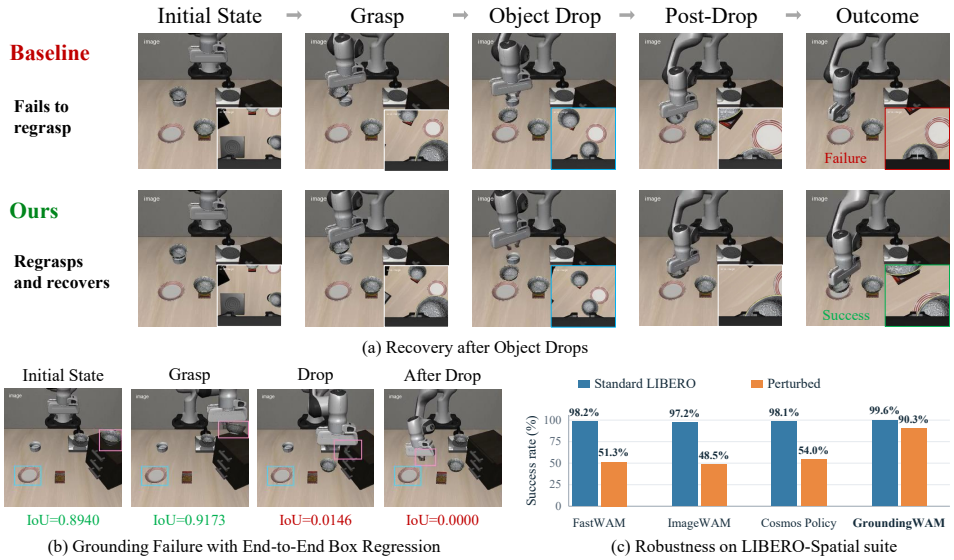
GROUNDINGWAM: LEARNING WHERE TO FOCUS IN WORLD ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models (WAMs) achieve high success rates on manipulation benchmarks, yet can struggle when execution departs from familiar demonstration trajectories. Following an unexpected object drop, a policy may fail to relocalize the target and resume manipulation. We probe this grounding weakness through auxiliary localization supervision: the predicted target object location can remain near the gripper even after the object has moved elsewhere. We interpret this behavior as a consequence of the strong correlations between object positions, robot states, and task progress in successful demonstrations, which allow models trained with end-to-end regression objectives to exploit predictable patterns without consistently grounding their predictions in current visual evidence. Motivated by this observation, we introduce **GroundingWAM**, which uses task-relevant spatial and language annotations to directly supervise internal attention during action generation, guiding action queries toward relevant image regions and instruction tokens. Trained on successful demonstrations without additional recovery demonstrations, GroundingWAM achieves 99.3% success on standard LIBERO and 87.1% on LIBERO-Plus benchmark. On our object-release setting, it improves success from 47.79% to 64.71%, with a gain of 41.75 percentage points on LIBERO-Spatial. Component comparisons and rollout visualizations support internal grounding supervision as a practical way to strengthen manipulation beyond familiar execution patterns.



(a) Recovery after Object Drops

(b) Grounding Failure with End-to-End Box Regression

(c) Robustness on LIBERO-Spatial suite

Figure 1: **Robust grounding under execution disturbances.** (a) The baseline WAM often fails to ground the target after an object drop, while GroundingWAM recovers from the same initial scene. (b) End-to-end box regression is not reliable to solve the grounding problem, often leaving predictions near the gripper after the target moves away. (c) Performance on LIBERO-Spatial with disturbance, current WAMs can suffer from object drop, while GroundingWAM maintains high success rate.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

World Action Models (WAMs) connect visual world modeling with action generation, transferring the generative priors of pretrained visual models into robot control. Recent advances in generative robot policies have demonstrated strong performance on manipulation benchmarks (Yuan et al., 2026; Zhang et al., 2026; Li et al., 2026; Kim et al., 2026). However, these models can struggle to adapt when execution deviates from familiar demonstration trajectories. In our experiments, WAMs struggle to resume manipulation after an unexpected object drop: the robot fails to regrasp the displaced object and sometimes continues toward the placement location despite no longer holding it. These observations raise a fundamental question: *do these models ground their actions in the current scene, or rely on familiar patterns that remain predictive along successful demonstrations?*

To investigate this question, we conduct controlled object-drop experiments on LIBERO (Liu et al., 2023), inducing the gripper to release an object during transport. This intervention preserves the instruction, target identity, and object appearance while disrupting the spatial relationship between the object and the gripper. Completing the task therefore requires the policy to use current visual evidence to relocalize the object and adjust its behavior. As shown in Figure 1(c), the resulting performance gap is shared by several representative WAMs. Fast-WAM, ImageWAM, and Cosmos Policy (Yuan et al., 2026; Zhang et al., 2026; Kim et al., 2026) report standard Spatial success rates of 98.2%, 97.2%, and 98.1%, respectively, yet achieve only 51.25%, 48.50%, and 54.00% under our controlled perturbations. Together with the observed failures to retrieve displaced objects, These results suggest that high nominal success does not necessarily imply strong failure recovery capabilities.

A natural approach to improving failure recovery is to strengthen the model's grounding capability. To this end, we introduce bounding-box regression as an auxiliary localization objective and jointly train it with the WAM's original visual and action objectives. Although the predicted boxes can accurately follow an object during nominal manipulation, they can often remain near the gripper after the object falls. Figure 1(b) illustrates this discrepancy: target-bowl 2D IoU is 0.8940 and 0.9173 in the initial and grasped states, but falls to 0.0146 and 0 after the drop. The prediction follows a familiar gripper–object relationship even when it no longer agrees with the observed scene. This finding indicates that end-to-end localization supervision alone does not ensure that the model learns to ground its predictions in the object's actual position.

We argue that this difficulty arises in part from the *strong correlations* embedded in successful manipulation demonstrations. During transport, object positions are tightly coupled with gripper positions, while robot configuration and task progression provide additional cues about the expected object state. Consequently, minimizing a box regression loss can favor reproducing these predictable relationships without consistently identifying the object from visual observations. Action learning is subject to the same ambiguity: correlations that explain demonstrated actions may become unreliable once execution departs from the training distribution. This interpretation is consistent with causal confusion in imitation learning (De Haan et al., 2019; Wen et al., 2020). It motivates our central insight: grounding supervision should directly shape how the model uses task-relevant evidence within its internal computation, beyond specifying the outputs it should predict.

Motivated by this insight, we propose **GroundingWAM**, a framework that strengthens visual grounding through *Grounding Attention Distillation* (GAD). GAD transfers task-relevant spatial and language annotations into direct supervision of internal attention during action generation. Its action-to-vision and action-to-text paths guide action queries toward task-relevant image regions and instruction phrases in selected layers and attention heads. These constraints encourage the action-generation process to attend to both spatial and language evidence, rather than merely predicting an output that can be driven by spurious correlations in the training data. GroundingWAM retains the original visual and action prediction objectives and requires grounding annotations only during training, allowing the policy to operate with its standard inputs at deployment.

We evaluate GroundingWAM across standard manipulation, environmental variations, and execution disturbances. Trained on successful demonstrations *without* additional recovery demonstrations, our method achieves 99.3% success on LIBERO and 87.1% in LIBERO-Plus. On the controlled LIBERO-Spatial object-drop evaluation, it improves success from 48.50% to 90.25%, a gain of 41.75 percentage points. Component comparisons further demonstrate the improvements brought by GAD and auxiliary box regression over the baseline, while rollout visualizations show attention

following displaced objects and remaining allocated to task-relevant regions. These results demonstrate the value of supervising internal representation for learning robust manipulation policies from successful demonstrations.

Our contributions are summarized as follows:

- We investigate recovery from execution disturbances in WAMs and find that high success rates on standard benchmarks can conceal poor recovery from unexpected object drops. We identify target localization failures in unfamiliar states as a key weakness and show that auxiliary box regression alone is insufficient to address this problem.
- We propose GroundingWAM, which incorporates attention supervision over task-relevant visual regions and instruction phrases into WAM training, guiding action generation to attend to the target and its currently observed state.
- Our experiments show that GroundingWAM improves task success under our object-release protocol from 48.50% to 90.25% on LIBERO-Spatial without additional recovery data, while achieving 99.3% success on LIBERO and 87.1% on the full LIBERO-Plus benchmark.

2 Related Work

World Action Models for Robotic Manipulation. World action models use future prediction to learn representations for robot control. Earlier approaches extract actions from generated video plans (Du et al., 2023), jointly predict images and actions (Wu et al., 2024; Cheang et al., 2024), or condition inverse dynamics on forecasted visual states (Tian et al., 2025). VideoVLA transfers pretrained video generation to joint visual and action forecasting (Shen et al., 2026). WorldVLA unifies image and action generation autoregressively (Cen et al., 2025), while Motus combines understanding, video generation, and action experts (Bi et al., 2026). Cosmos Policy adapts a video model to predict actions, future observations, and values (Kim et al., 2026), and LingBot-VA jointly learns frame prediction and policy execution (Li et al., 2026). Being-H0.7 and Fast-WAM retain predictive learning while avoiding future-frame generation at inference (Luo et al., 2026; Yuan et al., 2026). ImageWAM instead transfers image-editing priors into action generation (Zhang et al., 2026). Building on these advances, we study how grounding supervision helps WAMs use current visual evidence when execution departs from familiar demonstration trajectories.

Robust Policy Learning for Robotic Manipulation. Robust manipulation requires coverage of unfamiliar states and representations that support control. DAgger and DART collect corrective supervision through on-policy expert labeling and disturbance injection, respectively (Ross et al., 2011; Laskey et al., 2017). MOO and RoboGround incorporate grounded object information (Stone et al., 2023; Huang et al., 2025), while SpatialVLA and 3D-VLA incorporate spatial representations and 3D perception for control (Qu et al., 2025; Zhen et al., 2024). SlotVLA learns task-relevant object slots and relations (Hanyu et al., 2025), and ECoT supervises reasoning about plans, objects, and robot states before action prediction (Zawalski et al., 2024). These approaches demonstrate the value of structured information for policy learning.

Reliable execution also depends on how representations are learned and used. Causal Confusion studies unreliable dependencies in demonstrations (De Haan et al., 2019); copycat studies show how temporal correlations can favor repeating previous actions over responding to current evidence (Wen et al., 2020; 2021). Explanation constraints guide classifiers toward relevant inputs (Ross et al., 2017), while OREO addresses causal confusion through object-aware regularization (Park et al., 2021). GuidedVLA supervises object-centric structure, skills, and geometry within the VLA action decoder (Jia et al., 2026); Artificial Foveated Perception addresses perceptual shortcuts in robotic foundation models (Sun et al., 2026), and AGRA aligns predictive representations with action grounding (Qiu et al., 2026). We examine whether auxiliary localization supervision establishes robust grounding in WAMs. Box predictions can remain near the gripper after object drops, suggesting that output-level regression retains familiar correlations. This diagnosis motivates GroundingWAM's direct supervision of task-relevant visual and language attention within action generation.

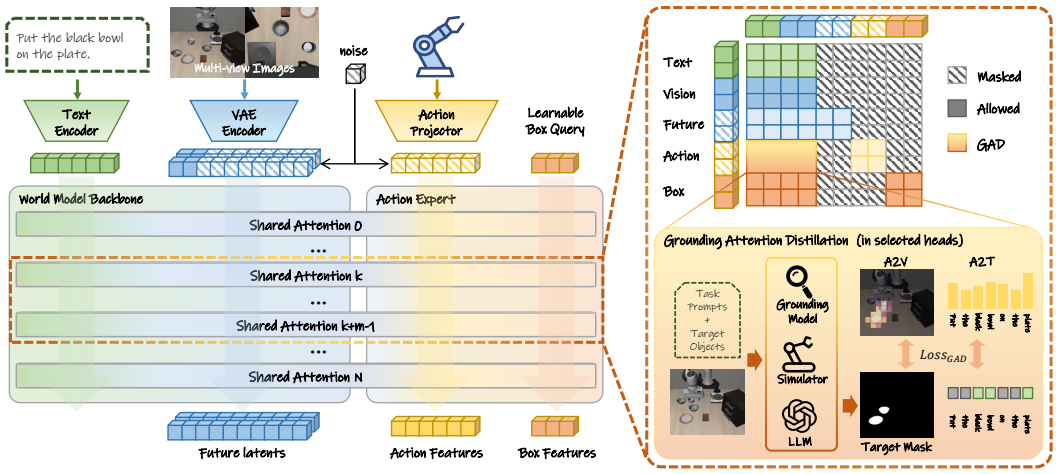


Figure 2: **GroundingWAM overview.** The visual backbone and action expert interact through masked joint attention. Learned box queries share the action expert's transformer layers and use a separate 2D prediction head. Grounding attention distillation supervises action-to-vision (A2V) and action-to-text (A2T) attention in selected heads, guiding action queries toward task-relevant regions and instruction phrases. The annotation providers shown are alternatives; our experiments use simulator-derived regions and task-level phrase annotations. These labels are required only during training.

3 METHOD

To address grounding failures in world action models under unexpected execution perturbations, we first introduce an end-to-end auxiliary grounding loss that supervises object localization through learned box queries within the action expert (Section 3.1). Its limitations in tracking displaced objects motivate *grounding attention distillation* (GAD), which directly supervises action attention to task-relevant visual regions and instruction phrases (Section 3.2). We then describe how these objectives are jointly optimized and how the policy operates at inference (Section 3.3). Figure 2 summarizes the overall architecture.

3.1 AUXILIARY GROUNDING THROUGH BOX PREDICTION

A world action model generates action chunks conditioned on current images, robot states, and language instructions while learning to predict future observations. We consider a mixture-of-transformers architecture (Liang et al., 2024), where a visual backbone and an action expert exchange information through joint attention.

To improve visual grounding, we first explore auxiliary localization supervision by appending eight learnable box queries to the action expert. These queries share transformer parameters with action tokens and attend to language and current visual features. A separate head predicts normalized 2D boxes and objectness scores. Using set prediction with bipartite matching (Carion et al., 2020), the auxiliary loss $\mathcal{L}_{\text{box}}$ combines matched coordinate regression with objectness supervision and jointly optimizes the box branch, action expert, and visual backbone. Details are provided in Appendix B.

However, auxiliary box regression alone does not ensure robust grounding. As shown in Section 4.2, predictions can remain near the gripper after an object drops, suggesting reliance on object–gripper correlations in successful demonstrations. Action learning is susceptible to the same bias: fitting recurring demonstration patterns need not produce representations that generalize when these patterns break. This motivates grounding attention distillation (GAD), which directly supervises intermediate representations through task-relevant attention.

3.2 Grounding Attention Distillation

GAD encourages action queries to attend to the visual regions and instruction phrases that specify the task. It provides supervision inside the action expert, where the model selects evidence for generating actions. We formulate this supervision through two complementary paths: action-to-vision (AV) and action-to-text (AT).

Grounding targets. GAD supervises attention to manipulated objects and task destinations in both the current observation and the instruction. We denote their visual patch indices by $\mathcal{V}_*$ and instruction token indices by $\mathcal{T}_*$. In our experiments, $\mathcal{V}_*$ comprises patches overlapping 2D boxes of target objects, while $\mathcal{T}_*$ is identified from target entity phrases. Visual targets follow the objects in each observation, and all action queries within a chunk share the same targets.

Action-to-vision and action-to-text supervision. At each selected layer and head, let A_{qj} denote the attention probability from action query q to key j, normalized over all allowed keys. We define the attention mass assigned to the visual and language targets as

$$m_q^{\mathrm{AV}} = \sum_{j \in \mathcal{V}_*} A_{qj}, \qquad m_q^{\mathrm{AT}} = \sum_{j \in \mathcal{T}_*} A_{qj}. \tag{1}$$

Action-to-vision (AV) supervision encourages attention to the observed locations of task-relevant objects and destinations, while action-to-text (AT) supervision emphasizes the instruction phrases that specify them.

Grounding attention distillation. GAD encourages each action query to allocate sufficient attention to task-relevant visual and language evidence. We apply a thresholded attention-mass penalty:

$$\mathcal{L}_p = \frac{1}{|\mathcal{Q}_p|} \sum_{q \in \mathcal{Q}_p} \max\left\{0, \log \frac{\tau_p + \epsilon}{m_q^p + \epsilon}\right\}, \qquad p \in \{\mathrm{AV}, \mathrm{AT}\}, \tag{2}$$

where $\mathcal{Q}_p$ denotes the action queries with valid targets for path p, m_q^p is the corresponding target attention mass, τ_p is its threshold, and $\epsilon = 10^{-8}$. The penalty is applied independently to each query and vanishes once its target mass reaches the threshold, allowing attention to distribute freely within the target set. Both objectives use attention probabilities normalized over all allowed keys. We set $\tau_{\mathrm{AV}} + \tau_{\mathrm{AT}} < 1$ to allow attention to other contextual information. Together, these objectives complement output-level localization supervision by directly guiding attention toward the spatial and semantic evidence needed for action generation. Our experiments further suggest that supervising intermediate representations yields more reliable grounding under execution perturbations than end-to-end auxiliary localization supervision alone.

3.3 Training and Inference

We jointly train visual prediction, action generation, and auxiliary grounding without additional perturbation data. The AV and AT paths define the attention supervision objective:

$$\mathcal{L}_{\mathrm{GAD}} = \lambda_{\mathrm{AV}} \mathcal{L}_{\mathrm{AV}} + \lambda_{\mathrm{AT}} \mathcal{L}_{\mathrm{AT}}. \tag{3}$$

We optimize this objective jointly with the visual and action flow-matching losses, $\mathcal{L}_{\mathrm{vis}}$ and $\mathcal{L}_{\mathrm{act}}$, and the auxiliary localization loss $\mathcal{L}_{\mathrm{box}}$. The trainable visual backbone and action expert receive gradients from these objectives, while the text encoder and VAE remain frozen. The box queries and prediction head are optimized jointly with them. Loss weights, supervised layers and heads, and optimization settings are given in Appendix B.

At inference, the policy conditions on the instruction, current images, and robot state to denoise an action chunk. It executes a prefix of that chunk and replans from the next observation. Grounding annotations and attention losses are used only for training; action generation requires neither an external grounding model nor predicted box coordinates. The resulting policy retains the WAM's standard observation-to-action interface.

4 EXPERIMENTS

Our experiments investigate how grounding supervision affects the behavior of WAMs beyond successful demonstration trajectories. We first diagnose the limitations of nominal task success and auxiliary localization, then evaluate GroundingWAM on standard manipulation, environmental variations, and object-position disturbances during execution. Component comparisons and attention analyses connect these behavioral results to the choice of supervision.

4.1 EXPERIMENTAL SETUP

4.1.1 EVALUATION BENCHMARKS

LIBERO and LIBERO-Plus. We evaluate nominal task success on the Spatial, Object, Goal, and Long suites of LIBERO (Liu et al., 2023), covering spatial relationships, object identities, task goals, and long-horizon manipulation. Each suite contains ten tasks, and each task is evaluated from 50 initial states, yielding 2,000 episodes in total. To further assess robustness, we evaluate LIBERO-Plus (Fei et al., 2026), which introduces variations in camera viewpoints, robot initial states, language instructions, lighting, backgrounds, sensor noise, and object layouts. We cover all 10,030 scenarios with one trial per scenario and report category-wise and overall success rates. Both evaluations use policies trained on the original LIBERO demonstrations, without additional recovery data.

Object-Release Perturbation. We construct a controlled evaluation based on LIBERO in which a grasped object is unexpectedly released during transport. Of the 40 tasks, 37 involve operations suitable for this intervention. These tasks contain 43 eligible transport targets, as some require transporting multiple objects. For each target, we measure transport progress by its displacement toward the destination, normalized by the initial object-to-destination distance. Once the object is stably grasped and lifted, reaching a prescribed progress threshold triggers forced gripper opening for ten control steps, while the remaining action dimensions stay policy-controlled. The policy then replans from the current observation and continues within the remaining episode budget. This intervention preserves the instruction and target identity while testing the policy's ability to respond to object displacement. We evaluate eight progress thresholds for each target, yielding 344 conditions and 1,720 episodes per model across five initial states. Detailed settings are provided in Appendix A.2.

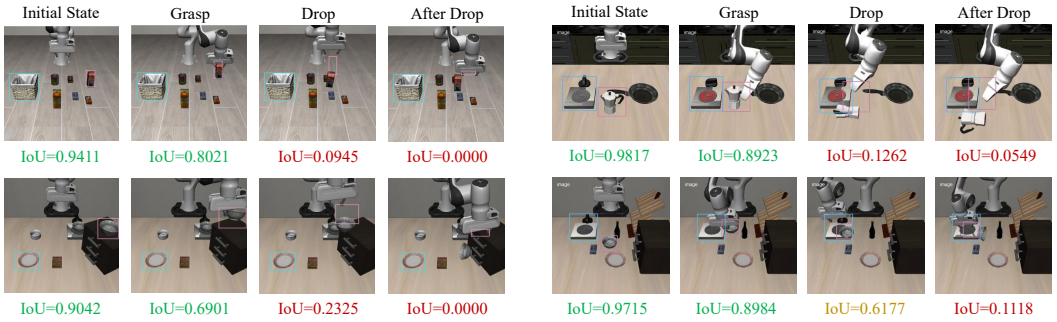
4.1.2 IMPLEMENTATION DETAILS

We conduct experiments on both FastWAM (Yuan et al., 2026) and ImageWAM (Zhang et al., 2026) frameworks. For our main experiments, we initialize the visual backbone from pretrained FLUX.2 4B (Black Forest Labs, 2025) weights and train the action expert and auxiliary grounding modules from scratch. Training uses demonstrations from all four LIBERO suites without additional recovery demonstrations. The text encoder and VAE remain frozen, while the visual backbone and action expert are jointly optimized.

We replay demonstrations in the simulator to extract masks and bounding boxes for task-relevant objects and destinations, and identify their corresponding text tokens from the task descriptions. We train for ten epochs on eight GPUs with a global batch size of 96, applying attention supervision to six heads in each of five selected intermediate layers. Further implementation details are provided in Appendix B.

4.2 DIAGNOSING GROUNDING FAILURES IN WAMS

High nominal success can conceal weak recovery. Despite their high success rates on standard LIBERO tasks, existing WAMs often struggle to recover when an object is unexpectedly dropped during execution. We examine this gap on LIBERO-Spatial, where our intervention disrupts the spatial relationship between the gripper and target while preserving the task instruction and scene configuration. FastWAM, ImageWAM, and Cosmos-Policy report standard success rates of 98.2%, 97.2%, and 98.1%, respectively (Yuan et al., 2026; Zhang et al., 2026; Kim et al., 2026). Under our object-release protocol, their success rates fall to 51.25%, 48.50%, and 54.00% over 400 episodes per model (Figure 1c). This consistent degradation across three WAMs shows that strong



(a) FastWAM with Box Auxiliary Prediction (b) ImageWAM with Box Auxiliary Prediction

Figure 3: **Grounding failure with end-to-end box regression.** Two trajectories each from box-supervised (a) FastWAM and (b) ImageWAM, spanning initial state, grasp, drop, and post-drop observations. Pink boxes show predictions associated with the transported object; cyan boxes show predictions for another task-relevant object or destination. After a drop, pink boxes often remain near the gripper and target IoU decreases.

performance on nominal tasks does not necessarily translate into effective recovery from execution disturbances. As illustrated in Figure 1a, a policy may continue toward the placement destination without retrieving the dropped object, suggesting insufficient grounding of subsequent actions in the current scene.

Output-level localization does not ensure visual grounding. We next examine whether introducing localization as an auxiliary objective resolves this problem. Figure 3 shows FastWAM and ImageWAM policies trained with auxiliary box supervision: predictions track targets during grasping but can remain near the gripper after the objects fall. In the four observations of Figure 1b, target-bowl 2D IoU drops from 0.8940 and 0.9173 before release to 0.0146 and 0 afterward. These observations suggest that end-to-end box regression on manipulation demonstrations is susceptible to fitting dataset-specific correlations, such as the coupling between object and gripper positions, without acquiring robust visual grounding. These findings motivate directly introducing grounding supervision at the level of intermediate representations. We next evaluate whether this approach improves task performance and robustness beyond auxiliary box regression alone.

4.3 Manipulation Performance

Standard manipulation and generalization. GroundingWAM achieves **99.25%** success on LIBERO (1,985/2,000 episodes), with success rates of at least 98.40% across all four suites (Table 1a), demonstrating strong performance across diverse manipulation tasks. This performance extends to environmental and instruction variations: without additional adaptation, GroundingWAM achieves **87.08%** success across all 10,030 LIBERO-Plus scenarios (8,734 successes; Table 1b). It reaches 98.51% under lighting changes and 90.63% under language variations, while maintaining 85.16% and 86.30% under changes to robot initial states and object layouts, respectively. Together, these results show that grounding supervision supports strong nominal control and generalization across a range of visual, linguistic, and physical variations. We additionally apply our grounding supervision to FastWAM. On standard LIBERO, success increases from 97.3% for our reproduced baseline to 98.1%, providing additional evidence of its applicability.

Recovery across manipulation tasks. Table 2 reports task completion over the complete release grid. GroundingWAM improves overall success from 47.79% for ImageWAM to **64.71%**, adding 291 successful episodes. The largest gain occurs on Spatial, where success rises from 48.50% to **90.25%**. FastWAM and Cosmos-Policy also achieve substantially lower Spatial success under the same intervention rules. GroundingWAM learns this recovery behavior from successful demonstrations without additional training on object drops, supporting the value of internal grounding supervision when execution departs from the demonstrated progression. Table 3 further reports drop rates and, among the remaining confirmed drops, rates of regrasping and regrasping with task suc-

Table 1: Success (%) on standard LIBERO and LIBERO-Plus. Baselines retain their published training and evaluation protocols; We use 50 episodes per LIBERO task and one trial per LIBERO-Plus scenario (10,030 total).

(a) Standard LIBERO

Method	Spatial	Object	Goal	Long	Average
OpenVLA-OFT (Kim et al., 2025)	97.6	98.4	97.9	94.5	97.1
$\pi_{0.5}$ (Physical Intelligence, 2026)	98.8	98.2	98.0	92.4	96.9
Fast-WAM (Yuan et al., 2026)	98.2	**100.0**	97.0	95.2	97.6
ImageWAM (4B) (Zhang et al., 2026)	97.2	99.2	98.8	98.4	98.4
Cosmos Policy (Kim et al., 2026)	98.1	**100.0**	98.2	97.6	98.5
LingBot-VA (Li et al., 2026)	98.5	99.6	97.2	**98.5**	98.5
GroundingWAM	**99.6**	**100.0**	**99.0**	98.4	**99.3**

(b) LIBERO-Plus

Method	Camera	Init.	Lang.	Light	Backgr.	Noise	Layout	Overall
OpenVLA-OFT (Kim et al., 2025)	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
$\pi_{0.5}$ (Physical Intelligence, 2026)	75.4	77.5	85.6	96.9	**94.6**	89.7	85.7	85.7
Fast-WAM (Yuan et al., 2026)	16.4	44.5	68.9	78.2	53.7	37.7	60.7	51.5
Cosmos Policy (Kim et al., 2026)	75.8	63.3	81.7	96.5	88.9	92.7	82.2	82.2
ImageWAM (Zhang et al., 2026)	80.8	50.3	**91.4**	98.1	85.5	93.8	80.5	83.1
GAM (Han et al., 2026)	**83.1**	70.0	84.8	97.2	94.3	**95.3**	79.1	85.5
GroundingWAM	77.0	**85.2**	90.6	**98.5**	88.6	87.2	**86.3**	**87.1**

Table 2: Success (%) under object-release perturbations. Overall pools all 1,720 episodes across 344 conditions.

Method	Spatial	Object	Goal	Long	Overall
FastWAM	51.25	33.00	53.21	51.72	47.50
ImageWAM	48.50	39.50	53.93	49.84	47.79
Cosmos-Policy	54.00	39.25	52.14	51.25	49.24
GroundingWAM	**90.25**	**44.00**	**60.71**	**63.44**	**64.71**

Table 3: Drop: % of all episodes. Regrasp and R&S: % of eligible drops (Appendix A.2).

Method	Drop	Regrasp	R&S
FastWAM	95.99	42.53	29.68
ImageWAM	97.73	47.32	31.46
Cosmos-Policy	98.02	44.03	33.33
GroundingWAM	97.33	**59.35**	**51.21**

Table 4: Component analysis. Success (%) on the fixed LIBERO-Plus subset: 2,000 shared scenarios, with 500 episodes per suite.

Method	Spatial	Object	Goal	Long	Overall
ImageWAM	86.00	89.60	78.80	80.80	83.80
Box-only	91.80	89.20	**81.20**	84.20	86.60
GAD-only	**94.80**	92.20	**81.20**	81.60	87.45
GroundingWAM	94.00	**95.40**	80.60	**85.20**	**88.80**

Table 5: Component comparison on all 400 Spatial release episodes.

Method	Success (%)
ImageWAM	48.50
Box-only	73.00
GAD-only	81.00
GroundingWAM	**90.25**

cess. Details of the release intervention, recovery criteria, and evaluation denominators are provided in Appendix A.2.

4.4 Ablation Study

We compare four configurations on environmental variations and execution disturbances: ImageWAM baseline, Box-only, GAD-only, and complete GroundingWAM. Here, GAD-only denotes attention supervision without an auxiliary box head. LIBERO-Plus comparisons use an outcome-independent subset of 2,000 shared scenarios, with 500 per suite (Table 4). The release comparison covers all ten Spatial tasks at eight fixed transport thresholds and five initial states, yielding 400 episodes per model without selecting tasks or release times based on success (Table 5). Appendix A.4 details the Plus subset selection.

On LIBERO-Plus, Box-only improves success from 83.80% to 86.60%, while GAD-only reaches 87.45% and complete GroundingWAM achieves 88.80%. The distinction is more pronounced under object-release perturbations: Box-only reaches 73.00%, GAD-only achieves 81.00%, and Ground-

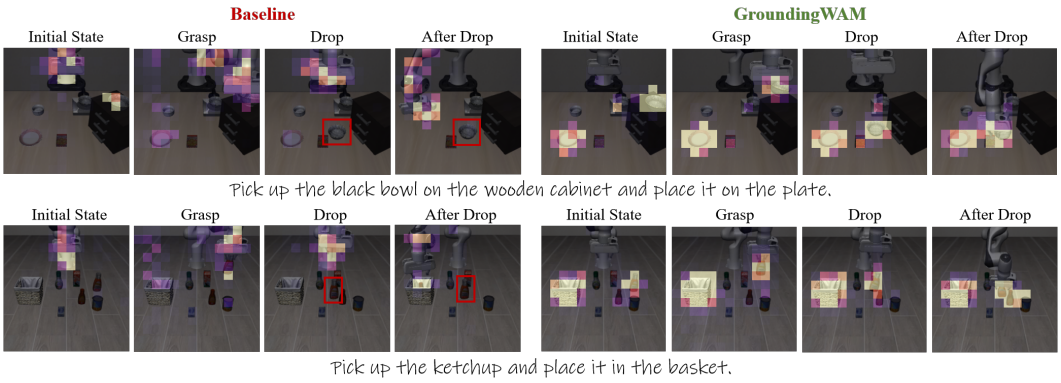


Figure 4: **Visual attention under object-drop perturbations.** Baseline WAM (left) and GroundingWAM (right) at the initial, grasp, drop, and post-drop stages. Tasks: placing a bowl on a plate (top) and ketchup in a basket (bottom). Red boxes mark the observed positions of dropped objects.

ingWAM reaches 90.25%. Auxiliary localization therefore provides a useful training signal, but the GAD configuration supports stronger recovery even without a box head. Together, the two evaluations support directly constraining internal information use beyond supervising localization outputs.

4.5 Attention Behavior during Manipulation

Spatial allocation after an object drop. Figure 4 compares action-to-vision attention in two manipulation tasks. The baseline often attends near the gripper or to surrounding regions, whereas GroundingWAM retains attention near the displaced object and its destination after the drop. These examples suggest that GroundingWAM maintains task-relevant attention when the familiar object–gripper relationship is disrupted.

5 Discussion and Limitations

Successful manipulation demonstrations contain strong correlations between object locations, gripper states, and task progression, allowing auxiliary box regression to fit recurring patterns without acquiring robust visual grounding. Action learning is susceptible to the same bias, as these correlations can predict demonstrated actions while becoming unreliable under execution disturbances. Motivated by this insight, we design GAD to directly guide internal attention toward task-relevant visual and language information. Improvements in task success and recovery from perturbations, together with attention visualizations, demonstrate the effectiveness of our approach.

Grounding is one component of reliable manipulation. Under our object-release protocol, GroundingWAM improves performance across all four suites, with the largest gains on Spatial, consistent with our focus on grounding. However, the smaller gains on other suites suggest that robust recovery requires capabilities beyond grounding. Completing tasks after regrasping, understanding task requirements, and achieving compositional generalization in long-horizon tasks remain important directions for future work.

6 Conclusion

We introduce GroundingWAM, which complements output-level prediction with Grounding Attention Distillation to guide action generation toward task-relevant visual and language information. Motivated by the limitations of auxiliary localization under strong correlations in successful demonstrations, our approach directly supervises internal information use. Improvements in task success, environmental generalization, and recovery from execution perturbations demonstrate its effectiveness. These findings highlight the importance of learning how to use task-relevant evidence, alongside learning to predict actions and future observations, for robust world action models.

AI USE STATEMENT

Generative AI tools assisted with literature retrieval, research discussion and experimental planning, code implementation and debugging, data processing and analysis, figure-authoring code, and manuscript drafting, translation, and internal review. The authors take full responsibility for the final manuscript, code, references, and interpretation of the results.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix B describe the learning objectives and grounding annotation construction. Appendix A documents the training and inference configurations, object-release protocol, scenario selection, metric aggregation, and visualization procedures. Results taken from prior work are distinguished from evaluations conducted in this study.

ETHICS STATEMENT

This work aims to improve the reliability of robotic manipulation under execution disturbances. It uses existing manipulation benchmarks and demonstration datasets. Applications of the proposed method should incorporate task-specific safety assessment and appropriate operational safeguards.

REFERENCES

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026.

Black Forest Labs. FLUX.2: Frontier visual intelligence. Official blog post, 2025. URL `https://bfl.ai/blog/flux-2`.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Pim De Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. *Advances in neural information processing systems*, 32, 2019.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38574–38583, 2026.

Jisang Han, Seonghu Jeon, Jaewoo Jung, René Zurbrügg, Honggyu An, Tifanny Portela, Marco Hutter, Marc Pollefeys, Seungryong Kim, and Sunghwan Hong. Geometric action model for robot policy learning. *arXiv preprint arXiv:2606.17046*, 2026.

Taisei Hanyu, Nhat Chung, Huy Le, Toan Nguyen, Yuki Ikebe, Anthony Gunderman, Duy Nguyen Ho Minh, Khoa Vo, Tung Kieu, Kashu Yamazaki, et al. Slotvla: Towards modeling of object-relation representations in robotic manipulation. *arXiv preprint arXiv:2511.06754*, 2025.

Haifeng Huang, Xinyi Chen, Yilun Chen, Hao Li, Xiaoshen Han, Zehan Wang, Tai Wang, Jiangmiao Pang, and Zhou Zhao. Roboground: Robotic manipulation with grounded vision-language priors. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22540–22550. IEEE, 2025.

Xiaosong Jia, Bowen Yang, Zuhao Ge, Xian Nie, Yuchen Zhou, Cunxin Fan, Yufeng Li, Yilin Chai, Chao Jing, Zijian Liang, et al. Guidedvla: Specifying task-relevant factors via plug-and-play action attention specialization. *arXiv preprint arXiv:2605.12369*, 1, 2026.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. Dart: Noise injection for robust imitation learning. In *Conference on robot learning*, pp. 143–156. PMLR, 2017.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.

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.

Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-h0.7: A latent world-action model from egocentric videos. *arXiv preprint arXiv:2605.00078*, 2026.

Jongjin Park, Younggyo Seo, Chang Liu, Li Zhao, Tao Qin, Jinwoo Shin, and Tie-Yan Liu. Object-aware regularization for addressing causal confusion in imitation learning. *Advances in Neural Information Processing Systems*, 34:3029–3042, 2021.

Physical Intelligence. openpi: LIBERO evaluation and training. Official repository documentation, 2026. URL `https://github.com/Physical-Intelligence/openpi/blob/main/examples/libero/README.md`. Accessed September 21, 2026.

Lu Qiu, Yizhuo Li, Yi Chen, Yuying Ge, Yixiao Ge, and Xihui Liu. Making foresight actionable: Repurposing representation alignment in world action models. *arXiv preprint arXiv:2606.12217*, 2026.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Yichao Shen, Fangyun Wei, Zhiying Du, Yaobo Liang, Yan Lu, Jiaolong Yang, Nanning Zheng, and Baining Guo. Videovla: Video generators can be generalizable robot manipulators. *Advances in neural information processing systems*, 38:95597–95621, 2026.

Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul Wohlhart, Sean Kirmani, Brianna Zitkovich, Fei Xia, et al. Open-world object manipulation using pre-trained vision-language models. *arXiv preprint arXiv:2303.00905*, 2023.

Xiatao Sun, Yuan Zhuang, Mateo Sanchez Lopez Negrete, Matei-Victor Coldea, Chen Liang, Haoyang Zhang, Che Liu, Ziyao Zeng, Shawn Li, Qian Wang, et al. Artificial foveated perception for mitigating shortcut learning in robotic foundation models. *arXiv preprint arXiv:2607.10655*, 2026.

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 92033–92052, 2025.

Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting copycat agents in behavioral cloning from observation histories. *Advances in Neural Information Processing Systems*, 33:2564–2575, 2020.

Chuan Wen, Jierui Lin, Jianing Qian, Yang Gao, and Dinesh Jayaraman. Keyframe-focused visual imitation learning. *arXiv preprint arXiv:2106.06452*, 2021.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, volume 2024, pp. 10641–10662, 2024.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

Yuyang Zhang, Wenyao Zhang, Zekun Qi, He Zhang, Haitao Lin, Jingbo Zhang, Yao Mu, Xiaokang Yang, Wenjun Zeng, and Xin Jin. Imagewam: Do world action models really need video generation, or just image editing? *arXiv preprint arXiv:2606.19531*, 2026.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024.

A Additional Experimental Details

This appendix provides the model configuration, object-release protocol, LIBERO-Plus evaluation details, and procedures for the box and attention visualizations in the main text.

A.1 Full GroundingWAM Configuration

The full model initializes its visual backbone from FLUX.2 Klein 4B foundation weights. The action expert, proprioceptive projection, and eight auxiliary 2D box queries are newly initialized; training starts from the foundation model rather than a pretrained robot policy. It trains for ten epochs (35,270 updates) with batch size 96 and learning rate 10^{-4}. Training takes approximately 22 hours on eight NVIDIA A800-SXM4 GPUs with 80 GB memory each. No perturbed or recovery trajectories are added to training. Appendix B details the annotations and optimization objectives.

The full configuration uses the capped attention-mass objective in Eq. 2 for AV and AT, with thresholds $(0.4, 0.1)$ and coefficient 0.001 for each path. Layers $\{8, 10, 12, 14, 16\}$ and heads $\{0, 4, 8, 12, 16, 20\}$ receive supervision. The visual, action, and box coefficients are 0.5, 1, and 0.2.

Table 6: Rollout settings for GroundingWAM and the ImageWAM-based component comparisons.

Setting	Value
Policy observations	Global and wrist RGB, language, 8-D proprioception
Image preprocessing	224×224 per camera; horizontal concatenation
Predicted / executed action steps	16 / 12
Diffusion sampling steps	10
Language context capacity at inference	512 tokens
Action dimension / control frequency	7 / 20 Hz
Initial settling period	30 control steps
Spatial / Object / Goal action budget	400 steps per rollout
Long action budget	700 steps per rollout
Action ensembling	Disabled
GroundingWAM standard evaluation	40 tasks $\times$ 50 rollouts; seed 42
Full release grid / Spatial ablation	Initial states 0–4; base seed 42
Box evaluation	Current observation at each replan; confidence ≥ 0.5

A.2 Online Release Intervention

Task and target coverage. The complete release evaluation in Table 2 covers 37 of the 40 LIBERO tasks. The three excluded Goal tasks ask the robot to open the middle drawer, turn on the stove, or push a plate; they do not require a lifted grasp-and-carry operation. Each eligible task contributes its explicitly transported objects. Six Long tasks transport two objects, giving 16 targets in that suite and 43 task–target pairs overall. For a multi-object task, each target is evaluated in a separate rollout; the original full-task success predicate is retained.

Each target is evaluated at progress thresholds $\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.65\}$ and initial states 0–4, with episode seed $42+$ initial-state index. Table 7 gives the resulting 344 conditions and 1,720 episodes per model. Long tasks 0, 1, 4, 6, 7, and 8 each contribute two targets; the other eligible tasks contribute one. Task IDs follow the zero-based benchmark order. No task or progress threshold is selected according to model success for this evaluation.

Geometric trigger. We release a task-relevant object during its own policy-controlled transport. Let $\mathbf{x}_0$ be its initial position, $\mathbf{x}_t$ its current position, and $\mathbf{x}_g$ the destination reference obtained from the task's placement predicate. Transport progress is

$$p_t = \frac{(\mathbf{x}_t - \mathbf{x}_0)^\top (\mathbf{x}_g - \mathbf{x}_0)}{\|\mathbf{x}_g - \mathbf{x}_0\|_2^2}. \tag{4}$$

The horizontal coordinates define this projection when the horizontal span is at least 5 cm; near-vertical transfers use three-dimensional coordinates. Destination references follow the corresponding simulator object or site, including moving drawers. The original Spatial-2, Object-0, and Goal-1

Table 7: Coverage of the complete object-release evaluation. Each task–target pair uses eight progress thresholds and five initial states. The Spatial row also defines the component comparison in Table 5.

Suite	Tasks	Task–target pairs	Conditions	Episodes
Spatial	10	10	80	400
Object	10	10	80	400
Goal	7	7	56	280
Long	10	16	128	640
Total	37	43	344	1,720

task specifications retain their fixed initial horizontal reference. The trigger additionally requires contact with both fingers, an object lift of at least 3 cm relative to its initial height, and three consecutive stable-grasp steps. The object's placement predicate must remain unsatisfied, lateral deviation must not exceed 10 cm, and progress must not exceed the destination. A zero progress threshold means the first eligible stable lifted grasp. Progress measures transport geometry rather than elapsed episode time.

Release and recovery. At the first eligible trigger, the gripper is commanded open for ten control steps (0.5 s); the other six action dimensions remain policy-controlled. The override persists across planning updates. Immediately after the intervention, any remaining pre-intervention action queue is discarded and the policy replans from the cur rent observation. Each rollout receives at most one release and retains its original task budget: 400 action steps for Spatial, Object, and Goal, and 700 for Long, after a 30-step settling period.

A drop is confirmed when bilateral grasp is absent and the target is at least 2 cm below its trigger height, within the release interval and the following ten steps. A regrasp is recorded after grasp loss when bilateral contact recurs for three consecutive steps after the forced-open interval. Tables 2 and 5 report final task success over all evaluated episodes. In Table 3, the drop rate uses all 1,720 episodes as its denominator. For the last two columns, eligible episodes are confirmed drops excluding those in which the entire task successfully terminates within the ten forced-open control steps, including the tenth step. Satisfying only the released object's placement condition does not qualify for exclusion. The resulting denominators are 1,425, 1,456, 1,458, and 1,449 for FastWAM, ImageWAM, Cosmos-Policy, and GroundingWAM, respectively. The numerators count a subsequent recorded regrasp (R), or both that regrasp and final task success (R&S), respectively.

Pairing and coverage. For each task–target–progress condition, FastWAM, ImageWAM, Cosmos-Policy, and GroundingWAM use the same initial simulator states and per-episode seeds. Each policy generates trajectory under the shared intervention rule. Overall success pools all 1,720 episodes; tasks with two transport targets contribute twice as many conditions as single-target tasks. Initial states are reused across progress thresholds.

Fixed Spatial component comparison. Table 5 uses all ten Spatial tasks, all eight transport thresholds, and initial states 0–4 with base seed 42, giving 80 conditions and 400 episodes per model. The Box-only and GAD-only evaluations reuse the baseline's fixed conditions and inference settings without searching for favorable release times or selecting tasks.

Inference settings for the release comparison. For each baseline, we retain the inference settings of its released policy. All models are evaluated under the same object-release protocol.

A.3 Box Evaluation and Visualization

Ground truth and assignment. We evaluate 2D boxes in the global camera at each policy replanning observation, before executing the newly predicted action. The simulator labels are axis-aligned image rectangles enclosing projections of three-dimensional OBBs fitted by principal component analysis to the object's MuJoCo box-geometry corners. The same projection defines the training la-

bels. These are geometric localization labels, distinct from visible segmentation masks or complete object meshes. Objects without valid geometry or projection are excluded from the ground-truth set.

We retain valid predictions with objectness confidence at least 0.5 and perform one-to-one Hungarian matching against the task's objects of interest, maximizing IoU with a center-distance tie-break. Assignment does not predict semantic instance identity. An unmatched ground-truth object receives IoU zero. The baseline and attention-only models have no Box head, so their localization scores are not applicable.

Qualitative examples. The recovery sequence in Figure 1(a) compares the ImageWAM baseline with the GAD + box configuration, while the GroundingWAM bars in panel (c) report the complete model. The two policies start from the same initial scene and both experience a confirmed drop; the baseline fails, while the GAD + box policy regrasps the object and completes the task. Each panel shows the corresponding policy's own trajectory.

Figure 3 shows auxiliary-box predictions from FastWAM on Object-7 (milk) and Spatial-9 (black bowl), and from ImageWAM on Long-2 (moka pot) and Goal-1 (black bowl). Neither box-only model uses attention supervision. FastWAM's three-dimensional predictions are projected into the global camera; ImageWAM predicts global-camera two-dimensional boxes directly. Each panel uses predictions computed from its current observation. Predicted boxes are associated with task objects by per-observation set matching, so a query index need not retain the same object identity across stages. Pink rectangles denote the transported target and cyan rectangles another task-relevant object or destination. Both are predictions; target IoU is computed against the simulator-projected box.

A.4 LIBERO-PLUS EVALUATION

Full benchmark evaluation. We evaluate all 10,030 LIBERO-Plus scenarios, grouped as 2,402 Spatial, 2,518 Object, 2,591 Goal, and 2,519 Long. Each uses one rollout, initial state index 0, and seed 42, following the official one-trial-per-task evaluation instructions. The policy is not fine-tuned on LIBERO-Plus, and no additional forced-release intervention is applied. These suite allocations differ from those in the original benchmark paper's appendix, while retaining its total of 10,030 scenarios. Scores from other publications follow their reported evaluation protocols.

Table 8: GroundingWAM on all 10,030 LIBERO-Plus scenarios. Each scenario contributes one episode.

Group	Successes / trials	Success (%)
Spatial	2,253/2,402	93.80
Object	2,344/2,518	93.09
Goal	2,055/2,591	79.31
Long	2,082/2,519	82.65
Camera viewpoint	1,231/1,599	76.99
Robot initial state	1,320/1,550	85.16
Language instruction	1,393/1,537	90.63
Lighting	1,125/1,142	98.51
Background texture	953/1,076	88.57
Sensor noise	1,396/1,601	87.20
Object layout	1,316/1,525	86.30
Overall	8,734/10,030	**87.08**

Outcome-independent subset for component comparison. The component-comparison subset contains 2,000 scenarios: 500 per suite, 50 per original task, and 71 or 72 per variation category within each suite. A fixed ordering with selection seed 20260912 covers available difficulty levels within task/category groups and allocates remaining slots according to capacity. Selection is independent of model outcomes. It is stratified rather than a uniform sample of all scenarios: levels 1–2 constitute 41.45% of the subset versus 38.34% of the full benchmark. All four configurations in Table 9 use the identical list, one rollout per scenario, initial state index 0, seed 42, and the full evaluation's inference settings.

Table 9: Comparison on the same scenarios on the 2,000-scenario subset (%). Each suite contains 500 episodes. This subset is distinct from the full-benchmark score.

Configuration	Spatial	Object	Goal	Long	Overall
ImageWAM	86.00	89.60	78.80	80.80	83.80
Box-only	91.80	89.20	81.20	84.20	86.60
GAD-only	94.80	92.20	81.20	81.60	87.45
GroundingWAM	94.00	95.40	80.60	85.20	88.80

A.5 Attention Visualization Protocol

Tasks and observations. Figure 4 uses Spatial-9 (*pick up the black bowl on the wooden cabinet and place it on the plate*) and Object-4 (*pick up the ketchup and place it in the basket*). The initial-state indices are 0 and 4, respectively, with episode seeds 42 and 46. Both tasks use release progress 0.2 and the intervention rules in Appendix A.2. The baseline is the ImageWAM policy, and the GroundingWAM panels use the complete model. The two policies start from the same task-specific initial state and subsequently generate their own trajectories.

Stage selection. The initial-state panel uses the first planning observation. The grasp panel uses the final planning observation before the release trigger, and the drop panel uses the first planning observation at or after a confirmed drop. The post-drop panel uses the last planning observation before the first regrasp, or the last recovery-attempt observation when no regrasp occurs. Each heatmap is paired with the RGB image used for that planning step. Stages are aligned by events rather than by a common elapsed time. Red rectangles highlight the dropped target's observed position and are display annotations, not predicted boxes or inputs to the policy.

Attention extraction. At each replan, we compute attention probabilities from the normalized, position-encoded queries and keys over all allowed keys. The maps average the 16 action queries, excluding box queries, at layers 8, 10, 12, 14, and 16 and heads 0, 4, 8, 12, 16, and 20 at the final denoising step. The same layer and head selection is used for both models.

Display normalization. Each camera contributes a 14×14 visual grid. Let $\bar{A}_j$ denote the averaged full-key probability on patch j, and let $\mathcal{V}_{\text{all}}$ contain the 392 current-image patches from both cameras. The displayed density is $392\bar{A}_j / \sum_{k\in\mathcal{V}_{\text{all}}} \bar{A}_k$, with a common color range $[0, 8]$ across policies and observations. A value of one corresponds to uniform allocation over the two views. The figure displays the global-camera portion without renormalizing it separately or rescaling individual frames. The patch grid is overlaid on the corresponding image; it is not a pixel-level segmentation prediction.

B Grounding Supervision and Model Details

B.1 Backbone and Attention Connectivity

GroundingWAM uses a FLUX.2 Klein 4B visual backbone and a 1,024-dimensional action expert following the ImageWAM architecture (Zhang et al., 2026). The model has five double-stream blocks followed by twenty single-stream blocks, with 24 attention heads and head dimension 128. In the double-stream blocks, language and image tokens have separate projections within the visual backbone. In the single-stream blocks, they form one backbone stream. The action expert retains its own projections and feed-forward parameters in both cases. The streams interact through masked joint attention while retaining separate projection weights.

Each 224×224 image contributes a 14×14 token grid. The global and wrist views are concatenated horizontally, giving 392 current-image tokens. A learned projection inserts the 8-dimensional proprioceptive state immediately after the valid language tokens. For the connectivity below, T denotes this language/proprioception prefix, V the current visual tokens, F the noisy future-image tokens, A the noisy action tokens, and B the auxiliary box queries. Rows are queries and columns are keys,

with one indicating an allowed attention block:

	T	V	F	A	B
T	1	1	0	0	0
V	1	1	0	0	0
F	1	1	1	0	0
A	1	1	0	1	0
B	1	1	0	0	1

(5)

Padded language keys are additionally masked. The same block connectivity is used in both transformer stages. Action inference conditions on the current observation without future-image tokens. Box queries share action-expert parameters but have a separate token-type embedding and positional identity, and their predictions cannot be read by action queries.

For every attention head, scores use normalized queries and keys with positional encodings. If $\mathcal{K}_q$ is the set of allowed keys for query q, the probability used by GAD is

$$A_{qj} = \frac{\exp\Big(\langle Q_q, K_j\rangle/\sqrt{d}\Big)}{\sum_{k\in\mathcal{K}_q}\exp\Big(\langle Q_q, K_k\rangle/\sqrt{d}\Big)}, \qquad j \in \mathcal{K}_q, \tag{6}$$

and zero otherwise. Here d is the head dimension. GAD extracts entries from these probabilities without changing the forward attention mask or its denominator.

B.2 Entity, Language, and Region Annotations

Training labels cover 40 tasks, 2,000 demonstrations, and 338,575 frames. A task-level annotation associates explicit noun phrases with simulator instance identities and roles. The 92 entity groups comprise 83 manipulated-object or destination groups and nine reference groups. The AV and AT target unions use the manipulated-object and destination groups, including articulated task targets; reference groups are excluded. Multi-object tasks retain all relevant entities throughout a demonstration because no temporal action-stage labels are supplied.

Entity phrases are mapped to the subword tokens whose character spans overlap the annotated phrase in the instruction. The correspondence follows the text encoder's tokenizer and prompt format. Padding, special tokens, proprioception, and pronouns are excluded from the target spans. Plural phrases can map to multiple simulator instances, in which case their visual target is the union of those instances' regions. The phrase masks identify token positions in contextualized representations, rather than assuming that an individual embedding contains an isolated object representation. All annotated instructions fit within the text encoder's context window.

Visual labels project each named object's simulator geometry through the global camera and normalize the resulting image box to $[0, 1]^4$. A patch enters the visual target set whenever it overlaps a valid task-relevant box with positive area. The mapping uses image-token coordinates, so global-camera patches are selected from the complete dual-view grid. Wrist-image patches are available to the policy but receive no direct region supervision. Box validity represents geometric projection validity, not visibility under occlusion. The masks therefore provide coarse spatial targets rather than instance segmentation.

Three drawer phrases have only parent-cabinet geometry. We exclude these entities from AV region targets while retaining their phrase tokens for AT. Similarly, a purely spatial destination without a projected region contributes no visual box. The two target sets are constructed separately, allowing language supervision to remain available when an entity lacks a valid visual region.

B.3 Supervised Attention and Loss Averaging

We supervise global layers $\{8, 10, 12, 14, 16\}$ and heads $\{0, 4, 8, 12, 16, 20\}$. All five layers belong to the single-stream stage, whose global numbering follows the five double-stream blocks. AV and AT use the same layer–head pairs and the 16 actual action queries, excluding the eight appended box queries and any padded action positions. For each query, AV sums probability over the relevant global-view patches, while AT sums probability over the relevant instruction tokens.

For each path and selected layer, we evaluate Equation 2 separately for every eligible sample, head, and action query, then divide their sum by the number of eligible entries. A sample contributes to AV only if its relevant visual union is nonempty, and to AT only if its phrase union is nonempty. We then average equally over selected layers. An empty valid set contributes zero for that path while retaining the other objectives. The AV and AT thresholds are 0.4 and 0.1, respectively, and each path has weight 0.001.

The logarithmic penalty saturates at zero above its threshold. It does not prescribe the relative attention among multiple target objects, the shape of attention within an annotated box, or zero attention outside the target. Because the probability denominator includes all allowed keys, increasing relevant mass still changes the competition with unannotated keys. Other heads are trained through the original objectives without a direct GAD loss.

B.4 Auxiliary 2D Box Prediction

The auxiliary branch has eight learned queries and a learned type embedding. A two-layer head with hidden dimension 256 and GELU activation outputs four sigmoid-normalized coordinates $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ and one objectness logit per query. Labels are the valid current global-camera object-of-interest boxes. The box and region targets derive from the same simulator geometry.

Given $Q = 8$ predictions $\{(\widehat{b}_i, z_i)\}_{i=1}^{Q}$ and $0 < m \le Q$ targets $\{b_j\}_{j=1}^{m}$, Hungarian matching selects an injective assignment π minimizing $\sum_j \|\widehat{b}_{\pi(j)} - b_j\|_1$. If $\mathcal{M}$ is the matched query set, the sample loss is

$$\mathcal{L}_{\text{box}}^{(t)} = \frac{1}{4m} \sum_{j=1}^{m} \|\widehat{b}_{\pi(j)} - b_j\|_1 + \frac{1}{m} \sum_{i \in \mathcal{M}} \text{BCE}(z_i, 1) + \frac{0.1}{Q - m} \sum_{i \notin \mathcal{M}} \text{BCE}(z_i, 0). \tag{7}$$

The final term is omitted when $m = Q$. When $m = 0$, the loss is the mean negative-objectness BCE across all queries. BCE denotes binary cross-entropy with logits, and the batch loss averages over samples. Predictions are permutation-invariant sets; query indices do not encode a fixed semantic object identity. Box queries do not receive the action-query GAD terms. Box-loss gradients reach the shared backbone and action-expert parameters, while the attention mask prevents direct action–box token interaction.

B.5 Optimization and Inference

We initialize the visual backbone from FLUX.2 Klein 4B foundation weights, retain the pretrained frozen VAE and text encoder, and newly initialize the action expert, proprioceptive projection, and box branch. The full model is trained jointly from these initializations, rather than fine-tuning a pretrained robot policy. Training uses AdamW with learning rate 10^{-4}, cosine decay, weight decay 0.01, gradient norm clipping at 1.0, batch size 96, and seed 42. Ten epochs correspond to 35,270 updates. The visual, action, and box loss coefficients are 0.5, 1.0, and 0.2, respectively; the AV and AT settings are specified above. Training hardware and duration are reported in Appendix A.1.

The visual and action objectives regress flow fields with independent Gaussian noise and independently sampled noise levels. For clean target x, noise $\varepsilon \sim \mathcal{N}(0, I)$, and noise level σ, the noisy input is $(1-\sigma)x + \sigma\varepsilon$ and the regression target is $\varepsilon - x$. Both flow losses use a noise schedule with shift 5 and timestep-dependent weighting. GAD is optimized jointly with these objectives. The frozen VAE and text encoder supply input features; the visual backbone, action expert, proprioceptive projection, and box branch are optimized jointly with the action-attention objective in Equation 3.

Inference uses ten action denoising steps, a 16-step prediction horizon, and execution of the first 12 actions before replanning. Action denoising conditions on language, current images, and proprioception, without future-image tokens or predicted box coordinates. No ground-truth boxes, entity spans, region masks, or external annotation model are required by the deployed policy.