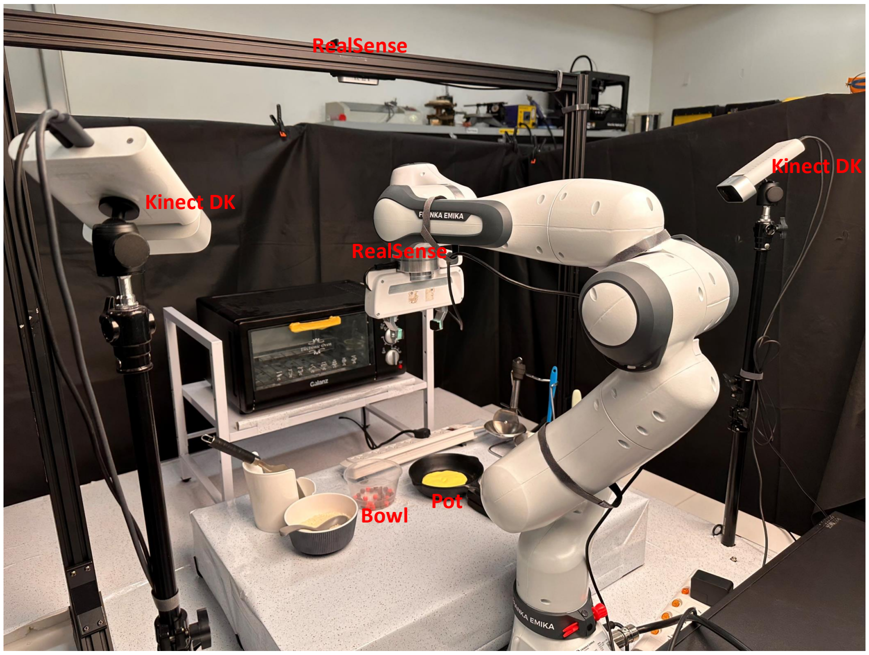

Figure 8: Our real-world scene with four cameras and a Franka Emika robot arm.

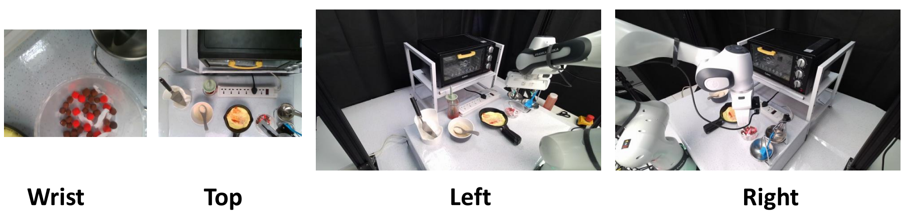

Figure 9: An example of four views.

Table 5: STP post-pre-training hyperparameters on simulation environments.

	MetaWorld	Franka-Kitchen	DMControl	Adroit	Trifinger
total epochs	50	100	50	50	50
warmup epochs	5	5	5	5	5
effective batch size	1024	128	2048	1024	1024
number of demonstrations	25	25	100	100	100
frame interval	4	4	4	4	16

Table 6: Policy training hyperparameters on simulation environments.

		MetaWorld	Franka-Kitchen	DMControl	Adroit	Trifinger
epochs		100	480	100	100	100 / 1000
batch size	frozen	256	32	256	256	32
	fine-tuning	64	32	64	64	16
learning rate	frozen	0.001	0.001	0.001	0.001	0.0001
	fine-tuning	0.00005	0.0001	0.00005	0.00005	0.0001

Table 7: Evaluation schemes on simulation environments.

Benchmark	Observation Space	History Window Size	Camera ViewPoints	Base Seeds	Number of Trajectories
Metaworld	RGB + prop.	3	top_cap2	100, 200, 300	25
Franka-Kitchen	RGB + prop.	1	left, right	123, 124, 125	50
DMControl	RGB	3	0	100, 200, 300	25
Adroit	RGB + prop.	1	vil_camera	100, 200, 300	25
Trifinger	RGB + prop.	1	default	10	25

Table 8: The success rate for each task on simulation bechmarks.

Assembly 94.7	Bin-Picking 97.3	Button-Press 94.7	Drawer-Open 100.0	Hammer 100.0
Sliding Door 96.0	Turning Light on 72.7	Opening Door 39.0	Turning Knob 31.3	Opening Microwave 29.0
Relocate 49.3	Reorient-Pen 77.3	Finger-Spin 69.6	Cheetah-Run 71.9	Reacher-Hard 87.7
Walker-Stand 95.9	Walker-Walk 89.0	Reach-Cube 85.3	Push-Cube 70.6	

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes, the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   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 discusses the limitations of the work performed by the authors.

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: [Yes]

   Justification: The paper does not include theoretical results.

   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 fully disclose all the information needed to reproduce results.

   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: We will release all codes and model weights on github.

   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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specify all the training and test details.

   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: [No]

   Justification: In our experiments, different seeds are primarily used for rendering different initial frames. Therefore, our evaluatation is comprehensive and sufficient, while our comparisons are absolutely fair.

   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 provide sufficient information on the computer resources.

   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: Our paper aligns with these.

   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: [NA]

    Justification: There is no societal impact of the work performed.

    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: The paper poses no such risks.

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: Yes.

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: [NA]

Justification: The paper does not release new assets.

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: The paper does not involve crowdsourcing nor research with 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: The paper does not involve crowdsourcing nor research with 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.