IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Single-View 3D-Aware Representations for Reinforcement Learning by Cross-View Neural Radiance Fields

Daesol Cho[1,*], Seungyeon Yoo[2,*], Dongseok Shim[3], and H. Jin Kim[2]

***Abstract*—Reinforcement learning (RL) has enabled robots to develop complex skills, but its success in image-based tasks often depends on effective representation learning. Prior works have primarily focused on 2D representations, often overlooking the inherent 3D geometric structure of the world, or have attempted to learn 3D representations that require extensive resources such as synchronized multi-view images even during deployment. To address these issues, we propose a novel RL framework that extracts 3D-aware representations from single-view RGB input, without requiring camera pose or synchronized multi-view images during the downstream RL. Our method employs an autoencoder architecture, using a masked Vision Transformer (ViT) as the encoder and a latent-conditioned Neural Radiance Fields (NeRF) as the decoder, trained with cross-view completion to implicitly capture fine-grained, 3D geometry-aware representations. Additionally, we utilize a time contrastive loss that further regularizes the learned representation for consistency across different viewpoints, which enables viewpoint-robust robot manipulations. Our method significantly enhances the RL agent's performance both in simulation and real-world experiments, demonstrating superior effectiveness compared to prior 3D-aware representation-based methods, even when using only single-view RGB images during deployment. Project page: `https://sincro-ral.github.io/`.**

***Index Terms*—Reinforcement learning, representation learning, visual learning.**

I. Introduction

REINFORCEMENT learning (RL) has empowered an embodied agent such as a robot to acquire complex skills. However, its capability heavily depends on the representation of the underlying systems, especially in the image domain. In other words, central to image-based RL is the challenge of representation learning, where the goal is to distill high-dimensional visual data into compact, informative features that capture the essence of the environment. Effective representation learning schemes for image-based RL enable agents to interpret and act upon visual data more efficiently, facilitating faster convergence and improving performance in tasks such as robotic manipulation.

Manuscript received: March, 31, 2025; Revised July, 4, 2025; Accepted September, 12, 2025.

This paper was recommended for publication by Editor Kober Jens upon evaluation of the reviewers' comments. This work was supported by the Technology Innovation Program funded by the Ministry of Trade, Industry & Energy(MOTIE, Korea) (RS-2024-00423940).

*Daesol Cho and Seungyeon Yoo contributed equally to this work.

[1]Daesol Cho is with School of Interactive Computing, Georgia Institute of Technology, Georgia, USA `chodaesol@gmail.com`

[2]Seungyeon Yoo and H. Jin Kim are with the Department of Aerospace Engineering, Seoul National University, Seoul, Republic of Korea `syeon.yoo@snu.ac.kr; hjinkim@snu.ac.kr`

[3]Dongseok Shim is with the Interdisciplinary Program in AI, Seoul National University, Seoul, Republic of Korea `tlaehdtjr01@snu.ac.kr`

Digital Object Identifier (DOI): see top of this page.

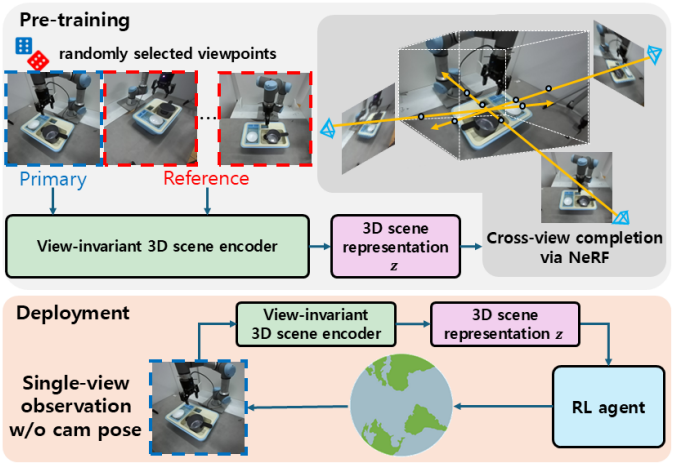


Fig. 1: During pre-training, SinCro learns a view-invariant 3D scene encoder by leveraging cross-view completion with a few randomly selected reference images from different viewpoints via NeRF. During deployment, it utilizes the pre-trained 3D scene encoder for downstream RL, relying solely on single-view RGB images without a camera pose and NeRF rendering.

Many previous works on image-based RL that have focused on learning efficient representation can be roughly categorized into several approaches: pre-training an image encoder via contrastive objectives [1], [2], employing data augmentations [3], [4], using autoencoders for reconstruction [5], [6], and leveraging in-the-wild internet-scale datasets [7], [8]. While these methods are effective and widely utilized, they typically treat visual inputs as 2D grids, overlooking the structured 3D geometric information inherently present in the 3D world. Such a lack of 3D awareness forces the embodied agent to rely on view-specific features such as surface-level pixel patterns or 2D shapes that are unique to the particular perspective, hindering its ability to adapt to different viewpoints. Furthermore, this limitation compels the policy network to implicitly infer 3D actions from 2D visual inputs (2D-to-3D mapping), rather than leveraging 3D-aware representations that are better suited for mapping directly to 3D actions (3D-to-3D mapping). Therefore, learning 3D-aware representations from 2D image inputs is crucial for achieving superior task performance, particularly when precise 3D spatial information inference is critical.

Recent approaches have attempted to learn 3D representation [9], [10], [11]. However, these methods typically require not only expert demonstration data but also calibrated cameras for accurate depth projection during deployment. Other prior works have explored different RGB-only approaches [12], [13], [14], by mapping multi-view images into a single latent feature and providing it with a volume rendering network in

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

neural radiance field (NeRF) [15] to reconstruct the 3D world. Despite these advancements, they often rely on the object-level mask or still require synchronized multi-view images along with camera pose information even during downstream RL. All of these constraints can be a significant burden in real-world robotics applications, where the requirement of camera pose and multi-view setups are impractical. This motivates our central question: under a practical deployment setting, *can we distill a 3D-consistent latent representation that boosts RL policy performance?* To overcome these challenges, a single-view 3D-aware representation inference framework that relies solely on RGB images is necessary. It is particularly beneficial in practical situations where a brief, one-time multi-view data capture with calibrated cameras is available during pre-training, but the robot has to rely on single-view images without a camera pose to perform the task during deployment.

To develop such an effective 3D-aware representation given a finite, in-domain dataset, we propose a **Sin**gle-view 3D-aware implicit representation inference framework for RL by performing **Cro**ss-view completion via NeRF (**SinCro**). Specifically, it adopts an autoencoder structure, trained only with RGB supervision, and consists of two stages: (1) pre-training a masked ViT-based [16] 3D scene encoder through a latent-conditioned NeRF decoder, and (2) deploying only the pre-trained 3D scene encoder in downstream RL tasks. The encoder utilizes a pixel masking strategy [17] with cross-view completion for 3D geometry-aware representation, and the NeRF decoder leverages multi-view reconstruction to capture inherent, essential 3D information of the environment. Additionally, we further regularize the intermediate representation to ensure consistency across different viewpoints by applying a time contrastive loss [18]. Combining all proposed components for 3D-aware representation, our method outperforms prior works in both simulation and real-world downstream RL tasks. Further ablation studies and analyses validate that the proposed method is crucial to perform the single-view inference successfully and enables us to obtain representations that are view-invariant and robust to the viewpoint changes.

In summary, this work has the following key contributions:

- We present SinCro, a 3D-aware representation-based RL framework. To the best of our knowledge, this is the first work that enables the extraction of 3D-aware implicit representations only with single-view RGB images during downstream RL.
- SinCro learns 3D geometry-aware and view-invariant representations of the scene by leveraging a NeRF-based cross-view completion and contrastive learning, enabling single-view inference even from novel viewpoints.
- The proposed method achieves superior downstream RL results both in simulation and real-world, and we qualitatively demonstrate that the learned representation provides an implicit understanding of the 3D world.

II. Related Works

A. 2D Representation Learning for RL

Prior works have been proposed to develop an efficient, effective representation learning strategy for RL in the image domain. Some prior approaches formulate representation learning as encoder pre-training via auxiliary learning objectives [19], [20], [21], self-supervised reconstruction [22], [5], [6], masked image modeling [5], [6], [23], and contrastive learning [18], [1], [2]. Other approaches have introduced objectives specialized for decision-making such as predicting future states from the current state [24], [25], training value functions [7], [8], or data augmentations [3], [4]. However, these works do not consider the innate 3D structure of the environment, which leads the network to lack 3D geometry awareness and depend on implicit 2D-to-3D mapping. In this work, we utilize the NeRF-based 3D scene representation learning to enhance the 3D understanding of the image feature.

B. 3D Scene Representation

Building on recent progress in robot learning and computer vision, several approaches have emerged that leverage multiple cameras to capture multi-view images for vision-based control [18], [26], [27]. While most of them directly utilize multi-view images as inputs, they do not perform explicit reconstruction of the 3D world, which is crucial for 3D understanding. Some recent prior works have attempted to explicitly model the 3D space [9], [28], [10], [11], [29], [30], but they require calibrated cameras to get depth images and project the queried pixel into a 3D space during deployment. Other prior works propose to learn implicit 3D-aware representation [12], [13], [14] by reconstructing the 3D world via NeRF [15]. However, multiple cameras with camera poses are still required to infer the 3D-aware representation and downstream behavioral learning. Furthermore, some of them even require semantic masks for all pixels in the image for semantic [13] or object-level reconstruction [12]. In this work, we propose a NeRF-based cross-view completion for 3D scene representation learning, trained only with RGB supervision. It enables the inference of an 3D scene representation from single-view images, without requiring camera poses during downstream RL.

III. Preliminary

A. Visual Reinforcement Learning

In visual RL, we assume a Partially Observable Markov Decision Process (POMDP) $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, r, \gamma)$, where $\mathcal{S}$ is the state space of the underlying system, $\mathcal{O}$ is the image observation space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the environment dynamics, r is the reward function, and γ is the discount factor. The RL objective is to discover a policy $\pi : \mathcal{O} \to \Delta(\mathcal{A})$ that maximizes the expected return $\mathbb{E}_{\pi,\mathcal{P}}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$. Since we utilize the 3D scene encoder $\Omega : \mathcal{O} \to \mathcal{Z}$ that maps the image observation into the latent representation, the RL-relevant networks such as the policy and critic are modified to take $\Omega(\mathcal{O})$ instead of raw image $\mathcal{O}$.

B. Neural Radiance Fields

The idea behind the neural radiance fields (NeRF) [15] is to model a 3D scene by predicting a learnable continuous volumetric radiance field. It is represented by differentiable rendering function F_θ that maps a 3D location $\mathbf{x}$ and viewing

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

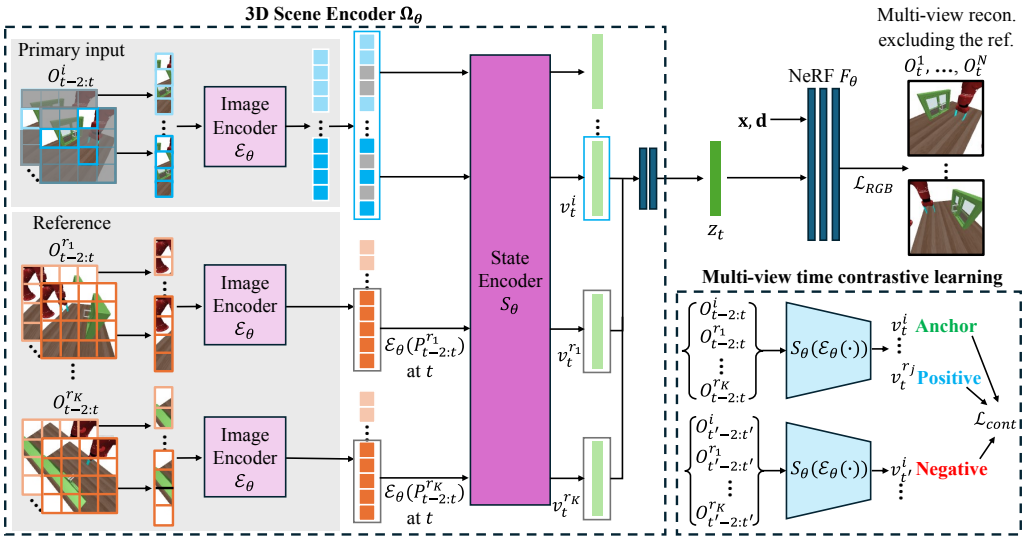


Fig. 2: 3D scene encoder Ω_θ takes masked primary images and K reference images as inputs to extract the latent scene representation z_t. It is trained by cross-view completion via NeRF, and time contrastive learning is applied to regularize z_t to be view-invariant. Once the pre-training is finished, only Ω_θ will be used for RL, and NeRF will no longer be used.

direction $\mathbf{d}$ to a color $\mathbf{c}$ and a density σ, i.e. $F_\theta(\mathbf{x}, \mathbf{d}) = (\mathbf{c}, \sigma)$. To render an image from a specific viewpoint, NeRF aggregates the color information of a camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, and computes the expected color $C(\mathbf{r})$ as follows:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\,\sigma(\mathbf{r}(t))\,\mathbf{c}(\mathbf{r}(t), \mathbf{d})\,dt, \quad \text{where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\,ds\right). \tag{1}$$

where $\mathbf{o}$ is the camera center, $T(t)$ is the accumulated transmittance, and t_n, t_f are pre-defined near and far depth bounds, respectively. Then, F_θ is optimized by pixel-level RGB supervision

$$\mathcal{L}_{RGB} = \sum_{\mathbf{r}_{i,k}} \|\hat{C}(\mathbf{r}_{i,k}) - C(\mathbf{r}_{i,k})\|_2^2 \tag{2}$$

where $\mathbf{r}_{i,k}$ denotes the sampled ray k from camera view i. Even though NeRF shows impressive results in 3D scene reconstruction, its key limitation is the assumption of a static scene. Some prior works propose methodologies to model the dynamic scenes [31], [32], [33], but they usually assume a single video input. In other words, the scene should be uniquely determined given a specific timestep t. However, in the context of RL, the scene at a specific timestep t could differ across every episode, making the prior works unavailable. To address this issue, we extract essential information from a few images of the current scene and use it as a latent condition for NeRF to reconstruct the scene.

IV. Method

This section demonstrates the details of our method. It consists of two stages, pre-training NeRF for representation learning and downstream RL. In section IV-A, we propose a cross-view completion using a latent-conditioned NeRF model that learns 3D geometry-aware scene representations. In section IV-B, time contrastive loss is proposed to regularize the representation. In section IV-C, we introduce an RL algorithm that leverages the 3D aware representation extracted by the pre-trained encoder from single-view input.

A. 3D-aware Representation Learning with Cross-View Completion via NeRF

To obtain an implicit 3D-aware scene representation and address the dynamic scenes, we train a 3D scene encoder Ω_θ that maps the image observations to a latent scene representation z for each timestep and learns a rendering function F_θ based on the latent z, i.e., $F_\theta(\mathbf{x}, \mathbf{d}, z)$. Specifically, we employ a pretext task of cross-view completion [34]. It involves reconstructing an input image with masked sections by utilizing the visible content and referring to unmasked reference images from different viewpoints. This approach is expected to enhance the model's 3D understanding.

a) Overview: The overall framework is shown in Figure 2. Given a dataset of episodic rollouts from N viewpoints, $O^i_{t-2:t}$ denotes three consecutive images (to capture trajectory history) from the i^{th} viewpoint at times $t-2:t$. Reference images $O^{r_j}_{t-2:t}$ are defined similarly for the j^{th} viewpoint. For each training iteration, we randomly select one primary viewpoint $O^i_{t-2:t}$ and K different reference viewpoints $O^{r_1}_{t-2:t}, \ldots, O^{r_K}_{t-2:t}$. We split these images into non-overlapping patches $P^i_{t-2:t}, P^{r_1}_{t-2:t}, \ldots, P^{r_K}_{t-2:t}$ and randomly mask a fraction m (75%) of the primary patches, $P^i_{m,t-2:t}$. This masking design forces the model to learn both 3D geometry-aware information and contexts within the masked viewpoint by cross-view completion. Next, $P^i_{m,t-2:t}, P^{r_1}_{t-2:t}, \ldots, P^{r_K}_{t-2:t}$ each passes through a shared ViT-based image encoder $\mathcal{E}_\theta$. The outputs are then concatenated and processed by a ViT-based state encoder $\mathcal{S}_\theta$, producing state features $v^i_t, v^{r_1}_t, \ldots, v^{r_K}_t$. Finally, these features yield the latent scene representation z_t, which F_θ uses to synthesize multi-view images conditioned on z_t.

b) Details on the image encoder: The shared image encoder $\mathcal{E}_\theta$ follows the standard ViT structure, but it is

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

modified to meet the requirement for the downstream RL. Specifically, to deal with the consecutive images, we add 1D learnable parameters representing each timestep along with 2D sinusoidal positional embeddings for the patches. Then, the patches $P^i_{m,t-2:t}, P^{r_1}_{t-2:t}, \cdots, P^{r_K}_{t-2:t}$ are independently passed through transformer blocks in ViT.

c) Details on the state encoder: The ViT-based state encoder $\mathcal{S}_\theta$ takes the outputs from the image encoder and fuses them across viewpoints and timesteps. For the primary images, we use all timesteps to highlight temporal information relevant to RL, such as the agent's movement. For the reference images, we only include patches from the latest timestep t to focus on 3D geometric information from different viewpoints, while improving memory and computational efficiency. Mask tokens [17] fill the positions of any masked patches, and we add 1D learnable parameters to distinguish primary vs. reference images, along with each timestep's token and 2D positional embeddings for the patches. After $\mathcal{S}_\theta$ processes these concatenated features, we obtain $v^i_t, v^{r_1}_t, \ldots, v^{r_K}_t$ at t. Averaging and projecting these through a shallow MLP yields the scene representation z_t, followed by L2 normalization. For notational simplicity, we define $z_t = \Omega_\theta(O^i_{t-2:t}, O^{r_1}_{t-2:t}, \cdots, O^{r_K}_{t-2:t})$, encompassing the entire pipeline from inputs to the scene representation.

d) NeRF decoder: Leveraging reference images, F_θ is trained to reconstruct the masked primary image (cross-view completion) conditioned on the latent scene representation z_t. Since NeRF inherently models 3D scenes and the cross-view completion task requires understanding the geometric relationship between primary and reference images, it naturally encourages stronger 3D structural understanding in the 3D scene encoder than typical 2D CNN-based encoders in visual RL algorithms. Also, to further enhance the 3D awareness of z_t, we additionally employ multi-view reconstruction, unlike prior masked modeling works [17], [34].The model reconstructs images from *all viewpoints excluding the reference images*, since reconstructing the reference images (which are already provided as input) would result in trivial self-reconstruction. This design enables the 3D scene encoder Ω_θ to capture essential 3D information by combining cross-view details from the reference images with information from other viewpoints not included in its inputs.

B. Regularization for Viewpoint-Invariance

In addition to the cross-view completion, we propose to regularize z_t by applying a multi-view time contrastive loss [18] to the state features $v^i_t, v^{r_1}_t, ..., v^{r_K}_t$ to ensure the 3D scene encoder Ω_θ is view-invariant. Specifically, the multi-view time contrastive loss encourages a pair of simultaneously observed state features from different viewpoints to be closer to each other, while repulsing state features from the same viewpoint but different timesteps. We set v^i_t as an anchor and randomly select a state feature from $v^{r_j}_t$, where $j \in \{1, \cdots, K\}$, as a positive, and set $v^i_{t'}$, where t' indicates a timestep distant from t, as a negative. Then, the objective can be represented as follows:

$$\mathcal{L}_{\text{cont}} = \max\left(\left\|\boldsymbol{v}^i_t - \boldsymbol{v}^{r_j}_t\right\|_2^2 - \left\|\boldsymbol{v}^i_t - \boldsymbol{v}^i_{t'}\right\|_2^2 + \alpha, 0\right) \tag{3}$$

where α is the margin that encourages dissimilar pairs and positive pairs to be distant. Finally, the overall loss function is formulated as

$$\mathcal{L}_{total} = \mathbb{E}_{\substack{t\sim\mathcal{T} \\ i\in\mathcal{V}}, \substack{r_1\cdots r_K\sim\mathcal{V} \\ (\neq i)}}\left[\mathcal{L}_{\widehat{RGB}}\right] + \mathbb{E}_{\substack{t,t'\sim\mathcal{T} \\ i\in\mathcal{V}}, \substack{r_j\sim\mathcal{V} \\ (\neq i)}}\left[\lambda_{cont}\mathcal{L}_{\text{cont}}\right] \tag{4}$$

where

$$\mathcal{L}_{\widehat{RGB}} = \sum_{\mathbf{r}_{i,k}} \left\|\hat{C}(\mathbf{r}_{i,k};\, z_t) - C(\mathbf{r}_{i,k})\right\|_2^2. \tag{5}$$

Each $\mathcal{T}$, $\mathcal{V}$ denotes timesteps and viewpoints in the dataset, r_j ($j \in \{1, \cdots, K\}$) is randomly selected reference viewpoints, and $\lambda_{cont} = 0.0004$. Since we have encouraged the 3D scene encoder Ω_θ to be not only 3D geometry-aware but also view-invariant, Ω_θ is capable of performing inference solely with single-view images, which is practically desirable for the downstream robotic tasks via RL.

C. Reinforcement Learning with 3D-aware representation

Once we train the 3D scene encoder Ω_θ, it is exploited as a 3D-aware implicit representation extractor for the downstream RL algorithm, requiring only single-view input and no camera pose. The 3D scene encoder Ω_θ takes K times replicated $O^i_{t-2:t}$ instead of using reference images from different viewpoints, i.e. $z_t = \Omega_\theta(O^i_{t-2:t}, \left[O^i_{t-2:t}\right] * K)$, where $*$ denotes replication. Then, the RL-relevant networks such as the policy and critic take z_t as an input observation. During the downstream RL process, we do not apply masking and freeze the encoder's weights.

In simulation settings, we adopt DrM [35] for the downstream online RL algorithm, which is built on top of DrQ-v2 [3]. It utilizes the dormant ratio of the neural network for active exploration-exploitation scheduling and shows state-of-the-art performance in the visual RL domain. In real-world settings, we adopt an offline RL algorithm which does not require any environment interactions during training, because performing hundreds of thousands of steps in the real world is practically impossible. Therefore, we pre-train an offline RL agent, FQL [36], with the dataset collected for the NeRF pre-training and then evaluate the learned policy in the real world. Since we consider a deployment setting with observation images from a single viewpoint, we randomly select a single viewpoint at the beginning of each episode and keep it fixed for that entire episode in simulation (online RL). In the real world (offline RL), there is no episode rollout, so we randomly select a viewpoint at every training iteration.

V. Experiment

In simulation settings, we pre-train and evaluate the proposed method for each environment in the Meta-world [37], closely following the prior work [13], while slightly modifying the environment to make it look more realistic and rich in

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

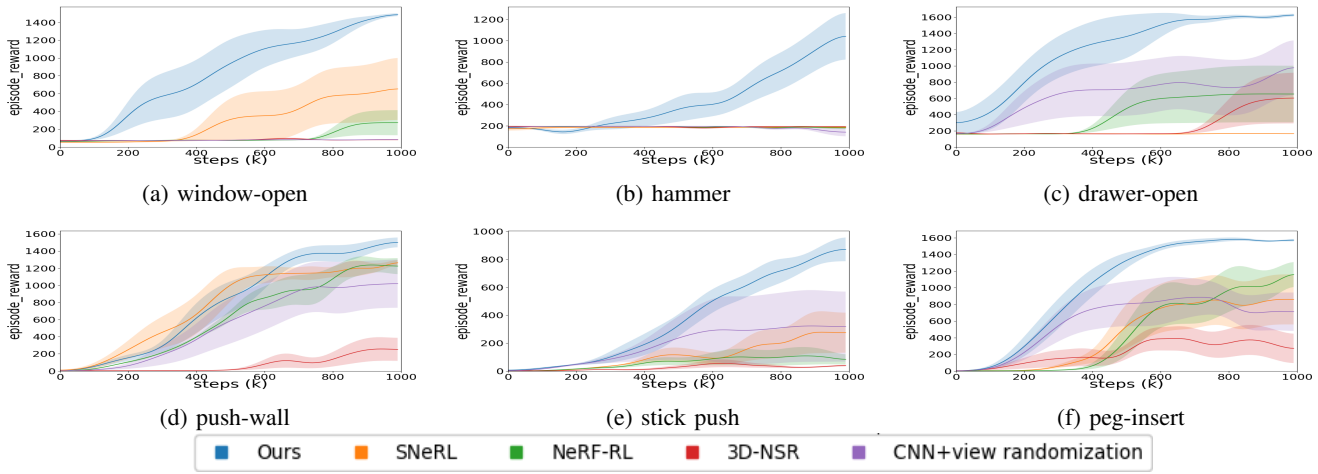


Fig. 3: Comparison of our method and baselines in simulation environments. The shaded regions represent a standard deviation across five seeds. Our method with single-view input outperforms baselines with multi-view inputs. Note that most of the NeRF-based baselines fail to perform the task with single-view input, so we do not include the results of these.

texture compared to the prior work. **In real-world settings,** we use a UR3 robot with a Robotiq 2F-85 gripper. We chose six tasks (drawer-open, window-open, hammer, push-wall, stick-push, peg-insert) in the Meta-world and designed three tasks for UR3 environments. *1) UR3 Pot Pick&Place*: The robot should pick a bowl and place it inside a pot. *2) UR3 Object Covering*: The robot should cover a cube with a side of 6 cm by manipulating a towel. *3) UR3 Kettle*: The robot should grasp the handle of a kettle and place it on top of the stove. In all environments, objects-of-interest are randomly initialized for every evaluation. The observation is a single-view RGB image and action is a 4 DoF end-effector position and gripper control for both simulation and real-world settings. For the dataset, we recorded around 100 episodic videos across six distinct viewpoints ($N=6$) (simulation: 6 mono cameras, real-world: 3 stereo cameras), ranging from random behavior to near-expert one. Specifically, we utilized the scripted policies provided in the official Meta-World repository [37] for simulation data, and we teleoperate the UR3 with a 3Dconnexion SpaceMouse (3DoF for end-effector position, 1DoF for gripper open/close) for the real-world data. We used two reference images ($K=2$) during NeRF pre-training. It took one day for 300K gradient updates on an NVIDIA A6000, using about 16 GB of GPU memory. For inference, the encoder runs at approximately 52 Hz. Downstream online/offline RL training took 36 hours on an NVIDIA A5000, and the entire network required about 4 GB of GPU memory.

To evaluate z_t's effectiveness in downstream RL, we compare our method with some prior 3D-aware representation-based RL methods, which can be summarized as follows:

NeRF-RL [12] – it performs NeRF-based 3D reconstruction by leveraging object masks for object-level reconstruction, which are required both in pre-training and deployment.

SNeRL [13] – it performs NeRF-based 3D reconstruction while distilling the feature field of DINO [38] and semantic labels of each pixel into the latent representation.

3D-NSR [14] – it learns **3D N**eural **S**cene **R**epresentations by performing self-reconstruction of multi-view images via NeRF while enforcing time contrastive loss for view-invariancy. SNeRL, NeRF-RL, and 3D-NSR require camera pose information and synchronized images from multiple viewpoints during the downstream RL process.

TABLE I: Conceptual comparisons. **Reconstruction**: whether the method performs self-reconstruction or cross-view reconstruction. **Supervision source**: the external supervision source of the representation learning objective. **Without camera pose**: the requirement for camera viewpoint information during deployment. **Single-view**: whether the 3D-aware representation can be inferred with single-view input during deployment.

	Pre-training		Downstream RL deployment	
	Reconstruction	Supervision source	Without camera pose	Single-view
SNeRL	Self	RGB, Feature Field, Semantic Label	✗	✗
NeRF-RL	Self	RGB, Object mask	✗	✗
3D-NSR	Self	RGB	✗	✗
SinCro(ours)	Cross-view	RGB	✓	✓

CNN+view randomization – as a 2D representation baseline, it constructs a naive 2D CNN as an encoder instead of the 3D scene encoder and NeRF-based 3D reconstruction. We apply random cropping for data augmentation to align with the recent image-based RL training recipe. It performs RL with single-view input while randomly selecting the viewpoint during RL, the same as SinCro is trained.

For all baselines, we concatenate proprioceptive states such as the XYZ position of the end-effector with the learned representation from each method to account for the robot's internal state, and use this combined data as input for the RL agent. While proprioceptive data is inherently view-agnostic, it serves a complementary role, focusing solely on the robot's dynamics. On the other hand, the learned 3D-aware representation remains crucial for understanding and interacting with the environment and object-of-interest. In simulation and real-world settings, we utilize DrM and FQL as the downstream RL algorithm with the same training process for a fair comparison. The conceptual comparison of the baselines and SinCro is shown in Table I.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE II: Real-world offline RL results with default viewpoints from the dataset and unseen arbitrarily perturbed viewpoints. The numbers are task success rates and standard deviations obtained by five seeds.

Task \ Method	SNeRL	NeRF-RL	3D-NSR	CNN+view Randomization	**Ours**
	Default Views				
Covering	0.40±0.16	0.35±0.25	0.30±0.12	0.10±0.20	**0.85±0.11**
Pot Pick&Place	0.25±0.10	0.80±0.05	0.55±0.19	0.25±0.30	**0.81±0.05**
Kettle	0.05±0.10	0.05±0.10	0.15±0.10	0.02±0.03	**0.84±0.09**
	Perturbed Views				
Covering	0.30±0.20	0.15±0.10	0.25±0.19	0.10±0.20	**0.65±0.20**
Pot Pick&Place	0.40±0.28	0.30±0.26	0.45±0.10	0.25±0.30	**0.63±0.08**
Kettle	0.05±0.10	0.00±0.00	0.15±0.10	0.05±0.10	**0.48±0.23**

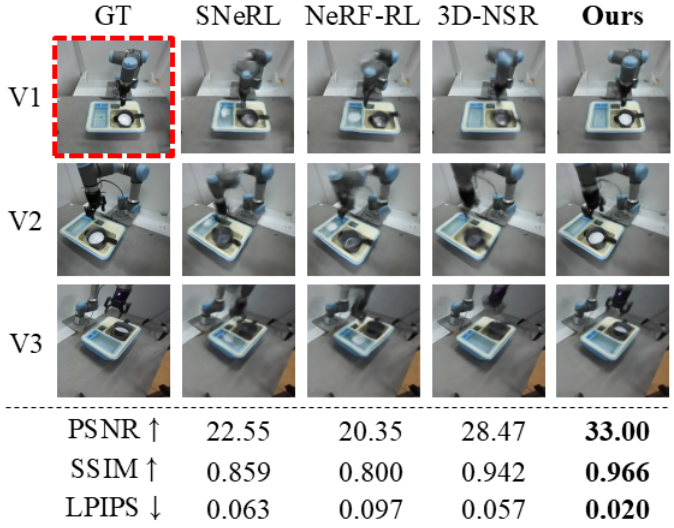
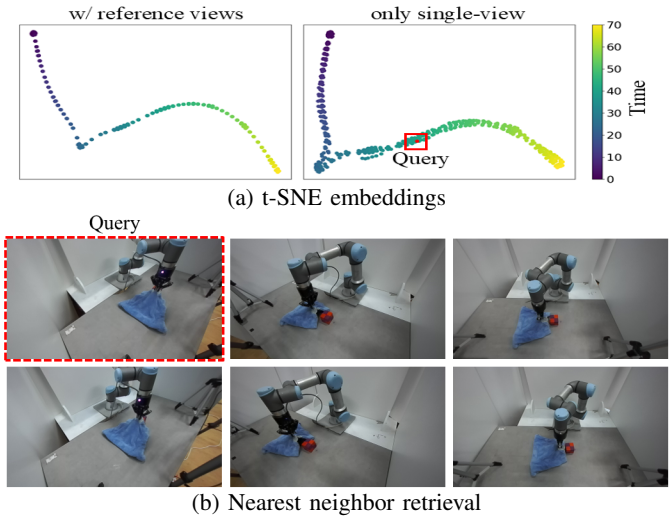


Fig. 4: 3D volume rendering results of the *UR3 Pot Pick&Place* environment (best viewed in the digital version). All methods take single-view input from V1 (outlined in red). SinCro demonstrates its ability to accurately localize the object-of-interest in the scene and achieves more consistent quantitative results than baselines. Note that we only visualize three among all viewpoints due to the page limit.

A. RL experiments & 3D scene representation validation

a) RL experiments: The downstream RL results for simulation and real-world settings are shown in Figure 3 and the upper rows of Table II. Since our method and CNN+view randomization utilize only single-view input, we evaluate these with each viewpoint used in the NeRF pre-training phase and average the results from all viewpoints. Compared to other baselines that utilize multi-view images to infer 3D-aware representation such as SNeRL, NeRF-RL, and 3D-NSR, the proposed method consistently shows superior downstream RL performances in both simulation and the real world, despite using single-view input. Since the proposed method mostly outperforms these baselines rather than just being comparable, it supports the significance of the proposed architecture and training framework for 3D geometry-aware representation. In the case of CNN+view randomization, it shows performance degradation compared to our method since it has to learn every single representation from each different viewpoint due to the lack of 3D awareness. It shows that the proposed 3D geometry-aware, view-invariant representation is crucial for consistent, reliable downstream RL performances.



(a) t-SNE embeddings

(b) Nearest neighbor retrieval

Fig. 5: Viewpoint-invariance analysis in *UR3 Object Covering* environment. (a) t-SNE embeddings of videos from six distinct viewpoints are aligned with similar timesteps across different viewpoints, even when extracted without reference images. (b) The nearest neighbor search for the query image (outlined in red) retrieves temporally aligned images from each viewpoint.

b) 3D scene representation: To validate whether the proposed method can effectively extract the essential information required to represent the 3D scene and perform downstream RL, we compare the 3D volume rendering results of the proposed method and baselines. Note that these results are indirect validations for the 3D understanding of our latent representation z_t, and high-quality image synthesis is not the primary objective. Given a task-solving trajectory, the evaluation is performed with a single-view input to consider a single-camera deployment in real-world settings.

As shown in Figure 4, the proposed method demonstrates superior scene-representing capabilities. SinCro successfully reconstructs the core components of the scene from all viewpoints and consistently achieves superior quantitative results, even though only a single-view image is provided. It can accurately position the bowl and robot arm, whereas baselines struggle to localize these key elements, often exhibiting issues such as jittering, and teleportation. This represents that SinCro implicitly captures the spatial information of the 3D world. Considering the baselines exploit additional supervision sources during pre-training or camera poses during deployment, the superior 3D spatial awareness of SinCro demonstrates the effectiveness of the proposed framework in capturing detailed 3D dynamic scenes, even with just RGB supervision and single-view inference.

B. Viewpoint-Invariance and Robustness

a) View-invariant 3D scene representation: To validate whether the latent scene representation z_t is viewpoint-invariant, we plot t-SNE embeddings of z_t in Figure 5. We collect videos of a task-solving trajectory from six different viewpoints and infer z_t for all timesteps and viewpoints with 1) multi-view input, and 2) single-view input. The results

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

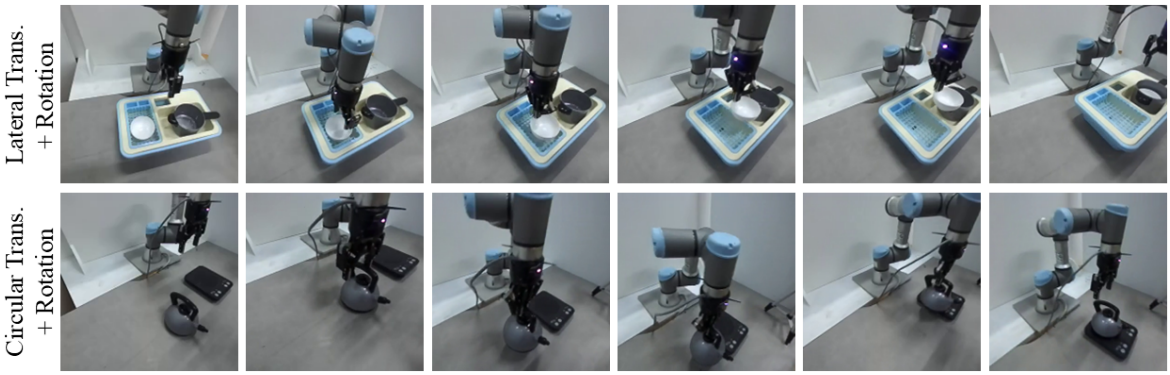
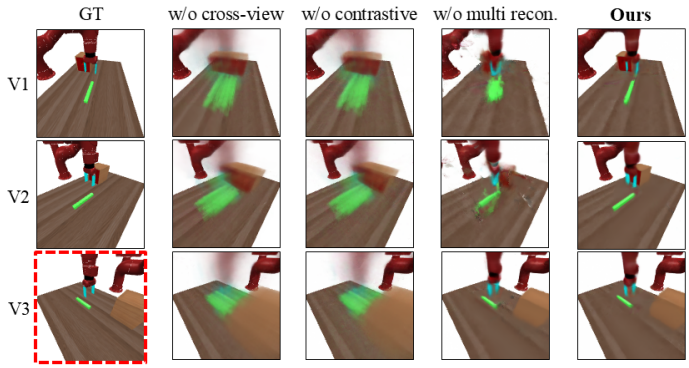
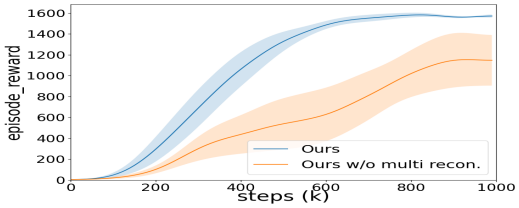


Fig. 6: Robustness to viewpoint changes. SinCro shows robust performance under various camera perturbations via translation and rotation (Bottom rows of Table II). Demonstration videos are available in the supplementary materials.



(a) Visual comparison for ablation study



(b) RL evaluation

Fig. 7: Ablation study in peg-insert environment (best viewed in the digital version). (a) Rendering results of single-view inference with a primary input V3 (outlined in red). Note that we only visualize three viewpoints in the dataset due to the page limit. (b) RL evaluation by ablating multi-view reconstruction. The performances for **w/o cross-view or contrastive** overlap to zero, so we do not plot these results.

demonstrate locally smooth and clear temporal progress of all trajectories, while the representations are closely aligned with similar timesteps across different viewpoints, even with single-view input. To further validate the view-invariance, we randomly select an image from arbitrary timestep and viewpoint, then retrieve the nearest neighbors in the latent scene representation space based on the Euclidean distance metric. The retrieved images from each viewpoint are temporally aligned, highlighting that the latent scene representation remains consistent regardless of the viewpoints.

b) Robustness to Viewpoint Changes: To further explore the benefits of 3D geometry-aware representation, we consider a viewpoint-robust control setup in the real world. We use a hand-held camera, which is continuously perturbed to the novel viewpoints (Figure 6). We evaluate the learned RL policy in Section V-A under these camera perturbations. As shown in the bottom rows of Table II, our method is surprisingly robust to the viewpoint changes even though it does not encounter any images from these perturbed viewpoints during training. It showcases the advantages of our 3D-aware representation.

C. Ablation Study

To investigate the contribution of each proposed component, we compare the single-view-based rendering results and downstream RL performance in the simulation setting by removing each one.

Cross-view completion and contrastive learning – we ablate reference images (**w/o cross-view**) or objective function (3) (**w/o contrastive**) during pre-training. Without one of these, the RL agent always fails to perform the task, so we do not include the RL results in Figure 7b. As shown in Figure 7a, the absence of these components leads to collapsed reconstructions, appearing as blurred images. It indicates the importance of both cross-view completion, which enhances the 3D scene encoder's understanding of the 3D geometry of the environment, and contrastive learning, which offers crucial guidance in distinguishing different scenes (timesteps). Together, they provide complementary benefits during pre-training, ensuring that the 3D-aware representation integrates both spatial and temporal aspects, ultimately enabling the RL agent to accomplish the task.

Multi-view reconstruction – the absence of multi-view reconstruction (**w/o multi recon.**) leads to RL performance degradation. This is because the 3D scene encoder appears to lose some 3D scene awareness, which is essential for effective policy learning. For example, the failure to localize key elements, such as the robot gripper, unobserved parts in the primary image (see V1, V2), and the accurate position of the green peg, impacts the downstream RL tasks. It suggests that multi-view reconstruction plays a critical role in improving the encoder's understanding of 3D geometry, eventually enhancing the downstream RL performance.

VI. Conclusion

In this work, we considered a 3D-aware implicit representation-based RL framework where the agent should learn how to perform the given task by leveraging information from multiple viewpoints. We proposed SinCro, which can

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

extract 3D geometry-aware representation while enabling a single-view inference without synchronized calibrated cameras during deployment. We have shown that the proposed method outperforms the baselines in qualitative and quantitative ways. However, SinCro still has some limitations. For example, although multi-view data collection is required only once, the need for synchronized and calibrated cameras during NeRF pre-training may entail additional human effort compared to a single camera setup. A promising future direction is to replace these prerequisites with multiple videos captured from a moving camera at varying viewpoints. Also, even though our method has shown a promising way for single-view 3D representation, the current approach has difficulty in generalization, such as object categories and backgrounds, due to its scene-specific training scheme. For more general application, we could consider additional 3D scene, object generation methods such as [39], [40] for data augmentation.

REFERENCES

[1] M. Laskin, A. Srinivas, and P. Abbeel, "Curl: Contrastive unsupervised representations for reinforcement learning," in *International conference on machine learning*. PMLR, 2020, pp. 5639–5650.

[2] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.

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

[4] D. Yarats, I. Kostrikov, and R. Fergus, "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels," in *International conference on learning representations*, 2021.

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

[6] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik, "Masked visual pre-training for motor control," *arXiv preprint arXiv:2203.06173*, 2022.

[7] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang, "Vip: Towards universal visual reward and representation via value-implicit pre-training," *arXiv preprint arXiv:2210.00030*, 2022.

[8] D. Ghosh, C. A. Bhateja, and S. Levine, "Reinforcement learning from passive data via latent intentions," in *International Conference on Machine Learning*. PMLR, 2023, pp. 11 321–11 339.

[9] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3d diffuser actor: Policy diffusion with 3d scene representations," *arXiv preprint arXiv:2402.10885*, 2024.

[10] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki, "Act3d: 3d feature field transformers for multi-task robotic manipulation," in *7th Annual Conference on Robot Learning*, 2023.

[11] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox, "Rvt-2: Learning precise manipulation from few demonstrations," *arXiv preprint arXiv:2406.08545*, 2024.

[12] D. Driess, I. Schubert, P. Florence, Y. Li, and M. Toussaint, "Reinforcement learning with neural radiance fields," *Advances in Neural Information Processing Systems*, vol. 35, pp. 16 931–16 945, 2022.

[13] D. Shim, S. Lee, and H. J. Kim, "Snerl: Semantic-aware neural radiance fields for reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 31 489–31 503.

[14] Y. Li, S. Li, V. Sitzmann, P. Agrawal, and A. Torralba, "3d neural scene representations for visuomotor control," in *Conference on Robot Learning*. PMLR, 2022, pp. 112–123.

[15] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[16] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[17] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[18] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain, "Time-contrastive networks: Self-supervised learning from video," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 1134–1141.

[19] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto, "Reinforcement learning with prototypical representations," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 920–11 931.

[20] H. Liu and P. Abbeel, "Aps: Active pretraining with successor features," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6736–6747.

[21] ——, "Behavior from the void: Unsupervised active pre-training," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 459–18 473, 2021.

[22] A. V. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine, "Visual reinforcement learning with imagined goals," *Advances in neural information processing systems*, vol. 31, 2018.

[23] Y. Seo, J. Kim, S. James, K. Lee, J. Shin, and P. Abbeel, "Multi-view masked world models for visual robotic manipulation," in *International Conference on Machine Learning*. PMLR, 2023, pp. 30 613–30 632.

[24] Y. Seo, K. Lee, S. L. James, and P. Abbeel, "Reinforcement learning with action-free pre-training from videos," in *International Conference on Machine Learning*. PMLR, 2022, pp. 19 561–19 579.

[25] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *arXiv preprint arXiv:1912.01603*, 2019.

[26] B. Chen, P. Abbeel, and D. Pathak, "Unsupervised learning of visual 3d keypoints for control," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1539–1549.

[27] K. Hsu, M. J. Kim, R. Rafailov, J. Wu, and C. Finn, "Vision-based manipulators need to also see from their hands," *arXiv preprint arXiv:2203.12677*, 2022.

[28] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 785–799.

[29] S. Qian, K. Mo, V. Blukis, D. F. Fouhey, D. Fox, and A. Goyal, "3d-mvp: 3d multiview pretraining for robotic manipulation," *arXiv preprint arXiv:2406.18158*, 2024.

[30] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy," *arXiv preprint arXiv:2403.03954*, 2024.

[31] A. Cao and J. Johnson, "Hexplane: A fast representation for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 130–141.

[32] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 318–10 327.

[33] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa, "K-planes: Explicit radiance fields in space, time, and appearance," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 479–12 488.

[34] P. Weinzaepfel, V. Leroy, T. Lucas, R. Brégier, Y. Cabon, V. Arora, L. Antsfeld, B. Chidlovskii, G. Csurka, and J. Revaud, "Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion," *Advances in Neural Information Processing Systems*, vol. 35, pp. 3502–3516, 2022.

[35] G. Xu, R. Zheng, Y. Liang, X. Wang, Z. Yuan, T. Ji, Y. Luo, X. Liu, J. Yuan, P. Hua, *et al.*, "Drm: Mastering visual reinforcement learning through dormant ratio minimization," *arXiv preprint arXiv:2310.19668*, 2023.

[36] S. Park, Q. Li, and S. Levine, "Flow q-learning," *arXiv preprint arXiv:2502.02538*, 2025.

[37] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on Robot Learning*. PMLR, 2020, pp. 1094–1100.

[38] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9650–9660.

[39] B. Ma, H. Gao, H. Deng, Z. Luo, T. Huang, L. Tang, and X. Wang, "You see it, you got it: Learning 3d creation on pose-free videos at scale," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 2016–2029.

[40] Y. Hong, K. Zhang, J. Gu, S. Bi, Y. Zhou, D. Liu, F. Liu, K. Sunkavalli, T. Bui, and H. Tan, "Lrm: Large reconstruction model for single image to 3d," *arXiv preprint arXiv:2311.04400*, 2023.