Cue the Flow: Steering Flow-Matching Policies for Open-World Delivery Manipulation

Anonymous Author(s)
Affiliation
Address
email

Abstract: Open-world delivery requires mobile manipulators to follow free-form user instructions and manipulate potentially novel objects. Existing dual-system approaches use high-level grounding models to convert language into grounded visual prompts, but their low-level controllers can remain brittle under noisy perception, dynamic scenes, and contact-rich interactions. We instead use a pretrained flow-matching vision-language-action model as the low-level control interface, leveraging its reactivity and robustness to environmental changes while treating the grounding output as a spatial cue for policy steering. Our key insight is that the pretrained VLA already provides a strong manipulation prior, while the spatial cue supplies the missing target information needed to guide actions under novel language–object mappings. Concretely, we introduce a lightweight cue-conditioned adapter. The adapter is first trained with contrastive objectives to produce salient and spatially discriminative cue representations, and is then supervised to predict a diagonal affine transformation over the generated action chunk, aligning policy steering with the cued target. Across tabletop and mobile-base settings, our method improves instruction following and manipulation success on both in-domain and out-of-domain objects, achieving up to near $2\times$ improvement in average task success rate with negligible inference overhead.

Keywords: Policy Steering, Mobile Manipulation; Generalization

1 Introduction

Consider a delivery robot receiving the request: "Give me the takeout bag with the name Joseph on it." We focus on the **manipulation phase** of this task, where the robot must turn an open-ended instruction into a physical action: parsing the language, grounding the referred object in a potentially cluttered scene, and executing a reliable trajectory toward it. While the underlying manipulation skill may be simple, such as pick-and-place, learning a practical policy for open-world delivery remains difficult. As shown in Figure 1(a), the core challenge is a substantial train–deployment mismatch: the policy is trained in fixed environments with a limited vocabulary of objects and instructions, yet deployed on a mobile robot in open-world settings with novel objects, unseen referring expressions, and variable scene configurations.

Large-scale pretrained robot policies [1, 2, 3, 4] have substantially improved policy robustness under visual perturbations [5], reducing the brittleness caused by changes in scene appearance and environment layout. Yet a central challenge remains: how to extend a policy to novel language–object mappings while preserving its previously learned manipulation capability. One promising direction is a dual-system design that decouples explicit grounding from robot control [6, 7, 8, 9]. The high-level grounding module, or System 2, interprets the instruction and converts it into spatial abstractions, often represented as visual prompts [10, 11]. The low-level controller, or System 1, then conditions on these prompts to generate executable robot actions.

Submitted to the 10th Conference on Robot Learning (CoRL 2026). Do not distribute.

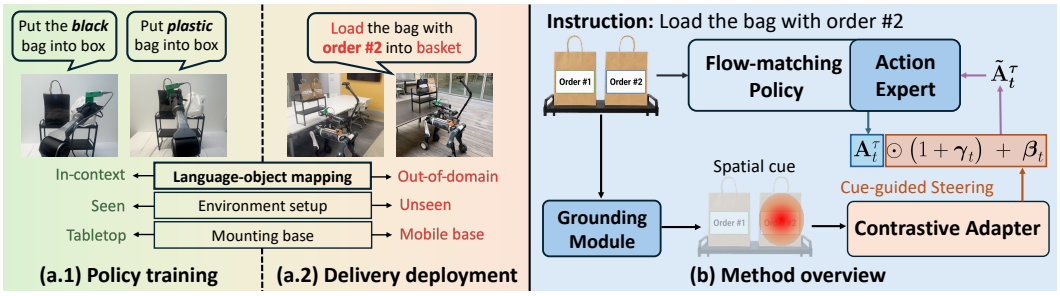


Figure 1: **(a.1)** The policy is trained on in-context demonstrations collected in a fixed environment. **(a.2)** In real-world delivery, however, the robot must execute manipulation under changing environments and novel language–object mappings. **(b)** We focus on improving policy generalization to these novel mappings. A grounding module converts the open-ended instruction into a spatial cue, which conditions a contrastive adapter to predict affine transformations over the frozen flow-matching policy's action output, steering the generated action chunk toward the target object.

Paradigms	VLA [3]	Dual + low-level [7]	**Dual + VLA (ours)**
Learned control	✓	✗	✓
Precise perception	✓	✗	✓
Explicit grounding	✗	✓	✓
OOD generalization	Limited	Strong	Strong

Table 1: **Comparison of paradigm designs.**

However, many existing dual-system methods still depend on optimization-based or geometric motion generation [12, 7, 13]. Such controllers can be fragile in open-world delivery settings: perception may be noisy, the target object may move during execution, and a fixed plan may not recover from grounding or localization errors. This motivates replacing the geometric low-level controller with a pretrained robot policy, especially a flow-matching vision-language-action model [14, 15, 16]. Since these models are trained on diverse robotic demonstrations, they provide learned priors for object interaction, contact-rich manipulation, and reactive correction. They therefore offer a practical low-level control interface for the dual-system design, while preserving the explicit grounding ability of the high-level module. Table 1 compares these paradigms.

Yet integrating flow-matching VLAs into a dual-system design is not straightforward. The visual prompt produced by System 2 must be converted into a form that can reliably steer a policy trained to act from its own visual-language representations. Prior work [9, 17] explores full policy training with visual prompts as additional inputs. While effective in controlled settings, this strategy still depends on the VLA's internal spatial reasoning ability, which is known to be limited [18, 19]. More critically, input-level prompting does not specify how the prompt should modulate the action generation process. After being encoded with the rest of the visual observation, the prompt is only implicitly represented, without a direct mechanism that encourages the generated trajectory to align with the System 2 guidance. The key challenge is therefore not merely how to provide a visual prompt to a VLA, but how to turn that grounded signal into actionable guidance during flow-based action generation.

In this paper, we address this challenge by treating the grounding output as a spatial cue that directly guides action generation. Our key insight is that a pretrained flow-matching VLA already knows how to manipulate: it can generate feasible trajectories for objects in the workspace, but may not reliably determine which object to act on under unseen language–object mappings. Thus, the missing component is not a new manipulation policy, but a mechanism for specifying the intended target. We therefore keep the VLA frozen and introduce a lightweight cue-conditioned adapter that uses the System 2 generated spatial cue to steer the frozen policy's action output. The adapter first learns to encode the explicit cue into a salient and spatially discriminative representation, using contrastive objectives to discourage reliance on scene-specific shortcuts. It then conditions on this cue representation to predict a diagonal affine transformation over the generated action chunk, shifting the frozen policy's output toward the cued object while preserving its learned manipulation prior. Figure 1(b) provides an overview of the full pipeline.

Our contributions are threefold. First, we analyze the policy-learning challenges posed by open-world delivery and propose a dual-system framework that combines explicit language–object grounding with learned low-level action generation. Second, we develop a lightweight cue-conditioned adapter for frozen flow-matching VLAs, which steers generated action chunks through a diagonal affine

transformation while avoiding full policy fine-tuning. Third, we validate the proposed method across three in-domain and out-of-domain object settings. Our method consistently outperforms all baselines, achieving relative gains of 86% and 44% over the best-performing baselines in average task success rate under tabletop and mobile-base settings, respectively.

2 Related Work

Flow-matching VLAs. A central goal in robot learning is to build generalist policies that can perform diverse tasks and transfer across environments, objects, and embodiments. Vision-language-action (VLA) models [1, 2, 15, 14] address this goal by mapping visual observations and language instructions directly to robot actions, often after training on large-scale robot and human demonstration data. Recent flow-matching VLAs [15, 14, 16] further model action generation as a learned continuous flow conditioned on fused visual-language features, allowing them to represent complex and potentially multimodal action distributions [20]. Although flow-matching VLAs have begun to be used within dual-system architectures [21, 22], how to effectively steer their action generation toward novel language–object mappings remains underexplored.

Policy steering. Policy steering improves pretrained generative policies by modifying their sampling or refinement process while keeping the base policy fixed [23]. Existing methods typically guide generated actions with external objectives or auxiliary models, steering outputs toward task, safety, or user-specified constraints. Model-predictive refinement methods [24, 25] use learned dynamics models to improve task performance or enforce safety, while human-in-the-loop approaches [26, 27, 28] refine actions using user-provided subgoals, corrections, or preferences. Classifier- and dynamics-guided methods [29, 30] further leverage latent visual dynamics models [31, 32] to bias actions toward desired outcomes. Unlike prior steering methods that primarily use external signals to refine in-domain behavior, our method treats spatial cues as target-specification signals and learns cue-conditioned transformations over frozen VLA action chunks, enabling better generalization to out-of-domain language–object mappings.

Visual prompting for robotic policies. Explicit visual representations provide a useful interface for guiding robotic manipulation. One line of work uses visual marks to connect a vision-language model (VLM) planner with a low-level controller [7, 6, 8], where the controller translates visual predictions into executable actions. However, such controllers can be brittle in open-world settings where perception is noisy and reactive control is needed. Another line of work treats actions as language [33, 34], using the VLM to directly output executable robot actions, but these methods depend heavily on the grounding precision and reasoning reliability of the VLM. Other approaches [13, 9, 35] inject visual prompts into the VLA input stream, yet the prompt's influence on action generation remains implicit. In contrast, our method uses a spatial cue separated from the input image as an explicit target-specification signal, steering a frozen VLA by directly modulating its generated action chunk.

3 Method

We present the problem setup and detail method design in this section. Section 3.1 formulates generalization to novel language–object mappings as a policy-steering problem. Section 3.2 introduces the overall dual-system framework. Section 3.3 describes the spatial-cue-conditioned adapter for steering the frozen policy. Section 3.4 provides the implementation details.

3.1 Problem Definition

Consider two deployments of the same frozen flow-matching policy. In the first, both the instruction and the referred object fall within the training distribution, and the policy can execute the task successfully. In the second, the target object is replaced by a novel object and the instruction is changed accordingly. The underlying motor behavior may remain the same, such as reaching, grasping, and placing, but the policy must now associate the instruction and visual scene with an unseen target. The central challenge is therefore not to acquire a new manipulation skill, but to

redirect an existing manipulation prior toward the intended object during action generation. We formulate this as a policy-steering problem: given a frozen flow-matching VLA and an explicit spatial cue produced by a grounding module, the goal is to modulate the generated action chunk so that it follows the cued target while preserving the base policy's learned manipulation capability.

3.2 Dual-System Design

Flow-matching VLAs generate action chunks using an action expert conditioned on features from a vision-language backbone. While this end-to-end grounding is effective for in-distribution tasks, it can become brittle when the policy encounters novel instructions referring to unseen objects. To reduce this uncertainty, we adopt a dual-system design that separates language–object grounding from action generation. The high-level grounding module converts diverse free-form instructions into a unified spatial cue, allowing the downstream policy to be steered by explicit target information rather than relying solely on implicit language grounding.

System 2: Grounding module. System 2 converts open-ended user instructions into a structured **spatial cue** for the downstream VLA, specifying the 2D target location in the workspace camera view. We first use a grounding model to predict spatial references for the target object, typically in the form of bounding boxes. An external segmentation model [36, 37] then converts these references into a pixel-level target mask. From this mask, we compute the object centroid and construct a Gaussian heatmap $\mathbf{h}_t$. This heatmap is not injected as an image-level visual prompt but used as an explicit spatial cue. All components in System 2 are frozen and used for zero-shot inference.

System 1: VLA as controller interface. To preserve the pretrained manipulation capability of the VLA while enabling it to be steered by the spatial cue from System 2, we freeze the VLA policy and attach a lightweight adapter. The adapter predicts a cue-conditioned diagonal affine transformation that modulates the action chunk produced by the frozen policy during flow integration. Specifically, let $\mathbf{A}_t^\tau$ denote the generated intermediate action chunk at integration time $\tau \in [0, 1]$. The adapter predicts two diagonal modulation terms, $\boldsymbol{\gamma}_t$ and $\boldsymbol{\beta}_t$, and computes the steered action chunk as

$$\tilde{\mathbf{A}}_t^\tau \;=\; \mathbf{A}_t^\tau \odot \big(1 + \boldsymbol{\gamma}_t\big) \;+\; \boldsymbol{\beta}_t, \tag{1}$$

where $\odot$ denotes element-wise multiplication. The scaling term $\boldsymbol{\gamma}_t$ reweights the frozen policy's action output, while the shift term $\boldsymbol{\beta}_t$ provides an additive correction. Together, they define a transformation that steers the generated action chunk toward the target object.

3.3 Contrastive Adapter

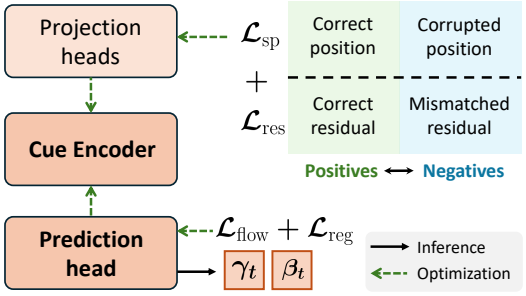


Figure 2: **Illustration of adapter learning.**

Learning an effective adapter is challenging because the spatial cue is aligned with the target object in the observation. Without additional constraints, the adapter may exploit shortcut correlations from in-domain training scenes or memorize scene-specific visual patterns, rather than using the cue as the source of target specification. To strengthen the correspondence between cue location and action transformation, we design the adapter with two components: a cue encoder that learns salient and spatially discriminative features from the explicit cue, and a prediction head that maps these features to affine transformation parameters for steering the frozen policy output. Figure 2 illustrates the adapter architecture and training pipeline.

Cue encoder learning. The spatial cue specifies the target location but does not by itself encode object semantics or task context. We therefore train a cue encoder $\mathcal{E}$ to transform the cue $\mathbf{h}_t$ into a representation suitable for action steering. The learned representation is encouraged to satisfy two properties: **spatial discrimination**, preserving where the target lies relative to the action representation, and **cue salience**, emphasizing cue-relevant features over incidental visual patterns.

Let $\mathcal{E}$ denote the cue encoder. To make the cue representation spatially discriminative, we require the action representation to align more closely with its corresponding spatial cue than with spatially shifted versions of the same cue. We first encode the Gaussian heatmap and project it into a cue representation, $\mathbf{z}_c = \mathcal{P}_c(\mathcal{E}(\mathbf{h}_t))$, where $\mathcal{P}_c$ is a projection head. We also construct an action representation, $\mathbf{z}_a = \mathcal{P}_a(\mathbf{a}_t, \mathcal{E}(\mathbf{h}_t))$, where $\mathbf{a}_t$ denotes the action-expert feature and $\mathcal{P}_a$ is the corresponding projection head. For each positive cue representation $\mathbf{z}_c^i$, we construct a set of *spatial-shift negatives* $\{\mathbf{z}_{c,k}^-\}_{k=1}^K$ by perturbing the heatmap peak on the patch grid while keeping the underlying scene fixed. These negatives retain the same visual context but specify different target locations, forcing the cue representation to capture spatial position rather than appearance context. The spatial contrastive loss for sample i is

$$\mathcal{L}_{\text{sp}}^i = -\log \frac{\exp(\mathbf{z}_a^i \cdot \mathbf{z}_c^i/\tau)}{\sum_{j=1}^B \exp(\mathbf{z}_a^i \cdot \mathbf{z}_c^j/\tau) + \sum_{k=1}^K \exp(\mathbf{z}_a^i \cdot \mathbf{z}_{c,k}^-/\tau)}, \tag{2}$$

where B is the batch size and τ is the softmax temperature. This objective contrasts the correct cue location against both in-batch cues and spatially corrupted cues, encouraging the encoder to learn features that are sensitive to the target location rather than scene-level appearance.

Since the adapter also conditions on action-related features, it may learn shortcuts that are not driven by the spatial cue itself. We therefore introduce a residual cue representation that isolates the effect of the cue. For each sample, we compute two encodings: one from the real heatmap, $\mathcal{E}(\mathbf{h}_t)$, and one from a zero heatmap, $\mathcal{E}(\mathbf{0})$. Their difference captures the cue-induced residual, which is then projected into the same representation space as the action feature: $\mathbf{z}_{\text{res}} = \mathcal{P}_\delta\big(\mathcal{E}(\mathbf{h}_t) - \mathcal{E}(\mathbf{0})\big)$, where $\mathcal{P}_\delta$ is a projection head. This reinforce the adapter to learn cue-relevant action transformations. Therefore, for each sample i, we introduce the residual contrastive loss:

$$\mathcal{L}_{\text{res}}^i = -\log \frac{\exp(\mathbf{z}_a^i \cdot \mathbf{z}_{\text{res}}^i/\tau)}{\sum_{j=1}^B \exp(\mathbf{z}_a^i \cdot \mathbf{z}_{\text{res}}^j/\tau)}. \tag{3}$$

Affine transformation learning. Given the encoded cue, a prediction head outputs the modulation terms $\boldsymbol{\gamma}_t$ and $\boldsymbol{\beta}_t$, which define the diagonal affine transformation in Eq. (1). We train this prediction head under the same flow-matching supervision as the base policy, so that the adapter learns to steer the frozen policy's action chunks while preserving the pretrained action distribution. To discourage overly large corrections, we regularize the magnitude of the modulation terms:

$$\mathcal{L}_{\text{reg}} = \|\boldsymbol{\gamma}_t\|_2^2 + \|\boldsymbol{\beta}_t\|_2^2. \tag{4}$$

The full training objective is therefore

$$\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda_{\text{sp}}\,\mathcal{L}_{\text{sp}} + \lambda_{\text{res}}\,\mathcal{L}_{\text{res}} + \lambda_{\text{reg}}\,\mathcal{L}_{\text{reg}}, \tag{5}$$

where $\mathcal{L}_{\text{flow}}$ trains the affine transformation under the base flow-matching supervision, while the auxiliary terms align the transformation with the spatial cue and regularize its magnitude to avoid corrupting the pretrained action distribution.

3.4 Implementation Details

Training. We train the adapter on the same demonstrations used to learn the flow-matching policy, supplemented with ground-truth spatial cues for target-location supervision. The adapter is implemented as a lightweight sidecar to the action expert, modulating generated action chunks while keeping the base policy frozen. We also evaluated LoRA [38] on the action expert as an alternative for learning cue-conditioned action transformations. In our low-data setting, LoRA tended to overfit to the training cue–action pairs, likely because its updates are directly coupled to the action-expert representations, and did not yield robust cue-steering behavior.

Deployment. During deployment, the adapter uses the same residual cue formulation as in training: it takes both the System 2 heatmap $\mathbf{h}_t$ and a zero heatmap $\mathbf{0}$, and predicts the modulation terms from their residual cue feature. For efficiency, the adapter adds about 0.02% parameters relative to the base policy and introduces negligible latency. System 2 is also invoked once at the beginning of each task rollout, unless substantial scene changes require the spatial cue to be recomputed.

Method	Tabletop				Mobile Base			
	Setting I	Setting II	Setting III	Average	Setting I	Setting II	Setting III	Average
Base [14]	70.8	45.8	0.0	38.9	33.3	50.0	8.3	30.5
Base-L [14]	37.5	8.3	29.2	25.0	8.3	0.0	8.3	5.5
MOKA [7]	20.8	25.0	8.3	18.0	16.7	8.3	0.0	8.3
VP-VLA [9]	58.3	45.8	4.2	36.1	58.3	66.7	8.3	44.4
Ours	**83.3**	**66.7**	**66.7**	**72.2**	**83.3**	**66.7**	**41.7**	**63.9**

Table 2: **Success rates (%) comparison under tabletop and mobile-base conditions across three settings.** (I) seen objects with paraphrased instructions, (II) unseen objects with in-domain instructions, and (III) unseen objects with novel instructions. Best results in each column are **bolded**.

4 Experiments

4.1 Experimental Setup

Pipeline. For policy adaptation, we collect 80 demonstration trajectories using two training objects, a black bag and a plastic bag, without any label tags. We fine-tune $\pi_{0.5}$ [14] on this dataset with a standard training recipe and use the resulting model as the base policy for our method. During deployment, we use Qwen3-VL-2B [39] and SAM2.1 [36, 37] as the grounding module. These model choices allow the full pipeline to run simultaneously on a single RTX 5080 GPU, with the VLA operating continuously at 15,Hz. Experiments are conducted with a 6-DoF robotic arm and a wheeled dog robot, which provides vibration robustness and flexible hardware integration.

Evaluation. We evaluate under three settings. **I. Seen objects with paraphrased language** uses objects from the fine-tuning dataset with perturbed prompts. **II. Unseen objects with in-domain language** uses target objects absent from the collected trajectories, while keeping object names within the pretrained $\pi_{0.5}$ vocabulary, such as "red bag". **III. Unseen objects with novel language** further introduces novel referring expressions, such as "bag with order number 1", representing the most challenging setting. For each task, we curate six prompts and two spatial configurations. Each prompt is generated from a compositional template, such as "<grasp-action> <target-object> and <place-action> into the box," where the grasping action, target object, and placing action vary across trials. The placement location is fixed, since delivery tasks can use a deterministic handover or drop-off location. Because all training demonstrations are collected on tabletop, deployment on the mobile base introduces an additional shift in viewpoint and scene geometry. Task success is measured as a binary outcome indicating whether the robot correctly completes the user instruction. The full set of evaluation tasks, prompts, and spatial configurations is provided in the supplementary material.

Baselines. We compare our method against the following baselines:

- **Base** [14]. We fine-tune pretrained $\pi_{0.5}$ on our demonstrations using one fixed language prompt per task, corresponding to a standard single-VLA setting.
- **Base-L**. We fine-tune pretrained $\pi_{0.5}$ with multiple language prompts per trajectory, using the same paraphrased prompts as in Setting I. This aligns the training prompt distribution more closely with the evaluation prompts and tests whether language augmentation improves generalization.
- **MOKA** [7]. MOKA represents a dual-system approach with a high-level planner and a geometric low-level controller. It relies on the planner to produce accurate 2D keypoints and waypoints as affordance representations, which are then deprojected into 3D space for controller execution.
- **VP-VLA** [9]. VP-VLA is also a dual-system approach that uses a VLA as the controller interface. Unlike our method, it injects the generated visual prompt as additional policy input and trains the VLA with an auxiliary grounding objective to focus on the prompted region.

4.2 Results

Quantitative results. Table 2 summarizes the quantitative results. Moving the robot arm from a stable tabletop setup to a mobile base degrades performance for all methods, due to changes

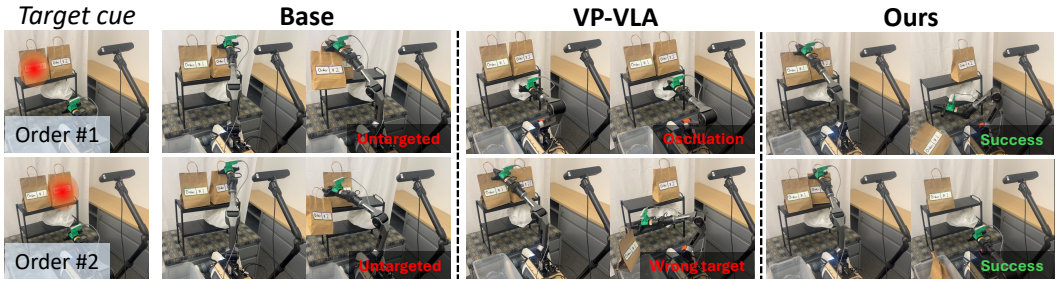


Figure 3: **Qualitative rollouts under the unseen-object with novel-language setting (Setting III).** The leftmost column shows the System 2 spatial cue, where the red heatmap indicates the target object. For each method, two temporal keyframes are shown. The Base policy produces untargeted behavior, while VP-VLA either oscillates or reaches the wrong object. Our method follows the spatial cue and successfully manipulates the specified target in both task examples.

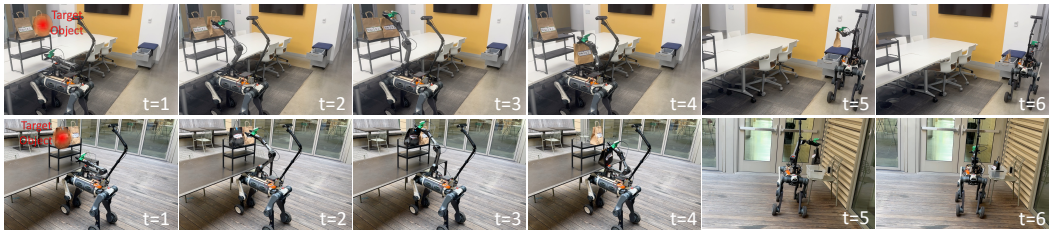


Figure 4: **Extension to a complete delivery pipeline.** We show example rollouts that combine manual navigation with autonomous manipulation. The red heatmap indicates the correctly grounded target object. After the mobile base is manually navigated to the workspace, our policy performs autonomous cue-conditioned manipulation to complete the task.

in viewpoint and base instability. resembling real-world delivery and often producing near-zero success for several baselines. The Base policy performs well on in-domain tasks and shows some generalization to object-level out-of-domain cases. We attribute this to the pretrained VLA prior, where the relevant visual-language patterns are likely represented in large-scale pretraining data. However, this prior is insufficient when both the target object and the referring expression are out of distribution. For Base-L, adding diverse language prompts during fine-tuning introduces a tradeoff: it reduces in-domain success but improves over Base on object-language-level out-of-domain tasks. Yet, we observe that this gain mainly comes from a biased motion trajectory that occasionally reaches the correct target, rather than from reliable language-conditioned grounding.

For the dual-system baselines, MOKA fails when the System 2 planner produces inaccurate visual prompts. This reflects the difficulty of generating dense spatial guidance for the full execution trajectory from a high-level planner alone. In our setup, the challenge is further amplified by the camera configuration: the forward-facing camera observes a relatively large workspace with the robot also in view, increasing perspective variation and visual clutter, and making accurate localization more difficult. VP-VLA performs comparably to the Base policy in Settings I and II, but still struggles in Setting III. This suggests that auxiliary grounding supervision alone does not reliably align the generated action trajectory with the visual prompt under stronger distribution shifts. In contrast, our method achieves better performance across all settings. We attribute this improvement to two design choices: our framework requires only sparse spatial cues from System 2, which are easier to obtain than full trajectory-level reasoning; and the learned adapter uses these cues to steer the frozen policy output while preserving the pretrained VLA prior and fine-tuned manipulation skill.

Qualitative results. Figure 3 compares qualitative rollouts in the unseen-object with novel-language setting on the mobile base. The Base policy produces untargeted trajectories, often reaching toward both bags on the shelf. VP-VLA shows stronger visual conditioning, but still lacks stable cue–action alignment, leading to oscillatory behavior or reaching toward the wrong object. This indicates that visual-prompt conditioning alone provides limited improvement under this stronger distribution shift and may also degrade manipulation stability. In contrast, our method consistently guides the frozen policy toward the specified target and enables successful manipulation. Figure 4 further demonstrates

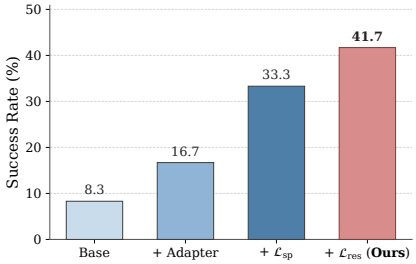


Figure 5: **Effectiveness of components.**

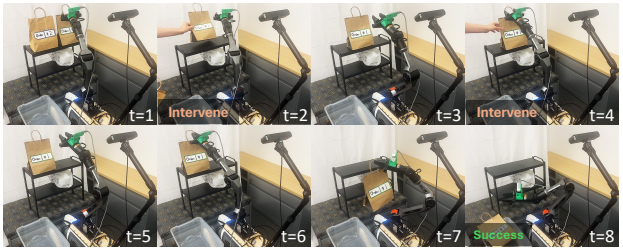


Figure 6: **Reactivity to dynamically moving objects.**

that the cue-conditioned manipulation skill transfers across diverse scenes. The policy remains robust to environmental changes during execution and can be integrated with mobile-base navigation. Full video demonstrations and policy execution details are provided in the supplementary material.

4.3 Analysis

Component effectiveness. Figure 5 evaluates the effectiveness of the proposed learning components by comparing their performance under Setting III. Training the adapter only with the flow-matching loss and spatial cue features yields limited gains, indicating that cue conditioning alone is insufficient. Adding the spatial and residual contrastive losses consistently improves performance, confirming the importance of learning cue representations that are both spatially discriminative and salient.

Effect of spatial cue format. We further study the effect of different spatial cue formats. In addition to the soft Gaussian heatmap, we evaluate a binary target mask, defined as a segmentation silhouette at patch resolution, and an axis-aligned bounding box, which captures coarse object extent. The Gaussian heatmap and binary mask yield similar behavior and performance, indicating that localized target cues are sufficient for effective steering. In contrast, the bounding box performs worse, likely because its coarse boundaries can direct the robot toward invalid or non-manipulable object regions. This suggests that fine-grained spatial cues are more effective than the coarse box-level one.

Reactivity to dynamically changing environments. Figure 6 demonstrates the benefit of using a large-scale VLA as the System 1 controller by perturbing the target object's position during execution. With the spatial cue grounded only once at the beginning of the rollout, our method still reacts to environmental changes without re-querying the high-level planner. This reactivity is inherited from the underlying VLA and preserved by our adapter, highlighting an advantage over conventional geometric controllers that rely on fixed trajectories or repeated replanning.

5 Conclusion

We introduced a spatial-cue-steered dual-system framework for open-world delivery manipulation. By replacing the geometric low-level controller with a pretrained flow-matching VLA, our framework improves robustness to visual perturbations and reactivity to dynamic environments. It converts System 2 outputs into action-steering signals via a lightweight adapter while keeping the VLA frozen, preserving its manipulation prior and improving generalization to novel language–object mappings. Across tabletop and mobile manipulation evaluations, our method improves in-domain and out-of-domain performance with negligible additional inference cost, showing that spatial-cue steering provides a practical interface between open-vocabulary grounding and reactive VLA control.

6 Limitations

Our method depends on the underlying flow-matching policy and remains sensitive to out-of-distribution affordance regions and object positions, constraining the mobile base's stopping location and orientation relative to the target. Future work will address more complex delivery scenarios, including human interaction and handover, item selection from densely clustered bags, and tighter integration with navigation policies to improve manipulation reliability.

References

[1] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model, 2024. URL https://arxiv.org/abs/2406.09246.

[2] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL https://arxiv.org/abs/2307.15818.

[3] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

[4] X. Chen, Y. Chen, Y. Fu, N. Gao, J. Jia, W. Jin, H. Li, Y. Mu, J. Pang, Y. Qiao, Y. Tian, B. Wang, B. Wang, F. Wang, H. Wang, T. Wang, Z. Wang, X. Wei, C. Wu, S. Yang, J. Ye, J. Yu, J. Zeng, J. Zhang, J. Zhang, S. Zhang, F. Zheng, B. Zhou, and Y. Zhu. Internvla-m1: A spatially guided vision-language-action framework for generalist robot policy, 2025. URL https://arxiv.org/abs/2510.13778.

[5] W. Pumacay, I. Singh, J. Duan, R. Krishna, J. Thomason, and D. Fox. The colosseum: A benchmark for evaluating generalization for robotic manipulation, 2024. URL https://arxiv.org/abs/2402.08191.

[6] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation, 2024. URL https://arxiv.org/abs/2409.01652.

[7] F. Liu, K. Fang, P. Abbeel, and S. Levine. Moka: Open-world robotic manipulation through mark-based visual prompting, 2024. URL https://arxiv.org/abs/2403.03174.

[8] W. Yuan, J. Duan, V. Blukis, W. Pumacay, R. Krishna, A. Murali, A. Mousavian, and D. Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics, 2024. URL https://arxiv.org/abs/2406.10721.

[9] Z. Wang, Y. Chen, Y. Liu, J. Ye, P. Chen, C. Lu, S. Liu, B. Yu, and J. Jia. Vp-vla: Visual prompting as an interface for vision-language-action models, 2026. URL https://arxiv.org/abs/2603.22003.

[10] A. Bar, Y. Gandelsman, T. Darrell, A. Globerson, and A. Efros. Visual prompting via image inpainting. *Advances in neural information processing systems*, 35:25005–25017, 2022.

[11] J. Yang, H. Zhang, F. Li, X. Zou, C. Li, and J. Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v, 2023. URL https://arxiv.org/abs/2310.11441.

[12] L. Manuelli, W. Gao, P. Florence, and R. Tedrake. kpam: Keypoint affordances for category-level robotic manipulation, 2019. URL https://arxiv.org/abs/1903.06684.

[13] R. Zheng, Y. Liang, S. Huang, J. Gao, H. D. III, A. Kolobov, F. Huang, and J. Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies, 2025. URL https://arxiv.org/abs/2412.10345.

[14] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

[15] NVIDIA, :, J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. J. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, J. Jang, Z. Jiang, J. Kautz, K. Kundalia, L. Lao, Z. Li, Z. Lin, K. Lin, G. Liu, E. Llontop, L. Magne, A. Mandlekar, A. Narayan, S. Nasiriany, S. Reed, Y. L. Tan, G. Wang, Z. Wang, J. Wang, Q. Wang, J. Xiang, Y. Xie, Y. Xu, Z. Xu, S. Ye, Z. Yu, A. Zhang, H. Zhang, Y. Zhao, R. Zheng, and Y. Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

[16] M. Shukor, D. Aubakirova, F. Capuano, P. Kooijmans, S. Palma, A. Zouitine, M. Aractingi, C. Pascal, M. Russi, A. Marafioti, S. Alibert, M. Cord, T. Wolf, and R. Cadene. Smolvla: A vision-language-action model for affordable and efficient robotics, 2025. URL https://arxiv.org/abs/2506.01844.

[17] D. Kim, S. Jan, H. Park, and D. Lim. Vca: Vision-click-action framework for precise manipulation of segmented objects in target ambiguous environments, 2026. URL https://arxiv.org/abs/2602.23583.

[18] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang, and X. Li. Spatialvla: Exploring spatial representations for visual-language-action model, 2025. URL https://arxiv.org/abs/2501.15830.

[19] Y. Feng, W. Zhang, Y. Wang, H. Luo, H. Yuan, S. Zheng, and Z. Lu. Spatial-aware vla pretraining through visual-physical alignment from human videos, 2025. URL https://arxiv.org/abs/2512.13080.

[20] C. Pan, G. Anantharaman, N.-C. Huang, C. Jin, D. Pfrommer, C. Yuan, F. Permenter, G. Qu, N. Boffi, G. Shi, and M. Simchowitz. Much ado about noising: Dispelling the myths of generative robotic control, 2026. URL https://arxiv.org/abs/2512.01809.

[21] L. X. Shi, B. Ichter, M. Equi, L. Ke, K. Pertsch, Q. Vuong, J. Tanner, A. Walling, H. Wang, N. Fusai, A. Li-Bell, D. Driess, L. Groom, S. Levine, and C. Finn. Hi robot: Open-ended instruction following with hierarchical vision-language-action models, 2025. URL https://arxiv.org/abs/2502.19417.

[22] C. Li, J. Wen, Y. Peng, Y. Peng, F. Feng, and Y. Zhu. Pointvla: Injecting the 3d world into vision-language-action models, 2025. URL https://arxiv.org/abs/2503.07511.

[23] M. Nakamoto, O. Mees, A. Kumar, and S. Levine. Steering your generalists: Improving robotic foundation models via value guidance, 2025. URL https://arxiv.org/abs/2410.13816.

[24] K. Nakamura, L. Peters, and A. Bajcsy. Generalizing safety beyond collision-avoidance via latent-space reachability analysis. In *Robotics: Science and Systems XXI*. Robotics: Science and Systems Foundation, June 2025. doi:10.15607/rss.2025.xxi.113. URL http://dx.doi.org/10.15607/RSS.2025.XXI.113.

[25] Y. Song, L. Le, Y.-H. Park, J. Wang, J. Shi, L. Liu, J. Gu, E. Eaton, D. Jayaraman, and K. Daniilidis. Omniguide: Universal guidance fields for enhancing generalist robot policies, 2026. URL https://arxiv.org/abs/2603.10052.

[26] H. Cai, Z. Peng, and B. Zhou. Predictive preference learning from human interventions, 2025. URL https://arxiv.org/abs/2510.01545.

[27] Y. Wu, R. Tian, G. Swamy, and A. Bajcsy. From foresight to forethought: Vlm-in-the-loop policy steering via latent alignment, 2025. URL https://arxiv.org/abs/2502.01828.

[28] Z. Peng, W. Mo, C. Duan, Q. Li, and B. Zhou. Learning from active human involvement through proxy value propagation, 2025. URL https://arxiv.org/abs/2502.03369.

[29] M. Du and S. Song. Dynaguide: Steering diffusion polices with active dynamic guidance, 2025. URL https://arxiv.org/abs/2506.13922.

[30] Z. Sun and S. Song. Latent policy barrier: Learning robust visuomotor policies by staying in-distribution, 2025. URL https://arxiv.org/abs/2508.05941.

[31] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels, 2019. URL https://arxiv.org/abs/1811.04551.

[32] P. Wu, A. Escontrela, D. Hafner, K. Goldberg, and P. Abbeel. Daydreamer: World models for physical robot learning, 2022. URL https://arxiv.org/abs/2206.14176.

[33] X. Li, L. Xu, M. Zhang, J. Liu, Y. Shen, I. Ponomarenko, J. Xu, L. Heng, S. Huang, S. Zhang, and H. Dong. Crayonrobo: Object-centric prompt-driven vision-language-action model for robotic manipulation, 2025. URL https://arxiv.org/abs/2505.02166.

[34] A. J. Hancock, X. Wu, L. Zha, O. Russakovsky, and A. Majumdar. Actions as language: Fine-tuning vlms into vlas without catastrophic forgetting, 2025. URL https://arxiv.org/abs/2509.22195.

[35] S. Lee, S. Mo, and W.-S. Han. Bring my cup! personalizing vision-language-action models with visual attentive prompting, 2026. URL https://arxiv.org/abs/2512.20014.

[36] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

[37] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, J. Lei, T. Ma, B. Guo, A. Kalla, M. Marks, J. Greer, M. Wang, P. Sun, R. Rädle, T. Afouras, E. Mavroudi, K. Xu, T.-H. Wu, Y. Zhou, L. Momeni, R. Hazra, S. Ding, S. Vaze, F. Porcher, F. Li, S. Li, A. Kamath, H. K. Cheng, P. Dollár, N. Ravi, K. Saenko, P. Zhang, and C. Feichtenhofer. Sam 3: Segment anything with concepts, 2026. URL https://arxiv.org/abs/2511.16719.

[38] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

[39] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, W. Ge, Z. Guo, Q. Huang, J. Huang, F. Huang, B. Hui, S. Jiang, Z. Li, M. Li, M. Li, K. Li, Z. Lin, J. Lin, X. Liu, J. Liu, C. Liu, Y. Liu, D. Liu, S. Liu, D. Lu, R. Luo, C. Lv, R. Men, L. Meng, X. Ren, X. Ren, S. Song, Y. Sun, J. Tang, J. Tu, J. Wan, P. Wang, P. Wang, Q. Wang, Y. Wang, T. Xie, Y. Xu, H. Xu, J. Xu, Z. Yang, M. Yang, J. Yang, A. Yang, B. Yu, F. Zhang, H. Zhang, X. Zhang, B. Zheng, H. Zhong, J. Zhou, F. Zhou, J. Zhou, Y. Zhu, and K. Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

A Method Details

A.1 MOKA

MOKA was originally designed for tabletop manipulation with top-down observations and 2D planar affordance prediction, which does not directly match our third-person shelf-picking setting with deformable bag objects and non-planar interactions. Rather than fully re-engineering MOKA for this domain, we evaluate its high-level planning capability under an oracle execution assumption: a prediction is considered successful if the generated 2D affordance points correspond to semantically correct and physically feasible interaction locations.

For consistency across methods, we standardize the perception stack with Qwen3-VL [39] and SAM2.1 [36], and adapt the prompting strategy to our shelf-manipulation tasks. We also evaluate MOKA under its 3-shot in-context configuration, corresponding to the strongest setting reported in the original work. Despite these adaptations and the oracle execution assumption, performance remains limited due to target misclassification, incorrect grasp-point prediction, and infeasible proposed motions. These results indicate that MOKA's 2D affordance representation has limited transferability to our shelf-based delivery manipulation setting.

A.2 VP-VLA

The original VP-VLA method is designed for autoregressive VLAs and is not directly formulated for flow-matching policies. We adapt its visual-prompting mechanism by providing the generated visual prompt as an additional visual input to the VLA and training the policy with auxiliary grounding supervision. This baseline differs from our method primarily in how the System 2 visual signal is used: VP-VLA treats it as an input-level prompt, whereas our method converts the spatial cue into an action-steering signal that modulates the frozen policy output. For consistency across methods, we use the same perception stack with Qwen3-VL [39] and SAM2.1 [36].

A.3 Ours

We rewrite each user instruction into a standardized command format compatible with the VLA's training distribution, removing task-irrelevant details while keeping the target specification. For demonstrations involving mobile navigation, we divide execution into three stages. First, the mobile base remains stationary while the robot arm performs autonomous manipulation. Once the gripper grasps the target object, policy inference is paused and a hard-coded arm motion moves the object to a predefined holding pose for stable transport. The grasping moment is detected when the gripper jaw width falls below a fixed threshold. Second, we manually teleoperate or use map-based autonomous navigation to guide the mobile base toward the target placement location. Third, policy inference is resumed to allow the robot to autonomously complete the placement stage. The demonstrations are recorded in both indoor and outdoor environments, with diverse backgrounds and lighting conditions. We observe that policy performance and steering effectiveness decrease under these out-of-domain conditions, but the model still shows a consistent steering trend and retains the ability to identify target objects and complete the tasks.

B Experiment Details

B.1 Hardware Setup

Our experimental platform consists of a DEEP Robotics Lynx M20 Pro wheeled quadruped integrated with an AgileX Robotics PiPER robotic arm. We use a Stereolabs ZED 2 stereo camera for the third-person view and an Intel RealSense D435i depth camera as a wrist-mounted camera on the manipulator, although depth and stereo measurements are not used in this paper. A Jetson AGX Orin mounted on the robot coordinates communication between the robot hardware and a remote workstation running the policy on an RTX 5080 GPU. For outdoor experiments, network access is

provided by USB tethering the Jetson to a mobile phone, which serves as a 5G uplink. The final hardware layout is shown in Figure 7.

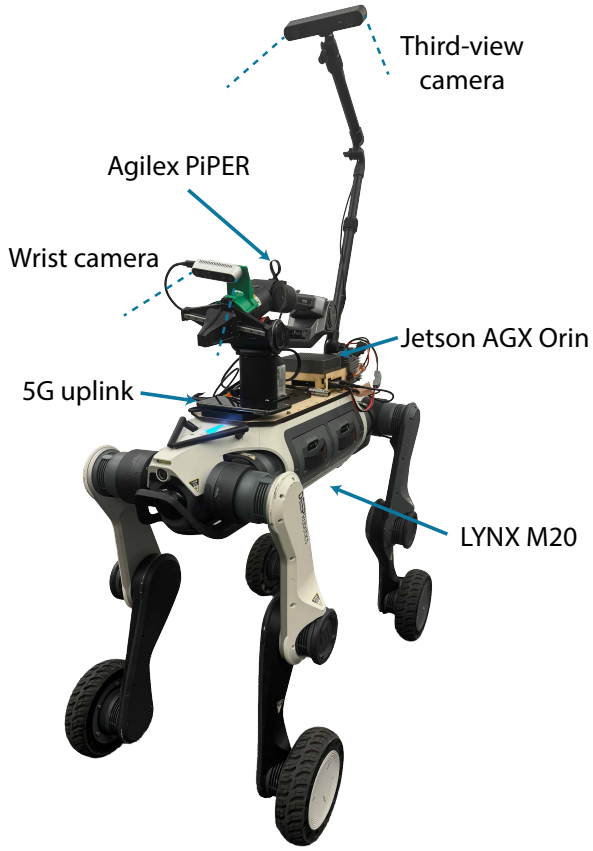


Figure 7: Robot hardware setup.

The Jetson AGX Orin serves as a lightweight communication bridge rather than an inference device. It captures both camera streams and transmits them to the remote workstation through local network communication. The workstation receives the video streams, runs the policy, and sends action commands back through the same network path to the Jetson, which forwards them to the robot arm. This setup incurs approximately 80ms round-trip latency, modestly higher than wired operation.

B.2 Evaluation Setup

To evaluate robustness to linguistic variation and compositional generalization, we construct a fixed set of prompt templates covering the three evaluation settings described in the main paper. Prompts are generated compositionally by varying grasp verbs, object descriptors, and placement phrasing while preserving the same underlying pick-and-place task semantics. Representative prompt templates are shown in Table 3.

In addition to language variation, we evaluate robustness under diverse spatial configurations and object layouts. Figure 8 shows representative scenes from each evaluation setting. Across configurations, we vary object placement, relative spacing, distractor positions, and camera viewpoints while preserving the same high-level pick-and-place objective. The seen-object setting reuses training objects in novel arrangements, whereas the unseen-object settings introduce novel targets and increasingly challenging referring expressions. All configurations are held fixed across compared methods to ensure consistent evaluation conditions.

Evaluation Setting	Prompt Templates
Setting I & II	Pick the <descriptor> bag and put it into the box Grab <descriptor> bag and place it in the box Take the <descriptor> bag and drop it into the box Move the <descriptor> bag into the box Put <descriptor> bag into the box Place the <descriptor> bag in the box
Setting III	Put bag with order number <value> into the box Grab bag with number <value> into the box Pick order number <value> bag and place it into the box

Table 3: **List of prompts used for evaluation.** For seen-object evaluations, <descriptor> refers to objects observed during training, i.e., "black" or "plastic." For unseen-object evaluations, it refers to novel descriptors such as "brown" or "red." In the novel-language setting, <value> denotes ordinal identifiers, i.e., 1, 2, or 3, used to evaluate compositional reference grounding under previously unseen instruction patterns.

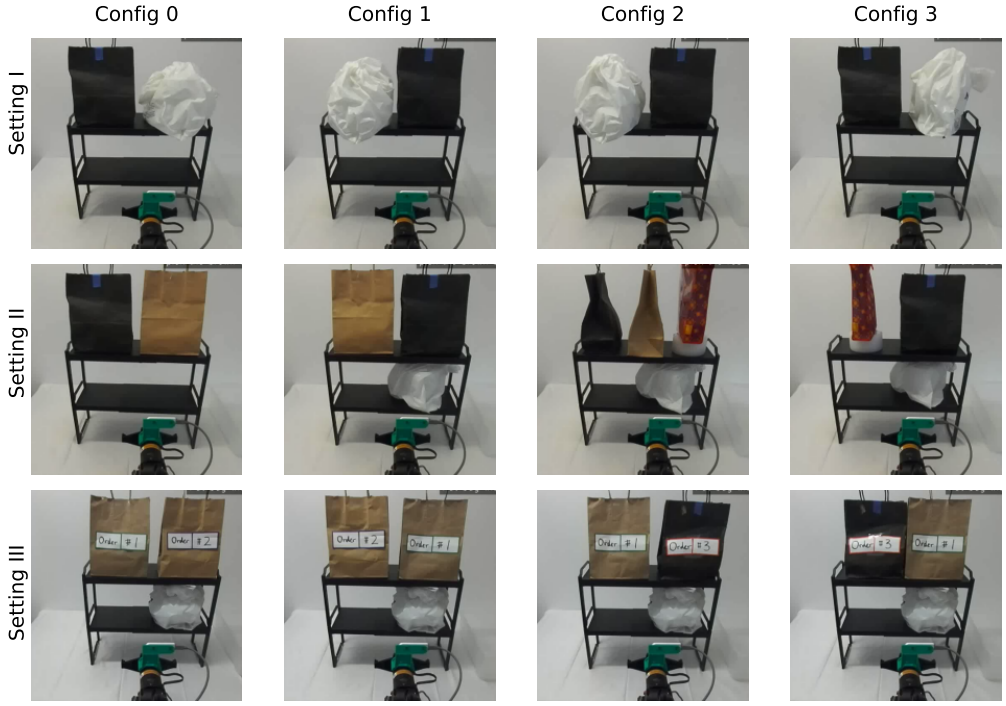
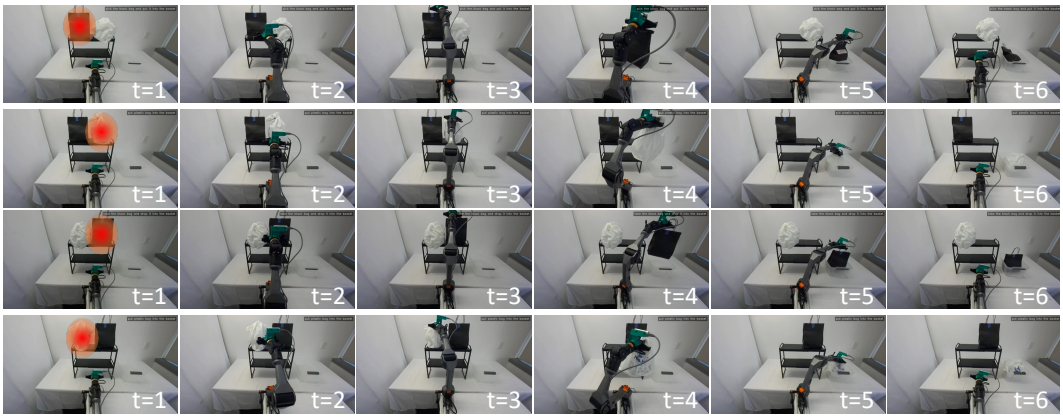


Figure 8: **Representative evaluation configurations across the three experimental settings.** Rows correspond to the evaluation regimes described in the main paper: Setting I uses seen objects with paraphrased language, Setting II uses unseen objects with known language, and Setting III uses unseen objects with novel language. Columns show different spatial layouts used during evaluation, varying object placement, distractor arrangement, and scene geometry. In Setting I, Configurations 0–1 correspond to the black-bag task, while Configurations 2–3 correspond to the plastic-bag task. All methods are evaluated on the same prompts and spatial configurations.

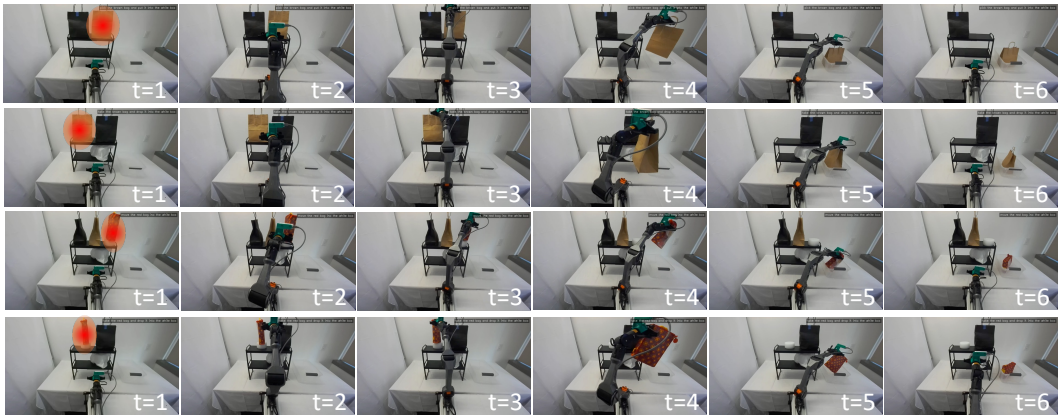
C Additional Results

Figure 9 provides additional visual comparisons, where the red heatmap indicates the correctly grounded target object. Under the same spatial object layouts, our method consistently steers the policy toward the spatial cue across all three evaluation settings. We observe that the policy often produces similar actions in the early execution steps, but as the end-effector approaches the objects, the generated trajectory is progressively redirected toward the cued target.

Setting I



Setting II



Setting III

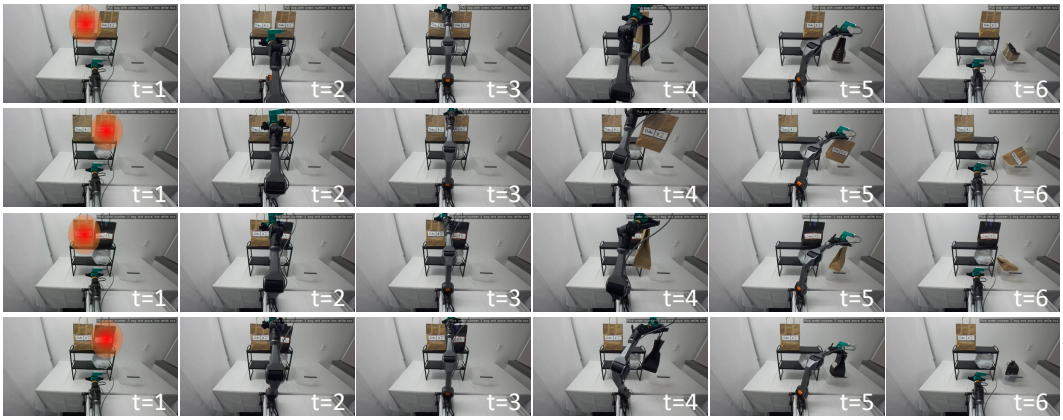


Figure 9: **Example demonstrations of policy steering across the three evaluation settings.** The red heatmap indicates the correctly grounded target object.

D Failure Cases

Our method can fail in two scenarios. First, it cannot recover when the underlying manipulation prior is unreliable. Noisy rollouts, accumulated execution errors, or environmental changes can still cause the frozen policy to deviate from a feasible manipulation trajectory. Second, the steering strength is applied consistently throughout the rollout, which can over-modulate the policy output when precise actions are required. This may lead to manipulation errors such as overshooting or unstable grasping.

E Video Demonstrations

The supplementary material includes a paper overview video and the corresponding full policy rollout demonstrations.