Token Bottleneck: One Token to Remember Dynamics

Taekyung Kim[1] **Dongyoon Han**[1] **Byeongho Heo**[1] **Jeongeun Park**[2] **Sangdoo Yun**[1]
[1]NAVER AI Lab [2]Korea University
{taekyung.k, dongyoon.han, bh.heo, sangdoo.yun}@navercorp.com baro0906@korea.ac.kr

Abstract

Deriving compact and temporally aware visual representations from dynamic scenes is essential for successful execution of sequential scene understanding tasks such as visual tracking and robotic manipulation. In this paper, we introduce Token Bottleneck (ToBo), a simple yet intuitive self-supervised learning pipeline that squeezes a scene into a bottleneck token and predicts the subsequent scene using minimal patches as hints. The ToBo pipeline facilitates the learning of sequential scene representations by conservatively encoding the reference scene into a compact bottleneck token during the squeeze step. In the reconstruction step, we guide the model to capture temporal dynamics by predicting the target scene using the bottleneck token along with few target patches as hints. This design encourages the vision backbone to embed temporal dependencies, thereby enabling understanding of dynamic transitions across scenes. Extensive experiments in diverse sequential tasks, including video label propagation and robot manipulation in simulated environments demonstrate the superiority of ToBo over baselines. Moreover, deploying our pre-trained model on physical robots confirms its robustness and effectiveness in real-world environments. We further validate the scalability of ToBo across different model scales. Code is available at https://github.com/naver-ai/tobo.

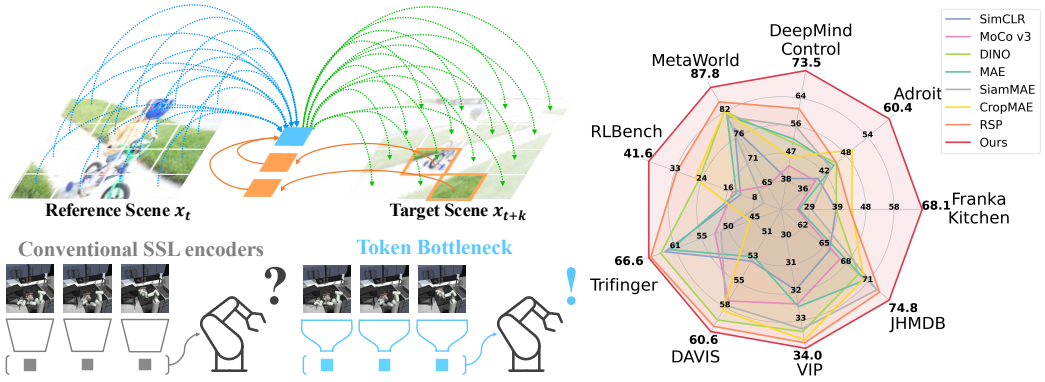
1 Introduction

With the increasing interest in deploying machines in real-world environments, ensuring seamless perception and interaction with their surroundings has emerged a crucial challenge. These operations are inherently sequential in nature, requiring the ability to trace objects (e.g., visual tracking) and predict future actions (e.g., manipulation) based on current and immediate past observations. Such understanding of the surrounding environments primarily depends on vision backbones. Therefore, a strong and robust backbone capable of generalizing across diverse tasks and environments is essential for effective sequential scene understanding.

Self-supervised learning (SSL) of visual representations has been highlighted as pivotal research in vision domains, with the pre-trained models being widely adopted for effective backbone deployment. A series of studies have introduced promising recipes for learning image [3, 4, 6, 7, 9, 17, 21] and video representations [39, 47] without labeled data. However, these studies primarily focus on understanding entire scenes or videos, which poses limitations for sequential scene understanding, as it requires capturing temporal changes across consecutive scenes and conservatively encoding the visual states of observed scenes.

To address the challenges, a sequence of studies [13, 19, 25] have attempted to incorporate correspondence learning into the MAE [21] framework, aiming to retain its strong localization capability while enabling the model to match corresponding regions across consecutive scenes. However, we observe that such additional considerations have a limited impact on the quality of scene representations and may result in suboptimal performance in sequential scene understanding tasks, such as robotic manipulation (§3.2). This limitation arises since recognizing temporal changes alone is insufficient;

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) Illustration of Token Bottleneck (ToBo) (b) Overall performance comparison

Figure 1: (a) We describe the underlying mechanism of our **Token Bottleneck (ToBo)** pipeline during pre-training, which conservatively encode a reference scene into a bottleneck token and predict the subsequent target scene based on a scarce target patches and the bottleneck token. ToBo facilitates learning the capability of temporal progression recognition and preservation of observed information (top). Therefore, using bottleneck tokens from the current and recent past observations enables the robot to better understand its current state (bottom). (b) Our method significantly surpasses previous self-supervised visual representation learning methods designed for static [4, 6, 9, 21] and dynamic scenes [19, 25, 52] on various robot manipulation and locomotion tasks.

these tasks require the ability to summarize the essential information from each scene without loss, while preserving temporal cues within the summarized representation.

In this paper, we introduce Token Bottleneck (ToBo), a simple yet effective SSL approach that intuitively facilitates the conservative summarization of observed scenes while enabling effective recognition of temporal evolution within the summarized representations. As illustrated in Fig. 1a, ToBo squeezes a reference scene into a bottleneck token and then predicts the subsequent target scene using only a minimal set of patches as hints. This design enforces strong reliance on the bottleneck token, encouraging the vision backbone to capture essential scene information. Moreover, predicting the target scene from the bottleneck token implicitly embeds temporal dependencies, guiding the vision backbone to generate representations capable of capturing dynamic transitions across consecutive scenes.

We conduct comprehensive experiments to assess the effectiveness of our pre-training pipeline in comparison with existing self-supervised learning methods. We evaluate our method on various sequential understanding tasks, including manipulation tasks in simulated environments and video label propagation tasks, surpassing baselines [4, 6, 9, 13, 19, 21, 25, 52] with significant gaps (see Fig. 1b). Furthermore, we deploy our pre-trained models on real-world robots, demonstrating strong generalization performance in unseen physical environments. Finally, we validate the scalability of our approach by observing consistent performance gains across various model scales.

2 Related Work

Self-supervised learning on a static scene. Self-supervised learning (SSL) approaches have been widely explored in the image domain. Contrastive learning approaches [3, 6, 8, 9, 20] aim to learn useful representations by maximizing the similarity between positive pairs derived from a static scene through strong augmentations. Although these methods excel in facilitating a cohesive understanding of images, they suffer from limited localization capabilities [30], essential for action prediction in robotics. On the other hand, masked image modeling (MIM) [1, 2, 21, 30, 31, 53] has recently gained attention for its promising capacity to learn visual representations through predictive learning. Inspired by masked language modeling (MLM) in transformers [11], BEiT [2] extends MLM into the vision domain, adopting an external offline tokenizer. MAE [21] and SimMIM [53] showcase efficient MIM by directly reconstructing masked input pixels without any tokenizer. However, these approaches do not incorporate mechanisms for capturing temporal progression during pre-training.

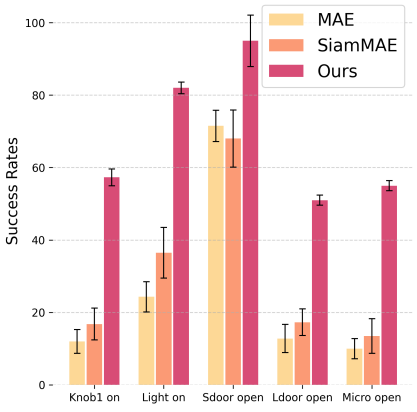


Figure 2: **Comparative analysis for motivation.** We compare robot manipulation performance using MAE and SiamMAE as visual backbones. While SiamMAE employ temporal correspondence to the limitation of MAE, its improvement over MAE remains limited.

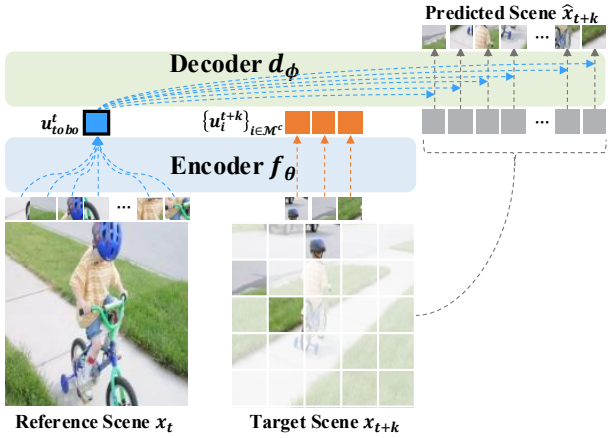


Figure 3: **Overview of our Token Bottleneck (ToBo).** Our ToBo reconstructs the masked patches from the *bottleneck token* representation of the reference scene $\mathbf{x}^t$ and extremely scarce patches from the target scene $\mathbf{x}^{t+k}$. Such extreme scarcity leads the decoder d_ϕ to rely heavily on the reference scene $\mathbf{x}^t$, facilitating the preservation of observed information in the *bottleneck token*.

Self-supervised learning on dynamic scenes. Recent studies have focused on enhancing the recognition of dynamic transitions. SiamMAE [19] proposes visual representation learning methods that utilize dynamic scenes. CropMAE [13] introduce a simple augmentation strategy that enables the generation of dynamic scenes even from a single static image. On the other hand, RSP employs stochastic frame prediction tasks along with masked autoencoding. Several works have also explored applying these techniques to embodied agents and robotic manipulation. For example, VC-1 [36], MVP [43], and Dasari et al. [10] adopt MAE objectives for visual pretraining, while STP [54] builds on SiamMAE with a reference masking strategy. On the other hand, some prior works investigate representation learning with annotated supervision. Theia [45] distills representations from large scale pre-trained teacher networks, some of which are trained with annotation supervision, into student models. MPI [27], Voltron [28], and R3m [37] explore language-driven representation learning, leveraging an auxiliary textual guidance through manually annotated data. In contrast, we focus on self-supervised learning directly from raw dynamic scenes without any guidance from annotations.

3 Method

3.1 Preliminary

Masked autoencoding. Given a scene image, we patchify the image into N non-overlapping $p\times p$-size patches $\{\mathbf{x}_i\}_{i=1}^N$ where $\mathbf{x}_i \in \mathbb{R}^{3p^2}$. We randomly select a masked patch set $\mathcal{M} \subset \{1, 2, ..., N\}$ with a ratio $r \in (0, 1)$ where $|\mathcal{M}| = \lfloor rN \rfloor$. The remaining patches $\{\mathbf{x}_i\}_{i\in\mathcal{M}^c}$ fed into the encoder f_θ, becoming spatial representations $\{\mathbf{u}_i\}_{i\in\mathcal{M}^c}$ where $\mathbf{u}_i \in \mathbb{R}^d$ for encoder dimension d. Note that a learnable CLS token $e_{[CLS]}$ is also encoded with spatial representations as a part of the encoding process. The encoded tokens are reconstructed to N tokens by substituting masked positions to a mask token $\mathbf{m} \in \mathbb{R}^d$. i.e. $\mathbf{u}_i \leftarrow \mathbf{m}$ for $i \in \mathcal{M}$. The decoder d_ϕ gets $\{\mathbf{u}_i\}_{i=1}^N$ as input and predicts the masked image patches $\{\hat{\mathbf{x}}_i\}_{i\in\mathcal{M}}$ using encoded tokens.

3.2 Motivation

In this section, we discuss the pros and cons of previous self-supervised learning (SSL) approaches from a sequential scene understanding perspective, which motivated our method.

Limited temporal evolution awareness of MAE. MAE [21] has been recognized for its strong localization capability, leading several studies [10, 36, 43] to adopt the recipe. This stems from its design that enforces the autoencoder to predict missing information based on available prior information (i.e., visible patches). This pipeline implicitly encourages the encoder to facilitate interactions among the

remaining sparse tokens, thereby enhancing its localization capability. However, since MAE performs predictive learning on a single static scene, the encoder is not explicitly trained to handle dynamic transitions over time, leading to limited performance in sequential scene understanding tasks (Fig. 2). Moreover, a recent study reveals that MAE falls short in learning broader contexts [30], leading to representations with a limited cohesive understanding of observed scenes. These limitations further constrains its potential to effectively understand sequential scenes.

Suboptimal impacts of SiamMAE in sequential scene understanding. To alleviate the chronic limitations of static scene-based SSL approaches, SiamMAE [19] builds a non-trivial correspondence matching problem by randomly sampling two dynamic scenes from sequential data. The core principle involves propagating patches from the reference scene to their corresponding locations in the target scene. Applying this guidance with a cross-attention layer-based decoder encourages fine-grained patch-wise similarity between target patches and reference patches. This process ultimately enforces the encoder to generate similar representations for corresponding patches. However, while SiamMAE enables capturing correspondences among consecutive scenes, despite being built upon the MAE framework, its impact over MAE is marginal or even negative in some sequential scene-based tasks (Fig. 2). In these tasks, since actions are predicted through a policy network based on the estimated visual states of both the observed and immediate past scenes, this suggests that considering temporal evolution recognition is insufficient for sequential scene understanding, and a conservative summarization of the observed scenes is essential.

3.3 The Proposed Method - Token Bottleneck (ToBo)

Our claim. Our goal is to achieve representations optimized for resolving sequential scene-based tasks. In light of the discussions in §3.2, we extend our focus beyond simply recognizing temporal evolution; we consider the conservative summarization of observed scenes in a way that also effectively embeds temporal dynamics within the summarized representation.

To this end, we present Token Bottleneck (ToBo), a self-supervised visual representation learning pipeline that enables these capabilities through a token bottleneck mechanism. ToBo consists of two key steps: squeezing a scene into a single token, which we denote as the *bottleneck token*, and reconstructing information from this token. Suppose a reference scene and a target scene are given. In the squeeze step, visual information from the reference scene is compactly encoded into the bottleneck token. Subsequently, in the reconstruction step, we guide the model to predict the target scene using the bottleneck token, with only a minimal set of patches from the target scene provided as hints. In this situation, the model cannot precisely reconstruct the target scene based solely on the limited hints, which strengthens the reliance of the reconstruction step on the bottleneck token. This design yields two advantages: (1) the bottleneck token should preserve essential information from the reference scene, and (2) such information should be encoded in a way that enables recognition of temporal dynamics when interleaved with the hints from the target scene. Eventually, our goal can be achieved by optimizing the objective of the Token Bottleneck pipeline. The overall description of our pipeline is depicted in Fig. 3.

Overall pipeline formulation. Suppose we sample a reference scene $\mathbf{x}^t \in \mathbb{R}^{3\times H\times W}$ and a target scene $\mathbf{x}^{t+k} \in \mathbb{R}^{3\times H\times W}$ with a temporal gap k. We patchify $\mathbf{x}^t$ and $\mathbf{x}^{t+k}$ into N non-overlapping patches $\{\mathbf{x}_i^t\}_{i=1}^N$ and $\{\mathbf{x}_i^{t+k}\}_{i=1}^N$, respectively. The reference scene patches $\{\mathbf{x}_i^t\}_{i=1}^N$ are fed into an encoder f_θ, yielding spatial representations $\{\mathbf{u}_i^t\}_{i\in\mathcal{M}^c}$. We use the CLS token output from this encoding process as the bottleneck token $\mathbf{u}_{tobo}$, which will be guided to compactly summarize the reference scene. The target scene $\{\mathbf{x}_i^{t+k}\}_{i=1}^N$ is masked with an extremely high ratio $r \in (0, 1)$, where $\mathcal{M} \subset \{1, 2, ..., N\}$ and $|\mathcal{M}| = \lfloor rN \rfloor$. The unmasked target patches $\{\mathbf{x}_i^{t+k}\}_{i\in\mathcal{M}^c}$ are processed by the same encoder f_θ, producing $\{\mathbf{u}_i^{t+k}\}_{i\in\mathcal{M}^c}$ for the target scene. We then concatenate the bottleneck token $\mathbf{u}_{tobo}$ with the target representations $\{\mathbf{u}_i^{t+k}\}_{i\in\mathcal{M}^c}$ and fill mask tokens $\mathbf{m}$ for missing regions $i \in \mathcal{M}$. These are passed to the decoder d_ϕ, which predicts the masked image patches $\{\hat{\mathbf{x}}_i^{t+k}\}_{i\in\mathcal{M}}$ by using $\mathbf{u}_{tobo}$ and $\{\mathbf{u}_i^{t+k}\}_{i\in\mathcal{M}^c}$. Due to the extremely high masking ratio applied to the target scene, the decoder d_ϕ proactively rely on $\mathbf{u}_{tobo}$, which enable the encoder f_θ to conservatively summarize the reference scene in a way that facilitates temporal reasoning when compared to the target hints. We minimize the reconstruction loss throughout the training as follows:

$$\mathcal{L}_{\text{ToBo}} = \sum_{i\in\mathcal{M}} d(\hat{\mathbf{x}}_i^{t+k}, \mathbf{x}_i^{t+k}), \tag{1}$$

Table 1: **Experimental results on vision-based robot policy learning on Franka Kitchen.** We report the performance of imitation learning agents on Franka Kitchen [18], which are trained upon representations from the ViT-S/16 model pre-trained on Kinetics-400 [29] dataset. The success rates (%) are reported for all the tasks. We underline the second-best performance. We report the gains of our method over the second-best baseline.

Tasks	SimCLR*	MoCo v3*	DINO*	MAE*	SiamMAE*	RSP*	CropMAE	ToBo
Knob1 on	25.3±2.1	11.5±3.9	27.0±3.2	12.0±3.3	16.8±4.4	31.0±2.4	<u>31.5</u>±5.3	**57.0**±2.0
Light on	<u>55.8</u>±6.4	24.3±5.0	44.3±6.5	24.3±4.2	36.5±7.0	44.5±5.6	54.0±11.2	**82.0**±1.6
Sdoor open	72.3±2.8	66.5±3.2	77.0±5.0	71.5±4.3	68.0±7.9	<u>82.5</u>±2.7	77.0±8.1	**95.0**±7.1
Ldoor open	17.0±2.9	10.3±2.1	16.5±2.5	12.8±3.9	17.3±3.7	<u>28.8</u>±4.8	25.5±5.7	**51.0**±1.4
Micro open	23.3±2.8	14.3±2.5	28.5±4.8	10.0±2.8	13.5±4.8	30.3±5.6	<u>32.5</u>±4.1	**55.0**±1.4

* Indicates results reported by Jang et. al. [25].

Table 2: **Experimental results on vision-based robot policy learning on CortexBench.** The performance of imitation learning agents on CortexBench [36] is reported, where the agents are trained upon representations from the ViT-S/16 model pre-trained on the Kinetics-400 [29] dataset. We report the normalized score for DeepMind Control Suite (DMC) and success rates (%) for other tasks. We report the gains of our method over the second-best baseline.

Tasks	SimCLR*	MoCo v3*	DINO*	MAE*	SiamMAE*	RSP*	CropMAE	ToBo
Adroit	40.4±3.3	39.6±4.3	45.6±6.2	39.6±4.3	44.0±6.6	45.6±4.6	<u>50.0</u>±5.1	**60.4**±2.2
MetaWorld	78.4±5.2	65.4±8.0	82.4±5.8	65.4±8.0	81.1±6.3	<u>84.5</u>±6.6	82.4±5.8	**87.8**±4.6
DMC	39.7±2.9	43.7±3.2	50.9±1.5	43.7±3.2	56.0±2.9	<u>61.6</u>±3.4	46.4±1.1	**73.5**±0.9
TriFinger	63.3±3.3	53.3±1.6	64.2±3.5	53.3±1.6	52.1±7.6	<u>66.2</u>±0.8	46.3±1.7	**66.5**±1.0

* Indicates results reported by Jang et. al. [25].

where $d(\cdot)$ is a distance function; we use cosine distance for the pre-training.

Decoder structure. Previous methods in dynamic SSL [13, 19, 25] utilize cross-attention layers as a core component for learning temporal evolution awareness, placing them within the decoders to guide the encoder to learn representations that effectively capture correspondences. These approaches leverage a hybrid structure of cross-attention layers, self-attention layers, and multi-layer perceptron (MLP) layers. In contrast, ToBo employs self-attention layers to ensure that the decoder exclusively attends to the given information during the reconstruction step, with MLP layers for progressive transformation from representation embedding spaces into the pixel space.

4 Experiment

In this section, we focus on demonstrating the effectiveness of our pre-training pipeline through fair comparisons with existing self-supervised learning methods. To this end, we evaluate our method on sequential tasks, including video label propagation tasks [26, 41, 58] and vision-based policy learning for robotic manipulation and locomotion across various simulated environments [18, 24, 36]. We extend our validation to real-world settings by deploying our pre-trained model on physical robots, showcasing its transferability. We further investigate the scalability of our method. In the appendix, we validate our claim regarding the importance of extremely high masking ratios to the target scene, present qualitative comparisons of manipulation processes against baseline methods, and show provide demonstrations of real-world manipulation tasks.

4.1 Experimental Setup

Implementaion details. We follow the evaluation protocol of Jang et. al. [25] for both video label propagation and vision-based policy learning on simulated environments. To ensure fair comparisons with the baselines, we also pre-train our method on Kinetics-400 for 400 epochs. Detailed explanation for both pre-training and evaluation are provided in the Appendix.

Baselines. We compare the performance of our method with conventional self-supervised learning (SSL) methods for visual representations including SimCLR [6], MoCo v3 [9], DINO [4], and MAE [21] We also consider previous dynamic scene SSL methods, i.e., SiamMAE [19], RSP [25],

Table 3: **Experimental results on vision-based robot policy learning on RLBench.** We report the performance of imitation learning agents on RLBench [24], which are trained upon representations from the ViT-S/16 model pre-trained on Kinetics-400 [29] dataset. The success rates (%) are reported for all the tasks. We report the gains of our method over the second-best baseline.

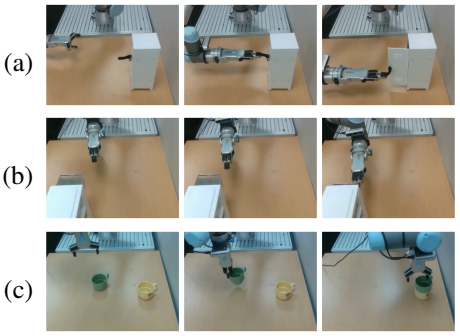
Tasks	SimCLR*	MoCo v3*	DINO*	MAE*	SiamMAE*	RSP*	CropMAE	ToBo
Button	7.4±2.6	11.4±4.1	24.7±1.5	6.4±2.2	6.1±2.3	28.4±3.0	26.9±6.7	**41.2**±7.4
Phone	34.6±6.6	36.2±3.4	32.0±5.5	37.7±1.9	5.4±0.5	48.0±4.6	16.6±3.8	**52.3**±3.2
Umbrella	5.8±3.3	13.2±1.5	28.1±1.4	10.0±1.2	4.0±0.0	37.3±3.0	37.5±8.8	**42.2**±6.9
Wine	11.0±2.1	8.7±0.7	31.4±1.5	10.0±2.1	8.7±0.8	31.9±2.3	33.2±0.2	**35.4**±3.8
Rubbish	5.2±1.2	6.7±0.8	12.9±1.5	6.2±3.2	3.5±0.9	18.5±1.1	20.6±1.7	**37.0**±6.1

* Indicates results reported by Jang et. al. [25].

Table 4: **Performance on real-world vision-based robot policy learning.** Success rates (%) of imitation learning agents on three manipulation tasks: Cabinet Opening, Drawer Closing, and Cup Stacking. Agents are trained with ViT-S/16 representations pre-trained on Kinetics-400 [29] for 400 epochs. The results demonstrate the generalizability of ToBo in real-world.

Method	Cabinet Opening	Drawer Closing	Cup Stacking
SiamMAE [19]	20.0	55.0	50.0
RSP [25]	25.0	65.0	55.0
CropMAE [13]	0.0	25.0	20.0
ToBo (ours)	**65.0**	**75.0**	**80.0**

Figure 4: **Real-world robot trajectories.** Initial, intermediate, and final states of the robot during (a) Cabinet Opening, (b) Drawer Closing, and (c) Cup Stacking.



and CropMAE [13]. We validate the impacts of explicitly learning state representations over these approaches.

4.2 Vision-based robot policy learning in simulated environments

We evaluate our method through imitation learning on robot manipulation and locomotion tasks across various simulated environments. Specifically, we evaluate five tasks from both the Franka Kitchen and RLBench benchmarks. Moreover, we consider two, five, five, and two tasks from Adroit [44], MetaWorld [56], DeepMind Control Suite (DMC) [46], and TriFinger [49] from the CortexBench benchmark, respectively.

Franka Kitchen. We present a comparison between our method and the baselines on vision-based robot policy learning in the Franka Kitchen environment in Table 1. The results demonstrate that our method significantly outperforms all the baselines across all tasks. Notably, our method achieves over 20% improvements in success rates on all tasks, except for the *Light on* task. This highlights the effectiveness of explicitly encoding visual state representation for vision-based robot policy learning.

CortexBench. We compare our method with the baselines for the vision-based robot manipulation and locomotion tasks in the Adroit, MetaWorld, DeepMind Control (DMC), and Trifinger environments in Table 2. The results show that our method achieves superior performance compared to the baselines across all tasks. In particular, our method surpasses the second-best performance with success rate gains of 11.9%p on DMC and 10.4%p on Adroit.

RLBench. Table 3 showcases the robot manipulation performance on five demonstration tasks in the RLBench environment. Notably, our method consistently exceeds all baselines across the five tasks. Moreover, the degraded performance of MAE and SiamMAE further highlights the significance of state representation learning for the robot backbones.

Table 5: **Results on video label propagation.** We report performances on video segmentation, video part segmentation, and pose tracking tasks from DAVIS [41], VIP [58], and JHMDB [26] benchmarks, respectively. For all methods, we report the performance with the representations pre-trained on the Kinetics-400 [29] dataset for 400 epochs.

	DAVIS			VIP	JHMDB	
Method	$\mathcal{J}\&\mathcal{F}_m$	$\mathcal{J}_m$	$\mathcal{F}_m$	mIoU	PCK@0.1	PCK@0.2
SimCLR [6]	53.9	51.7	56.2	31.9	37.9	66.1
MoCo v3 [9]	57.7	54.6	60.8	32.4	38.4	67.6
DINO [4]	59.5	56.5	62.5	33.4	41.1	70.3
MAE [21]	53.5	50.4	56.7	32.5	43.0	71.3
SiamMAE [19]	58.1	56.6	59.6	33.3	44.7	73.0
RSP [25]	60.1	57.4	62.8	33.8	44.6	73.4
CropMAE [13]	58.6	55.8	61.4	33.7	42.9	71.1
ToBo (ours)	**60.6**	**58.4**	**63.0**	**34.0**	**47.0**	**74.8**

(a) Semantic Part Propagation

(b) Object Propagation (c) Pose Tracking

Figure 5: **Qualitative results for video label propagation.** We provide examples of predicted propagation of our model on video object segmentation, video part segmentation, and pose tracking benchmarks. The leftmost images indicate the ground-truth annotations. We visualize the propagated results corresponding to 25, 50, and 100% ratio of the videos.

4.3 Vision-based Robot Policy Learning in Real-world Environments

Quantitative comparison. To validate the robustness of our method in real-world environments, we further investigate SSL methods on real-world robot manipulation tasks. Specifically, we design three demonstration tasks: *Cabinet Opening*, *Drawer Closing*, and *Cup Stacking*. For each task, We collect 50 demonstration episodes for training and 10 demonstration episodes for evaluation for imitation learning. Following the training protocol used in simulated environments, we train the policy network using a standard behavior cloning loss. The experimental results for each individual task are reported in Table 4. We first observe that our method exceeds SiamMAE [19], RSP [25], and CropMAE [13] on all three tasks. Specifically, our method improves 40%p, 10%p, and 25%p over the baselines on the *Cabinet Opening*, *Drawer Closing*, and *Cup Stacking* tasks, respectively. While previous SSL methods on dynamic scenes struggle with tasks that require relatively high precision, like cabinet opening tasks, our method even successfully executes the task with a considerable success rate. This showcases that models pre-trained by our method can be robustly transferred to real-world environments.

Qualitative comparison. To illustrate the actual manipulation processes, we present the robot trajectories from successful demonstrations for three real-world manipulation tasks in Fig. 4. Specifically,

Table 6: **Comparison with robot representation learning models.** We compare the performance of our method with robot representation learning methods across multiple simulated manipulation tasks. We categorize the methods into self-supervised learning, supervised learning through foundation models, and supervised learning with auxiliary language guidance. Despite the unbalanced training and evaluation setup, ToBo surpasses the RRL models on MetaWorld. Moreover, ToBo exceeds self-supervised RRL models despite of a smaller model with a significantly smaller amount of data. These results demonstrate the effectiveness of the representations learned by ToBo in diverse robot manipulation tasks.

Method	#Param	Dataset	#Seen frames	Adroit	MetaWorld	Franka Kitchen
Supervision through Foundation Models						
Theia† [45]	52.9M	Theia dataset	14.4B*	66.0	86.1	-
Supervision with Auxiliary Language Guidance						
R3M [37]	25.6M	Ego4D [16]	0.8B	65.0	69.2	53.1
MVP [43]	21.7M	MVP dataset	4.8B	-	84.6	-
Voltron‡ [28]	21.7M	SS-v2 [15]	0.3B	-	68.7	70.5
MPI‡ [27]	21.7M	Ego4D [16]	0.1B	-	85.7	76.5
Self-supervised Learning						
R3M° [37]	25.6M	Ego4D [16]	0.8B	45.6	67.0	47.2
data4robotics [10]	86.0M	Kinetics-700 [5]	0.5B	-	87.0	55.0
VC-1 [36]	86.0M	Ego4D+N [16, 36]	1.0B	50.0	86.4	-
ToBo (ours)	21.7M	Kinetics-400 [29]	0.2B	60.4	87.8	68.0

† Uses additional compression layers. ‡ Uses multi-head attention pooling layers for integrating spatial tokens.
° Excludes language guidance from the vanilla recipe. * Includes data for distillation models [12, 32, 42, 55].

the initial states of the physical robot are depicted in the left scenes, while the right scenes show the final states of the demonstrations. The middle scenes illustrate the intermediate states of the demonstrations. Our model clearly succeeds in all the tasks. We also compared the trajectories with the baselines in the Appendix.

4.4 Video Label Propagation

We perform comparative analyses on the video label propagation tasks. We consider the video object segmentation, video part segmentation, and pose tracking tasks from DAVIS [41], VIP [58], and JHMDB [26]. We follow the evaluation protocol in Jang et. al [25]. We present the quantitative evaluation in Table 5. Our method demonstrates superior performance compared to all the baselines across the video label propagation tasks. We also provide qualitative results in Fig. 5, where our method effectively traces visual appearances across various video label propagation tasks. These visualizations highlight that our method maintains robust object identity, part consistency, and pose continuity. The strong performance in both quantitative and qualitative evaluations further demonstrate the effectiveness of our approach in capturing the temporal evolution of visual appearance across consecutive scenes.

5 Discussion

Comparison with robot representation learning models We further compare our method with recent robot representation learning (RRL) models, categorized by their supervision types: self-supervised learning [10, 36], supervision via foundation model outputs [45], and supervision with auxiliary language annotations [27, 28, 37, 43]. Table 6 shows the reported performance of RRL models across several simulated robot manipulation benchmarks [18, 44, 56]. Here, our model is based on a ViT-Small architecture trained on Kinetics-400 for 400 epochs. Notably, despite having the smallest number of parameters and the second smallest amount of training data, and using no annotation-based supervision, our method achieves the highest score on MetaWorld. In particular, Theia is trained by distilling knowledge from five large-scale foundation models (CLIP large [42], Depth Anything large [55], DINOv2 large [38], Segment Anything huge [32], and ViT huge [48]), which are collectively trained on 14.3 billion annotated samples. It also employs convolution-based compression layers during evaluation. Surpassing Theia under such an unbalanced training and evaluation setup is noteworthy. Furthermore, the performance gap between R3M with and without

Table 7: **Performance with vision-language models.** We compare the performance of our method with vision-language models on Franka Kitchen. Despite using a smaller model, significantly less pre-training data, and no auxiliary textual guidance from manually annotated data, ToBo consistently outperform the other models across all tasks.

Method	#Param	Dataset	#Seen frames	Knob1 on	Light on	Sdoor open	Ldoor open	Micro open
CLIP* [42]	149.3M	WebImageText	12.8B	23.0	29.5	69.5	13.5	22.0
DINOv2 [38]	22.1M	LVD-142M	4.3B	25.5	38.0	82.0	15.5	20.0
SigLIP* [57]	203M	WebLI	2.1B	17.5	38.5	75.0	8.5	16.5
SigLIP2* [50]	375M	WebLI	40B	11.0	23.5	58.5	11.0	18.0
ToBo (Ours)	21.7M	Kinetics-400	0.2B	**57.0**	**82.0**	**95.0**	**51.0**	**55.0**
Gain				+ 31.5	+ 43.5	+ 13.0	+ 35.5	+ 33.0

* The model is trained using textual guidance with manually annotated data.

language guidance highlights the substantial benefit of auxiliary language supervision. Even with such unfairness in the training setup, our method outperforms R3M, MVP, Voltron, and MPI on MetaWorld. It also surpasses R3M on Franka Kitchen, despite significant differences in training data and model size. Compared to self-supervised RRL models, our method outperforms all the models. It surpasses much larger models such as VC-1 and data4robotics, despite being trained on a significantly smaller amount of data. Given the minimal number of parameters and training scale, these results demonstrates the effectiveness and efficiency of our proposed method for robot manipulation tasks.

Comparison with Vision-Language Models We compare our method with vision-language models widely used either as backbones across various domains or as vision towers in large language models. For fair evaluation, we follow the same evaluation protocol used in the main paper. We evaluate CLIP [42], DINOv2 [38], SigLIP [57], and SigLIP2 [50] in the Franka Kitchen benchmark, as shown in Table 7. Despite having the smallest number of learnable parameters and being exposed to the smallest number of seen frames during pre-training, ToBo achieves consistently superior performance, outperforming the baselines by margins at least 13.0%p to the maximum 43.5%p. These performance gaps are notable given that all baselines except DINOv2 use language supervision from manually annotated data. These results demonstrate the effectiveness of ToBo in summarizing visual observations for sequential scene understanding tasks.

Ablation Study on Mask Ratio of Target Scenes To verify our claim that extremely scarce information from target scenes forces the decoder to rely highly on the stored visual scene information of the reference scene, we conduct an ablation study varying the mask ratio of target scenes. We pre-train the models on the Kinetics-400 [29] dataset for 100 epochs and evaluate three tasks on Franka Kitchen. As shown in Figure 6, the effectiveness of our proposed method increases as the masking ratio of target scenes increases until 0.9, verifying our claim that scarce target scene information facilitates the exploitation of the compressed reference information. Besides, the models pre-trained with a masking ratio of 0.95 yield degraded performance in some tasks, demonstrating that minimal clues are necessary for the prediction of the missing information.

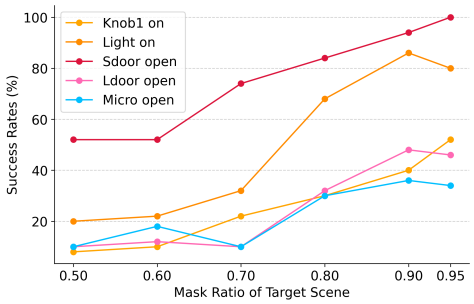


Figure 6: **Varying the masking ratio of target scenes.** We vary the masking ratio from 0.5 to 0.95 and pre-train ViT-S/16 models on the Kinetics-400 [29] dataset for 100 epochs.

Scalability. We investigate the scalability of our ToBo beyond ViT-S/16 by pre-training ViT-B/16 and ViT-L/16 on Kinetics-400 [29] for 100 epochs. We evaluate the pre-trained models on vision-based robot policy learning on Franka Kitchen [18] using three different seeds. We compare our method with MAE, SiamMAE, and RSP. Table 8 presents the mean and standard deviation across all seeds. We observe that models pre-trained with ToBo consistently achieving the best performance across all five tasks, exhibiting significant improvements over the second-best results. These demonstrate the scalability of our method.

Table 8: **Scalability of our method.** We report the performance of vision-based robot policy learning on Franka Kitchen [18], which are trained upon representations from the ViT-B/16 and ViT-L/16 model pre-trained on Kinetics-400 [29] dataset for 100 epochs. The success rates (%) are reported for all the tasks. We underline the second-best performance. We report the gains of our method over the second-best baseline. We conduct evaluations using three different seeds.

Arch.	Method	Knob1 on	Light on	Sdoor open	Ldoor open	Micro open
ViT-B/16	MAE [21]	18.7±1.2	21.3±4.6	70.0±2.0	17.3±2.3	15.3±2.3
	SiamMAE [19]	18.0±2.0	34.0±2.0	80.7±3.1	18.7±1.2	19.3±6.1
	RSP [25]	<u>24.7</u>±3.1	<u>51.7</u>±9.1	<u>87.3</u>±2.3	<u>23.3</u>±7.6	<u>26.7</u>±2.3
	ToBo (ours)	**46.7**±6.4	**78.7**±7.6	**95.3**±1.2	**47.3**±5.0	**37.3**±4.6
	Gain	+ 22.0	+ 27.0	+ 8.0	+ 24.0	+ 10.6
ViT-L/16	MAE [21]	19.3±7.6	33.3±2.3	61.3±6.4	16.0±2.0	14.0±2.0
	SiamMAE [19]	20.7±3.1	34.0±4.0	76.0±2.0	12.7±6.4	22.0±0.0
	RSP [25]	<u>26.7</u>±2.3	<u>48.0</u>±2.0	<u>88.0</u>±2.0	<u>22.7</u>±8.3	<u>23.3</u>±4.2
	ToBo (ours)	**54.7**±5.0	**75.3**±4.2	**94.0**±3.5	**50.0**±2.0	**42.7**±6.1
	Gain	+ 28.0	+ 27.3	+ 6.0	+ 27.3	+ 19.4

Comparison of training and inference flops We conducted FLOPs evaluation for both training and inference to quantitatively compare the computational cost of each model, as summarized in Table 9. During inference, all models use the same backbone architecture and input resolution without any input masking, resulting in identical inference FLOPs at the same model scale (e.g., 4.6 GFLOPs for ViT-Small). During training, ToBo, MAE [21], and SiamMAE [19] show similar computational costs while RSP [25] requires substantially more computation of 32.5 GFLOPs due to its complex decoding mechanisms. When considering computational costs with downstream performance (e.g., performance in Franka Kitchen [18]), these results further support the effectiveness of ToBo, which achieves a strong balance between efficiency and performance.

Table 9: Comparison of training FLOPs and downstream performance in Franka Kitchen.

Method	Training FLOPs (GFLOPs)	Franka Kitchen (%)
MAE [21]	13.0	26.1
SiamMAE [19]	13.1	30.4
RSP [25]	32.5	43.4
ToBo	15.9	**68.1**

6 Conclusion

We have introduced Token Bottleneck (ToBo), a self-supervised visual representation learning method designed for sequential scene understanding. The backbones for sequential scene-based tasks should effectively preserve visual information from observations while facilitating the recognition of temporal progression across sequential scenes. While conventional self-supervised learning (SSL) methods have proven promising impacts in visual representation learning, they primarily focus on understanding static images or entire videos, often lacking embeds for handling dynamic transitions in sequential tasks. Recent SSLs aim to address this by adapting correspondence learning in dynamic scenes. However, their patch-wise representations of observations are often suboptima for subsequent policy networks, especially in tasks like robotic manipulation. To this end, ToBo introduces a simple yet effective pipeline that facilitates conservative summarization of the observed scene into a bottleneck token while enable capturing of dynamic transitions through the bottleneck token. Through extensive experiments in various sequential understanding tasks including manipulation tasks and video label propagation tasks, we verified the superiority of ToBo over conventional SSL methods and previous dynamic scene SSL methods. Furthermore, applying ToBo in real-world settings demonstrates its robustness and generalization capability.

Limitation Due to the resource constrains, we did not check the scalability of our method beyond huge scale and explore beyond the commonly used input resolution. Additionally, our study focused on a simplest setting involving two dynamic scenes to learn temporal dynamics.

References

[1] 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. 2

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2021. 2

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Proceedings of NeurIPS*, 2020. 1, 2

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021. 1, 2, 5, 7

[5] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset, 2022. 8

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 2020. 1, 2, 5, 7

[7] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, pages 15750–15758, 2021. 1

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2

[9] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021. 1, 2, 5, 7

[10] Sudeep Dasari, Mohan Kumar Srirama, Unnat Jain, and Abhinav Gupta. An unbiased look at datasets for visuo-motor pre-training. In *Conference on Robot Learning*. PMLR, 2023. 3, 8

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 8, 16

[13] Alexandre Eymaël, Renaud Vandeghen, Anthony Cioppa, Silvio Giancola, Bernard Ghanem, and Marc Van Droogenbroeck. Efficient image pre-training with siamese cropped masked autoencoders. In *European Conference on Computer Vision*, pages 348–366. Springer, 2025. 1, 2, 3, 5, 6, 7

[14] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3299–3309, 2021. 16

[15] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 8

[16] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall,

Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18995–19012, 2022. 8

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray Kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In *NeurIPS*, pages 21271–21284. Curran Associates, Inc., 2020. 1

[18] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019. 5, 8, 9, 10, 15, 16, 18

[19] Agrim Gupta, Jiajun Wu, Jia Deng, and Li Fei-Fei. Siamese masked autoencoders. In *Advances in Neural Information Processing Systems*, 2023. 1, 2, 3, 4, 5, 6, 7, 10, 15, 16

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 2

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022. 1, 2, 3, 5, 7, 10, 15, 16

[22] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 16

[23] Allan Jabri, Andrew Owens, and Alexei A. Efros. Space-time correspondence as a contrastive random walk. In *Advances in Neural Information Processing Systems*, 2020. 19

[24] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. 5, 6, 15, 17, 18

[25] Huiwon Jang, Dongyoung Kim, Junsu Kim, Jinwoo Shin, Pieter Abbeel, and Younggyo Seo. Visual representation learning with stochastic frame prediction. In *Proceedings of the 41st International Conference on Machine Learning*, pages 21289–21305. PMLR, 2024. 1, 2, 5, 6, 7, 8, 10, 15, 16, 17

[26] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *Proceedings of the IEEE international conference on computer vision*, 2013. 5, 7, 8, 19

[27] Zeng Jia, Bu Qingwen, Wang Bangjun, Xia Wenke, Chen Li, Dong Hao, Song Haoming, Wang Dong, Hu Di, Luo Ping, Cui Heming, Zhao Bin, Li Xuelong, Qiao Yu, and Li Hongyang. Learning manipulation by predicting interaction. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024. 3, 8

[28] Siddharth Karamcheti, Suraj Nair, Annie S. Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. In *Robotics: Science and Systems (RSS)*, 2023. 3, 8

[29] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 5, 6, 7, 8, 9, 10, 15, 16

[30] Taekyung Kim, Sanghyuk Chun, Byeongho Heo, and Dongyoon Han. Learning with unmasked tokens drives stronger vision learners. *European Conference on Computer Vision (ECCV)*, 2024. 2, 4

[31] Taekyung Kim, Byeongho Heo, and Dongyoon Han. Morphing tokens draw strong masked image models. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[32] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 8

[33] Z. Lai and W. Xie. Self-supervised learning for video correspondence flow. In *BMVC*, 2019. 19

[34] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. In *NeurIPS*, 2019. 19

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 16

[36] 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? In *Advances in neural information processing systems*, 2023. 3, 5, 8, 15, 17, 18

[37] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *6th Annual Conference on Robot Learning*, 2022. 3, 8, 16

[38] 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. 8, 9

[39] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11205–11214, 2021. 1

[40] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. In *Proceedings of the 39th International Conference on Machine Learning*, pages 17359–17371. PMLR, 2022. 16

[41] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 5, 7, 8, 18

[42] 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 *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 8, 9

[43] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Proceedings of The 6th Conference on Robot Learning*, pages 416–426. PMLR, 2023. 3, 8

[44] 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. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018. 6, 8

[45] Jinghuan Shang, Karl Schmeckpeper, Brandon B. May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. 2024. 3, 8

[46] Yuval Tassa, Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, and Nicolas Heess. dm_control: Software and tasks for continuous control. *arXiv preprint arXiv:2006.12983*, 2020. 6, 17

[47] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in neural information processing systems*, 2022. 1

[48] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 516–533. Springer, 2022. 8

[49] Frederik Träuble, Andrea Dittadi, Manuel Wuthrich, Felix Widmaier, Peter Vincent Gehler, Ole Winther, Francesco Locatello, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. The role of pretrained representations for the OOD generalization of RL agents. In *International Conference on Learning Representations*, 2022. 6

[50] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 9

[51] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019. 19

[52] Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Revaud Jérôme. Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. In *NeurIPS*, 2022. 2

[53] 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 *ICCV*, 2022. 2

[54] Jiange Yang, Bei Liu, Jianlong Fu, Bocheng Pan, Gangshan Wu, and Limin Wang. Spatiotemporal predictive pre-training for robotic motor control, 2024. 3

[55] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024. 8

[56] 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 *Conference on Robot Learning*, 2020. 6, 8, 17

[57] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11975–11986, 2023. 9

[58] Qixian Zhou, Xiaodan Liang, Ke Gong, and Liang Lin. Adaptive temporal encoding network for video instance-level human parsing. In *Proceedings of the 26th ACM international conference on Multimedia*, 2018. 5, 7, 8, 18

Appendix

- §A: Further analysis on the hyperparameter choices in the Token Bottleneck pre-training pipeline
- §B: Manipulation trajectory visualization of real-world demonstrations
- §C: Implementation details for pre-training and evaluation

A Further Analysis

In this section, we examine how hyperparameter choices in the Token Bottleneck (ToBo) pre-training pipeline affect performance, regarding the number of bottleneck tokens, impact of temporal difference, We pre-train ViT-B/16 for 100 epochs on Kinetics-400 [29] throughout the ablation studies. The comparisons are done on five Franka Kitchen imitation-learning tasks [18].

Ablation study on the number of bottleneck tokens We vary the number of bottleneck tokens in {1, 2, 4, 8}. As shown in Table Aa, using a single token consistently yields the best performance across tasks. This demonstrates that conservative summarization without auxiliary storage better captures the current observation and thus improves action prediction in robotics.

Ablation study on temporal difference We further vary the maximum temporal gap between frames among {48, 96, 144}. As shown in Table Ab, moderate temporal differences encourage the model to learn dynamic scene evolution, since shorter gaps lack meaningful change whereas overly longer gaps disrupt temporal coherence

Ablation study with no temporal difference We apply our method using the same frame for both source and target scenes. As shown in Table Ac, our method still works even without temporal difference, surpassing other baselines (e.g., MAE [21], SiamMAE [19], and RSP [25]) with significant margin across tasks. However, its overall performance degrades compared to original ToBo, reflecting the loss of supervision from temporal change. This highlights the importance of temporal contrast for effective pre-training of ToBo.

Ablation study on multiple source frames We compare ToBo to a variant pre-trained with multiple source frames. Specifically, we randomly sample four source frames and pre-train for 100 epochs under the same recipe as ToBo. As shown in Table Ad, this multi-frame variant surpasses prior baselines (e.g., MAE [21], SiamMAE [19], and RSP [25]) in most of the tasks. However, despite requiring higher pre-training cost, it underperforms compared to ToBo across all robotics tasks. These results suggest that while it is possible to extend ToBo to multi-frame settings, such naive extension may encounter potential new challenges, leading to suboptimal performance.

B Manipulation trajectory visualization of Real-world Demonstrations

We showcase the robot manipulation trajectories for the SiamMAE [19], RSP [25], and our model as robot backbones in the same episode on the real-world environment for each task. In Figure A, Specifically, the leftmost scenes depicts the initial states of the physical robot, while the rightmost scenes show the final states of the demonstrations. As shown in Fig. A, while SiamMAE and RSP fail to execute the manipulation tasks, our method successfully completes them within the same episode. We also provide videos of these demonstrations in the supplementary material.

C Implementation Details

We provide implementation details for pre-training and evaluation. Specifically, we present the evaluation protocols for vision-based robot policy learning on each simulated environment (i.e., Franka Kitchen [18], CortexBench [36], RLBench [24]) and real-world environment. Then, we explain experimental setups for video label propagation tasks.

Table A: **Ablation studies Token Bottleneck pre-training**. We vary hyperparameters of the Token Bottleneck (ToBo) pre-training pipeline. We report the success rates (%) on five imitation learning tasks from the Franka Kitchen benchmark [18]. All models are ViT-B/16 and are pre-trained for 100 epochs on Kinetics-400 [29]. We mark our default settings in gray.

(a) **Number of bottleneck tokens**

# bottleneck tokens	Knob1 on	Light on	Sdoor open	Ldoor open	Micro open	Mean
1	**46.7**	**78.7**	**95.3**	**47.3**	**37.3**	**61.1**
2	31.0	54.0	74.0	26.0	24.0	41.8
4	28.0	24.3	78.0	28.0	22.0	36.1
8	10.0	20.0	56.0	26.0	9.3	24.3

(b) **Temporal difference**

Maximum temporal difference	Knob1 on	Light on	Sdoor open	Ldoor open	Micro open	Mean
48	40.7	78.7	96.0	44.0	35.3	58.9
96	**46.7**	**78.7**	95.3	**47.3**	37.3	**61.1**
144	36.0	69.3	**97.3**	46.7	**39.3**	57.7

(c) **Ablation with no temporal difference**

Method	Knob1 on	Light on	Sdoor open	Ldoor open	Micro open	Mean
MAE [21]	18.7	21.3	70.0	17.3	15.3	28.5
SiamMAE [19]	18.0	34.0	80.7	18.7	19.3	34.1
RSP [25]	24.7	51.7	87.3	23.3	26.7	42.7
ToBo (no temporal difference)	41.0	72.0	89.3	32.7	32.0	53.5
ToBo	46.7	78.7	95.3	47.3	37.3	**61.1**

(d) **Ablation on multiple source frames**

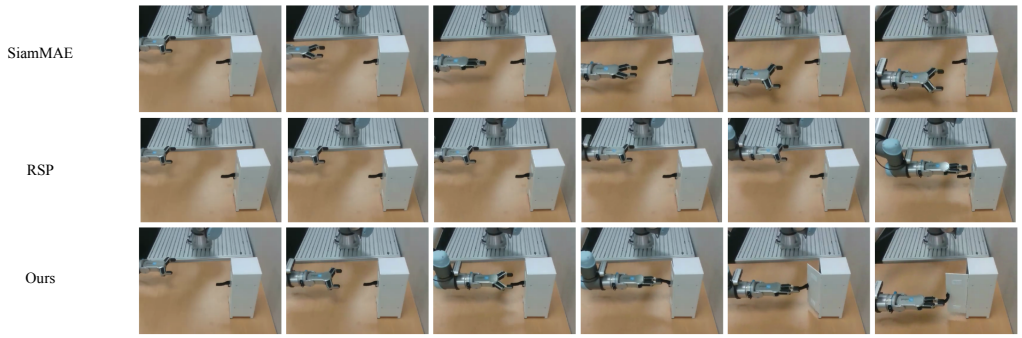
Method	Knob1 on	Light on	Sdoor open	Ldoor open	Micro open	Mean
MAE [21]	18.7	21.3	70.0	17.3	15.3	28.5
SiamMAE [19]	18.0	34.0	80.7	18.7	19.3	34.1
RSP [25]	24.7	51.7	87.3	23.3	26.7	42.7
ToBo (w/ multi-frame)	28.7	60.7	92.7	20.7	32.0	46.9
ToBo	46.7	78.7	95.3	47.3	37.3	**61.1**

C.1 Pre-training

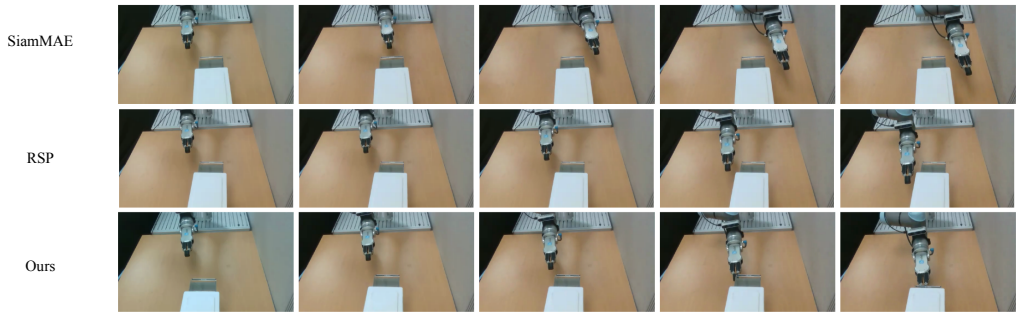
We pre-train ViT-S/16 [12] on Kinetics-400 [29] for 400 epochs for the main comparison, while we pre-train ViT-S/16, ViT-B/16, and ViT-L/16 for 100 epochs for analyses. We employ repeated sampling [22, 14] with a factor of 2 so that the models are indeed pre-trained for 200 epochs. We use AdamW optimizer [35] with a batch size of 1536, comprising dynamic scenes with a resolution of 224×224. These scenes are randomly sampled from videos at a rate of 30 FPS, with a temporal index gap ranging from 4 to 96. We simply apply random resized crop and horizontal flip to the scenes, aligning the cropping region across the reference and target scenes. To drive the learning mechanism of our proposed method, we randomly mask the target scenes with an extremely high masking ratio of 0.9. Our decoder is composed of eight vision transformer blocks, i.e., each block contains self-attention layers and multi-layer perceptrons. We follow the default hyperparameters of the baselines for their pre-training on Kinetics-400 [29]. We adopt a siamese masked autoencoding loss [19] as an auxiliary objective to enhance learn patch-level correspondence learning.

C.2 Vision-based Robot Policy Learning

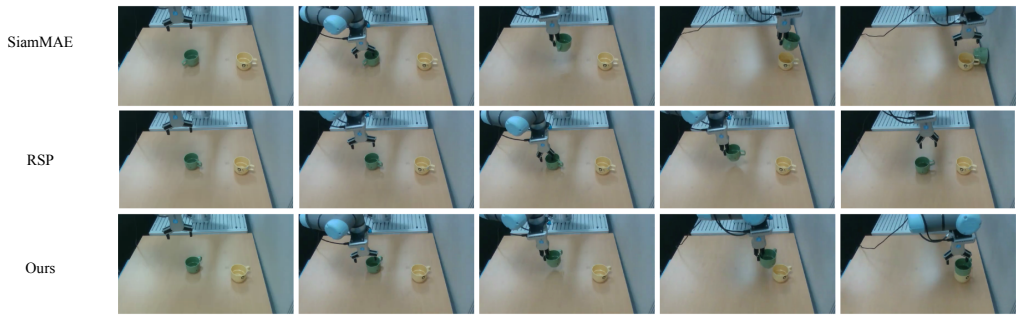
Franka Kitchen. We validate models pre-trained by our method and other baselines in five imitation learning tasks from the Franka Kitchen benchmark [18]. Our experiments mainly follow the imitation learning evaluation setup in Jang et. al. [25], which builds upon [37, 40]. Specifically, we employ an agent comprising a frozen backbone initialized with pre-trained models and a policy network consisting of a two-layer MLP, with a batch normalization layer applied at the input stage. We define the state representation for the policy network as the combination of the visual representation and the robot's proprioception. For the perception, we employ either a left or right camera with a 224×224 resolution while omitting depth. The policy network is trained with a standard behavior cloning loss. Training for each demonstration task progresses for 20,000 steps, with a periodic online



(a) Cabinet Opening



(b) Drawer Closing



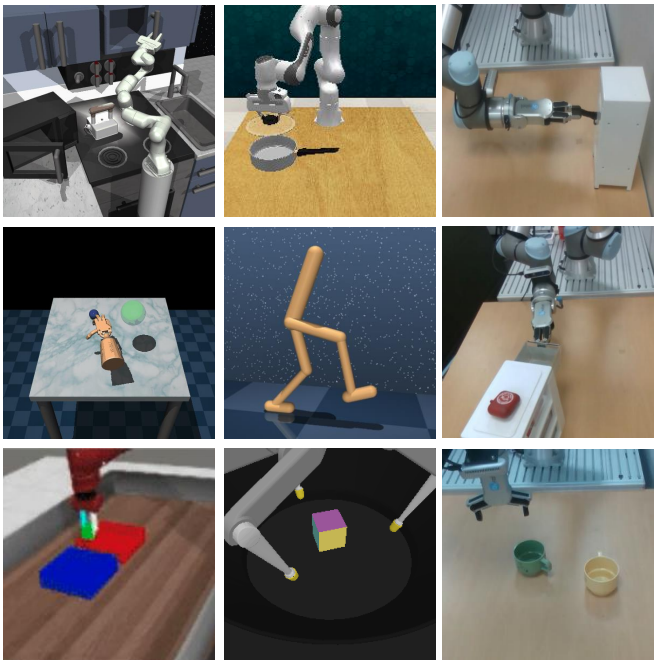
(c) Cup Stacking

Figure A: **Sampled Trajectories from Real World Experiment**. We visualize the manipulation trajectories of ToBo, RSP, and SiamMAE on physical robot manipulation tasks in real-world environments (i.e., *Cabinet Opening*, *Drawer Closing*, and *Cup Stacking*). Our ToBo successfully demonstrates all tasks, which aligns with the quantitative performance comparisons results.

evaluation in the simulated environment every 1,000 steps. We evaluate the highest success rates of each demonstration across four different seeds and report its average with a 95% confidence interval.

RLBench. We consider five manipulation tasks from RLBench [24]. Follow the evaluation setup in Jang et. al. [25], we generate 100 demonstrations and utilize them for training the agent. We employ a front camera with a 224×224 resolution. Point cloud information is excluded throughout all experiments. We employ the end-effector controller with path planning. We evaluate the highest success rates of each demonstration across four different seeds.

CortexBench. We evaluate the models on four simulated environments from CortexBench [36]. We consider two, five, five, and two demonstrations from Adroit, DeepMind Control (DMC) [46], Meta-World [56], and Trifinger, respectively. Proprioceptive data is utilized except the DMC benchmark. We mainly follow the experimental setups in Jang et. al. [25], which builds upon [36]. For each task, we train the agent for 100 epochs, with a periodic online evaluation in the simulated environment



(a) Simulated environments (b) Real-world

Figure B: **Visualization of environments for robot policy learning evaluation.** We validate the effectiveness of our method on (a) simulated environments (e.g., Franka Kitchen [18], CortexBench [36], RLBench [24] and (b) real-world environments. We design real-world environments with physical robots to evaluate how the algorithm handles given tasks.

(a) Cabinet Opening (b) Drawer Closing (c) Cup Stacking

Figure C: **Task Description for Real-world Environments.** We illustrates the objectives of physical robot manipulation tasks in the real-world. Yellow arrows indicate the target actions for each task.

every 5 epochs. We report the normalized score for DMC and the highest success rates for other tasks. We conduct demonstration tasks for five different seeds and report its average with a 95% confidence interval.

Real-world Environments. We evaluate our proposed method in real-world robotic imitation learning tasks using a UR5e manipulator equipped with a parallel gripper. The policy operates at a control frequency of 5 Hz, executing actions defined as delta end-effector poses and gripper's state, with specific parameterizations for each task: (dx, dy) for drawer closing, (dx, dy, gripper open/close) for cabinet opening, and (dx, dy, dz, gripper) for cup stacking. The system employs joint position control at 50 Hz, with a numerical inverse kinematics (IK) solver running in the background to calculate the end-effector's pose to the joint position. Our training dataset consists of 50 demonstrations for cabinet opening and cup stacking and 30 demonstrations for drawer closing. We train the two-layer MLP policy for 100 epochs without incorporating proprioceptive states, using a top-front camera view with a resolution of 224×224. The final performance is evaluated based on the reported average success rate across tasks. Figure C provides visual examples of the three tasks under consideration.

Video label propagation. We conduct comparative analyses for video label propagation on video object segmentation on DAVIS [41], video part segmentation on VIP [58], and pose tracking on

JHMDB [26]. Following the evaluation protocols in the previous studies [51, 34, 33, 23], we employ k-nearest neighbor inference, maintain a queue of length m to provide temporal context, and restrict the set of source nodes within a spatial radius r. Additionally, we perform a grid search to optimize evaluation hyperparameters for each method and report the best results.

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: We clearly state our claims in the abstract and introduction sections.

   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: We discuss the limitations at the end of the paper

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

   Justification: The paper does not include theoretical result.

   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: We explain the implementation details in the Appendix.

   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 provide a link to our source code.

   Guidelines:

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

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We explain them in the implementation details in the Appendix

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report the experimental results with error bars.

   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 explain them in the implementation details in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our research conform with the NeurIPS Code of Ethics.

   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: We properly cited all the papers from which we used code, data, and models.

    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.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.