Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

DISC: Decoupling Instruction from State-Conditioned Control via Policy Generation

Hanxiang Ren[‡,§,¶,*] Pei Zhou[§,¶,*] Xunzhe Zhou[§,¶] Yanchao Yang[§,¶,†]
[‡]Zhejiang University [§]The University of Hong Kong [¶]TranscEngram

***Abstract*—Language-conditioned manipulation policies typically process instructions and observations through shared network parameters. This *task-state entanglement* provides a pathway for *observation leakage* – networks learn scene-to-action shortcuts that bypass language grounding entirely. *DISC* eliminates this failure structurally. Rather than conditioning a universal policy on language, *DISC* uses a hypernetwork to generate the *entire parameter set* of a task-specific visuomotor policy from the instruction alone. The generated policy never directly accesses language; therefore, its task-awareness must come from the language. Consequently, observation leakage has no pathway to emerge. On the other hand, generating coherent high-dimensional policy weights is itself a challenging problem. We address it with a two-stage hypernetwork whose refinement stage embeds the structure of gradient-based optimization as a feed-forward inductive bias, producing globally consistent parameters without actual gradient computation. Trained entirely from scratch on standard data budgets, *DISC* outperforms all entangled baselines on LIBERO-90 and Meta-World, with advantages that widen on complex, long-horizon tasks – and surpasses the large-scale pretrained π_0 despite using no external pretraining data. On a real-world benchmark where all tasks share identical visual context, *DISC* substantially outperforms entangled alternatives, directly confirming that language-generated policy parameters, not visual shortcuts, drive behavior. The hypernetwork further learns a semantically structured parameter manifold that enables few-shot adaptation from minimal demonstrations and robust generalization across paraphrased instructions. Our code is available at: https://github.com/ReNginx/DISC.**

I. Introduction

Reliable task-conditioned manipulation requires robots to treat the task specification as the source of task identity, rather than as a weak hint that can be overridden by familiar visual scenes. Language-conditioned manipulation policies must process two fundamentally different inputs: an instruction that specifies *what task to perform* (fixed throughout an episode) and an observation that captures *what the current state is* (evolving continuously). Current architectures conflate these roles by processing both through shared representations – a design we term *task-state entanglement*. We argue this creates two compounding failures. *First,* shared parameters face optimization tension between language grounding (requiring semantic understanding) and visuomotor control (requiring spatial precision). *Second,* and more critically, entangled architectures enable *observation leakage*: visual context often correlates with task identity in limited robotics datasets, allowing the network to map scenes directly to actions while bypassing language entirely. For example, in a shared kitchen scene, a policy may learn the frequent pan-placement behavior and omit the instructed stove-activation subgoal. The result is policies that exploit visual shortcuts rather than ground instructions – succeeding on familiar scenes but failing when instructions change or visual context becomes ambiguous. Evidence from the broader vision-language community confirms this pattern, manifesting as modality competition [21, 37], visual neglect [23], and entity grounding failures [1, 24].

We propose *DISC*, which takes architectural decoupling to its logical conclusion: a task specification, instantiated here as language, generates the *entire parameter set* of a task-specific visuomotor policy. A hypernetwork processes instructions alone to produce policy weights; the generated policy processes visual observations alone using those weights. This separation provides a structural guarantee against observation leakage – no task-agnostic parameters exist where shortcuts could emerge, and task semantics influence behavior only through generated weights. A central challenge is that naive weight generation produces incoherent parameters. *DISC* addresses this with a two-stage architecture. A Weight Initialization Network first maps the language embedding to a semantically informed point in parameter space. An Iterative Refinement Module then improves this initialization by mimicking the structure of gradient-based optimization – forward evaluation, error estimation, and parameter correction – while remaining fully feed-forward at inference time. This optimization-inspired inductive bias enables the hypernetwork to produce globally coherent policy parameters without incurring the cost of actual gradient computation, bridging the gap between the expressiveness of full policy generation and the quality demands of precise visuomotor control.

Despite training entirely from scratch, *DISC* achieves 94.3% on LIBERO-90 and 92.2% on Meta-World, outperforming the strongest trained-from-scratch entangled baseline by 7.7% on LIBERO-90. For context, *DISC* surpasses even the pretrained π_0 (91.6%) and remains competitive with $\pi_{0.5}$ (95.7%) – demonstrating that architectural inductive biases can recover much of the benefit attributed to large-scale pretraining. The advantage widens on complex, long-horizon tasks, precisely where task-state entanglement is most damaging. On a real-world combinatorial benchmark where visual context is shared across all 9 tasks, *DISC* achieves 86.4% versus 78.5% for the best entangled baseline, confirming that task-specific generated parameters, not visual shortcuts, drive behavior. The decou-

[*]Equal contribution. [†]Corresponding author.

pled architecture further enables superior few-shot adaptation, robust grounding across paraphrased instructions, and a semantically structured parameter manifold where related tasks cluster closely by functional meaning.

II. Related Work

Multi-Task Manipulation Policies. Current language-conditioned manipulation methods process instructions and observations through shared network components and differ mainly in their fusion mechanism. Transformer-based approaches [5, 16, 42, 45, 43] tokenize and concatenate multimodal inputs into unified sequences [39], as exemplified by RT-1 [4], Gato [27], Octo [38], VQ-BeT [34], and BAKU [9] – where the same W_Q, W_K, W_V projections and feed-forward weights operate on both language and visual tokens. Text-aware variants such as OTTER [12] condition visual features on language, but retain shared action-prediction parameters. Diffusion-based methods [10, 36] instead cast control as conditional denoising, pioneered by Diffusion Policy [6]. Architectures have evolved from U-Nets [32] with FiLM conditioning [26] to Transformer-based backbones such as DiT [25] and BESO [29], with RDT-1B [20] and MoDE [30] scaling through pre-training and mixture-of-experts tuning, while DP3 [41], 3D Diffuser Actor [15], and PointVLA [17] enhance geometric grounding via 3D representations. Flow-matching models π_0 [3] and $\pi_{0.5}$ [14] further improve generalization through large-scale pre-training. Despite this diversity, all these methods entangle instruction and state within shared processing stages. Concatenation-based transformers apply identical weights to both modalities throughout the network. Cross-attention and FiLM-based architectures [26] introduce partial separation – FiLM affects vision only through affine modulation – but language still controls only a small fraction of computation, while the vast majority of parameters remain task-agnostic and can learn shortcut mappings that bypass language entirely. As analyzed in Section III-A, this task-state entanglement creates two problems: optimization tension between competing functions within shared parameters, and observation leakage that enables the network to map visual context directly to actions without grounding language.

Hypernetworks for Policy Generation. An alternative paradigm uses hypernetworks [2, 8, 13, 44] – networks trained to output the weights of another – to generate task-specific policies. HyPoGen [28], HyperZero [31], and Hyper-GoalNet [44] demonstrate this for multi-task or goal-conditioned manipulation. These methods motivate separating task specification from policy execution, but differ in conditioning and generator design: HyPoGen and HyperZero use compact MLP-style generators, while Hyper-GoalNet conditions on goal images and current observations rather than language-only specifications. *DISC* addresses both the framework question – decoupling instruction from state-conditioned control by generating task-specific target policy parameters from language – and the architecture question, introducing an optimization-inspired hypernetwork whose iterative refinement enables feed-forward generation of high-quality policy parameters. Although adaptable to meta-learning [7, 33, 35], we focus on multi-task manipulation policy learning.

III. Preliminary

We target multi-task robotic manipulation, where an agent must execute diverse tasks specified by natural language instructions. Each task is defined by an instruction $l \in \mathcal{J}$, and at each timestep, the agent receives an observation $o_t = (I_t, s_t)$ comprising an RGB image and proprioceptive state, from which it produces a continuous action $a_t \in \mathcal{A}$.

Notably, the instruction and observation serve qualitatively different roles. The instruction specifies *what task to perform* – it defines the goal and remains fixed throughout an episode. The observation captures *what the current state is* – it evolves continuously and directly constrains which action is appropriate. A well-designed policy shall treat these inputs according to their distinct roles, without conflating the two.

For learning, given a dataset $\mathcal{D}$ of expert demonstrations spanning N tasks, we train a policy via behavior cloning:

$$\mathcal{L}_{\text{BC}}(\theta) = \mathbb{E}_{(o_t,a_t,l)\sim\mathcal{D}}\left[\|\pi(o_t, l; \theta) - a_t\|_2^2\right] . \tag{1}$$

Current approaches learn an entangled policy $\pi(o, l; \theta)$ that processes both inputs through shared parameters – a design we term **task-state entanglement**. As we argue next, this architectural choice introduces fundamental learning difficulties that compromise language grounding.

A. Limitations of Task-State Entanglement

Task-state entanglement creates two distinct problems that undermine robust instruction following.

a) Problem 1: Competing Functions within Shared Parameters: A language-conditioned policy must serve two distinct roles: interpreting the instruction to understand what task to perform, and processing observations to predict appropriate actions. In entangled architectures, both roles are handled by the same parameters. For instance, transformer-based policies concatenate language and visual tokens:

$$\text{Output} = \text{Transformer}(\text{Concat}(T_l, T_v); \theta)\,; \tag{2}$$

Diffusion policies similarly condition on jointly embeddings:

$$a_t = \text{Denoise}(\epsilon_t \mid e_l, e_v; \theta)\,. \tag{3}$$

In both designs, shared parameters θ must simultaneously learn language understanding and visuomotor control. However, demands conflict: language understanding benefits from semantic abstraction and visual invariance, while visuomotor control requires sensitivity to precise spatial details. Thus, forcing both roles through shared parameters creates optimization tension that degrades either core function.

b) Problem 2: Observation Leakage Enables Shortcut Learning: A more severe failure arises from what we call **observation leakage**: visual observations often contain cues correlating with task identity. Specific objects, scene layouts, or robot configurations may appear predominantly in certain tasks. Given this, entangled architectures can learn to map visual patterns directly to actions – bypassing language. This

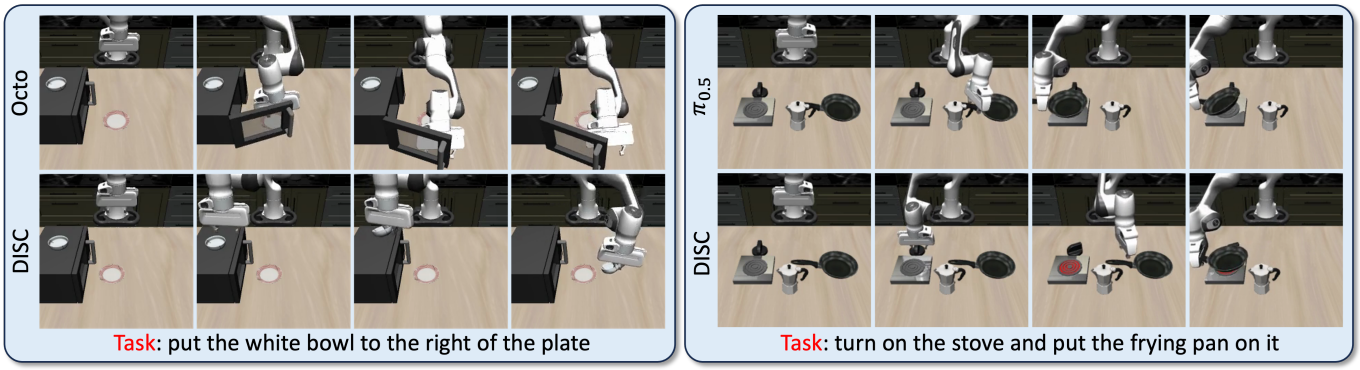

Fig. 1: **Behavioral Evidence of Task-State Entanglement.** Current policies trained with entangled architectures fail to ground language instructions faithfully. Left: given the instruction regarding "white bowl," Octo instead approaches the microwave – executing a behavior associated with a different task that shares similar visual context. Right: given the multi-step instruction "turn on the stove and put the frying pan on it," the large-scale pretrained $\pi_{0.5}$ skips the stove-activation subgoal and directly places the pan on the stove, matching a related pan-placement task under the same scene-level context. These failures illustrate observation leakage: the policies have learned scene-to-action mappings rather than instruction-to-action mappings. *DISC*, with its decoupled architecture, correctly executes both instructions by separating task specification from state-conditioned control.

shortcut minimizes training loss without requiring the network to ground linguistic meaning, yielding policies that succeed on familiar scenes but fail when instructions change.

Figure 1 demonstrates this failure. In the left example, Octo approaches the microwave when asked to put the white bowl to the right of the plate, executing a behavior linked to a similar visual context. In the right example, the pretrained $\pi_{0.5}$ is asked to turn on the stove and put the frying pan on it, but it performs only the pan-placement subgoal, consistent with a scene-level shortcut from the related task of putting the pan on the stove under the same scene context. These examples show that observation leakage can persist across model scales when language and state are fused during action fine-tuning. In contrast, our decoupled architecture executes the specified subgoals, confirming that architectural separation prevents such shortcuts. These problems compound to produce policies that are brittle to instruction variation, fail to generalize across visual contexts, and degrade on long-horizon tasks where grounding errors accumulate. To overcome these limitations, we propose treating language not as an input to be fused, but as a generative signal that produces the parameters of a task-specific visuomotor policy – an approach we detail next.

IV. Method

A. Decoupling Instruction from State-Conditioned Control

Our solution is to separate task-specification processing from state-conditioned control entirely. In this paper, the task specification is instantiated as a natural-language instruction. Rather than learning a single policy that fuses both modalities, our method, *DISC*, decomposes the problem into two stages: (1) a hypernetwork $\mathcal{H}_\phi$ processes the task specification to generate a complete set of policy weights, and (2) the generated policy π_{θ_π} maps observations to actions using only those weights – without any direct task-specification input.

Formally, given an instruction l as the task specification, we first obtain a task embedding via a frozen language encoder:

$$e_l = \Phi_L(l)\,. \tag{4}$$

The hypernetwork then generates policy parameters:

$$\theta_\pi = \mathcal{H}_\phi(e_l)\,. \tag{5}$$

Further, the generated policy performs visuomotor control:

$$a_t = \pi(o_t; \theta_\pi)\,. \tag{6}$$

This decomposition directly addresses both problems identified in Section III-A. *First,* the competing functions are assigned to separate networks: the hypernetwork learns to interpret the task specification, while the target policy learns to map observation embeddings to actions, eliminating optimization tension between task understanding and visuomotor control. *Second,* the direct action-mapping pathway for observation leakage is structurally removed. In entangled architectures, shortcuts arise because the model can learn to ignore the task specification and map observations directly to actions – both modalities feed into the same network, so bypassing one is easy. In *DISC*, the target policy has no direct access to the instruction at all. The only way for task information to influence behavior is through the generated parameters; the hypernetwork *must* encode task semantics into the weights because there is no alternative route.

Figure 2 illustrates this architecture. Observations o_t enter only at the target policy stage, maintaining complete separation from task-specification processing.

a) Contrast with Conditioning Approaches: Existing conditioning methods – concatenation [34, 38], cross-attention [18], and FiLM [6, 26] – integrate language and vision within shared processing stages, to different extents.

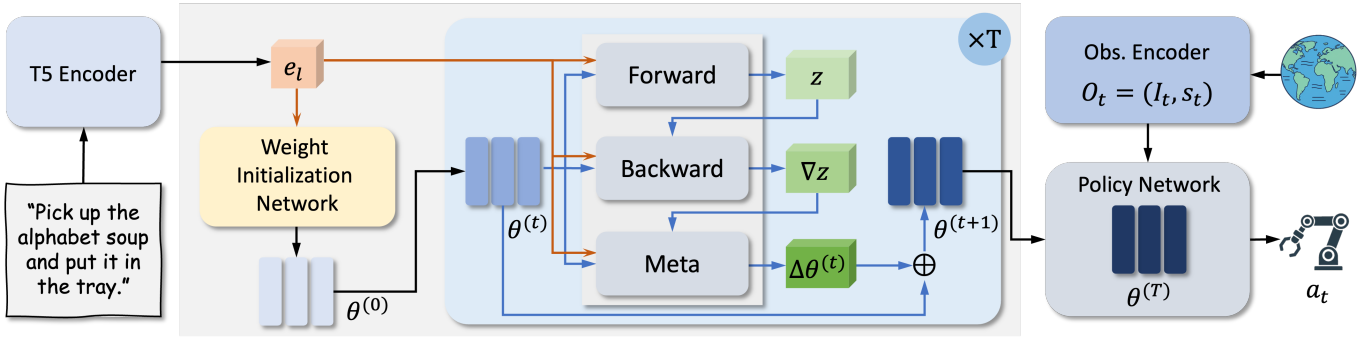

Fig. 2: **The *DISC* architecture.** The task specification, instantiated as a language instruction l, is encoded into embedding e_l and processed by the hypernetwork through two stages: (1) the Weight Initialization Network generates initial parameters $\theta_\pi^{(0)}$ (labeled as $\theta^{(0)}$), and (2) the learned iterative refinement module updates parameters over T steps to produce final $\theta^{(T)}$. The refinement module mimics optimization structure – forward, backward, and meta-update operations – but all transformations are learned, and the entire generation process is feed-forward. The generated parameters configure the target policy π_{θ_π}, a compact, task-specific visuomotor controller. Observations o_t enter only at the target policy stage: the task specification determines *which* policy executes, while the generated policy processes visual input independently, ensuring complete separation between task-specification processing and state-conditioned control.

Concatenation exhibits the most direct form of entanglement: language tokens T_l and visual tokens T_v are concatenated into a unified sequence and processed by the same transformer layers. *Cross-attention* uses modality-specific projections to compute queries from vision and keys/values from language. While these projections are separate, the attention operation produces fused features that are processed by subsequent shared layers, entangling the two modalities from that point onward. *FiLM* is closest in spirit to our approach: language affects vision through learned affine modulation $(\gamma(e_l)\cdot h_v+\beta(e_l))$, and no layer processes both modalities with shared parameters. *However,* language generates only a small fraction of the computation – the scale and shift values – while the vast majority of the visual backbone parameters remain shared across all tasks. These shared, task-agnostic parameters can still learn shortcut mappings from visual features to actions that bypass language conditioning entirely.

DISC takes this principle to its logical conclusion: *all* task-specific parameters of the target policy are generated from the task specification, instantiated here as a language instruction, leaving no space in the target policy where shortcuts could emerge. The policy's entire computational structure is determined by the task specification, ensuring that task semantics are embedded throughout the visuomotor controller rather than applied as a light modulation.

B. Hypernetwork Architecture

A core challenge is generating coherent, high-performing policy parameters from a task embedding induced by a language instruction. Naive direct regression from e_l to the full parameter vector θ_π fails because policy networks contain millions of interdependent parameters with complex functional relationships. Prior hypernetwork architectures (e.g., simple MLP generators) struggle to capture these dependencies, producing weights that lack the coherent structure required for effective control.

Our key architectural contribution is a hypernetwork that incorporates the *structural inductive bias of iterative optimization* while remaining *fully feed-forward at inference time*. The design is inspired by a simple observation: gradient-based optimization naturally produces coherent parameters via incremental updates respecting inter-layer dependencies. We embed this structure into the architecture itself – the hypernetwork *mimics* the computational pattern of optimization (forward pass, backward pass, parameter update), but *how* each of these operations refines the parameters is entirely learned from data. This achieves the coherence benefits of optimization-like refinement without incurring the computational cost of actual gradient computation at inference time.

a) Stage 1: Weight Initialization Network: The first stage generates a coarse initialization from the task embedding:

$$\theta_\pi^{(0)} = \text{WIN}_{\phi_1}(e_l)\,. \tag{7}$$

This positions the parameters in a well-conditioned region of the weight space – distinguishing, for instance, grasping tasks from pushing tasks – but lacks the fine-grained precision required for successful execution.

b) Stage 2: Learned Iterative Refinement: The second stage refines this initialization through T learned update steps:

$$\theta_\pi^{(t+1)} = \theta_\pi^{(t)} + \Delta\theta^{(t)}, \quad \text{where} \quad \Delta\theta^{(t)} = f_{\phi_2}(\theta_\pi^{(t)}, e_l)\,. \tag{8}$$

Importantly, this is *not* optimization in the traditional sense. The refinement module f_{ϕ_2} is a neural network with fixed, learned parameters ϕ_2. At inference time, generating policy parameters requires only T forward passes through f_{ϕ_2} – no gradients are computed with respect to any loss, no demonstration data is accessed, and no iterative solving occurs. The entire generation process is feed-forward.

Specifically, the refinement module is architecturally decomposed to mirror the structure of optimization: it computes a

latent forward representation z from the current parameters, estimates pseudo-gradient signals ∇z, and aggregates these into parameter updates $\Delta\theta^{(t)}$ (see Figure 2). However, unlike true optimization where gradients are computed analytically from a loss function, here the "forward," "backward," and "update" operations are all learned neural network components. The architecture provides the structural scaffold – the pattern of how information flows – while training determines the actual transformation at each stage. This learned refinement process produces globally coherent parameter updates, similar to how true gradients enforce consistency across layers, but adapted specifically to the encountered task distribution.

c) Target Policy: The generated parameters θ_π configure a compact visuomotor controller mapping observations to actions. Because the target policy is specialized to a single task, it can be significantly smaller than monolithic architectures that must accommodate the entire task distribution within one set of weights. In our implementation, the target policy is a lightweight MLP, enabling efficient inference suitable for high-frequency robotic control. More architectural details are provided in the appendix.

d) Training and Inference: The full framework is trained end-to-end by optimizing hypernetwork parameters $\phi = \{\phi_1, \phi_2\}$ using the behavior cloning loss:

$$\mathcal{L}_{\text{BC}}(\phi) = \mathbb{E}_{(o_t,a_t,l)\sim\mathcal{D}}\left[\|\pi(o_t; \mathcal{H}_\phi(\Phi_L(l))) - a_t\|_2^2\right] . \quad (9)$$

For each training sample, the hypernetwork generates policy parameters $\theta_\pi = \theta_\pi^{(T)}$ from the task specification, the target policy uses these parameters to predict actions, and the prediction error backpropagates through the entire generation process to update ϕ accordingly. At inference time, deployment is straightforward: given a new instruction l as the task specification, we encode it, run the hypernetwork forward to generate θ_π, and execute the target policy in the environment. The generation is a single feed-forward pass through the hypernetwork for each task, after which the compact target policy runs independently at control frequency without further overhead during execution.

C. Few-Shot Adaptation (Optional)

We call this adaptation optional because it is not required by the standard inference pipeline; it is used only when a novel task is accompanied by a few demonstrations. A practical benefit of *DISC*'s decoupled architecture is efficient adaptation to novel tasks from minimal demonstrations. The separation between the hypernetwork (which captures general knowledge about how language maps to policy structure) and the target policy (which executes a specific task) provides a natural decomposition for efficient adaptation.

When presented with a novel task with instruction l_{new} and a small number of demonstrations $\mathcal{D}_{\text{new}} = \{\xi_j\}_{j=1}^K$, adaptation proceeds in two stages. *First,* the hypernetwork generates by mapping the new instruction to policy parameters:

$$\theta_\pi^{\text{init}} = \mathcal{H}_\phi(\Phi_L(l_{\text{new}})) . \quad (10)$$

Since the hypernetwork has learned the structure of the task manifold during multi-task training, this initialization already encodes relevant task semantics – positioning the parameters near the task-optimal region rather than at a random starting point. *Second,* the generated target policy parameters are fine-tuned with the provided demonstrations:

$$\theta_\pi^* = \arg\min_{\theta_\pi} \sum_{\xi_j\in\mathcal{D}_{\text{new}}} \sum_{(o_t,a_t)\sim\xi_j} \|\pi(o_t;\theta_\pi) - a_t\|_2^2 . \quad (11)$$

The hypernetwork remains frozen throughout, preserving its learned meta-knowledge. This transforms adaptation from a global search over a large, entangled parameter space into a local refinement of a compact, well-initialized policy – requiring fewer demonstrations and fewer gradient steps to reach competent performance.

V. Experiment

We conduct experiments to validate that explicit decoupling of instruction processing from state-conditioned control through *DISC* improves language-conditioned robotic manipulation. Our evaluation addresses four key research questions:

1) **Multi-Task Efficiency (RQ1):** Does *DISC*'s decoupled architecture outperform state-of-the-art entangled methods on diverse manipulation tasks?
2) **Few-Shot Adaptation (RQ2):** Can the structured parameter manifold learned by *DISC* enable more effective adaptation to novel tasks compared to entangled representations?
3) **Language Grounding Quality (RQ3):** Does *DISC*'s decoupled architecture ground the semantic content of instructions – responding to task meaning rather than surface-level phrasing – and does the generated parameter space reflect meaningful task structure?
4) **Real-World Deployment (RQ4):** Does *DISC*'s decoupling principle transfer to physical robot settings, where real-world perception noise and scene ambiguity might affect precise language-driven control?

The experimental setup is detailed in Section V-A. We report multi-task performance in Section V-B, few-shot adaptation in Section V-C, language grounding and structure analysis in Section V-D, and real-world evaluation in Section V-E. Further implementation details are provided in the appendix.

A. Experimental Setup

a) Benchmarks: In simulation, we employ two standard benchmarks to test multi-task proficiency: **LIBERO-90** [19] and **Meta-World** [40]. For LIBERO-90, we train on 90 diverse language-conditioned tasks and evaluate under novel initial configurations. To analyze performance relative to task complexity, we categorize these tasks into **Easy**, **Medium**, and **Long-Horizon** groups based on the number of object interactions involved. For Meta-World, we adopt the ML45 protocol (45 goal-conditioned tasks) and group them by manipulation primitives (e.g., `pick&place`, `press&pull`) to assess semantic generalization across diverse mechanics. We utilize the standard dataset budget: 50 human demonstrations per task for LIBERO-90 and 100 expert demonstrations per

task for Meta-World. Detailed task lists and categorization criteria are provided in the appendix.

Real-World Benchmark. To rigorously evaluate whether language instructions – rather than visual cues – drive policy behavior, we construct a real-world testbed comprising 9 distinct combinatorial tasks. Figure 3 illustrates this experimental protocol. The setup involves picking one of 3 potential target objects and placing it into one of 3 specific containers, resulting in $3 \times 3 = 9$ unique language-conditioned tasks. As depicted in Figure 3 (a), the workspace is cluttered with distractors. The yellow arrows exemplify the task ambiguity that is resolved solely by language: for instance, the Red Apple corresponds to three distinct potential trajectories depending on the target container specified in the instruction. Figure 3 (b) shows the terminal state of a successful execution. Crucially, because the same objects and containers are present in every task, visual context alone cannot determine the correct behavior – making this benchmark a direct test of whether observation leakage drives the policy. We collect 100 demonstrations for each task. Additional environment details are deferred to the appendix.

b) Baselines: We benchmark *DISC* against a suite of state-of-the-art methods, categorized into two groups: (1) **Entangled Architectures**, including Transformer-based models such as Octo [38] and VQ-BeT [34], the text-aware VLA model OTTER [12], and Diffusion-based models like U-Net-based Diffusion Policy (DP) [6] and Transformer-based Diffusion (DiT) [6]; and (2) **Hypernetwork-based Architectures**, specifically HyPoGen [28] (abbreviated as H-Gen) and HyperZero [31] (H-Zero). To isolate the impact of architectural inductive biases, we train all baselines from scratch using identical visual (ResNet-18) and language (T5-small) backbones. Architectural details are provided in the appendix. Furthermore, to contextualize our performance against the current state-of-the-art, we include two large-scale pretrained models, π_0 [3] and $\pi_{0.5}$ [14], fine-tuned on LIBERO-90 as reference points.

c) Training Protocol: All experiments follow a standard Behavior Cloning (BC) paradigm. The aggregate training datasets consist of 4,500 episodes for LIBERO (50×90), 4,500 for Meta-World (100×45), and 900 for the real-world benchmark (100×9). Crucially, to isolate the impact of proposed architecture from capacity-related benefits, we standardize the model size across all trained-from-scratch methods to $\sim$30M parameters. For the reference models π_0 and $\pi_{0.5}$, we fine-tune from their official pretrained checkpoints. Complete training details are available in the appendix.

B. Multi-Task Performance (RQ1)

We evaluate whether decoupling instruction from state-conditioned control improves multi-task performance. Models are tested on the same tasks in the training set, but with novel initial configurations (unseen object positions, robot poses) to assess whether the policy generalizes based on language instructions rather than memorized scene-action associations.

TABLE I: Multi-task performance on LIBERO-90. Success rates (%) averaged over 20 episodes. *DISC* shows increasing advantages on complex tasks where task-state entanglement is most problematic. Best in **bold**, second-best underlined among methods trained from scratch. *Italicized values* denote fine-tuned results with large-scale pretraining.

Method	Easy	Medium	Long	Overall
Octo	92.3	89.0	79.3	84.7
VQ-BeT	89.5	86.7	84.3	85.9
DP	93.2	72.7	72.7	75.2
DiT	57.3	42.0	34.2	40.1
H-Zero	50.0	75.1	69.1	69.1
H-Gen	30.5	50.7	46.3	46.1
OTTER	96.4	87.9	83.2	86.6
DISC (Ours)	**97.3**	**95.3**	**92.7**	**94.3**
π_0 *(Pretrained + FT)*	*95.9*	*92.0*	*90.2*	*91.6*
$\pi_{0.5}$ *(Pretrained + FT)*	*96.4*	*97.0*	*94.4*	*95.7*

a) Results on LIBERO-90: Table I reveals a critical pattern supporting our hypothesis. Among trained-from-scratch entangled baselines, OTTER [12] is the strongest overall, reaching 86.6%, and performs particularly well on Easy tasks with 96.4%. *DISC* achieves 94.3% overall – a 7.7% absolute improvement over OTTER. When checking the gap against the best from each category, we see it correlate with task complexity: +0.9% on Easy tasks (97.3 vs. 96.4), +6.3% on Medium tasks (95.3 vs. 89.0), and +8.4% on Long-Horizon tasks (92.7 vs. 84.3). This pattern is consistent with our analysis of task-state entanglement: as task horizon grows, the opportunity for observation leakage increases – entangled policies accumulate grounding errors across sequential steps because visual shortcuts that work for early subgoals may not generalize to later ones. *DISC*'s decoupled architecture avoids this failure mode entirely, as the target policy has no mechanism to bypass its generated parameters. Remarkably, despite being trained entirely from scratch, *DISC* surpasses the large-scale pretrained model π_0 (91.6%) and remains competitive with $\pi_{0.5}$ (95.7%).

b) Complementary Failure Modes vs. $\pi_{0.5}$: A task-level comparison across all 90 LIBERO tasks reveals that *DISC* and $\pi_{0.5}$ exhibit *complementary* failure modes despite their close aggregate gap: 68 tasks are within 5%, *DISC* wins by $>5\%$ on 9 tasks, and $\pi_{0.5}$ wins by $>5\%$ on 13 tasks. *DISC*'s largest advantages appear on **multi-step tasks requiring precise language-grounded sequencing**: on "turn on the stove and put the frying pan on it" across two kitchen scenes, *DISC* achieves 92%/96% while $\pi_{0.5}$ drops to 30%/60% (**+62%/+36%** gaps), suggesting $\pi_{0.5}$ exploits scene-level visual priors rather than faithfully following the instruction. This indicates that even internet-scale pretraining does not eliminate observation leakage: action fine-tuning on limited robot data re-introduces shortcut-learning pathways through the entangled fusion of language and visual tokens. Conversely, $\pi_{0.5}$'s largest advantages appear on **fine-grained placement tasks** (e.g., "place book in front compartment"), where *DISC* scores 46–56% vs. $\pi_{0.5}$'s 75–95%. Since *DISC*

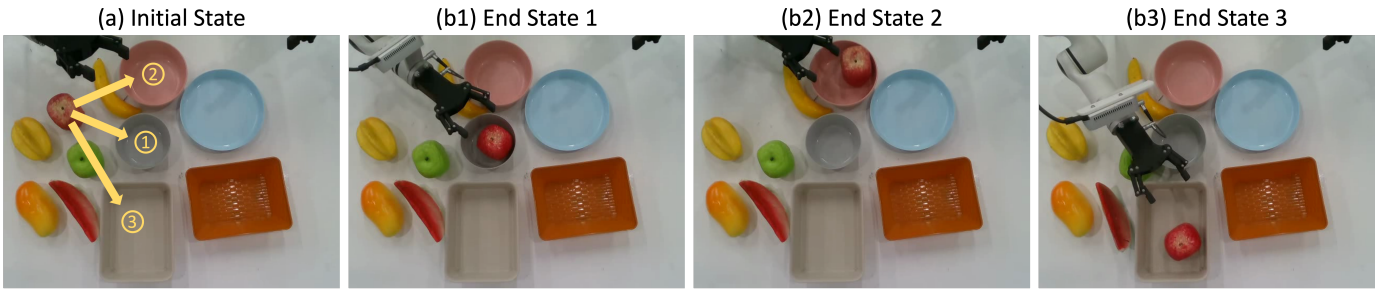

Fig. 3: **Real-World Combinatorial Benchmark Visualization.** (a) **Initial State:** The setup presents inherent visual ambiguity: the target object (e.g., Red Apple) has multiple potential destinations (indicated by arrows). (b) **Language-Conditioned Outcomes:** Starting from the *identical* initial state shown in (a), *DISC* successfully executes three distinct tasks specified by different language instructions: (b1) placing the apple into the **Small Gray Bowl**, (b2) the **Large Red Bowl**, and (b3) the **Light Gray Box**. Because the visual scene is identical in all three cases, the divergent behaviors can only arise from the language instruction – confirming that the generated policy parameters, not visual context, determine task execution.

TABLE II: Joint evaluation on LIBERO-Spatial, LIBERO-Object, and LIBERO-Goal. Success rates (%) are averaged over 50 episodes per task. π_0 and $\pi_{0.5}$ are fine-tuned from pretrained checkpoints, whereas *DISC* is trained from scratch.

Method	Spatial	Object	Goal	Overall
π_0	96.8	98.8	95.8	97.1
$\pi_{0.5}$	98.8	98.2	**98.0**	98.3
DISC	**99.2**	**99.4**	97.4	**98.7**

TABLE III: Meta-World ML45 multi-task performance by manipulation primitives. *DISC* maintains advantages across action types, achieving 92.2% overall success rate.

Task Category	Octo	VQ-BeT	DP	DiT	H-Zero	H-Gen	*DISC*
open&close	99.3	**100.0**	56.8	52.7	99.3	91.5	99.5
pick&place	83.0	69.0	19.5	19.0	81.5	66.5	**88.0**
press&pull	**93.4**	88.1	33.9	38.5	38.4	72.5	**93.4**
others	85.8	77.6	54.6	48.1	48.2	68.5	**88.7**
Overall	90.6	84.6	44.5	42.9	56.8	73.8	**92.2**

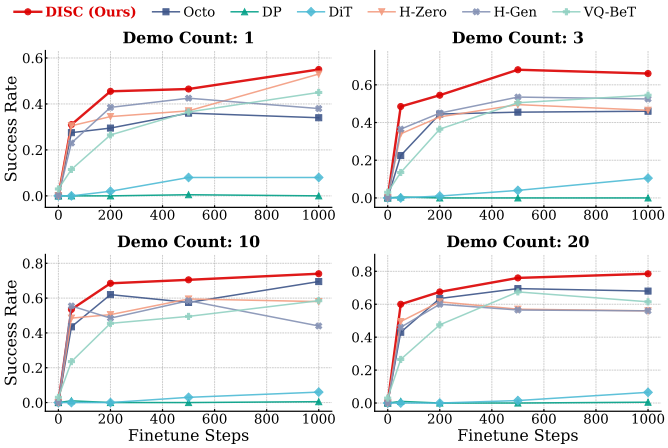

Fig. 4: Few-Shot Adaptation Efficiency on LIBERO-Spatial. The plots show success rate over 1000 fine-tuning steps, with $K \in \{1, 3, 10, 20\}$ demonstrations. *DISC* shows better sample efficiency than state-of-the-art baselines like Octo and DP.

achieves 96–100% on other book-placement tasks (left/back compartments), the gap reflects motor precision of the compact MLP target policy for specific spatial targets, rather than a language-grounding failure.

c) Broader LIBERO Suite Evaluation: We further test whether *DISC* remains competitive on the standard LIBERO-Spatial, LIBERO-Object, and LIBERO-Goal suites under a broader joint-training setting. As shown in Table II, *DISC* achieves the highest overall success rate of 98.7%, slightly exceeding both pretrained reference models despite using no large-scale pretraining. *DISC* performs best on LIBERO-Spatial (99.2%) and LIBERO-Object (99.4%), while $\pi_{0.5}$ obtains the highest score on LIBERO-Goal (98.0%). These results provide an additional cross-suite check that the decoupled architecture remains competitive beyond LIBERO-90, rather than being tailored to a single benchmark split.

d) Results on Meta-World: Meta-World results (Table III) confirm that *DISC*'s advantages generalize across benchmarks. While simple `open&close` tasks saturate near 100% for most methods, *DISC* shows clear improvements on tasks requiring precise object manipulation (`pick&place`: +5.0% over Octo) and complex action sequences (`others`: +2.9%). The consistent performance across categories demonstrates that decoupling instruction from state-conditioned control benefits diverse manipulation primitives across task categories, not just specific tasks in isolation.

C. Few-Shot Adaptation to Unseen Tasks (RQ2)

We evaluate the adaptation efficiency of *DISC* on held-out LIBERO-Spatial tasks. This experiment assesses whether the structured parameter manifold learned by our hypernetwork enables efficient transfer to novel tasks under varying data budgets ($K \in \{1, 3, 10, 20\}$ demonstrations).

a) Adaptation Protocol: Our adaptation protocol ensures a rigorous comparison across architectures. For Hypernetwork-based methods, we query the frozen hypernetwork with the new instruction to generate a policy initialization θ_π^{init}, then

fine-tune only the generated parameters. For entangled baselines, we adopt the standard LoRA [11] strategy. To maintain fairness, the trainable parameter count for LoRA adapters is matched ($\approx$ 0.74M) to that of our generated policies. All methods are optimized using the same loss and optimizer for 1,000 gradient steps. We report the success rate evaluated at key checkpoints across different fine-tuning steps, averaged over 20 evaluation episodes per task.

b) Results and Analysis: Figure 4 highlights *DISC*'s superior sample efficiency across diverse data regimes. Notably, in the **extreme few-shot setting** ($K = 1, 3$), *DISC* achieves non-trivial success rates where diffusion-based methods (DP, DiT) struggle near 0% performance. With very few demonstrations, entangled architectures lack sufficient data to disentangle task specification from state observation, causing the optimization to collapse onto spurious visual correlations. *DISC*'s decoupled architecture sidesteps this failure: because the hypernetwork has already encoded task semantics into the generated initialization, fine-tuning only needs to adjust visuomotor precision rather than learn task identity from scratch. Furthermore, *DISC* demonstrates **rapid convergence**: as seen in the $K = 10$ and $K = 20$ plots, our performance curve rises sharply within the first 200 steps, significantly outpacing baselines like Octo which exhibit a slower "warm-up" phase. This confirms that the hypernetwork provides a semantically informed initialization near the task-optimal region, transforming adaptation from a global search over an entangled parameter space into a local refinement of a compact, well-initialized policy. See the appendix for additional results.

D. Language Grounding and Structure Analysis (RQ3)

a) Semantic Grounding under Linguistic Variation: A central claim of *DISC* is that the hypernetwork learns to extract task semantics from language, not memorize surface-level token patterns. To test this, all methods are retrained under identical conditions using augmented language data: 50 paraphrased instructions per task (e.g., "pick up the red block" $\rightarrow$ "grab the crimson cube"), and then evaluated on 10 additional held-out descriptions not seen during training. This measures each architecture's ability to map diverse phrasings of the same task to functionally equivalent control behavior – a test of whether the model grounds task meaning rather than memorizes specific wording. As shown in Table IV, *DISC* achieves the highest performance on both benchmarks: 85.4% on LIBERO-90 (vs. Octo's 80.6%) and 91.8% on Meta-World (vs. Octo's 90.3%). This result follows naturally from *DISC*'s architecture. Because the hypernetwork must decode an instruction into a *complete set of target policy parameters* – the only channel through which language influences the task-specific action mapping – it is incentivized to extract the underlying task specification rather than relying on specific lexical patterns. In entangled architectures, language tokens and visual tokens share processing stages, allowing the network to form spurious associations between particular word tokens and visual features. When phrasing changes, these associations break.

TABLE IV: Semantic grounding under linguistic variation. All methods are retrained with paraphrased data and evaluated on held-out descriptions. *DISC* maintains higher success rates, indicating that the decoupled architecture grounds task meaning rather than memorizes specific phrasings.

Dataset	Octo	VQ-BeT	DP	DiT	H-Zero	H-Gen	*DISC*
LIBERO-90	80.6	76.4	34.5	29.7	66.4	45.9	**85.4**
Meta-World	90.3	85.9	49.5	23.3	79.2	83.1	**91.8**

Fig. 5: t-SNE visualization of generated policy parameters. Semantically similar tasks cluster based on functional meaning, indicating that the hypernetwork maps linguistic task specifications to a structured parameter manifold.

b) Parameter Space Visualization: Figure 5 visualizes the generated parameter manifold via t-SNE [22], revealing that the hypernetwork organizes policy parameters by functional semantics rather than visual similarity. We observe three key properties: (1) **Action-Type Separation:** A macroscopic distinction exists between *articulated object manipulation* (Tasks 1–8, central cluster) and *pick-and-place* tasks (Tasks 10–24, periphery), indicating the capture of high-level control primitives in the generated weights. (2) **Functional Grouping:** The central cluster further separates tasks by object affordance, distinguishing Cabinet/Drawer interactions (Tasks 5–8) from Stove and Microwave operations. (3) **Semantic Continuity:** Inverse operations, such as Turn On/Off Stove (Tasks 1–2) and Open/Close Microwave (Tasks 3–4), are mapped as nearest neighbors. This topology confirms that the hypernetwork has learned a mapping where semantic proximity in language corresponds to geometric proximity in the weight manifold. This structure directly explains the few-shot adaptation advantages observed in Section V-C: novel tasks with semantically similar instructions receive nearby initializations, reducing the distance that fine-tuning must traverse.

E. Real-World Deployment (RQ4)

The simulation experiments establish *DISC* 's advantages under controlled conditions. We now evaluate whether the decoupling principle transfers to physical robot settings, where richer visual correlations, increased perceptual noise, and narrower grounding margins exacerbate observation leakage.

We evaluate *DISC* on the combinatorial benchmark described in Section V-A, which provides a direct test of observation leakage on a physical robot. The 9 tasks are formed

TABLE V: Real-world performance on 9 combinatorial tasks. The setup involves picking one of 3 target objects into one of 3 containers in a cluttered scene where visual context is shared across all tasks. Success rates (%) averaged over 30 episodes. (Abbr: R.A.=Red Apple, G.A.=Green Apple, W.M.=Watermelon; S.G.=Small Gray, L.R.=Large Red, L.G.=Light Gray).

ID	Instruction (Obj → Container)	H-Zero	DP	Octo	*DISC*
01	R.A. → S.G. Bowl	73.3	60.0	73.3	**96.7**
02	R.A. → L.R. Bowl	10.0	23.3	53.3	**83.3**
03	R.A. → L.G. Box	13.3	33.3	70.0	**90.0**
04	G.A. → S.G. Bowl	3.3	43.3	80.0	**93.3**
05	G.A. → L.R. Bowl	6.7	56.7	**73.3**	53.3
06	G.A. → L.G. Box	0.0	76.7	83.3	**100.0**
07	W.M. → S.G. Bowl	36.7	53.3	**90.0**	**90.0**
08	W.M. → L.R. Bowl	16.7	**96.7**	93.3	86.7
09	W.M. → L.G. Box	26.7	66.7	**96.7**	83.3
	Average Success Rate (%)	18.5	57.0	78.5	**86.4**

by the Cartesian product of 3 distinct objects (Red Apple, Green Apple, Watermelon) and 3 target containers (Small Gray Bowl, Large Red Bowl, Light Gray Box), all presented simultaneously in a cluttered workspace with distractors. This setting is particularly revealing: because the same objects and containers are present in every task, visual context is nearly identical across all 9 tasks. A policy that has learned scene-to-action mappings through observation leakage will systematically confuse tasks that share visual context, while a policy whose behavior is determined by language-generated policy parameters will distinguish them correctly.

As shown in Table V, *DISC* achieves the highest average success rate, outperforming all baselines by clear margins. Qualitative analysis reveals distinct failure modes that align with our theoretical framework. HyperZero frequently fails due to *spatial precision errors* (e.g., "air grasps"), indicating that without the iterative refinement stage, the generated parameters lack sufficient fidelity for precise physical control – validating the architectural contribution of our optimization-inspired refinement module. Entangled baselines exhibit a qualitatively different failure: Octo and DP often achieve correct localization but suffer from *orientation misalignment*, suggesting that their shared representations have learned coarse scene-to-action associations rather than precise language-driven control. This is the signature of observation leakage: the policy partially succeeds by mapping visual context to approximate actions, but fails when the task requires fine-grained distinctions that only language can resolve. Notably, Octo performs well on visually distinctive tasks (e.g., Task 09, Watermelon → Light Gray Box) but degrades sharply on tasks requiring subtle container distinctions (e.g., Task 02, Red Apple → Large Red Bowl vs. Small Gray Bowl) – precisely the failure mode predicted when visual context rather than language determines behavior. *DISC* mitigates both failure modes: the decoupled architecture ensures that task identity is encoded entirely in the generated parameters, eliminating observation leakage, while the refinement module provides the parameter precision needed for reliable physical execution.

a) Real-World Failure Analysis: *DISC*'s only sub-baseline result is on Task 05 (G.A. → L.R. Bowl, 53.3%), where the gripper tends to grasp the apple at suboptimal positions, leading to drops during transport. Notably, *DISC* achieves 93.3% and 100% on the other two Green Apple tasks (04, 06) and 83.3% on the same target container with a different object (Task 02), confirming that the failure stems from execution-level imprecision rather than a language-grounding error. This indicates that the current bottleneck is the compact MLP target policy, which can be insufficient for physically demanding placements. Generating more expressive target policies (e.g., diffusion- or flow-based) is a promising direction for further improving precision and robustness.

Summary. Across simulation and real-world settings, our experiments confirm that decoupling instruction from state-conditioned control yields consistent improvements in multi-task performance, few-shot adaptation, semantic grounding, and physical deployment. These advantages scale with task complexity and visual ambiguity – precisely the conditions under which task-state entanglement is most damaging, and where *DISC*'s architectural guarantees are most valuable.

VI. Conclusion

The central finding of this work is that *where* language enters a policy architecture matters more than *how* it is encoded. Task-state entanglement is not merely a training problem solvable with more data or larger models – it is a structural property of shared-parameter designs. *DISC* demonstrates that generating task-specific policy parameters from language blocks the shared action-mapping pathway behind this entanglement, recovering instruction grounding, generalization, and few-shot adaptation beyond what scaling entangled architectures has achieved. The learned parameter manifold organizes tasks as geometric neighbors in weight space without explicit supervision, suggesting that decoupling unlocks structure entangled architectures cannot express. In practice, *DISC* is most attractive when reliable grounding, efficient control, and adaptation from limited robot data are primary requirements. Large pretrained VLAs remain complementary: they are well suited to data-rich and compute-rich settings where broad pretraining can be fully exploited.

Limitations. The main practical trade-off of *DISC* lies on the training side, where it can require more memory and longer optimization than comparable static policies. Under-specified instructions may also surface uncertainty rather than being silently resolved through observation leakage, motivating future work on more efficient refinement, uncertainty-aware clarification, and broader multi-modal extensions.

Acknowledgments

This work is supported by the Early Career Scheme of the Research Grants Council (RGC) grant # 27207224, the HKU-100 Award, a donation from the Musketeers Foundation, and in part by the JC STEM Lab of Autonomous Intelligent Systems funded by The Hong Kong Jockey Club Charities Trust.

References

[1] Iñigo Alonso, Gorka Azkune, Ander Salaberria, Jeremy Barnes, and Oier Lopez de Lacalle. Vision-language models struggle to align entities across modalities. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18846–18862, Vienna, Austria, July 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.findings-acl.965.

[2] Jacob Beck, Matthew Thomas Jackson, Risto Vuorio, and Shimon Whiteson. Hypernetworks in meta-reinforcement learning. In *Conference on Robot Learning*, pages 1478–1487. PMLR, 2023.

[3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[5] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

[6] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[8] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[9] Siddhant Haldar, Zhuoran Peng, and Lerrel Pinto. Baku: An efficient transformer for multi-task policy learning. *Advances in Neural Information Processing Systems*, 37: 141208–141239, 2024.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[12] Huang Huang, Fangchen Liu, Letian Fu, Tingfan Wu, Mustafa Mukadam, Jitendra Malik, Ken Goldberg, and Pieter Abbeel. Otter: A vision-language-action model with text-aware feature extraction. *arXiv preprint arXiv:2503.03734*, 2025.

[13] Yizhou Huang, Kevin Xie, Homanga Bharadhwaj, and Florian Shkurti. Continual model-based reinforcement learning with hypernetworks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 799–805. IEEE, 2021.

[14] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. pi_0.5: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[15] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024.

[16] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[17] Chengmeng Li, Junjie Wen, Yaxin Peng, Yan Peng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models. *IEEE Robotics and Automation Letters*, 11(3):2506–2513, 2026.

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

[19] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[20] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[21] Zhining Liu, Ziyi Chen, Hui Liu, Chen Luo, Xianfeng Tang, Suhang Wang, Joy Zeng, Zhenwei Dai, Zhan Shi, Tianxin Wei, Benoit Dumoulin, and Hanghang Tong. Seeing but not believing: Probing the disconnect between visual attention and answer correctness in VLMs. *arXiv preprint arXiv:2510.17771*, 2025.

[22] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[23] Ankan Mullick, Saransh Sharma, Abhik Jana, and Pawan Goyal. Text takes over: A study of modality bias in multimodal intent detection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24040–24070, Miami, Florida, USA, November 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.emnlp-main.1226.

[24] Anupam Pani and Yanchao Yang. Gaze-vlm: Bridging gaze and vlms through attention regularization for egocentric understanding. *arXiv preprint arXiv:2510.21356*, 2025.

[25] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[26] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[27] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[28] Hanxiang Ren, Li Sun, Xulong Wang, Pei Zhou, Zewen Wu, Siyan Dong, Difan Zou, Youyi Zheng, and Yanchao Yang. Hypogen: Optimization-biased hypernetworks for generalizable policy generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[29] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023.

[30] Moritz Reuss, Jyothish Pari, Pulkit Agrawal, and Rudolf Lioutikov. Efficient diffusion transformer policies with mixture of expert denoisers for multitask learning. *arXiv preprint arXiv:2412.12953*, 2024.

[31] Sahand Rezaei-Shoshtari, Charlotte Morissette, Francois R Hogan, Gregory Dudek, and David Meger. Hypernetworks for zero-shot transfer in reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9579–9587, 2023.

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[33] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR, 2016.

[34] Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Arty Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

[35] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[36] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[37] Jiaqi Tang, Yinsong Xu, Yang Liu, and Qingchao Chen. Shaping initial state prevents modality competition in multi-modal fusion: A two-stage scheduling framework via fast partial information decomposition. *arXiv preprint arXiv:2509.20840*, 2025.

[38] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

[41] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.

[42] Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint arXiv:2507.04447*, 2025.

[43] Pei Zhou and Yanchao Yang. Maxmi: A maximal mutual information criterion for manipulation concept discovery. In *European Conference on Computer Vision*, pages 88–105. Springer, 2024.

[44] Pei Zhou, Wanting Yao, Qian Luo, Xunzhe Zhou, and Yanchao Yang. Hyper-goalnet: Goal-conditioned manipulation policy learning with hypernetworks. *Advances in Neural Information Processing Systems*, 38:83438–83469, 2026.

[45] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.