Improving Compositional Generalization in Cross-Embodiment Learning via Mixture of Disentangled Prototypes

Ren Wang*
DCST, Tsinghua University
Beijing, China
xxlifelover@gmail.com

Xin Wang†
DCST, BNRist, Tsinghua University
Beijing, China
xin_wang@tsinghua.edu.cn

Tongtong Feng
DCST, Tsinghua University
Beijing, China
fengtongtong@tsinghua.edu.cn

Xinyue Gong
DCST, Tsinghua University
Beijing, China
xy-gong21@mails.tsinghua.edu.cn

Guangyao Li
DCST, Tsinghua University
Beijing, China
guangyaoli@tsinghua.edu.cn

Yu-Wei Zhan
DCST, Tsinghua University
Beijing, China
zhanyuweilif@gmail.com

Qing Li‡
DEE, Tsinghua University
Beijing, China
soleilor@mail.tsinghua.edu.cn

Wenwu Zhu†
DCST, BNRist, Tsinghua University
Beijing, China
wwzhu@tsinghua.edu.cn

Abstract

Cross-Embodiment Learning (CEL) aims to train a generalist policy model by integrating large-scale compositional interactions of heterogeneous agents and environments. However, the inherent conflict between the unbounded space of agent-environment combinations and a single unified policy model hinders generalization to unseen combinations. To address this challenge, we propose a novel Mixture of Disentangled Prototypes (MoDP) method to improve the compositional generalization in CEL. The key idea is to introduce a finite prototype space that bridges the gap between unbounded agent-environment combinations and a single policy model. Specifically, we design a dual-headed autoencoder and a compositional reconstruction loss to disentangle agent and environment features from interaction data, and map them into respective prototype spaces. We then introduce a connection-sensitivity-based pruning method to extract sub-networks from the pre-trained policy model, forming policy prototypes associated with specific agent-environment prototype pairs. Finally, a parameter-free routing mechanism adaptively integrates relevant policy prototypes for each input composition. Experiments in both standard and compositional settings demonstrate the effectiveness of our MoDP in enhancing the generalization capability of pre-trained policies.

CCS Concepts

• **Computing methodologies** → *Learning from demonstrations.*

*DCST is the abbreviation for Department of Computer Science and Technology.
†Corresponding author. BNRist is the abbreviation for Beijing National Research Center for Information Science and Technology.
‡DEE is the abbreviation for Department of Electrical Engineering.

MM '25, Dublin, Ireland

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3754499

Keywords

Generalist Embodiments, Prototype Learning, Feature Disentanglement, Mixture-of-Experts, Dynamic Environments

ACM Reference Format:
Ren Wang, Xin Wang, Tongtong Feng, Xinyue Gong, Guangyao Li, Yu-Wei Zhan, Qing Li, and Wenwu Zhu. 2025. Improving Compositional Generalization in Cross-Embodiment Learning via Mixture of Disentangled Prototypes. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746027.3754499

1 Introduction

Cross-Embodied Learning [10, 50] (CEL) seeks to train a unified policy applicable to heterogeneous robot entities and capable of interacting with various environments, leading to better robustness and generalization. The resulting well-trained model, known as the generalist embodied agent [37], is expected to drive advancements in fields such as home service, autonomous driving, and multi-agent collaboration [11, 59]. As collecting embodied data from a single robot embodiment remains costly and inefficient, CEL has become an increasingly prominent research focus.

Rapid progress in CEL is primarily driven by advances in both large-scale embodied datasets and unified model architectures. On the data side, recent efforts have significantly expanded the scale and diversity of interaction data [13]. For example, RoboNet [9] and HM3D [51] provide large-scale data for manipulation and navigation tasks, respectively, covering diverse embodiments and real-world environments. Open X-Embodiment [33] further integrates data from over 20 robot platforms and 500 tasks, serving as a comprehensive benchmark for CEL. On the model side, Transformer-based architectures have emerged as a promising foundation for generalist policy learning. RT-X [3, 61] demonstrates that scalable vision-language-action models trained on diverse embodied data can generalize across both tasks and embodiments. CrossFormer [10] and HPT [39] further explore modular designs that represent heterogeneous inputs as unified token sequences and

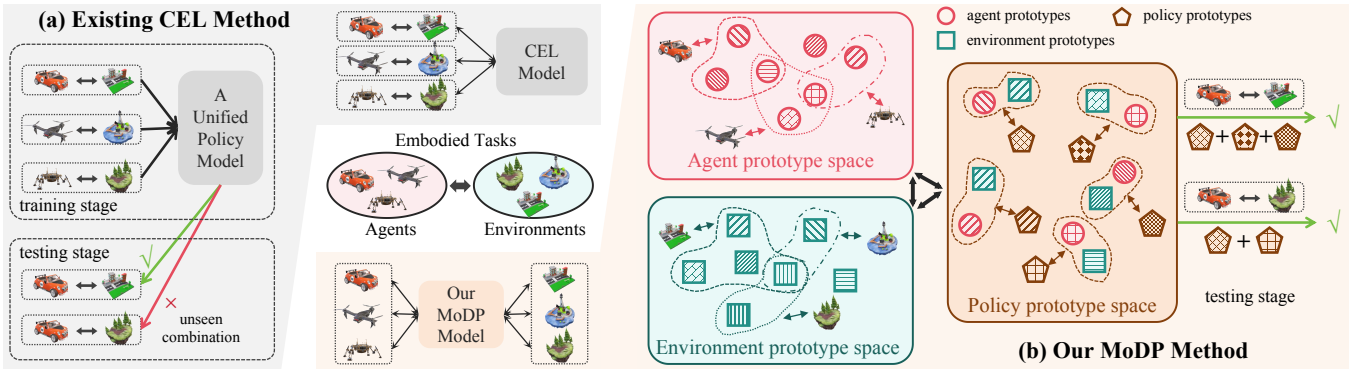


Figure 1: Illustration of the key idea of our MoDP method. (a) Existing cross-embodiment learning (CEL) approaches train a single generalist policy model across various combinations of agents and environments. The inherent conflict between the unbounded combination space and a single policy model poses a challenge to achieving compositional generalization. (b) Our MoDP method introduces a finite prototype space to disentangle agents, environments, and policies. Therefore, unseen combinations can be represented using known prototypes and routed to an appropriate mixture of policy prototypes.

share a common trunk network, improving adaptability to diverse embodiments.

The paradigm of training general-purpose models on large-scale aggregated data has been successfully validated in domains such as natural language processing (NLP) [17]. However, it poses new challenges for compositional generalization in the embodied domain. Fundamentally, NLP tasks involve discrete, symbolic inputs (e.g., words or tokens) that can be flexibly recombined under well-defined compositional rules, with training data often densely covering a wide range of such combinations [21]. In contrast, embodied tasks are grounded in continuous, high-dimensional interactions, where the agent's proprioception, environmental observation, and policy action are tightly coupled and interdependent. This intrinsic entanglement limits the transferability of learned skills to unseen combinations of agents and environments. As illustrated in Fig. 1 (a), the compositional generalization capability of current CEL approaches is limited by two key factors:

- **The combinatorial explosion of agents and environments.** The space of agent and environment is virtually unbounded, making exhaustive data coverage impractical.
- **The limited capacity of a single unified model.** Diversity in agent-environment combinations can introduce policy conflicts, limiting the positive transferability of a single model across compositions [10].

Compositional generalization is not a new topic and has been widely studied in language, vision, and reinforcement learning. Existing methods include symbolic rule-based approaches [22], modular architectures [60], and factorized policy representations [18]. For example, NMN [1] assumes visual reasoning can be decomposed into discrete submodules; Slot Attention [29] enforces object-centric representations via spatial disentanglement; and C-SWM [20] factorize dynamics and content for better policy transfer. In contrast, embodied tasks involve a tight coupling between agent, environment, and policy. Effectively disentangling these components and balancing the trade-off between the unbounded combination space and a unified model remains a valuable yet underexplored problem.

To address this problem, we innovatively propose a Mixture of Disentangled Prototypes (MoDP) method. As shown in Fig. 1 (b), the key idea is to introduce a finite prototype space that bridges the gap between unbounded agent-environment combinations and a unified policy. In prototype spaces, the mapping from unseen agent-environment combinations to their corresponding policies can be modeled as the mapping between known prototypes, enabling effective compositional generalization. Specifically, a dual-head autoencoder is designed to disentangle interaction data into agent and environment prototype spaces. Next, we identify policy prototypes by pruning sub-networks from a pre-trained generalist policy model by measuring the connection sensitivity of parameters. Finally, we introduce a parameter-free routing mechanism that adaptively assigns appropriate policies to each agent-environment combination. Experimental results under both standard and compositional settings validate the effectiveness of our method. Main contributions are summarized as follows:

- We propose a novel mixture of disentangled prototypes method to improve the compositional generalization of cross-embodiment learning.
- We design a compositional reconstruction loss and parameter-saliency-based pruning method to disentangle agent, environment, and policy prototypes.
- We evaluate our method under both generalist and constructed compositional settings, demonstrating its effectiveness in improving cross-embodiment generalization.

2 Related Works

2.1 Cross-Embodied Learning

Cross-Embodiment Learning focuses on enabling robots to generalize learned policies across different embodiments. Early work explored techniques such as conditioning on explicit representations of the embodiment [8], domain randomization and adaptation [12, 34], and modular policies [19, 52], which were applied to simpler scenarios like single-task manipulations. Subsequent research has focused on alignment methods to develop universal strategies

that can transfer across tasks, such as manipulation and navigation. Recent work includes the design of more powerful generalizable policy models for specific domains, such as RoboCat [2], GVA [15], and MagenticOne [14], as well as models that generalize across various types of scenarios, such as Octo [16], CrossFormer [10], HPT [39], etc. These methods focus on training a general policy model for heterogeneous agents but rarely address the generalization capability in unseen combinations of agents and environments. Our work bridges this gap and holds the potential to enhance the compositional generalization of existing CEL methods.

2.2 Compositional Generalization

Compositional generalization [28] (CG) has gained significant attention across various domains. In language and semantics, CG involves the ability of models to generalize to unseen combinations of words or sentences. For instance, Qiu et al. [36] explore CG in semantic parsing, highlighting challenges with larger models that still struggle with novel combinations. In reinforcement learning (RL), CG focuses on generalizing across combinations of tasks or entities. Mambelli et al. [32] address multi-object manipulation, while Zhao et al. [58] propose the HOWM model for object-oriented RL, improving generalization to dynamic settings. Instead, our work focuses on agent-environment compositional generalization in CEL, where models must handle complex, dynamic interactions between agents and their environments.

2.3 Disentangled Representation Learning

Disentangled Representation Learning (DRL) aims to separate underlying factors in data to enhance interpretability and generalization [43, 44, 46, 47]. In graph learning, advances include separating invariant and variant mechanisms to improve the generalization under distribution shifts [24–27, 56], leveraging neural architecture search to disentangle functional modules for continual and transferable optimization [55, 57], and employing large language models to extract factorized semantics in text-attributed graphs [35]. In recommendation systems, DRL has been applied to disentangle multi-intent user factors [30, 31], multimodal user–item factors [45], feedback signals [4, 42, 48], and sequential patterns [54, 55]. In generation and grounding, DRL is commonly used to disentangle identity from background and content from location, improving controllability and personalization [5–7, 49]. In this work, we explore a DRL strategy tailored to CEL, disentangling agent and environment prototypes from embodied interaction data at the feature level and policy prototypes at the architectural level, addressing the unique challenge of achieving compositional generalization.

3 Problem Formulation

In embodied AI, an agent α interacts with an environment ε through a closed loop of perception and action. Formally, Each interaction episode yields a trajectory $\tau = (s_1, a_1, s_2, a_2, \ldots, s_t, a_t, \ldots)$, where s_t and a_t denote the state observed at timestep t and the action executed by the agent, respectively. Due to the heterogeneity of agents and the complexity of environments, the state s_t can be a multimodal observation comprising proprioception, RGB images, 3D point clouds, and scalar feedback. These observations implicitly contain two types of latent information: agent-related information (e.g., embodiment, kinematics, or internal state) and environment-related information (e.g., scene layout, objects, dynamics).

We denote the agent and environment spaces as $\mathcal{A}$ and $\mathcal{E}$, respectively. An embodied task instance involving the interaction between agent $\alpha \in \mathcal{A}$ and environment $\varepsilon \in \mathcal{E}$ can be represented by their combination, i.e., $t_{\langle\alpha,\varepsilon\rangle} \in \mathcal{T}$, where $\mathcal{T} \subseteq \mathcal{A} \times \mathcal{E}$ denotes the overall embodied task space. By collecting demonstrations $\mathcal{T}_{\text{train}} \subset \mathcal{T}$ from various agents interacting with their environments, CEL aims to train a generalist policy model $\pi : s_{\langle\alpha,\varepsilon\rangle} \mapsto a_\alpha$ that maps observations to appropriate actions across diverse task instances. However, since the data reflect only a finite subset of possible agent-environment combinations, existing methods inevitably face the following combinational generalization challenge:

Definition 1 (Combinational Generalization): Given a training set $\mathcal{T}_{\text{train}} \subset \mathcal{A} \times \mathcal{E}$, where each element (α, ε) denotes a specific agent-environment pairing, the goal is to generalize to a set of unseen combinations in a testing set $\mathcal{T}_{\text{test}} \subset (\mathcal{A} \times \mathcal{E}) \setminus \mathcal{T}_{\text{train}}$:

$$\forall(\alpha', \varepsilon') \in \mathcal{T}_{\text{test}}, \quad \alpha' \in \mathcal{A}, \quad \varepsilon' \in \mathcal{E}, \quad (\alpha', \varepsilon') \notin \mathcal{T}_{\text{train}}. \quad (1)$$

That is, while the agent and environment have been seen individually during training, their specific combination has not. A model capable of compositional generalization must recombine learned behaviors across these independently observed components to handle novel pairings at test time.

As discussed in the Introduction, the challenge of compositional generalization in CEL exists at both the data and the model levels. From a data perspective, exhaustively collecting data from all possible agent-environment combinations is prohibitively expensive and practically infeasible. Moreover, heterogeneous agents are typically tied to specific environments, and indiscriminately integrating these interaction data may lead to overfitting to the specific agent-environment combinations rather than the policy itself, preventing generalization to new combinations.

From the model perspective, the challenge arises due to entangled representations of agent-specific and environment-specific factors. Existing CEL methods optimize a policy $\pi(a_t \mid s_t)$, where the state s_t contains both the agent's internal state and its external perception. However, the policy processes this composite input in a monolithic fashion. In the absence of mechanisms to explicitly disentangle the influences of α and ε on behavior, the learned policy becomes tightly coupled to the joint training distribution, and struggles to generalize when the composition shifts at test time.

4 Method

This paper proposes a novel Mixture of Disentangled Prototypes (MoDP) method to address the challenge of compositional generalization in CEL. The key idea is to represent the unbounded agent-environment combinations and the single policy as mixtures over a finite set of disentangled prototypes. As illustrated in Fig. 2, our MoDP consists of three main modules: the Agent and Environment Prototypes Disentanglement module, the Policy Prototype Disentanglement module, and the Disentangled Prototype Routing module. The first two modules are responsible for disentangling the three fundamental factors (agent, environment, and policy) into their respective prototype spaces. The final module then integrates these disentangled prototypes to construct generalized policies

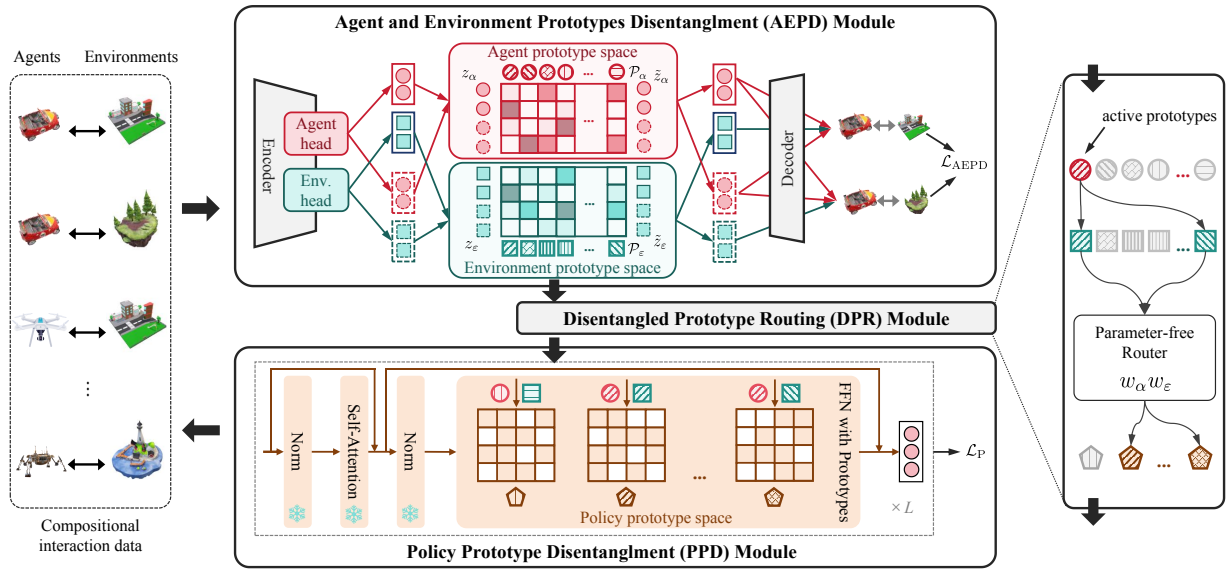


Figure 2: The framework of the proposed MoDP, consists of three modules: (1) AEPD module extracts disentangled agent-specific and environment-specific features from compositional interaction data and maps them into separate prototype spaces; (2) PPD module extracts policy prototypes from a pre-trained model via connection-sensitive pruning; and (3) DPR module adaptively routes prototype-aligned sub-policies for given agent-environment compositions.

capable of handling novel agent-environment combinations. The following subsections elaborate on each module in our framework.

4.1 Agent and Environment Prototypes Disentanglement (AEPD)

The purpose of the AEPD module is to learn a disentangled representation of the agent and environment by projecting their high-dimensional interaction embeddings [1] into two independent prototype spaces. Specifically, given an interaction embedding $x_{\langle\alpha,\varepsilon\rangle} \in \mathbb{R}^d$ extracted from a trajectory τ in task $t_{\langle\alpha,\varepsilon\rangle}$, we employ a dual-headed encoder $f(\cdot) = [f_\alpha(\cdot), f_\varepsilon(\cdot)]$ to decompose x into agent-specific and environment-specific representations $z_\alpha = f_\alpha(x) \in \mathbb{R}^{d_\alpha}$ and $z_\varepsilon = f_\varepsilon(x) \in \mathbb{R}^{d_\varepsilon}$, respectively.

Further, these latent features are softly projected onto two learnable prototype spaces. Let $\mathcal{P}_\alpha \in \mathbb{R}^{d_\alpha \times K_\alpha}$ denote the agent prototype matrix, containing K_α basis vectors as columns, and $\mathcal{P}_\varepsilon \in \mathbb{R}^{d_\varepsilon \times K_\varepsilon}$ denote the environment prototype matrix. For each input, we compute prototype-based representations through linear combination:

$$\tilde{z}_\alpha = \mathcal{P}_\alpha w_\alpha, \quad \tilde{z}_\varepsilon = \mathcal{P}_\varepsilon w_\varepsilon, \tag{2}$$

where w_α and w_ε are the attention weights, computed via a softmax over similarity scores between the input features and the corresponding prototypes:

$$w_\alpha = \text{softmax}(\mathcal{P}_\alpha^\top z_\alpha), \quad w_\varepsilon = \text{softmax}(\mathcal{P}_\varepsilon^\top z_\varepsilon). \tag{3}$$

This formulation allows each representation to be expressed as a convex combination of a small set of shared prototype vectors. Intuitively, the weights w_α and w_ε reflect how strongly the current instance activates each agent or environment prototype, thereby enabling the model to capture reusable, structured factors that generalize across combinations.

Now, we have decomposed the interaction state into two sets of features and represented them as combinations of prototypes in their respective prototype spaces. However, we still cannot guarantee the quality of these features, nor ensure that they have independently learned the agent-specific and environment-specific information. To address this issue, inspired by the idea of "proxy" [40], we design three constraints to ensure the **Representativeness**, **Uniqueness**, and **Simplicity** of the disentangled features.

- **Representativeness** requires that the disentangled agent and environment features effectively capture the underlying characteristics of the agent and the environment, respectively. To enforce this, we design a hybrid reconstruction loss:

$$\begin{aligned}\mathcal{L}_{\text{CR}} = \sum_{\substack{\varepsilon \neq \varepsilon' \\ (\alpha,\varepsilon),(\alpha,\varepsilon') \in \mathcal{T}_{\text{train}}}} & \left\| g(\tilde{z}_\alpha \oplus \tilde{z}_\varepsilon) - x_{\langle\alpha,\varepsilon\rangle} \right\|^2 \\ & + \lambda_{\text{CR}} \left\| g(\tilde{z}_\alpha \oplus \tilde{z}_{\varepsilon'}) - x_{\langle\alpha,\varepsilon'\rangle} \right\|^2,\end{aligned} \tag{4}$$

where $g(\cdot)$ is the decoder network, and $\oplus$ denotes feature concatenation. The first term is a reconstruction loss over all observed interactions, encouraging the fused agent and environment prototype features to accurately reconstruct the corresponding interaction embeddings. The second term is the compositional reconstruction loss, which enforces cross-instance consistency. Given an agent α observed in environment ε, and an environment ε' associated with another agent α', the agent prototype from $\langle\alpha,\varepsilon\rangle$ is recombined with the environment prototype from $\langle\alpha',\varepsilon'\rangle$ to reconstruct the embedding of $\langle\alpha,\varepsilon'\rangle$. Thus, the well-designed loss $\mathcal{L}_{\text{CR}}$ can

[1] The interaction embeddings x is derived by mapping heterogeneous state data s into a unified dimensional space, which serves as the pre-processing in CEL and is not the focus of our study.

encourage the learned prototypes to be both disentangled and composable across agent-environment combinations.

- **Uniqueness** requires each prototype to capture distinct, non-overlapping information about either the agent or environment. To this end, we impose an orthogonality constraint on the prototype-based agent and environment features:

$$\mathcal{L}_{\mathrm{o}} = \sum_{(\alpha,\varepsilon)\in\mathcal{T}_{\mathrm{train}}} \left\| \frac{\tilde{z}_\alpha^\top \tilde{z}_\varepsilon}{\|\tilde{z}_\alpha\| \cdot \|\tilde{z}_\varepsilon\|} \right\|^2. \tag{5}$$

  This constraint minimizes the cosine similarity between agent and environment features, encouraging the model to encode non-redundant, factor-specific information in each branch. This, in turn, improves both interpretability and compositional generalization.
- **Simplicity** follows the principle of Occam's razor, implying that each feature should rely on only a small subset of prototypes. To enforce this, we apply an ℓ_1-based sparsity regularization on the attention weights:

$$\mathcal{L}_{\mathrm{s}} = \sum_{(\alpha,\varepsilon)\in\mathcal{T}_{\mathrm{train}}} \left(\|w_\alpha\|_1 + \|w_\varepsilon\|_1\right). \tag{6}$$

The final loss function in the AEPD module is given by:

$$\mathcal{L}_{\mathrm{AEPD}} = \mathcal{L}_{\mathrm{CR}} + \lambda_{\mathrm{o}}\mathcal{L}_{\mathrm{o}} + \lambda_{\mathrm{s}}\mathcal{L}_{\mathrm{s}}, \tag{7}$$

where λ_{o} and λ_{s} are hyper-parameters controlling the strength of regularization. This loss encourages the learned agent and environment representations to be informative, disentangled, and interpretable through a set of shared prototype vectors.

4.2 Policy Prototype Disentanglement (PPD)

The AEPD module has effectively transformed the potentially unbounded combinations of agent-environment combinations into a finite set of agent and environment prototype compositions. The next challenge is to enable the policy model to generalize across these prototype compositions. A straightforward and intuitive approach is to directly train a generalist policy model, or fine-tune a pre-trained one, using all available combinations of agent and environment prototypes. However, this strategy suffers from two fundamental limitations:

- There is no explicit supervision on which agent-environment prototype pairs should behave similarly or differently;
- Not all parameters in the policy network transfer equally well across prototype pairs. Blindly sharing or adapting them may lead to negative transfer [10].

To address these limitations, the PPD module aims to build factorized sub-policies associated with specific prototype combinations instead of a single generalist policy. The key idea is to represent each sub-policy as a mask-induced sub-network pruned from the pre-trained policy model. Conditioned on a pair of agent and environment prototypes, each sub-network is selectively activated based on the semantic structure of the input interaction.

Concretely, let π_{base} denote the pretrained generalist policy model, instantiated as a Transformer with L layers, each including a feedforward network (FFN) with weights $\{W_\ell \in \mathbb{R}^{d_\ell \times d_\ell}\}_{\ell=1}^{L}$. For each agent-environment prototype pair $(p_\alpha^{(k)}, p_\varepsilon^{(l)})$, we learn a corresponding mask $\{M_{k,l}^{(\ell)} \in [0,1]^{d_\ell \times d_\ell}\}_{\ell=1}^{L}$, and the associated subnetwork as a policy prototype $\pi_{k,l}$:

$$\pi_{k,l}(x) := \pi_{\mathrm{base}}\left(x;\ \{M_{k,l}^{(\ell)} \odot W_\ell\}_{\ell=1}^{L}\right), \tag{8}$$

where $\odot$ denotes element-wise multiplication.

Inspired by prior work [23, 41], the mask values are computed based on the connection sensitivity of each parameter with respect to the current agent-environment prototype pair. Specifically, given a prototype feature pair $\tilde{z}_\alpha$ and $\tilde{z}_\varepsilon$, and their associated prototypes $(p_\alpha^{(k)}, p_\varepsilon^{(l)})$, the sensitivity of each parameter is estimated as the absolute product of the weight and its gradient:

$$S_{k,l}^{(\ell)} = \left| W_\ell \odot \frac{\partial \mathcal{L}(\pi_{\mathrm{base}}(\tilde{z}_\alpha, \tilde{z}_\varepsilon))}{\partial W_\ell} \right|. \tag{9}$$

This connection sensitivity reflects how strongly each parameter contributes to the policy output for a given prototype pair. The binary mask is then obtained by applying a threshold γ: $M_{k,l}^{(\ell)} = \mathbb{I}(S_{k,l}^{(\ell)} > \gamma)$, where $\mathbb{I}(\cdot)$ denotes the indicator function.

This process yields a set of policy prototypes aligned with the underlying factorized representation space. Each mask effectively captures which parts of the base model are responsible for expressing the behavior associated with a given agent-environment prototype combination. Each mask defines a submodel that acts as an expert policy tailored to a specific agent-environment prototype pair. This completes our prototype-based disentanglement of agents, environments, and policies. The following section presents how these modular components are integrated via prototype-guided mixture to realize generalization to novel combinations.

4.3 Disentangled Prototype Routing (DPR)

Based on the disentangled prototypes of the agent, environment, and policy, this DPR module integrates them to produce adaptive policy across diverse agent-environment combinations.

Given an input interaction x, the soft assignment weights over agent and environment prototypes obtained from the AEPD module naturally define a weighting over the full space of prototype pairs. Each policy prototype $\pi_{k,l}(x)$, as defined in the PPD module, corresponds to a masked sub-network of the shared pre-trained policy model, specialized to the prototype pair $(p_\alpha^{(k)}, p_\varepsilon^{(l)})$. The final action prediction is computed as a weighted combination of all sub-policies:

$$\hat{a} = \sum_{k=1}^{K_\alpha} \sum_{l=1}^{K_\varepsilon} w_\alpha^{(k)} w_\varepsilon^{(l)} \cdot \pi_{k,l}(x). \tag{10}$$

This parameter-free soft routing mechanism allows the model to interpolate between prototype-aligned behaviors and synthesize coherent policies, even for novel combinations unseen during training. The predicted action $\hat{a}$ is trained via standard imitation learning, with the loss function defined as mean squared error:

$$\mathcal{L}_{\mathrm{P}} = \ell(\hat{a}, a) = \|\hat{a} - a\|_2^2, \tag{11}$$

where a is the demonstrated action.

To enable continual refinement, all types of prototypes are updated using a momentum-based scheme that aggregates information across training batches. As a concrete example, each policy

Algorithm 1 Training Procedure of MoDP method

Require: Dataset $\mathcal{T}_{\text{train}} = \{(s, a)_{\langle\alpha,\varepsilon\rangle}\}$, learning rate η, momentum λ, the number of agent and environment prototypes K_α and K_ε

Ensure: Prototypes $\mathcal{P}_\alpha$, $\mathcal{P}_\varepsilon$, and $\{M_{k,l}^{(\ell)}\}$

Initialize shared encoder f, decoder g, pretrained policy π_{base}
Initialize agent/environment prototype matrices $\mathcal{P}_\alpha$, $\mathcal{P}_\varepsilon$
Initialize mask parameters $\{M_{k,l}^{(\ell)}\}$ for each prototype pair.
for each minibatch $B \subset \mathcal{T}_{\text{train}}$ **do**
  for each sample $(s, a) \in B$ **do**
    Extract interaction embedding x
    Compute disentangled features: $z_\alpha, z_\varepsilon \leftarrow f_\alpha(x), f_\varepsilon(x)$
    Compute prototype weights: w_α, w_ε via Eq. (3)
    Compute prototype-based features: $\tilde{z}_\alpha, \tilde{z}_\varepsilon$ via Eq. (2)
    Predict reconstructed embedding: $\hat{x} \leftarrow g(\tilde{z}_\alpha \oplus \tilde{z}_\varepsilon)$
    Compute reconstruction loss $\mathcal{L}_{\text{AEDP}}$ via Eq. (7)
    Compute action prediction $\hat{a}$ via Eq. (10)
    Compute imitation loss: $\mathcal{L}_{\text{P}}$ via Eq. (11)
    Update all parameters via gradient descent
    Momentum update prototypes $\mathcal{P}_\alpha$, $\mathcal{P}_\varepsilon$, and $\{M_{k,l}^{(\ell)}\}$ via Eq. (12)
  end for
end for

prototype mask $M_{k,l}^{(\ell)}$ is updated via an exponential moving average that reflects its contribution to the current prediction. Specifically, the gradient of the policy loss with respect to each mask is scaled by the corresponding mixture weight:

$$M_{k,l}^{(\ell)} \leftarrow \lambda M_{k,l}^{(\ell)} + (1-\lambda) \cdot w_\alpha^{(k)} w_\varepsilon^{(l)} \cdot \nabla_{M_{k,l}^{(\ell)}} \mathcal{L}_{\text{P}}, \tag{12}$$

where $\lambda \in [0, 1)$ is the momentum coefficient. This mechanism allows each sub-policy to accumulate experience gradually, with more relevant prototypes receiving stronger updates. In doing so, the model maintains a shared policy backbone while developing modular, reusable behavioral primitives aligned with latent agent and environment semantics.

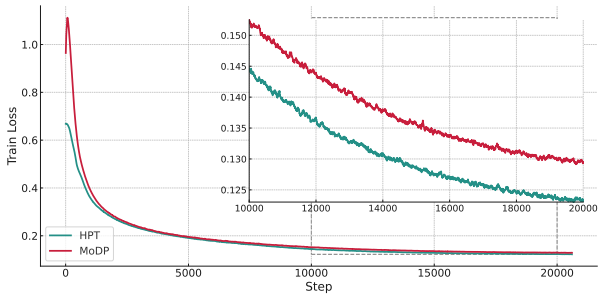
This soft routing and masked update mechanism enables the model to express complex, context-dependent behaviors as mixtures of reusable prototype-aligned sub-policies. More importantly, it equips the system with the ability to generalize to unseen combinations of agents and environments, so long as their constituent prototypes have been encountered during training. In this way, the DPR module bridges prototypes of agent, environment, and policy. The full procedure is outlined in Algorithm 1.

5 Experiments

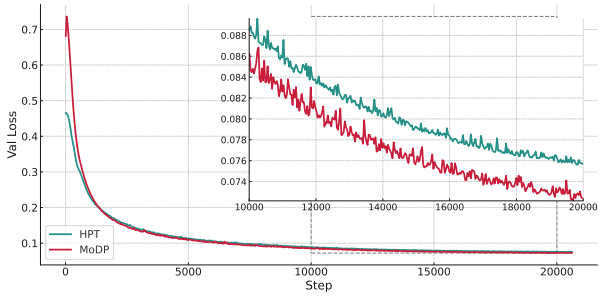
This section presents experiments to evaluate the effectiveness of the proposed MoDP method. Our goal is to assess MoDP's ability to learn robust and generalizable policies under diverse embodiment scenarios. In particular, we focus on three key questions: (1) Can MoDP achieve competitive performance in standard cross-embodiment settings compared to existing generalist policy baselines (Sec. 5.1)? (2) Does MoDP exhibit improved compositional generalization when encountering unseen agent-environment combinations (Sec. 5.2)? (3) What makes MoDP effective (Sec. 5.3)?

Table 1: Per-task success rate on 20 MetaWorld tasks.

Task	HPT		MoDP	
	finetune	scratch	finetune	scratch
assembly-v2	0.03	0.40	0.27	0.72
basketball-v2	0.00	0.63	0.40	0.70
bin-picking-v2	0.27	0.87	0.10	0.83
box-close-v2	0.10	0.40	0.23	0.53
button-press-topdown-v2	1.00	1.00	1.00	1.00
button-press-topdown-wall-v2	1.00	1.00	1.00	1.00
button-press-v2	0.50	0.80	0.83	1.00
button-press-wall-v2	1.00	1.00	0.97	1.00
coffee-button-v2	0.97	1.00	0.97	1.00
coffee-pull-v2	0.30	0.47	0.40	0.62
coffee-push-v2	0.50	0.63	0.60	0.87
dial-turn-v2	0.87	1.00	0.57	0.97
disassemble-v2	0.03	0.30	0.03	0.30
door-close-v2	0.23	1.00	1.00	1.00
door-lock-v2	0.83	0.97	0.73	0.90
door-open-v2	1.00	1.00	1.00	1.00
door-unlock-v2	0.80	1.00	1.00	1.00
drawer-close-v2	0.23	1.00	1.00	1.00
drawer-open-v2	0.97	1.00	1.00	1.00
faucet-open-v2	0.43	1.00	1.00	1.00
hand-insert-v2	0.07	0.13	0.10	0.23
Average	**0.53**	**0.79**	**0.68**	**0.84**



(a) Train Loss

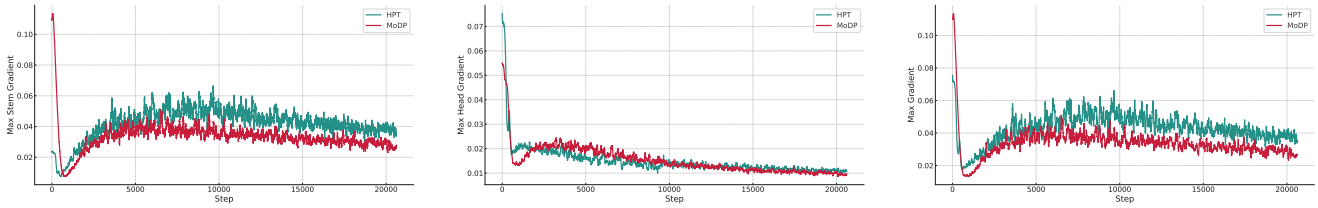


(b) Val Loss

Figure 3: Training and validation loss curves of MoDP (red) and HPT (green) over 20k steps. Insets show zoomed-in views of the later training stage.

5.1 Standard Cross-Embodiment Setting

We follow the evaluation protocol of HPT [39] to assess the standard performance of our MoDP in CEL. Experiments are conducted



(a) Max Stem Grad. (b) Max Head Grad. (c) Max Grad.

Figure 4: Comparison of gradient statistics between MoDP and HPT.

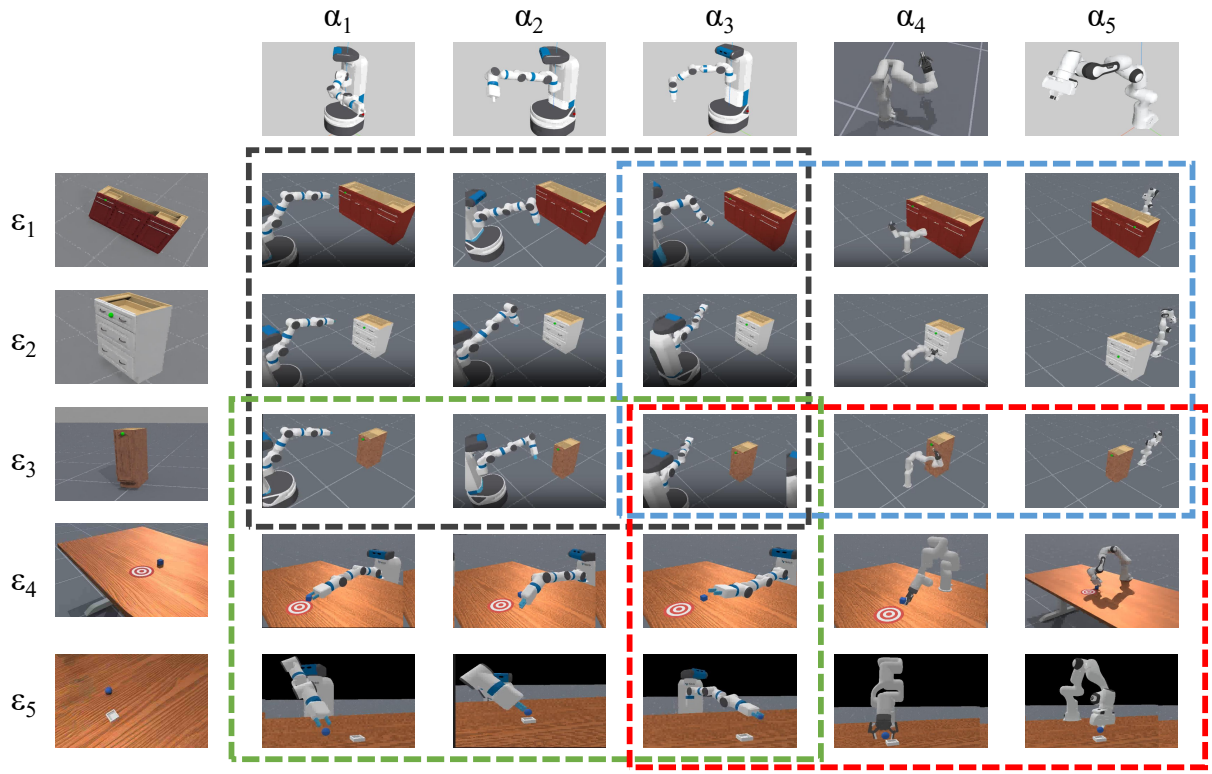


Figure 5: Compositional setting with 5 agents and 5 environments. The colored boxes indicate four types of compositions.

on 20 diverse tasks from the MetaWorld benchmark [53], covering a wide range of manipulation skills and embodiment variations. All models are trained for 20,000 steps using the AdamW optimizer with a learning rate of 1×10^{-5}, cosine annealing schedule, and weight decay of 1×10^{-4}. The batch size is set to 1024, and training is performed for up to 500 epochs with early stopping based on validation performance. In our MoDP, both the agent and environment prototype spaces contain 6 prototypes of dimension 64. The heads in the autoencoder are implemented as two-layer MLPs with ReLU activation, while the decoder consists of a single linear layer.

we compare two training manners in Table 1: one fine-tuned from the pre-trained HPT-base model with the trunk frozen, and the other trained from scratch. MoDP consistently demonstrates performance gains across a wide range of manipulation tasks in both manners, validating its effectiveness in standard CEL setting.

To investigate the generalization behavior during training, we visualize the training and validation loss curves in Fig. 3. MoDP converges faster than HPT, reaching low validation loss in fewer steps. While MoDP exhibits slightly higher training loss than HPT in the later stages, it consistently achieves lower validation loss. This suggests that MoDP generalizes better and avoids overfitting, likely due to its disentangled and mixed prototype strategy, which promotes more robust policy representations.

To further understand the differences in optimization dynamics, we compare the maximum gradients of the stem, head, and entire network in Fig. 4. Compared to HPT, MoDP exhibits a faster decline in gradient magnitudes and maintains consistently lower values in the later stages of training. This supports our hypothesis that not all parameters contribute positively to transfer. By selectively composing policy sub-networks through shared prototype spaces,

Table 2: Success rate under the compositional setting. Each group contains 5 held-out combinations.

In-domain	$(\alpha_2, \varepsilon_1)$	$(\alpha_3, \varepsilon_3)$	$(\alpha_2, \varepsilon_2)$	$(\alpha_3, \varepsilon_1)$	$(\alpha_1, \varepsilon_1)$	**Avg.**
HPT	0.40	0.70	0.53	0.51	0.80	**0.59**
MoDP	0.87	0.80	0.40	0.63	0.93	**0.73**
Cross-Agent-Domain	$(\alpha_5, \varepsilon_3)$	$(\alpha_4, \varepsilon_1)$	$(\alpha_4, \varepsilon_3)$	$(\alpha_5, \varepsilon_1)$	$(\alpha_4, \varepsilon_2)$	**Avg.**
HPT	0.63	0.50	0.02	0.13	0.52	**0.36**
MoDP	1.00	0.63	0.17	0.30	0.67	**0.55**
Cross-Env-Domain	$(\alpha_3, \varepsilon_3)$	$(\alpha_2, \varepsilon_3)$	$(\alpha_3, \varepsilon_4)$	$(\alpha_2, \varepsilon_4)$	$(\alpha_1, \varepsilon_5)$	**Avg.**
HPT	0.27	0.80	0.18	0.65	0.03	**0.39**
MoDP	0.53	0.78	0.30	0.67	0.40	**0.54**
Cross-Domain	$(\alpha_3, \varepsilon_3)$	$(\alpha_3, \varepsilon_5)$	$(\alpha_4, \varepsilon_5)$	$(\alpha_4, \varepsilon_3)$	$(\alpha_5, \varepsilon_4)$	**Avg.**
HPT	0.00	0.33	0.19	0.53	0.07	**0.22**
MoDP	0.20	0.51	0.30	0.67	0.23	**0.38**

MoDP avoids unnecessary parameter updates and achieves more stable optimization and better generalization.

5.2 Compositional Generalization Setting

To evaluate the compositional generalization capability of MoDP, we construct a challenging setting (Fig. 5) with 5 heterogeneous agents and 5 distinct environments based on the ManiSkill3 simulation environment [38]. The first three agents are modified variants of the Fetch robot with different joint ranges, while the last two are entirely different embodiments, XArm6-Robotiq and Panda. Similarly, the first three environments are variations of the OpenCabinet-Drawer-v1 task with differing drawer styles and orientations, and the remaining two are PushCube-v1 and PlaceSphere-v1 tasks.

Owing to the varying degrees of differences between these agents/ environments, we consider four evaluation settings by composing them into four groups: (1) In-Domain compositions (gray dashed box); (2) Cross-Agent-Domain compositions (blue dashed box); (3) Cross-Environment-Domain compositions (green dashed box); and (4) Cross-Domain compositions (red dashed box). This setup allows us to systematically test the model's ability to generalize across increasingly challenging compositional shifts.

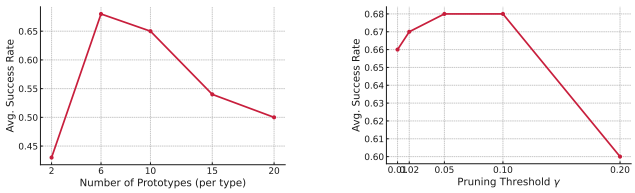
Table 2 presents the success rates under four levels of compositional generalization. Across all settings, our MoDP method consistently outperforms the HPT baseline. In the In-Domain group, both methods perform relatively well, with MoDP achieving a higher average of 0.73. As the compositional gap increases, performance drops are observed, especially in the Cross-Domain group, where HPT achieves only 0.22 on average. However, MoDP maintains robust performance across generalization settings, reaching 0.55 in Cross-Agent-Domain, 0.54 in Cross-Env-Domain, and 0.38 in the most challenging Cross-Domain case. These results highlight the effectiveness of MoDP's prototype-guided routing mechanism in handling unseen agent-environment compositions.

5.3 Ablation Studies

Ablation on components. Table 3 shows that removing any core component of MoDP leads to a performance drop, confirming their complementary roles. The absence of the compositional loss $\mathcal{L}_{AEPD}$ causes the largest drop (from 0.68 to 0.58), highlighting its importance for learning disentangled representations. Removing policy or agent-environment prototypes also degrades performance, showing that both disentangled feature modeling and policy routing are essential for compositional generalization.

Table 3: Ablation study on MoDP components.

Agent-Env Prototypes	Policy Prototypes	$\mathcal{L}_{AEPD}$	Avg. Success Rate
✓	✓		0.58
✓		✓	0.61
	✓	✓	0.65
✓	✓	✓	**0.68**



(a) Effect of prototype number. **(b) Effect of threshold γ.**

Figure 6: Ablation results on key hyperparameters.

The number of prototypes. Fig. 6 (a) demonstrates that the number of prototypes is a critical factor influencing performance. We adopt 6 prototypes as the default setting for the MetaWorld 20-task benchmark, as it yields the best results. Both too few and too many prototypes lead to performance drops, confirming that maintaining a compact and expressive prototype space is essential for effective compositional generalization.

The selection of threshold γ. Fig. 6 (b) illustrates the effect of varying the pruning threshold γ on average success rate. Performance improves as the threshold increases moderately, indicating that pruning helps identify relevant sub-network parameters. However, when the threshold becomes too large, performance drops significantly due to the removal of critical parameters. These results highlight the importance of selecting an appropriate pruning threshold to ensure the effectiveness of the policy prototypes. We adopt $\gamma = 0.05$ as the default setting.

6 Conclusion

This paper introduces MoDP, a Mixture of Disentangled Prototypes method for improving compositional generalization in CEL. By constructing compact prototype spaces for agents, environments, and policies, MoDP bridges the gap between the unbounded space of agent-environment combinations and a single unified policy model. Agent and environment representations are disentangled via a dual-headed autoencoder with compositional reconstruction loss, guided by constraints on representativeness, uniqueness, and simplicity. Policy prototypes are achieved through connection-sensitive pruning, extracting sub-networks from a shared policy backbone. All prototypes are updated via a momentum-based strategy for stable learning. Experimental results demonstrate that MoDP outperforms baselines across standard and compositional settings, marking a promising step toward scalable and generalizable embodied agents.

7 Acknowledgments

This work was supported by the National Natural Science Foundation of China No. 62222209, China Postdoctoral Science Foundation No. 2025M771553, National Key Research and Development Program of China No. 2023YFF1205001, Beijing National Research Center for Information Science and Technology under Grant No. BNR2023TD03006, and Beijing Key Lab of Networked Multimedia.

References

[1] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural Module Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 39–48.

[2] Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Manon Devin, Alex X. Lee, Maria Bauzá Villalonga, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, Antoine Laurens, Claudio Fantacci, Valentin Dalibard, Martina Zambelli, Murilo Fernandes Martins, Rugile Pevceviciute, Michiel Blokzijl, Misha Denil, Nathan Batchelor, Thomas Lampe, Emilio Parisotto, Konrad Zolna, Scott E. Reed, Sergio Gómez Colmenarejo, Jon Scholz, Abbas Abdolmaleki, Oliver Groth, Jean-Baptiste Regli, Oleg Sushkov, Thomas Rothörl, José Enrique Chen, Yusuf Aytar, Dave Barker, Joy Ortiz, Martin A. Riedmiller, Jost Tobias Springenberg, Raia Hadsell, Francesco Nori, and Nicolas Heess. 2023. RoboCat: A Self-Improving Generalist Agent for Robotic Manipulation. *arXiv preprint arXiv:2306.11706* (2023).

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong T. Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. 2023. RT-1: Robotics Transformer for Real-World Control at Scale. In *Proceedings of the Robotics: Science and Systems*.

[4] Hong Chen, Yudong Chen, Xin Wang, Ruobing Xie, Rui Wang, Feng Xia, and Wenwu Zhu. 2021. Curriculum disentangled recommendation with noisy multi-feedback. *Advances in Neural Information Processing Systems* 34 (2021), 26924–26936.

[5] Hong Chen, Xin Wang, Yipeng Zhang, Yuwei Zhou, Zeyang Zhang, Siao Tang, and Wenwu Zhu. 2024. Disenstudio: Customized multi-subject text-to-video generation with disentangled spatial control. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 3637–3646.

[6] Hong Chen, Yipeng Zhang, Xin Wang, Xuguang Duan, Yuwei Zhou, and Wenwu Zhu. 2024. DisenDreamer: Subject-driven text-to-image generation with sample-aware disentangled tuning. *IEEE Transactions on Circuits and Systems for Video Technology* 34, 8 (2024), 6860–6873.

[7] Hong Chen, Yipeng Zhang, Simin Wu, Xin Wang, Xuguang Duan, Yuwei Zhou, and Wenwu Zhu. 2023. Disenbooth: Identity-preserving disentangled tuning for subject-driven text-to-image generation. *arXiv preprint arXiv:2305.03374* (2023).

[8] Tao Chen, Adithyavairavan Murali, and Abhinav Gupta. 2018. Hardware Conditioned Policies for Multi-Robot Transfer Learning. In *Proceedings of the Advances in Neural Information Processing Systems*. 9355–9366.

[9] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. 2019. RoboNet: Large-Scale Multi-Robot Learning. In *Proceedings of the Annual Conference on Robot Learning*, Vol. 100. 885–897.

[10] Ria Doshi, Homer Rich Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. 2024. Scaling Cross-Embodied Learning: One Policy for Manipulation, Navigation, Locomotion and Aviation. In *Proceedings of the Conference on Robot Learning*, Vol. 270. 496–512.

[11] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. 2022. MineDojo: Building Open-Ended Embodied Agents with Internet-Scale Knowledge. In *Proceedings of the Advances in Neural Information Processing Systems*.

[12] Gilbert Feng, Hongbo Zhang, Zhongyu Li, Xue Bin Peng, Bhuvan Basireddy, Linzhu Yue, Zhitao Song, Lizhi Yang, Yunhui Liu, Koushil Sreenath, and Sergey Levine. 2022. GenLoco: Generalized Locomotion Controllers for Quadrupedal Robots. In *Proceedings of the Conference on Robot Learning*, Vol. 205. 1893–1903.

[13] Tongtong Feng, Xin Wang, Feilin Han, Leping Zhang, and Wenwu Zhu. 2024. U2udata: A large-scale cooperative perception dataset for swarm uavs autonomous flight. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 7600–7608.

[14] Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Erkang Zhu, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, Peter Chang, Ricky Loynd, Robert West, Victor Dibia, Ahmed Awadallah, Ece Kamar, Rafah Hosn, and Saleema Amershi. 2024. Magentic-One: A Generalist Multi-Agent System for Solving Complex Tasks. *arXiv preprint arXiv:2411.04468* (2024).

[15] Minghe Gao, Wendong Bu, Bingchen Miao, Yang Wu, Yunfei Li, Juncheng Li, Siliang Tang, Qi Wu, Yueting Zhuang, and Meng Wang. 2024. Generalist Virtual Agents: A Survey on Autonomous Agents Across Digital Platforms. *arXiv preprint arXiv:2411.10943* (2024).

[16] Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. 2024. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of the Robotics: Science and Systems*.

[17] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory F. Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. 2017. Deep Learning Scaling is Predictable, Empirically. *arXiv preprint arXiv:1712.00409* (2017).

[18] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *Proceedings of the International Conference on Learning Representations*.

[19] Wenlong Huang, Igor Mordatch, and Deepak Pathak. 2020. One Policy to Control Them All: Shared Modular Policies for Agent-Agnostic Control. In *Proceedings of the International Conference on Machine Learning*, Vol. 119. 4455–4464.

[20] Thomas N. Kipf, Elise van der Pol, and Max Welling. 2020. Contrastive Learning of Structured World Models. In *Proceedings of the International Conference on Learning Representations*.

[21] Brenden M. Lake and Marco Baroni. 2018. Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks. In *Proceedings of the International Conference on Machine Learning*, Vol. 80. 2879–2888.

[22] Yunshi Lan, Lei Wang, Jing Jiang, and Ee-Peng Lim. 2022. Improving Compositional Generalization in Math Word Problem Solving. *arXiv preprint arXiv:2209.01352* (2022).

[23] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. 2019. Snip: single-Shot Network Pruning based on Connection sensitivity. In *Proceedings of the International Conference on Learning Representations*.

[24] Haoyang Li, Xin Wang, Zeyang Zhang, Haibo Chen, Ziwei Zhang, and Wenwu Zhu. 2024. Disentangled graph self-supervised learning for out-of-distribution generalization. In *Forty-first International Conference on Machine Learning*.

[25] Haoyang Li, Xin Wang, Ziwei Zhang, Zehuan Yuan, Hang Li, and Wenwu Zhu. 2021. Disentangled contrastive learning on graphs. *Advances in Neural Information Processing Systems* 34 (2021), 21872–21884.

[26] Haoyang Li, Xin Wang, Xueling Zhu, Weigao Wen, and Wenwu Zhu. 2025. Disentangling invariant subgraph via variance contrastive estimation under distribution shifts. In *Forty-second International Conference on Machine Learning*.

[27] Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. 2022. Disentangled graph contrastive learning with independence promotion. *IEEE Transactions on Knowledge and Data Engineering* 35, 8 (2022), 7856–7869.

[28] Baihan Lin, Djallel Bouneffouf, and Irina Rish. 2023. A Survey on Compositional Generalization in Applications. *arXiv preprint arXiv:2302.01067* (2023).

[29] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. 2020. Object-Centric Learning with Slot Attention. In *Proceedings of the Advances in Neural Information Processing Systems*.

[30] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. 2019. Disentangled graph convolutional networks. In *International conference on machine learning*. PMLR, 4212–4221.

[31] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 483–491.

[32] Davide Mambelli, Frederik Träuble, Stefan Bauer, Bernhard Schölkopf, and Francesco Locatello. 2022. Compositional Multi-Object Reinforcement Learning with Linear Relation Networks. *arXiv preprint arXiv:2201.13388* (2022).

[33] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alexander Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew E. Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman,

Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Paul Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Haoshu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I. Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi Jim Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J. Joshi, Niko Sünderhauf, Ning Liu, Norman Di Palo, Nur Muhammad (Mahi) Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R. Sanketi, Patrick Tree Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham D. Sonawani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Liangwei Xu, Xuanlin Li, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, and Zipeng Lin. 2024. Open X-Embodiment: Robotic Learning Datasets and RT-X Models : Open X-Embodiment Collaboration. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 6892–6903.

[34] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. 2018. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 3803–3810.

[35] Yijian Qin, Xin Wang, Ziwei Zhang, and Wenwu Zhu. 2023. Disentangled representation learning with large language models for text-attributed graphs. *arXiv preprint arXiv:2310.18152* (2023).

[36] Linlu Qiu, Peter Shaw, Panupong Pasupat, Tianze Shi, Jonathan Herzig, Emily Pitler, Fei Sha, and Kristina Toutanova. 2022. Evaluating the Impact of Model Scale for Compositional Generalization in Semantic Parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 9157–9179.

[37] Scott E. Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. 2022. A Generalist Agent. *Transactions on Machine Learning Research* 2022 (2022).

[38] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. 2024. ManiSkill3: GPU Parallelized Robotics Simulation and Rendering for Generalizable Embodied AI. *arXiv preprint arXiv:2410.00425v1* (2024).

[39] Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. 2024. Scaling Proprioceptive-Visual Learning with Heterogeneous Pre-trained Transformers. In *Proceedings of the Advances in Neural Information Processing Systems*.

[40] Ren Wang, Haoliang Sun, Xiushan Nie, Yuxiu Lin, Xiaoming Xi, and Yilong Yin. 2023. Multi-View Representation Learning via View-Aware Modulation. In *Proceedings of the ACM International Conference on Multimedia*. 3876–3886.

[41] Ren Wang, Haoliang Sun, Xiushan Nie, and Yilong Yin. 2022. SNIP-FSL: Finding task-specific lottery jackpots for few-shot learning. *Knowledge-Based Systems* 247 (2022), 108427.

[42] Xin Wang, Hong Chen, Zirui Pan, Yuwei Zhou, Chaoyu Guan, Lifeng Sun, and Wenwu Zhu. 2025. Automated disentangled sequential recommendation with large language models. *ACM Transactions on Information Systems* 43, 2 (2025), 1–29.

[43] Xin Wang, Hong Chen, Si'ao Tang, Zihao Wu, and Wenwu Zhu. 2024. Disentangled Representation Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 12 (2024), 9677–9696.

[44] Xin Wang, Hong Chen, Yuwei Zhou, Jianxin Ma, and Wenwu Zhu. 2022. Disentangled representation learning for recommendation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 1 (2022), 408–424.

[45] Xin Wang, Hong Chen, and Wenwu Zhu. 2021. Multimodal disentangled representation for recommendation. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE Computer Society, 1–6.

[46] Xin Wang, Hong Chen, and Wenwu Zhu. 2023. Disentangled representation learning for multimedia. In *Proceedings of the 31st ACM International Conference on Multimedia*. 9702–9704.

[47] Xin Wang, Zirui Pan, Hong Chen, and Wenwu Zhu. 2025. Divico: Disentangled visual token compression for efficient large vision-language model. *IEEE Transactions on Circuits and Systems for Video Technology* (2025).

[48] Xin Wang, Zirui Pan, Yuwei Zhou, Hong Chen, Chendi Ge, and Wenwu Zhu. 2023. Curriculum co-disentangled representation learning across multiple environments for social recommendation. In *International Conference on Machine Learning*. PMLR, 36174–36192.

[49] Xin Wang, Zihao Wu, Hong Chen, Xiaohan Lan, and Wenwu Zhu. 2023. Mixup-augmented temporally debiased video grounding with content-location disentanglement. In *Proceedings of the 31st ACM International Conference on Multimedia*. 4450–4459.

[50] Jonathan Heewon Yang, Catherine Glossop, Arjun Bhorkar, Dhruv Shah, Quan Vuong, Chelsea Finn, Dorsa Sadigh, and Sergey Levine. 2024. Pushing the Limits of Cross-Embodiment Learning for Manipulation and Navigation. In *Proceedings of the Robotics: Science and Systems*.

[51] Naoki Yokoyama, Ram Ramrakhya, Abhishek Das, Dhruv Batra, and Sehoon Ha. 2024. HM3D-OVON: A Dataset and Benchmark for Open-Vocabulary Object Goal Navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 5543–5550.

[52] Heng You, Tianpei Yang, Yan Zheng, Jianye Hao, and Matthew E. Taylor. 2022. Cross-domain adaptive transfer reinforcement learning based on state-action correspondence. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, Vol. 180. 2299–2309.

[53] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. 2020. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. In *Proceedings of the Conference on Robot Learning* (2020-05-12). 1094–1100.

[54] Yipeng Zhang, Xin Wang, Hong Chen, and Wenwu Zhu. 2023. Adaptive disentangled transformer for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*. 3434–3445.

[55] Zeyang Zhang, Xin Wang, Haibo Chen, Haoyang Li, and Wenwu Zhu. 2024. Disentangled Dynamic Graph Attention Network for Out-of-Distribution Sequential Recommendation. *ACM Transactions on Information Systems* 43, 1 (2024), 1–42.

[56] Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, and Wenwu Zhu. 2023. Out-of-distribution generalized dynamic graph neural network with disentangled intervention and invariance promotion. *arXiv preprint arXiv:2311.14255* (2023).

[57] Zeyang Zhang, Xin Wang, Ziwei Zhang, Guangyao Shen, Shiqi Shen, and Wenwu Zhu. 2023. Unsupervised graph neural architecture search with disentangled self-supervision. *Advances in Neural Information Processing Systems* 36 (2023), 73175–73190.

[58] Linfeng Zhao, Lingzhi Kong, Robin Walters, and Lawson L. S. Wong. 2022. Toward Compositional Generalization in Object-Oriented World Modeling. In *Proceedings of the International Conference on Machine Learning*, Vol. 162. 26841–26864.

[59] Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. 2024. Towards Learning a Generalist Model for Embodied Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13624–13634.

[60] Allan Zhou, Vikash Kumar, Chelsea Finn, and Aravind Rajeswaran. 2024. Policy Architectures for Compositional Generalization in Control. *Reinforcement Learning Journal* 5 (2024), 2264–2283.

[61] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong T. Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. 2023. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of the Conference on Robot Learning*, Vol. 229. 2165–2183.