CGGT: Concept-Guided Grounded Training for Generalizable and Interpretable Reinforcement Learning in Robot Control

Anonymous Authors

***Abstract*— Visuomotor reinforcement learning (RL) has shown a strong capability in enabling robotic agents to perceive their environments and generate actions to accomplish downstream manipulation tasks. However, existing approaches often do shortcut learning, resulting in poor performance under out-of-distribution (OOD) conditions, while also inducing incomprehensible interpretations. To address these issues, we propose Concept-Guided Grounded Training (*CGGT*), a framework that leverages task-specific concepts from the environment to improve visuomotor learning, not only in terms of success rates but also in terms of generalization and interpretability. Specifically, our *CGGT* framework employs a single stage training scheme that integrates visual concept alignment at the feature extraction and spatial concept alignment at the decision-making network, along with the reward optimization. This approach enables robots to focus on action-relevant features and subsequently generate actions for task completion. Through our evaluations on eleven `robotsuite` tasks with multiple OOD scenarios in the `RL-Vigen` benchmark, *CGGT* outperforms state-of-the-art baselines in terms of generalization while also enabling failure inspection through concept visualizations. Finally, we also deploy our proposed method on the Franka Emika robot in four different tasks with real-world objects to test its feasibility in the real-world domain.**

I. Introduction

Training visuomotor control agents using deep reinforcement learning (RL) has recently been a compelling topic in robotics, thanks to its remarkable performance on various simulation benchmarks [1]–[3]. With visual observations as inputs, numerous auxiliary techniques have been introduced to improve generalization, such as data augmentation [4]–[6], contrastive learning [7], and representation learning approaches that leverage saliency masks [8], multi-view encoders [9], or spatial transformer networks [10]. While these methods enhance robustness by encouraging agents to learn invariant latent features, they primarily rely on self-exploration to extract task-relevant information, which often leads to unstable training dynamics and high variance in visual representations [4], [11]. Moreover, redundant features [12] and shortcut learning behaviors [13] remain prevalent, leading to policies that generalize poorly when deployed in visually perturbed, like out-of-distribution (OOD) scenarios.

Compounding these issues, deep RL policies are opaque, making it difficult to detect when shortcut learning affects the model's behavior [14], [15]. Previous works have explored interpretable approaches, such as decision trees, logic-based policies [16], [17], and reward-driven masking strategies [18]. However, these approaches struggle to scale to the complexity of visuomotor control. Within this realm, concept bottleneck models (CBMs) [19], [20] offer a promising alternative by introducing explicit intermediate concepts that improve transparency while supporting more stable and robust learning. Recent applications of CBMs in RL [21]–[23] highlight their potential for interpretability, though remaining limited to simplified and constrained environments, without any clear generalizations. Yet, applying concept-based learning in the context of robotic RL is substantially more challenging than in static image classification, as it must capture both spatial and temporal dynamics, which directly complicates concept design and reduces training efficiency.

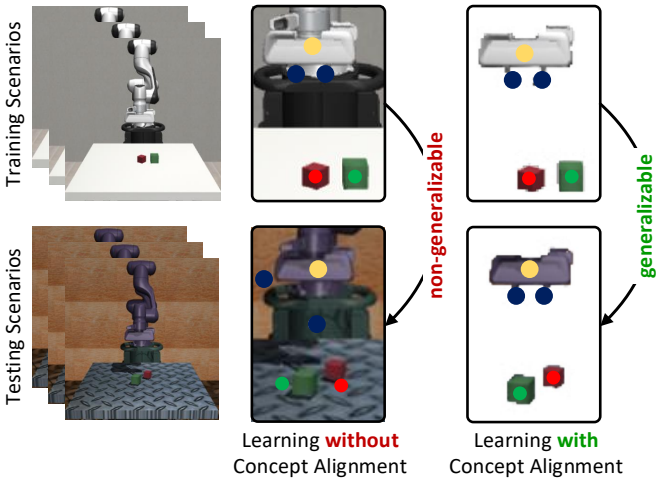


Fig. 1: Overview of *CGGT*: The learned RL agent with concept alignments is guided to concentrate on task-relevant objects, extracts correct spatial coordinates, and yields actions that maximize reward even with out-of-distribution (OOD) scenarios, showing strong generalization capabilities. Without concept alignments, the RL agent tends to overfit to training scenarios, eventually leading to poor performance when encountering OOD evaluations.

To apply CBMs to guide the deep RL model in extracting features in an interpretable manner, we introduce the Concept-Guided Grounded Training (*CGGT*) framework, which integrates visual and spatial concept alignments to direct the agent's attention toward task-relevant features. Specifically, *CGGT* employs a trainable masking mechanism that suppresses irrelevant regions in the visual domain, exposing only task-critical features and thereby improving generalization under visually perturbed conditions. The mask is thus aligned with segmentations of task-specific features, ensuring that the latent representation encodes only meaningful visual features as an essential step in handling the complexity of environments. Beyond visual alignment, *CGGT* leverages spatial information to extract compact bottleneck features that capture object relations for effective action generation. By doing so, both visual and spatial concepts serve as interpretable probes during OOD inference, allowing for the

inspection of whether the agent attends to the correct objects and maintains accurate spatial reasoning under distribution shifts. Our contributions are summarized as follows:

- We propose a visuomotor RL training framework that leverages both visual and spatial concepts to guide training to improve interpretability and generalization.
- We do comprehensive experiments in `robosuite` environment [24] to evaluate the performance of *CGGT* at three generalization levels from `RL-ViGen` benchmark [25] against to state-of-the-art approaches.
- We empirically demonstrate the feasibility and generalization of our proposed framework on a real-robot system with real-world objects.

II. Related Work

Generalizable Visual RL for Robot Control: Vision-based model-free RL has been widely applied in robotic control [1], [26]–[28]. Recent studies have tackled overfitting and improved generalization in RL through complementary strategies, including invariant representation learning [10], environment augmentation and normalization [4], [29]–[32], uncertainty quantification [33], and task decomposition for modular training [18], [34]. Data augmentation has been a central strategy, broadening the visual distribution to improve robustness. For example, Bertoin *et al.* [8] propose saliency-guided training, while Grooten *et al.* [18] introduce distraction masking, and Seo *et al.* [9] develop multi-view masked encoders for model-based RL. Another work of Yuan *et al.* [10] leverage multi-view representations with spatial transformer networks to capture invariant latent features. These approaches primarily emphasize learning representations that are robust to visual perturbations, but they often rely on self-exploration and can retain irrelevant information. In contrast, our method enables the model to identify and discard task-irrelevant features, leading to an improvement in generalization for visuomotor tasks in complex environments.

Reinforcement Learning with Interpretability: Interpretable RL is an active area within the field of explainable artificial intelligence, aiming to provide RL models with interpretable capabilities [14], [15], [35]. Recent approaches often encode deep RL policies using interpretable models such as decision trees [16], [36], [37] and logical programs [17], [38], [39]. Other studies leverage large language models to generate natural-language explanations of agent behavior [40], [41]. Bertoin *et al.* [8] employ saliency-guided masks, and Grooten *et al.* [18] introduce reward-based distraction masks to suppress irrelevant pixels and highlight task-relevant regions. Unlike prior approaches, our work utilizes supervised visual masks as extractable concepts to ensure that the robots focus on task-relevant objects, even when visual inputs change. In addition, spatial concept predictions provide an extra layer of interpretability, enabling the analysis of the agent's decisions in failure cases.

Concept Bottleneck Models as Learning Strategy: The concept of CBMs was initially proposed to improve the interpretability of deep learning models by introducing an intermediate, human-interpretable representation [19], [20]. These models follow a two-stage training approach, first extracting interpretable concepts from the input and then using these concepts for the final task prediction. Integrating the concepts of CBMs into RL-based problems, Zabounidis *et al.* [21] and Grupen *et al.* [22] apply CBMs in multi-agent settings, demonstrating that agents could achieve improved interpretability through concept representations while maintaining strong performance. Delfosse *et al.* [16] further propose SCoBots, which uses consecutive CBMs to extract interpretable concepts across sequential feature spaces. However, due to the complexity of defining and training heterogeneous concepts, SCoBots separates concept learning from RL, limiting its integration. Moreover, these prior works are primarily restricted to 2D game environments, which are much simpler than real-world robotic control in terms of concept learning and decision-making. In this work, we leverage and incorporate CBMs to guide robots to learn efficiently in complex spatial environments, thus enhancing both interpretability and generalization for visuomotor control.

III. Problem Formulation

A. Preliminaries

In general, the visual RL problem is formulated as a Markov Decision Process (MDP), denoted as $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma \rangle$, where $\mathcal{S}$ as the state space, $\mathcal{A}$ as the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ as the transition function, $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$ as the reward function, and $\gamma \in [0, 1]$ as the discount factor. A policy with learnable parameters θ, $\pi_\theta : \mathcal{S} \rightarrow \mathcal{A}$, learns to maximize the discounted return $\mathcal{R}_t = \mathbb{E}_{\Gamma \sim \pi} \left[\sum_{t=0}^{T} \gamma^t \, r(s_t, a_t) \right]$ along the visual state trajectory $\Gamma = (s_0, s_1, \ldots, s_T)$. Traditional approaches use an encoder, $f_\theta : \mathcal{S} \rightarrow \mathcal{Z}$, to encode visual state inputs into a latent representation $\mathcal{Z}$, thus deciding optimal actions through an actor network, $\phi_\theta : \mathcal{Z} \rightarrow \mathcal{A}$, to gain the discounted return.

In the scope of continuous control for robotic manipulation, the deep deterministic gradient policy with random exploration noise [28] is typically used with the main objective of optimizing the Bellman residual equation [42]:

$$R = r(s_t, a_t) + \gamma \max \left\{ Q^*_\theta \left(s_{t+1}, a^{'}_t \right) \right\} - Q_\theta(s_t, a_t), \quad (1)$$

where Q^* represents the optimal state-action value function, Q_θ is the parameterized critic network with learnable parameters θ, s_t and s_{t+1} are the two consecutive visual states, a_t denotes what the agent actually did at the state s_t, and $a^{'}_t$ denotes any possible action at the subsequent state s_{t+1}. The actor network is modeled as a deterministic policy, ϕ_θ, that aims to maximize the expected Q-value predicted by the critic network, which can be expressed as follows:

$$\mathcal{L}_\phi = \mathbb{E}_{s_t \sim D} \left[\mathbb{E}_{a_t \sim \phi_\theta} \left[-Q_\theta(s_t, a_t) \right] \right] \quad (2)$$

The training process of the RL agent depends on the exploration strategy to collect samples and optimize the networks based on Eq. 1 and Eq. 2. As a result, the visual RL agent is considered a black-box that processes the raw input visual states to provide the action distribution. Meanwhile, the generalizability of the RL model is contingent upon the

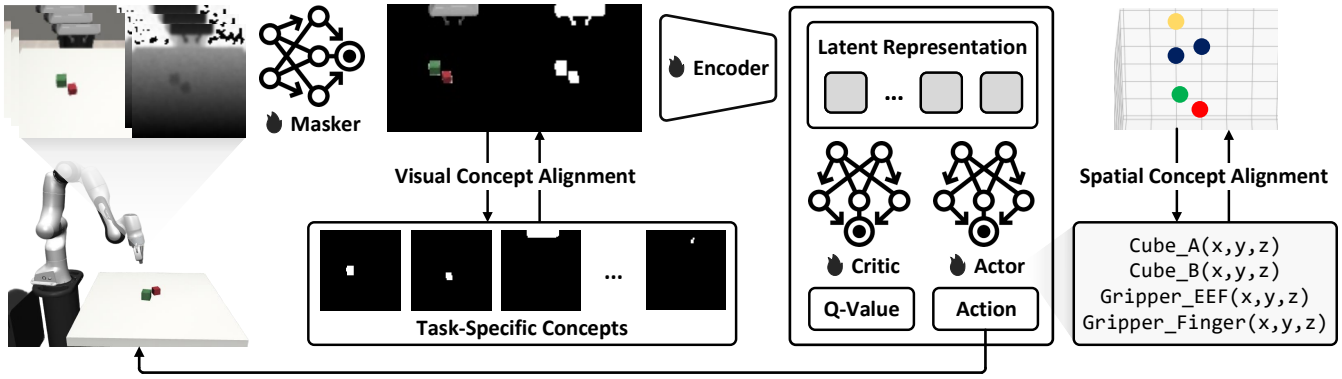


Fig. 2: Architecture of *CGGT*: Given a task definition, the robot first observes the environment to seek out task-relevant visual features through a trained masker, while being grounded with the alignment with task-specific concepts. The filtered observations are then encoded as a latent representation, which is fed to train an actor-critic network. While the critic network evaluates the quality of the action, the actor network generates robot actions that are aligned with extracted spatial concepts of task-relevant objects in the environment.

agent's ability to extract the Visual Perturbation-Invariant Representation (VPIR) [43], denoted as z^*:

$$z^* = f_\theta(s) = f_\theta(\overline{s}), \text{ with } s \in \mathcal{S},\ \overline{s} \in \overline{\mathcal{S}}, \tag{3}$$

where $\overline{s}$ represents the perturbed counterpart of the visual state s, which is transformed by data augmentation. However, to mitigate the shortcut problem, there is no indication of whether the agent extracts VPIR features thanks to the black-box nature, nor of the mechanism that guides the RL model in extracting VPIR features.

B. Assumptions & Problem Formulation

1) Assumptions: To formulate our problem, we first make the following key assumptions about the environment:

- *Assumption 1:* The robot environment must contain a set of objects that is sufficient for the robot to accomplish downstream manipulation tasks.
- *Assumption 2:* Visual disturbances, including colors, textures, blurriness, brightness, etc., must not affect the agent's performance in terms of generalization.

2) Problem Formulation: Thus, our goal is to guide the RL model to optimally focus on task-relevant features while suppressing irrelevant ones in the latent space z_t^* in order to improve generalization in OOD scenarios. We also promote the interpretability of the model through the visualization of the agent's focus on such invariant features across multiple visual variations. Therefore, to achieve robust VPIR shown in Eq. 3, the optimal policy is to learn to encode only from task-relevant information s^{Ω^+}:

$$z_t^* = f_\theta\left(s_t^{\Omega^+}\right) = f_\theta\left(\overline{s}_t^{\Omega^+}\right), \tag{4}$$

where Ω^+ represents the set of task-relevant objects, containing the necessary information for task accomplishment, and Ω^- denotes the remaining observable objects, which are considered task-irrelevant. Note that $s^{\Omega^+} \cup s^{\Omega^-} = s$ and $s^{\Omega^+} \cap s^{\Omega^-} = \varnothing$.

IV. Concept-Guided Grounded Training

A. Concept Bottleneck Methodology

We utilize the concept bottleneck to guide the visual encoder in extracting and transforming features into the concept space $\mathcal{C}$. The concept bottleneck network $\zeta : \mathcal{Z} \rightarrow \mathcal{C}$ projects the latent representation z into human-understandable concepts, whereas the shortcut problem can be identified through inspection of involved concepts in the decision-making process. We categorize the concepts into visual and spatial ones:

1) The visual concepts include the visual features extracted by the encoder, which comprise properties of objects, such as shapes and colors. These features are explicitly defined in the latent space z of the CBMs.
2) The spatial concepts capture the spatial relations of objects in the spatial domain, such as spatial coordinates or orientations of task-relevant objects, which are first and foremost information for robot control.

Specifically, we design concept alignment modules at the feature extraction and decision-making networks to address their internal differences in scales, data types, and concept variations.

B. Visual Concept Alignment

While focusing on Ω^+, Ω^- is large and complex, which makes the optimal policy learning cumbersome. To solve this issue, we design a masker network M_θ that learns to predict the mask of Ω^+, denoted as $m_{\text{predict}}^{\Omega^+}$, in RGB observations:

$$m_{\text{predict}}^{\Omega^+} \leftarrow M_\theta\left(s_t^{\text{D}}\right), \tag{5}$$

where s_t^{D} denotes depth observation, which is used to predict the mask in a stable manner under visual disturbances.

The predicted mask $m_{\text{predict}}^{\Omega^+} \in [0,1]$ assigns each pixel in the corresponding RGB observation s_t^{RGB} as a confidence score indicating its relevance to the task. The masker network M_θ is trained by minimizing the Binary Cross Entropy (BCE) distance between $m_{\text{predict}}^{\Omega^+}$ and the ground-truth segmentation mask $m_{\text{sim}}^{\Omega^+}$ of task-relevant objects in simulation, defined as $\mathcal{L}_{\text{visual}} = \mathcal{D}_{\text{BCE}}\left(m_{\text{predict}}^{\Omega^+}, m_{\text{sim}}^{\Omega^+}\right)$. Finally, we construct a filtered observation that retains only the task-relevant components by applying an element-wise product between the RGB observation and the predicted mask:

$$s_t^{\Omega^+} \leftarrow s_t^{\text{RGB}} \odot m_{\text{predict}}^{\Omega^+} \tag{6}$$

Algorithm 1: Concept-Guided Grounded Training

Input : $s \coloneqq$ RGB-D state
Output: $s^{\Omega^+} \coloneqq$ masked visual state
$m^{\Omega^+}_{\text{predict}} \coloneqq$ predicted mask

function `Masking(`s`)`
  $s^{\text{RGB}}, s^{D} = s$
  $m^{\Omega^+}_{\text{predict}} \leftarrow M_\theta(s^D)$ (Eq. 5)
  $s^{\Omega^+} \leftarrow s^{\text{RGB}} \odot m^{\Omega^+}_{\text{predict}}$ (Eq. 6)
  return $s^{\Omega^+}, m^{\Omega^+}_{predict}$

Input : $f_\theta, Q_\theta, \phi_\theta, \Psi_\theta, M_\theta \coloneqq$ initialized networks
$Q^{tgt}_\theta, f^{tgt}_\theta \leftarrow Q_\theta, f_\theta \coloneqq$ target networks
$T \coloneqq$ total timesteps
Output: $f_\theta, Q_\theta, \phi_\theta, \Psi_\theta, M_\theta \coloneqq$ trained networks

function `train(`$f_\theta, Q_\theta, \phi_\theta, \Psi_\theta, M_\theta, T$`)`
  for $t \in [1, \dots, T]$ **do**
    ▷ ------- collect samples -------
    $s^{\Omega^+}_t, \_ \leftarrow$ `Masking(`s_t`)`
    $a_t \sim \phi_\theta(\cdot | f_\theta(s^{\Omega^+}_t))$
    $s_{t+1}, r_t \sim \mathcal{P}(\cdot | s_t, a_t)$
    $\mathcal{B} \leftarrow \mathcal{B} \cup (s_t, a_t, r_t, s_{t+1}, m^{\Omega^+}_{\text{sim}}, c^{\text{spatial}}_{\text{sim}})$
    ▷ ------- update phase -------
    $\{s_b, a_b, r_b, s'_b, m^{\Omega^+}_{\text{sim}}, c^{\text{spatial}}_{\text{sim}}\} \sim \mathcal{B}$
    $s^{\Omega^+}_b, m^{\Omega^+}_{\text{predict}} \leftarrow$ `Masking(`s_b`)`
    $\overline{s}^{\Omega^+}_b \leftarrow \tau(s_b, \nu)$ (Augmentation)
    ▷ -- update actor and critics --
    $\theta_\phi \leftarrow \theta_\phi - \nabla_{\theta_\phi} \mathcal{L}_\phi$ (Eq. 2)
    for $N \in \{Q_\theta, f_\theta\}$ **do**
      $\theta_N \leftarrow \theta_N - \nabla_{\theta_N} \mathcal{L}_Q$ (Eq. 10)
    $\theta_{Q^{tgt}} \leftarrow (1-\tau)\theta_{Q^{tgt}} + \tau\theta_Q$ (Soft updates)
    ▷ ------- concept alignment -------
    for $N \in \{\Psi_\theta, \phi_\theta, f_\theta, M_\theta\}$ **do**
      $\theta_N \leftarrow \theta_N - \nabla_{\theta_N}(\mathcal{L}_{\text{spatial}} + \mathcal{L}_{\text{visual}})$
    $\theta_f \leftarrow \theta_f - \nabla_{\theta_f} \mathcal{L}_{\text{InfoNCE}}$
  return $f_\theta, Q_\theta, \phi_\theta, \Psi_\theta, M_\theta$

From Eq. 5 and Eq. 6, we can ensure that the encoder only selects the task-relevant objects to extract robust VPIR. Levaraging the invariance of s^{D}_t under augmentation, we also apply the predicted mask $m^{\Omega^+}_{\text{predict}}$ onto the perturbed state $\overline{s}^{\text{RGB}}_t$ to produce the perturbed observation of task-relevant objects $\overline{s}^{\Omega^+}_t \leftarrow \overline{s}^{\text{RGB}}_t \odot m^{\Omega^+}_{\text{predict}}$.

To further eliminate irrelevant visual features, including colors, textures, and brightness, we employ contrastive learning [44] to minimize the discrepancy between $f_\theta(s^{\Omega^+})$ and $f_\theta(\overline{s}^{\Omega^+})$ and to generate the compact and minimal VPIR z^*, as defined in Eq. 4. We adopt InfoNCE [45] to define the contrastive loss $\mathcal{L}_{\text{InfoNCE}}$ as follows:

$$-\log\left[\frac{\exp\left(\frac{\text{sim}(\mathbf{q}^T,\mathbf{k}^+)}{\tau}\right)}{\exp\left(\frac{\text{sim}(\mathbf{q}^T,\mathbf{k}^+)}{\tau}\right) + \sum_{i=0}^{M}\exp\left(\frac{\text{sim}(\mathbf{q}^T,\mathbf{k}^-_i)}{\tau}\right)}\right], \tag{7}$$

where $\text{sim}(\mathbf{q},\mathbf{k}) = \frac{\mathbf{q}^T\mathbf{k}}{\|\mathbf{q}\|\|\mathbf{k}\|}$ is the normalized dot product measure of the similarity between $\mathbf{q}$ and $\mathbf{k}$, and τ represents the learning temperature. In specific, we use the contrastive learning strategy to maximize the similarity between the query vector $\mathbf{q} = f_\theta(s^{\Omega^+}_t)$ and the positive key $\mathbf{k}^+ = f_\theta(\overline{s}^{\Omega^+}_t)$, while minimizing the similarities between the query and the representations of the next states of all other samples.

Instead of aligning concepts in the latent space, this approach constrains the latent representation to retain only task-relevant visual features. This means that the latent space is not directly interpretable at this step for robot control; the mask $m^{\Omega^+}_{\text{predict}}$ only provides a means to visualize the agent's focus during inference in OOD scenarios.

C. Spatial Concept Alignment

Let $c^{\text{spatial}} \in \mathcal{C}^{\text{spatial}}$ denote the Cartesian coordinates of important objects in the environment. The actor network, ϕ_θ, is formulated as a bottleneck projector $\zeta_\theta : \mathcal{Z} \to \mathcal{F}$ to compress the latent representation into feature space $\mathcal{F}$, and a decision maker $\Upsilon_\theta : \mathcal{F} \to \mathcal{A}$ that decides the action distribution. Mathematically:

$$\rho_t = \zeta_\theta(z^*_t), a_t \leftarrow \Upsilon_\theta(\rho_t). \tag{8}$$

Unlike CBMs, where the feature space $\mathcal{F}$ is explicitly aligned with the concept space $\mathcal{C}_{\text{spatial}}$, we contend that the spatial concept space is not inherently structured to support efficient network optimization. While features normalized by `LayerNorm` are uniformly distributed in the feature space, thus facilitating gradient descent during training [46], spatial coordinates rarely exhibit such properties. For instance, when the robot picks the cube, the cube's coordinates should remain unchanged until it is grasped, and the gripper's displacements vary across spatial directions. In brief, aligning two spaces with such disparate characteristics introduces unnecessary training complexity and degrades performance.

From Eq. 8, we introduce an auxiliary concept predictor $\Psi_\theta : \mathcal{F} \to \mathcal{C}_{\text{spatial}}$ to convert features into Cartesian coordinates to improve the learning efficiency for both concept learning and reward training. The output of the concept predictor, $c^{\text{spatial}}_{\text{predict}}$ is aligned with the spatial coordinates $c^{\text{spatial}}_{\text{sim}}$ stored in the replay buffer for each state, using mean squared error (MSE) as the spatial loss, as follows:

$$c^{\text{spatial}}_{\text{predict}} = \Psi_\theta(\rho_t), \tag{9a}$$

$$\mathcal{L}_{\text{spatial}} = \mathcal{D}_{\text{MSE}}(c^{\text{spatial}}_{\text{predict}}, c^{\text{spatial}}_{\text{sim}}). \tag{9b}$$

The spatial loss $\mathcal{L}_{\text{spatial}}$ in Eq. 9b incentivizes the bottleneck projector ζ_θ to produce the features ρ containing spatial information to predict the accurate spatial coordinates, focusing on the right features for generalization. As the critic network Q_θ is used to evaluate the action produced by the actor, we did not apply the spatial concept alignment, allowing the critic to look at multiple aspects of the environment. In addition, to stabilize the critic learning under augmentation, we adopt SVEA [4] framework to isolate the augmented and unaugmented data streams as a linear combination of two Bellman Residuals (Eq. 1) of $s^{\Omega^+}_t$ and $\overline{s}^{\Omega^+}_t$:

$$\mathcal{L}_Q = \alpha R^2(s^{\Omega^+}_t, a_t, r_t, s^{\Omega^+}_{t+1}) + \beta R^2(\overline{s}^{\Omega^+}_t, a_t, r_t, s^{\Omega^+}_{t+1}), \tag{10}$$

where α and β are the coefficients to balance the ratio of the two data streams. The critic target network Q^{tgt}_θ is updated

by an exponential moving average from Q_θ, avoiding the critic updates toward the moving target. We summarize the concept guided training in Alg. 1 and illustrate it in Fig. 2.

V. Evaluations & Ablation Analyses

A. Baselines & Experiment Setup

1) Baselines & Environments: To validate our proposed method's performance, we conduct experiments on eleven tasks in `robosuite` [24] at three generalization levels from the `RL-ViGen` benchmark [25] (Fig. 3) on the Franka Emika robot. We compare *CGGT*'s performance against state-of-the-art baselines, all with the backbone of DrQv2 [28]:

- SVEA [4]: a stabilized actor-critic framework with data augmentation.
- SGQN [8]: a framework that leverages salient maps to train agents to focus only on important locations.
- MaDi [18]: a method that applies masks on visual inputs to reduce the effect of distractions, improving the generalization of the trained agent.
- Maniwhere [10]: the most recent state-of-the-art method for visual RL in generalization.

Fig. 3: Training (`train`) and Evaluation Environments: `easy` includes changes of the background and object appearances, while `medium` alters light direction and arm color, and `hard` adds a more complex texture and overlay as visual disturbances.

Besides `train`, each evaluation mode (Fig. 3) consists of ten scenarios with distinct and visual changes across the evaluated tasks. Throughout our experiments, we run ten episodes on three random seeds for each scenario.

TABLE I: Training Specifications: Number of robot arms, frozen/trained DoFs, and training steps in eleven evaluated tasks.

Task \ Settings	No. of Arms	Frozen DoFs	Trained DoFs	Training Steps
`Lift`	1	3	4	1e6
`Door`	1	0	7	1e6
`TwoArmPegInHole`	2	0	14	1e6
`Stack`	1	3	4	3e6
`NutAssemblyRound`	1	3	4	3e6
`NutAssemblySquare`	1	3	4	3e6
`PickPlaceCan`	1	3	4	3e6
`PickPlaceBread`	1	3	4	3e6
`PickPlaceCereal`	1	3	4	3e6
`PickPlaceMilk`	1	3	4	3e6
`TwoArmLift`	2	4	10	3e6

2) Experiment Setup: We utilize RGB-D observations of size $(84, 84, 4)$ as input for all models. The RGB input is augmented with `random_shift` [47], `random_overlay` [5], and `random_color_jitter` [48]. For the depth input, we add Gaussian noise $\mathcal{N}(0, 0.005)$ and depth-dependent noise $\mathcal{N}(0, \texttt{depth\_scale})$ with Gaussian blur to mimic realistic depth images. Specifically, we train $1,000,000$ steps for simple tasks, such as `Lift`, `Door`, and `TwoArmPegInHole`, and for $3,000,000$ steps for the remaining challenging tasks. The training specifications are outlined in Tab. I. Note that for some tasks, we lock unnecessary robot degrees of freedom (DoFs) to promote training efficiency.

TABLE II: Generalization Results: Episode returns (μ, σ) on eleven evaluated `robosuite` tasks in the `RL-Vigen` benchmark.

Mode	SVEA [4]	SGQN [8]	MaDi [18]	ManiWhere [10]	**CGGT (Ours)**
`train`	301.46 ± 9.84	308.63 ± 14.21	312.13 ± 28.81	**314.38 ± 8.66**	310.25 ± 11.97
`easy`	108.44 ± 68.43	153.87 ± 81.68	149.13 ± 73.29	182.29 ± 64.37	**263.33 ± 66.61**
`medium`	40.19 ± 23.85	42.28 ± 25.50	48.86 ± 18.42	135.91 ± 77.53	**219.99 ± 94.89**
`hard`	18.47 ± 11.67	12.53 ± 19.91	16.86 ± 10.24	108.77 ± 68.11	**202.85 ± 99.31**

B. Generalization Results

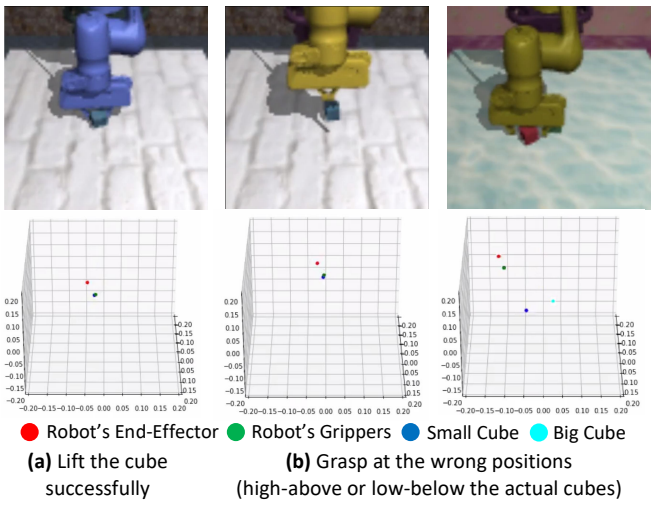
As shown in Tab. II, our proposed framework consistently outperforms all baselines across modes of evaluations. SVEA achieves only 35.82% of its training performance on average, while SGQN, MaDi, and ManiWhere lose nearly half of their returns. For example, SGQN drops from 308.63 ± 14.21 in training to 153.87 ± 81.68 in the `easy` setting, MaDi falls from 312.13 ± 28.81 to 149.13 ± 73.29, and Maniwhere declines from 314.38 ± 8.66 to 182.29 ± 64.37. The performance gap becomes more noticeable in `medium` and `hard` settings, where all baselines show substantial degradation, whereas *CGGT* maintains returns of 263.33 ± 66.61 in `easy` mode, 219.99 ± 94.89 in `medium` mode, and 202.85 ± 99.32 in `hard` mode, showing its strong generalization capabilities.

TABLE III: Concept-Pruning Results: Average episode returns (μ, σ) in concept settings across evaluated `robosuite` tasks.

Setting \ Mode	`train`	`easy`	`medium`	`hard`
$c_-^{\text{vis}}, c^{\text{spa}}$	269.08 ± 35.14	177.06 ± 64.18	163.86 ± 75.23	147.47 ± 79.12
$c_+^{\text{vis}}, c^{\text{spa}}$	309.51 ± 17.63	161.90 ± 70.69	130.63 ± 77.67	119.61 ± 66.95
$c_*^{\text{vis}}, c_-^{\text{spa}}$	283.58 ± 28.16	199.54 ± 48.42	40.89 ± 58.97	30.61 ± 68.44
$c_*^{\text{vis}}, c_+^{\text{spa}}$	312.51 ± 10.63	188.06 ± 64.15	168.50 ± 80.68	144.53 ± 68.97
Ours $(c_*^{\text{vis}}, c_*^{\text{spa}})$	310.25 ± 11.97	**263.33 ± 66.61**	**219.99 ± 94.89**	**202.85 ± 99.31**

C. Concept Pruning

We prune the task-specific visual and spatial concepts (c^{vis} and c^{spa}) to identify the optimal set and examine the effectiveness of these concepts in terms of generalization. We found that for each task, the gripper end effector, along with the gripper finger, is crucial, as are the important objects related to the task, such as the cube, peg, or pot. Starting from the optimal sets $\left(c_*^{\text{vis}}, c_*^{\text{spa}}\right)$, we add more features (e.g. robot base) to form the redundant sets $\left(c_+^{\text{vis}}, c_+^{\text{spa}}\right)$ or remove some



(a) Lift the cube successfully

(b) Grasp at the wrong positions (high-above or low-below the actual cubes)

Fig. 4: Interpretability of Downstream Manipulation Tasks: Three scenarios, including both **(a)** success and **(b)** failure cases, is inspected with spatial concepts predicted by *CGGT*.

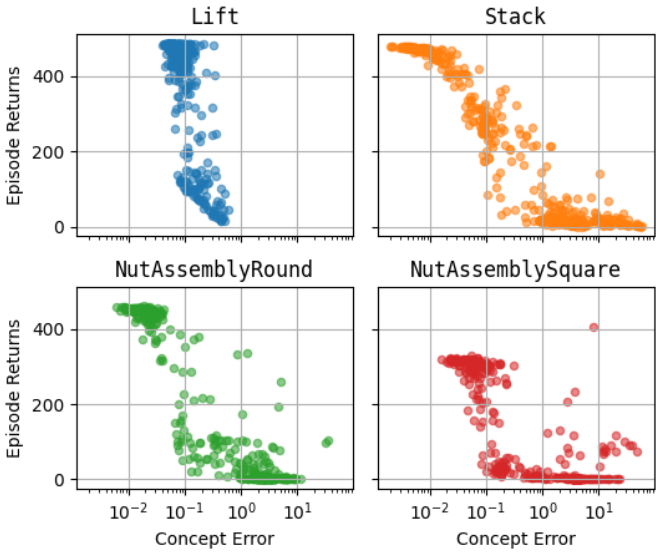


Fig. 5: Episode Returns *vs.* Concept Errors: The correlation between episode returns and spatial prediction errors in `Lift`, `Stack`, `NutAssemblyRound`, and `NutAssemblySquare` tasks while being evaluated in all scenarios.

of the features (e.g. gripper finger) to create the inadequate sets $\left(c_{-}^{\text{vis}}, c_{-}^{\text{spa}}\right)$. As shown in Tab. III, both redundant and inadequate concept sets degrade generalization. For example, using c_{-}^{vis} with optimal spatial concepts reduces `hard` mode performance from 202.85 ± 99.31 to 147.47 ± 79.12, while c_{+}^{vis} decreases it further to 119.61 ± 66.95. Similarly, removing key spatial concepts (c_{-}^{spa}) dramatically drops the `medium` return to 40.89 ± 58.97. These results confirm that the agent's generalization critically depends on focusing on relevant objects and extracting only the VIPR features. Notably, inadequate sets also reduce training performance due to the lack of essential information for task completion.

D. Interpretability

As aforementioned, concept predictions provide insights into agent behavior to explain task failures. For instance, in the cube-grasping task (Fig. 4), spatial concept predictions reveal why certain attempts fail. In successful cases, such as Fig. 4a, the gripper is correctly positioned, and all object locations are accurately predicted. In contrast, failures occur when the gripper is misaligned or spatial predictions are inaccurate (Fig. 4b and Fig. 4c), preventing successful lifts. We further quantify this relationship by correlating spatial prediction errors with obtained rewards across tasks. As shown in Fig. 5, higher prediction errors consistently correspond to lower rewards, with an average correlation of -0.613 ± 0.072 across eleven tasks, indicating a strong negative correlation that highlights the critical role of spatial alignment in improving OOD performance.

E. Ablation Studies

Next, we perform an ablation study to evaluate the importance of each component in *CGGT*. Specifically, we iteratively remove spatial alignment, visual alignment, contrastive learning, and masking from the *CGGT* framework. As reported in Tab. IV, the absence of either visual or spatial alignment results in a substantial drop in generalization performance, confirming their critical role in our framework. For instance, removing visual alignment reduces the average return in `medium` mode from 219.99 ± 94.89 to 125.86 ± 55.23, while removing spatial alignment decreases it further to 157.27 ± 61.93. In addition, contrastive learning contributes to more compact latent representations: without it, the return in `hard` mode drops from 202.85 ± 99.31 to only 42.48 ± 23.94. Notably, removing the mask results in the most degradation, where the `medium` performance collapses to 10.25 ± 2.09 and `hard` to 11.39 ± 3.35. Nevertheless, applying CBMs strictly with spatial alignment in the feature space ($\mathcal{F} \leftarrow \mathcal{C}_{\text{spatial}}$) reduces performance in both training and OOD evaluations, due to the discrepancy between the two spaces mentioned in Sec. IV-C.

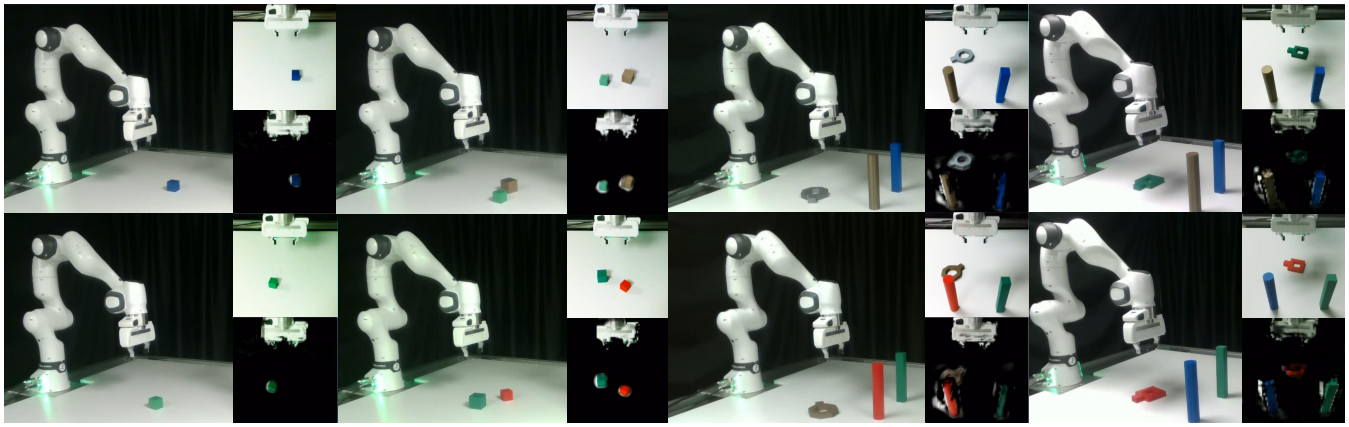
TABLE IV: Ablation Studies: Overall performance across eleven `robosuite` tasks. Episode returns (μ, σ) are reported for each evaluation mode among different ablations.

Ablation \ Mode	`train`	`easy`	`medium`	`hard`
Align spatial concept with features	159.12 ± 15.02	84.81 ± 32.14	48.61 ± 16.76	21.40 ± 11.94
without masking	272.81 ± 44.87	134.37 ± 57.99	10.25 ± 2.09	11.39 ± 3.35
without contrastive learning	292.84 ± 57.50	166.04 ± 88.96	56.85 ± 39.44	42.48 ± 23.94
without visual concept alignment	228.43 ± 21.77	174.43 ± 21.77	125.86 ± 55.23	111.64 ± 68.97
without spatial concept alignment	289.43 ± 55.18	206.93 ± 47.70	157.27 ± 61.93	139.86 ± 70.63
Full of *CGGT* (Ours)	**310.25 ± 11.97**	**263.33 ± 66.61**	**219.99 ± 94.89**	**202.85 ± 99.31**

VI. Real-Robot Experiments

A. Experiment Set-Up

We deploy the trained *CGGT* model with simulation scenarios across four different tasks: `Lift`, `Stack`,



(a) `Lift` **(b)** `Stack` **(c)** `NutAssemblyRound` **(d)** `NutAssemblySquare`

Fig. 6: Real-Robot Demonstrations: Experiment set-ups of different tasks in real-world settings, including **(a)** `Lift` tasks, **(b)** `Stack` tasks, **(c)** `NutAssemblyRound` tasks, and **(d)** `NutAssemblySquare` tasks. In each task, we also test the model's robustness and generalization with objects of different colors and layouts. The RGB images and corresponding masked observations are shown, respectively.

`NutAssemblyRound`, and `NutAssemblySquare`. The input of RGB-D observations from the Intel RealSense D435i RGB-D camera is pre-processed and streamed to a Robot Operating System (ROS2) node with the same settings as we used in simulation. To mitigate the depth noises, we apply the spatial, temporal, and hole-filling filters provided by `realsense-ros`. A ROS2 wrapper is developed for the RL model to publish action commands, while the `franky` library provides a control interface that subscribes to these actions and executes them asynchronously. For each task, objects of different colors are 3D-printed to construct five scenarios, as shown in Fig. 6, and each algorithm is tested five times per scenario.

TABLE V: Real-Robot Evaluations: Success rates (%) across four evaluated tasks on the Franka Emika robot with real-world objects.

Model \ Task	`Lift`	`Stack`	`NutRound`	`NutSquare`
ManiWhere [10]	28.00%	8.00%	0.00%	0.00%
CGGT	**48.00%**	**20.00%**	**12.00%**	**16.00%**

B. Evaluations on Real-Robot Experiments

As shown in Fig. 6, the masked views at the lower-right corners of each tested task well-capture key visual concepts, such as the gripper, cubes, pegs, and nuts. Tab. V reports the performance of the tested algorithms in the sim-to-real manner, and *CGGT* outperforms ManiWhere across all four tasks. However, the overall performance remains considerably below that achieved in simulation-based experiments since the model lacks knowledge of realistic robot–object interactions in the simulated environments. Moreover, we find that the simulation does not account for the built-in collision detection and safety constraints presented in the real Franka Emika manipulator.

C. Demonstration

Demonstration videos of evaluated tasks in both simulated scenarios and real-world settings on the Franka Emika robot are available in our supplementary materials.

VII. Conclusions

In this paper, we present *CGGT*, a generalizable and interpretable framework for visuomotor RL through grounded concept training. Specifically, *CGGT* leverages predefined visual and spatial concepts to guide the agent's attention toward task-critical objects and extract visual perturbation-invariant representations, thereby enhancing generalization in OOD scenarios and improving task completion. Our experiments demonstrate that *CGGT* improves generalization, interpretability, and real-world transfer across a range of tasks in simulation environments with different levels of generalization. Concept pruning and ablation studies further highlight the critical role of visual and spatial alignments in achieving these benefits. Through this, we deploy our proposed method on the Franka Emika manipulator with real-world objects, showcasing its generalization to real-world scenarios. These empirical results show that concepts in both visual and spatial domains highly benefit the performance of robot systems in both simulated and real-world settings.

References

[1] R. Jangir, N. Hansen, S. Ghosal, M. Jain, and X. Wang, "Look closer: Bridging egocentric and third-person views with transformers for robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3046–3053, 2022.

[2] T. Pham and A. Cangelosi, "Pay attention to what and where? interpretable feature extractor in vision-based deep reinforcement learning," *arXiv preprint arXiv:2504.10071*, 2025.

[3] T. Haarnoja, B. Moran, G. Lever, S. H. Huang, D. Tirumala, J. Humplik, M. Wulfmeier, S. Tunyasuvunakool, N. Y. Siegel, R. Hafner *et al.*, "Learning agile soccer skills for a bipedal robot with deep reinforcement learning," *Science Robotics*, vol. 9, no. 89, p. eadi8022, 2024.

[4] N. Hansen, H. Su, and X. Wang, "Stabilizing deep q-learning with convnets and vision transformers under data augmentation," *Advances in neural information processing systems*, vol. 34, pp. 3680–3693, 2021.

[5] N. Hansen and X. Wang, "Generalization in reinforcement learning by soft data augmentation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 611–13 617.

[6] Y. Huang, P. Peng, Y. Zhao, G. Chen, and Y. Tian, "Spectrum random masking for generalization in image-based reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 20 393–20 406, 2022.

[7] M. Dunion and S. V. Albrecht, "Multi-view disentanglement for reinforcement learning with multiple cameras," *arXiv preprint arXiv:2404.14064*, 2024.

[8] D. Bertoin, A. Zouitine, M. Zouitine, and E. Rachelson, "Look where you look! saliency-guided q-networks for generalization in visual reinforcement learning," *Advances in neural information processing systems*, vol. 35, pp. 30 693–30 706, 2022.

[9] 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.

[10] Z. Yuan, T. Wei, S. Cheng, G. Zhang, Y. Chen, and H. Xu, "Learning to manipulate anywhere: A visual generalizable framework for reinforcement learning," *arXiv preprint arXiv:2407.15815*, 2024.

[11] Z. Yuan, G. Ma, Y. Mu, B. Xia, B. Yuan, X. Wang, P. Luo, and H. Xu, "Don't touch what matters: Task-aware lipschitz data augmentation for visual reinforcement learning," *arXiv preprint arXiv:2202.09982*, 2022.

[12] R. Ngo, L. Chan, and S. Mindermann, "The alignment problem from a deep learning perspective," *arXiv preprint arXiv:2209.00626*, 2022.

[13] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, 2020.

[14] G. A. Vouros, "Explainable deep reinforcement learning: state of the art and challenges," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–39, 2022.

[15] T. Mesnard, T. Weber, F. Viola, S. Thakoor, A. Saade, A. Harutyunyan, W. Dabney, T. Stepleton, N. Heess, A. Guez *et al.*, "Counterfactual credit assignment in model-free reinforcement learning," *arXiv preprint arXiv:2011.09464*, 2020.

[16] Q. Delfosse, S. Sztwiertnia, M. Rothermel, W. Stammer, and K. Kersting, "Interpretable concept bottlenecks to align reinforcement learning agents," *Advances in Neural Information Processing Systems*, vol. 37, pp. 66 826–66 855, 2024.

[17] J. Sha, H. Shindo, Q. Delfosse, K. Kersting, and D. S. Dhami, "Expil: Explanatory predicate invention for learning in games," *arXiv preprint arXiv:2406.06107*, 2024.

[18] B. Grooten, T. Tomilin, G. Vasan, M. E. Taylor, A. R. Mahmood, M. Fang, M. Pechenizkiy, and D. C. Mocanu, "MaDi: Learning to Mask Distractions for Generalization in Visual Deep Reinforcement Learning," *The 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2024, uRL: https://arxiv.org/abs/2312.15339.

[19] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang, "Concept bottleneck models," in *International conference on machine learning*. PMLR, 2020, pp. 5338–5348.

[20] W. Stammer, M. Memmel, P. Schramowski, and K. Kersting, "Interactive disentanglement: Learning concepts by interacting with their prototype representations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 317–10 328.

[21] R. Zabounidis, J. Campbell, S. Stepputtis, D. Hughes, and K. P. Sycara, "Concept learning for interpretable multi-agent reinforcement learning," in *Conference on Robot Learning*. PMLR, 2023, pp. 1828–1837.

[22] N. Grupen, N. Jaques, B. Kim, and S. Omidshafiei, "Concept-based understanding of emergent multi-agent behavior," in *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022.

[23] H. Kohler, Q. Delfosse, R. Akrour, K. Kersting, and P. Preux, "Interpretable and editable programmatic tree policies for reinforcement learning," *arXiv preprint arXiv:2405.14956*, 2024.

[24] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, Y. Zhu, and K. Lin, "robosuite: A modular simulation framework and benchmark for robot learning," in *arXiv preprint arXiv:2009.12293*, 2020.

[25] Z. Yuan, S. Yang, P. Hua, C. Chang, K. Hu, and H. Xu, "Rl-vigen: A reinforcement learning benchmark for visual generalization," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[26] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. Pmlr, 2018, pp. 1861–1870.

[27] C. Tang, B. Abbatematteo, J. Hu, R. Chandra, R. Martín-Martín, and P. Stone, "Deep reinforcement learning for robotics: A survey of real-world successes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 27, 2025, pp. 28 694–28 698.

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

[29] J. Lyu, L. Wan, X. Li, and Z. Lu, "Understanding what affects the generalization gap in visual reinforcement learning: Theory and empirical evidence," *Journal of Artificial Intelligence Research*, vol. 81, pp. 1–42, 2024.

[30] E. Teoh, S. Patidar, X. Ma, and S. James, "Green screen augmentation enables scene generalisation in robotic manipulation," *arXiv preprint arXiv:2407.07868*, 2024.

[31] L. Li, J. Lyu, G. Ma, Z. Wang, Z. Yang, X. Li, and Z. Li, "Normalization enhances generalization in visual reinforcement learning," *arXiv preprint arXiv:2306.00656*, 2023.

[32] W. Pumacay, I. Singh, J. Duan, R. Krishna, J. Thomason, and D. Fox, "The colosseum: A benchmark for evaluating generalization for robotic manipulation," *arXiv preprint arXiv:2402.08191*, 2024.

[33] K. Nguyen, A. T. Le, J. Peters, and M. N. Vu, "Doublyaware: Dual planning and policy awareness for temporal difference learning in humanoid locomotion," *arXiv preprint arXiv:2506.12095*, 2025.

[34] Y. Hu, R. Wang, L. E. Li, and Y. Gao, "For pre-trained vision models in motor control, not all policy learning methods are created equal," in *International Conference on Machine Learning*. PMLR, 2023, pp. 13 628–13 651.

[35] A. Verma, V. Murali, R. Singh, P. Kohli, and S. Chaudhuri, "Programmatically interpretable reinforcement learning," in *International conference on machine learning*. PMLR, 2018, pp. 5045–5054.

[36] B. Fuhrer, C. Tessler, and G. Dalal, "Gradient boosting reinforcement learning," *arXiv preprint arXiv:2407.08250*, 2024.

[37] S. Marton, T. Grams, F. Vogt, S. Lüdtke, C. Bartelt, and H. Stuckenschmidt, "Sympol: Symbolic tree-based on-policy reinforcement learning," *arXiv e-prints*, pp. arXiv–2408, 2024.

[38] Q. Delfosse, H. Shindo, D. Dhami, and K. Kersting, "Interpretable and explainable logical policies via neurally guided symbolic abstraction," *Advances in Neural Information Processing Systems*, vol. 36, pp. 50 838–50 858, 2023.

[39] D. Trivedi, J. Zhang, S.-H. Sun, and J. J. Lim, "Learning to synthesize programs as interpretable and generalizable policies," *Advances in neural information processing systems*, vol. 34, pp. 25 146–25 163, 2021.

[40] L. Luo, G. Zhang, H. Xu, Y. Yang, C. Fang, and Q. Li, "End-to-end neuro-symbolic reinforcement learning with textual explanations," *arXiv preprint arXiv:2403.12451*, 2024.

[41] T. Kaufmann, J. Blüml, A. Wüst, Q. Delfosse, K. Kersting, and E. Hüllermeier, "Ocalm: Object-centric assessment with language models," *arXiv preprint arXiv:2406.16748*, 2024.

[42] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, pp. 9–44, 1988.

[43] T. Pham, X. Chi, K. Nguyen, M. Huber, and A. Cangelosi, "Deguv: Depth-guided visual reinforcement learning for generalization and interpretability in manipulation," *arXiv preprint arXiv:2509.04970*, 2025.

[44] R. Agarwal, M. C. Machado, P. S. Castro, and M. G. Bellemare, "Contrastive behavioral similarity embeddings for generalization in reinforcement learning," *arXiv preprint arXiv:2101.05265*, 2021.

[45] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[46] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[47] I. Kostrikov, D. Yarats, and R. Fergus, "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels," *arXiv preprint arXiv:2004.13649*, 2020.

[48] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2918–2928.