ROUTEWAM: HIERARCHICAL ROUTING OF HETEROGENEOUS FUTURE REPRESENTATIONS FOR ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained vision foundation models (VFMs) provide rich perceptual knowledge for robot learning, yet manipulation often requires complementary capabilities distributed across models with different specializations. Consolidating this knowledge into a unified representation may dilute specialized strengths, while retaining separate experts leaves open how to organize their use and adapt them to robot tasks. To address these challenges, we propose RouteWAM, a hierarchical mixture-of-experts policy that learns and routes heterogeneous future representations for robot manipulation. Specifically, six independent experts predict future appearance, semantics, 3D geometry, and motion from current observations and language instructions, supervised by targets extracted from future observations in robot demonstrations. We organize these experts into visual, geometric, and motion groups, separating within-group selection among related representations from cross-group combination of complementary information. Both routing levels consider the current task context and candidate future predictions, enabling content-aware, token-wise selection for action generation. Future prediction and action imitation are jointly optimized, allowing the experts, routers, and action policy to adapt together. Experiments on LIBERO and RoboCasa demonstrate consistent improvements over the compared methods. On RoboCasa, RouteWAM surpasses DIAL by 6.3% and 11.3% on pick-and-place and articulated-object tasks, respectively. Real-world pick-and-place evaluations further demonstrate improved robustness under appearance and scene changes. The code will be released soon.

1 INTRODUCTION

Building robots that can perceive and act in complex, unstructured environments has long been a central pursuit of embodied AI Xu et al. (2024). Among recent advances, visuomotor policy learning stands out as a promising paradigm, in which robots learn to translate visual observations directly into control actions Chi et al. (2025). Central to this paradigm are pretrained vision foundation models (VFMs) such as SigLIP Zhai et al. (2023), DINO Caron et al. (2021), and JEPA Assran et al. (2023), which supply transferable visual representations that endow robotic perception and control with growing generalizability.

However, even a single robotic task often requires complementary visual capabilities Majumdar et al. (2023). For example, pick-and-place relies on semantic understanding for target identification and instruction grounding Shridhar et al. (2022), 3D geometry for grasp alignment and stable contact Sundermeyer et al. (2021), and motion awareness for tracking object transfer Chen et al. (2025). These capabilities are typically distributed across semantic encoders and specialized geometry and motion models Ranzinger et al. (2024); Shang et al. (2024); Chen et al. (2025); Zhang et al. (2026); Yuan et al. (2026). Directly integrating multiple VFMs into the policy increases inference overhead, while distilling their knowledge into a unified representation offers a more scalable alternative but faces three challenges Ranzinger et al. (2024); Shang et al. (2024): preserving specialized capabilities, organizing their use, and adapting them to robot observations. **First**, capability preservation and utilization: distilling heterogeneous VFM features into a unified representation may dilute model-specific strengths and hinder the policy's selective use of task-relevant visual capabilities Shang et al. (2024); Wang et al. (2026b). **Second**, structured expert routing: even when these capabilities are

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.

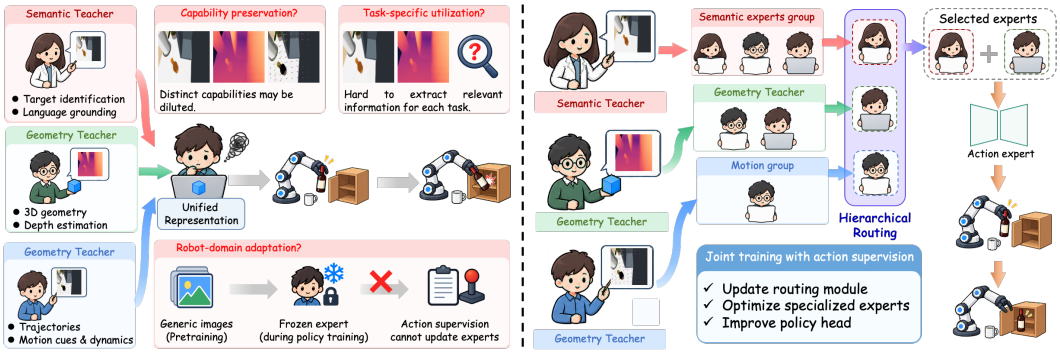


Figure 1: **Motivation and key idea of RouteWAM. Left:** Unified distillation may dilute specialized capabilities; action supervision cannot directly update frozen experts. **Right:** RouteWAM organizes future experts into visual, geometric, and motion groups, performs within- and cross-group routing, and jointly optimizes future prediction and action imitation.

retained in separate experts, how their outputs should be selected and combined remains a challenge. Visual experts can provide overlapping evidence for similar perceptual functions, as well as complementary information across functions Zamir et al. (2018); Liang et al. (2023), but flat routing does not explicitly distinguish these relationships. **Third**, robot-domain adaptation: beyond how experts are organized and routed, the experts themselves may need to adapt to robot observations. Some methods distill visual experts on generic images and freeze them during policy fine-tuning Wang et al. (2026b), which may limit their adaptation to task-relevant visual cues under domain shift Majumdar et al. (2023); Jiang et al. (2025).

To address these challenges, we propose RouteWAM, a hierarchical mixture-of-experts policy that learns and selects complementary future representations for robot manipulation, as shown in Fig. 1. Given the current observation and instruction, six separate experts predict future appearance, semantics, 3D geometry, and motion. Each expert is supervised by corresponding targets extracted from future observations in robot demonstrations. We organize these experts into visual, geometric, and motion groups, and dynamically combine their outputs to guide action generation.

RouteWAM performs routing in two stages. It first selects one expert within each of the visual and geometric groups, while the trajectory expert directly provides the motion candidate. It then selects which group-level candidates should condition the action policy. At both stages, each candidate is evaluated using the current observation and instruction together with its own predicted future features. Routing therefore depends on both the current task context and the future content predicted by each expert. Future prediction and action imitation are optimized jointly, allowing the experts, routers, and action policy to adapt together to robot demonstrations.

We evaluate RouteWAM on LIBERO Liu et al. (2023) and RoboCasa Nasiriany et al. (2024) against representative VLA and future-conditioned policies. On LIBERO, RouteWAM achieves the highest success rate among the compared methods on all four suites, improving the average from 92.6% for DreamVLA Zhang et al. (2026) to 94.7%. On RoboCasa, it outperforms DIAL Chen et al. (2026b) by 6.3% on pick-and-place tasks and 11.3% on articulated-object tasks under the same training-data setting. Controlled comparisons with flat routing and fixed expert combinations isolate the contribution of hierarchical selection, while further analyses examine knowledge grouping, routing decisions, and their changes during task execution.

2 Related Work

2.1 Visual Knowledge Transfer for Robot Learning

Pretrained vision foundation models (VFMs) provide reusable representations for robot control Majumdar et al. (2023). Multi-VFM transfer methods can be broadly divided into shared-representation and expert-based approaches. RADIO and Theia consolidate multiple teachers into a shared visual

encoder Ranzinger et al. (2024); Shang et al. (2024), whereas expert-based methods retain specialized representations for selective use. VER is the most related expert-routing method: its standard policy applies single-level routing and scalar CTA to visual experts distilled on generic images and frozen during policy fine-tuning, whereas RouteWAM jointly adapts future-prediction experts and performs candidate-conditioned hierarchical routing with coordinated HRA Wang et al. (2026b).

2.2 Future Representation Learning for Manipulation

Predictive methods for manipulation can be categorized into image prediction and structured future prediction. GR-1 and UP-VLA combine future image prediction with action learning Wu et al. (2024); Zhang et al. (2025), while Seer couples visual prediction with inverse dynamics Tian et al. (2025). Structured methods predict task-relevant cues, including dynamic regions, depth, semantics, and motion Zhang et al. (2026); Li et al. (2026). DreamWAM is the most related predictive method: it uses heterogeneous future supervision to shape a shared representation from which the action expert extracts task-relevant information, whereas RouteWAM retains independent future experts and explicitly selects task- and token-relevant candidates based on the current context and predicted content Yuan et al. (2026).

2.3 Fusion and Routing of Future Representations

Future representations can be integrated through representation-level interaction or explicit expert aggregation. DreamVLA uses block-wise attention masks, Faster-WAM aggregates features across network depths, and ProWAM modulates future features using execution progress and latent relevance Zhang et al. (2026); Zhao et al. (2026b); Zhu et al. (2026). FRAPPE combines parallel VFM-aligned experts through learned weights Zhao et al. (2026a). FRAPPE and ProWAM are the closest integration methods: unlike their weighted aggregation and progress-based hierarchy, respectively, RouteWAM performs candidate-conditioned, token-wise sparse routing first within perceptual groups and then across complementary groups Shazeer et al. (2017). This structure separates competition among related capabilities from composition across complementary capabilities, matching the expert-organization problem introduced in Section 1.

3 Methods

3.1 Problem Formulation and RouteWAM Overview

We consider learning a language-conditioned robot manipulation policy from expert demonstrations. The training dataset $\mathcal{D}$ consists of K demonstration trajectories. For any trajectory $\tau \in \mathcal{D}$, we denote $\tau = \big(\ell, \{(O_t, p_t, a_t)\}_{t=1}^{T}\big)$, where ℓ denotes the task instruction, O_t denotes the visual observation at time step t, p_t denotes the robot proprioceptive state, and a_t denotes the expert action. At each decision step, the policy predicts an action chunk $A_t = [a_t, \ldots, a_{t+H-1}]$ of horizon H. Given the current observation and task instruction, a vision-language backbone B_θ and a current-query bridge P_ϕ extract the action-conditioning tokens $C_t = P_\phi\big(B_\theta(O_t, \ell)\big)$. RouteWAM enhances these conditioning tokens with future information predicted from the current input. It then combines the enhanced tokens with the robot state p_t to generate an action chunk $\widehat{A}_t \sim \pi_\psi\left(\cdot \mid C_t^{+}, p_t\right)$, where C_t^{+} denotes the action-conditioning tokens augmented with future information. During training, future observations from the demonstration trajectories are used to construct appearance, semantic, geometric, and motion supervision. The ground-truth future information is used only as a training target. At inference time, the model receives only the current observation, task instruction, and robot state.

As illustrated in Fig. 2, RouteWAM predicts complementary appearance, semantic, geometric, and motion representations from the current conditioning tokens through independent future prediction branches. Modality-specific alignment modules transform these representations into action-relevant residuals. The Hierarchical Future-Latent MoE first selects experts within the visual and geometry groups, treats the trajectory expert as the motion candidate, and then selects two candidates from the resulting visual, geometry, and motion representations for token-wise weighted fusion. The augmented conditioning tokens are passed to the action model for action-chunk generation. The entire

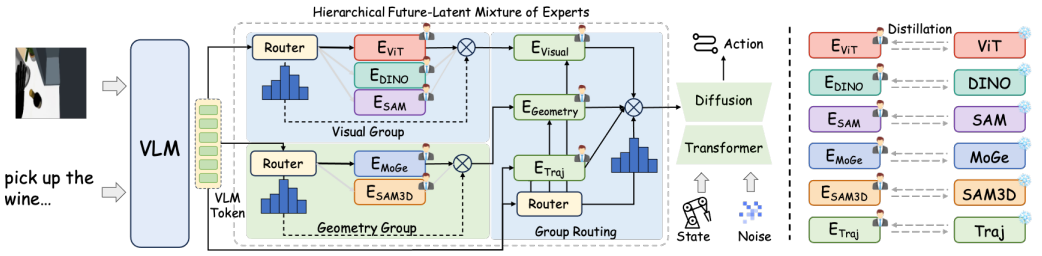


Figure 2: **Overview of RouteWAM.** The VLM encodes visual observations and language instructions into context tokens. Six future-latent experts predict heterogeneous future representations under supervision from frozen feature models. Their routed outputs condition the action expert. RouteWAM organizes experts into visual, geometric, and motion groups. For each conditioning token, Top-1 routing selects one expert from each of the visual and geometric groups, while the trajectory expert provides the motion candidate. Group-level Top-2 routing then selects and combines two candidates. Snowflakes denote frozen modules.

model is trained end to end by jointly optimizing future prediction and action imitation objectives:

$$\min_{\Theta} \mathbb{E}_{\tau \in \mathcal{D},\, t} \left[\lambda_{\text{act}} \mathcal{L}_{\text{act}} + \lambda_{\text{future}} \mathcal{L}_{\text{future}} \right],$$

where Θ denotes all trainable parameters of RouteWAM. At inference time, the model predicts future representations directly from the current input, selects and fuses relevant information through hierarchical routing, and continuously generates robot actions in a closed-loop manner.

3.2 Multi-Modal Future Prediction and Alignment

RouteWAM predicts future visual appearance and semantics ($\mathcal{E}_{\text{V}}$: RGB, DINOv3 Siméoni et al. (2025), SAM3 Carion et al. (2026)), 3D geometry ($\mathcal{E}_{\text{G}}$: MoGe-2 Wang et al. (2026a), SAM3D Chen et al. (2026a)), and motion dynamics ($\mathcal{E}_{\text{M}}$: Trajectory Karaev et al. (2025)) from the current conditioning tokens C_t. The six experts, collectively denoted by $\mathcal{E}$, predict at future offsets $\Delta = \{4, 8, 12, 16\}$.

Multi-modal future prediction. Each expert $e \in \mathcal{E}$ predicts $Z^e_{t,\delta} = W_e(C_t; q^e_\delta)$ with an independent predictor W_e and learnable horizon-specific queries q^e_δ, preserving the spatial structure of its corresponding representation. During training, a native-space decoder D_e produces $\widehat{Y}^e_{t,\delta} = D_e(Z^e_{t,\delta})$, supervised by targets $Y^e_{t,\delta}$ extracted from future observations $O_{t+\delta}$. RGB targets are appearance latents rather than raw pixels, while Trajectory targets encode motion from the current anchor to a future endpoint. We optimize $\mathcal{L}_{\text{future}} = \sum_{e \in \mathcal{E}} \lambda_e (\mathcal{L}^e_{\text{reg}} + \alpha \mathbf{1}[e \in \mathcal{C}] \mathcal{L}^e_{\cos})$, where $\mathcal{L}^e_{\text{reg}}$ is a regression loss normalized over valid future targets, and $\mathcal{L}^e_{\cos}$ is a feature-level cosine loss applied only to RGB, DINOv3, SAM3, and MoGe-2 ($\mathcal{C}$). The weights λ_e and α balance experts and loss terms, respectively. Future observations and their targets serve only as supervision and are never model inputs.

Future-to-action alignment. An expert-specific resampler S_e converts the multi-horizon predictions $Z^e_t = \{Z^e_{t,\delta}\}_{\delta \in \Delta}$ into fixed-size evidence tokens $E^e_t = S_e(Z^e_t) \in \mathbb{R}^{|\Delta| \times Q \times d_f}$, with $Q = 16$ and $d_f = 256$. An alignment module A_e then produces $(R^e_t, F^e_t) = A_e(C_t, \text{Flatten}(E^e_t))$, using C_t as queries and the $|\Delta|Q$ evidence tokens, flattened over the temporal and query axes, as keys and values. Here, R^e_t is a policy-conditioning residual matching the feature dimension of C_t, while F^e_t provides a low-dimensional routing feature for each current token. Resamplers and alignment modules have disjoint parameters across experts. Inference skips only the native-space decoders; all six latent prediction and alignment branches remain active.

3.3 Hierarchical Future-Latent Mixture of Experts

Although aligned expert residuals share an action-conditioning space, their relevance varies across tasks, manipulation phases, and token positions. Rather than placing all six experts in flat competition, we perform content-aware, token-wise routing within and across the semantic groups $\mathcal{G} = \{\mathrm{V}, \mathrm{G}, \mathrm{M}\}$.

Within-group expert selection. For token i, we project the current context into the routing space as $c_{t,i} = P_r(\mathrm{LN}_r(C_{t,i}))$. For each $g \in \{\mathrm{V}, \mathrm{G}\}$, the within-group router computes

$$\alpha^g_{t,i} = \mathrm{STTopK}_{k_g}\big(\{s_g(c_{t,i} + F^e_{t,i} + u_e)\}_{e\in\mathcal{E}_g}\big), \tag{1}$$

where u_e is an expert identity embedding and s_g is a group-specific scoring network. The scores consider both the current context $c_{t,i}$ and the candidate future content $F^e_{t,i}$, rather than the current observation and instruction alone. The singleton motion group requires no within-group selection, so we set $\alpha^{\mathrm{M}}_{t,i,\mathrm{Trajectory}} = 1$. Each group candidate is then expressed uniformly as $f^g_{t,i} = \sum_{e\in\mathcal{E}_g} \alpha^g_{t,i,e} F^e_{t,i}$ for $g \in \mathcal{G}$.

Cross-group candidate selection. The second routing level selects among the three group candidates:

$$\beta_{t,i} = \mathrm{STTopK}_{k_b}\big(\{s_b(c_{t,i} + f^g_{t,i} + b_g)\}_{g\in\mathcal{G}}\big), \tag{2}$$

where b_g is a group identity embedding and s_b is the cross-group scoring network. The final routing budget $(k_{\mathrm{V}}, k_{\mathrm{G}}, k_b) = (1, 1, 2)$ retains one expert within each of the visual and geometry groups, then selects two of the three group candidates.

Leaf-expert weighting and residual fusion. For an expert $e \in \mathcal{E}_g$, the two routing levels give the leaf weight $w^e_{t,i} = \beta^g_{t,i}\alpha^g_{t,i,e}$. The weighted residuals are injected into the current conditioning token:

$$C^+_{t,i} = C_{t,i} + \sum_{e\in\mathcal{E}} w^e_{t,i} R^e_{t,i}. \tag{3}$$

Here, $F^e_{t,i}$ provides evidence for routing, whereas $R^e_{t,i}$ conveys future information to the action model. Under the final sparse configuration, each token fuses exactly two leaf experts from different semantic groups.

Differentiable sparse routing. At each routing level, STTopK uses renormalized hard Top-K weights in the forward pass and a dense-softmax gradient in the backward pass. Specifically, given dense softmax weights p and their masked, renormalized Top-K counterpart p^{hard}, we use the straight-through estimator $\widetilde{p} = \mathrm{sg}(p^{\mathrm{hard}} - p) + p$, where sg denotes stop-gradient. This enables sparse forward fusion while allowing the routing networks and expert-specific alignment modules to be jointly optimized through the action objective.

3.4 Action Generation and End-to-End Optimization

Action generation and joint optimization. We retain the base policy's flow-matching action head, conditioned on C^+_t and the proprioceptive state p_t, to generate action chunks of horizon H. Its original flow-matching objective is denoted by $\mathcal{L}_{\mathrm{act}}$. The future predictors, alignment modules, routers, and action head are jointly optimized with

$$\mathcal{L} = \lambda_{\mathrm{act}}\mathcal{L}_{\mathrm{act}} + \lambda_{\mathrm{future}}\mathcal{L}_{\mathrm{future}}, \tag{4}$$

where λ_{act} and $\lambda_{\mathrm{future}}$ balance the two objectives. Future supervision encourages complementary predictions, while action supervision shapes their representations, alignment, and routing for control.

Hierarchical Routing Annealing. To prevent premature routing lock-in, we apply Hierarchical Routing Annealing (HRA) during joint training. HRA progressively reduces the routing budget $(k_{\mathrm{V}}, k_{\mathrm{G}}, k_b)$ from $(3, 2, 3)$ through $(2, 2, 3)$ to $(1, 1, 2)$, decreasing the number of fused leaf experts per token from six to five and finally to two. Unlike VER's CTAWang et al. (2026b), which anneals a single K for patch-wise routing, HRA jointly adjusts the within-group and cross-group budgets. The final configuration selects two experts from different semantic groups and is retained during late training and inference.

Inference. At inference, RouteWAM generates action chunks from (O_t, ℓ, p_t) without future-target extraction. The robot executes a prefix of each chunk and replans from the updated observation.

4 Simulation Experiments

4.1 Setup

Evaluation setup. We evaluate RouteWAM on the four LIBERO suites Liu et al. (2023) and RoboCasa GR1 Tabletop Nasiriany et al. (2024); Chen et al. (2026b). RoboCasa comprises 18 pick-and-place and six articulated-object tasks. For RoboCasa, we use the training data from DIAL's full-data comparison (1,000 demonstrations per task) and follow its evaluation protocol with 50 closed-loop episodes per task. We report success rates by LIBERO suite and RoboCasa task category. Future targets are used only for training supervision; at inference, the policy takes current RGB observations, language instructions, and proprioceptive states as inputs. Training and evaluation details are provided in Appendix A.

Baselines. We organize the comparison into three groups. First, we include the general-purpose visuomotor baseline Diffusion Policy (DP) Chi et al. (2025). Second, we compare with generalist vision-language-action policies, including Octo Team et al. (2024), OpenVLA Kim et al. (2024), and GR00T N1.6 NVIDIA et al. (2025). Third, we include methods that inject explicit spatial reasoning or predictive world knowledge into action generation: SpatialVLA Qu et al. (2025), CoT-VLA Zhang et al. (2026), DreamVLA Zhang et al. (2026), FLARE Zheng et al. (2025), and DIAL Chen et al. (2026b). The LIBERO baseline values follow the comparison collected by DreamVLA. RoboCasa baseline values follow the controlled comparison reported by DIAL.

4.2 Comparison with Existing Methods

Table 1: **The extended LIBERO experiments.** RouteWAM achieves the best or competitive performance across all tracks compared to previous approaches. The best results are **bolded**.

Method	Scores (%)				Avg.
	Spatial	Object	Goal	Long	
DP Chi et al. (2025)	78.3	92.5	68.3	50.5	72.4
Octo Team et al. (2024)	78.9	85.7	84.6	51.1	75.1
OpenVLA Kim et al. (2024)	84.7	88.4	79.2	53.7	76.5
SpatialVLA Qu et al. (2025)	88.2	89.9	78.6	55.5	78.1
CoTVLA Zhao et al. (2025)	87.5	91.6	87.6	69.0	83.9
DreamVLA Zhang et al. (2026)	97.5	94.0	89.5	89.5	92.6
RouteWAM	**98.2**	**96.1**	**92.3**	**92.5**	**94.7**

LIBERO. As shown in Table 1, RouteWAM achieves the highest success rate among the compared methods on all four LIBERO suites. It reaches 98.2% on Spatial and 96.1% on Object, compared with 97.5% and 94.0% for DreamVLA, respectively. The gains are larger on Goal and Long, where RouteWAM achieves 92.3% and 92.5%, exceeding DreamVLA by 2.8% and 3.0%. These results show consistent performance across different manipulation requirements, with improvements extending from spatial and object-centric tasks to goal-conditioned and long-horizon execution.

RoboCasa. As shown in Figure 3, RouteWAM achieves the highest success rates among the compared methods in both task categories using the same RoboCasa training data. It obtains 75.2% on pick-and-place tasks and 85.6% on articulated-object tasks, compared with 68.9% and 74.3% for DIAL, respectively. These correspond to improvements of 6.3% and 11.3%. The larger gain on articulated-object tasks highlights improved performance on multi-stage manipulation, where object placement is followed by closing a cabinet, drawer, or microwave.

4.3 Ablation Results

We conduct ablation studies on RoboCasa to evaluate the effects of the routing design, future-expert learning, and expert grouping. Each ablation setting is evaluated over five random seeds, $\{0, 1, 2, 3, 4\}$, and we report the mean and standard deviation across seeds. Results are reported as success rates on pick-and-place (PnP) and articulated-object (Artic.) tasks.

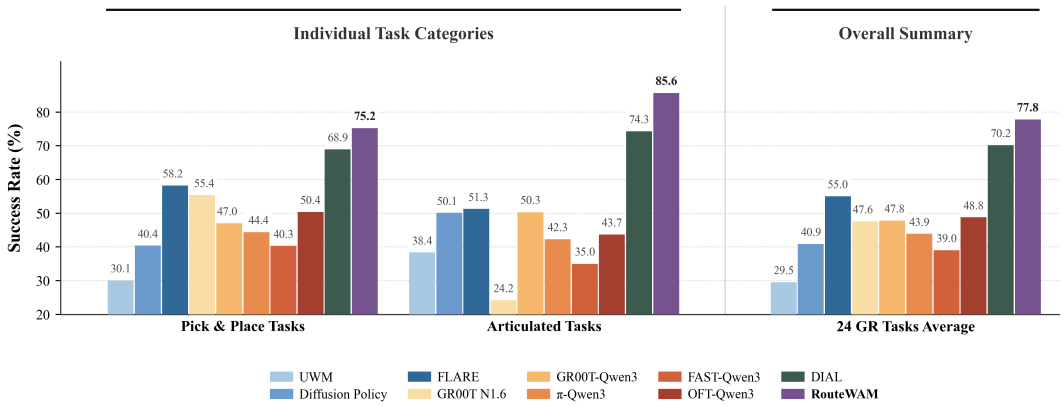


Figure 3: **Comparison on RoboCasa GR1 Tabletop.** Following the evaluation protocol of DIAL, we evaluate RouteWAM on 18 pick-and-place tasks and six articulated-object tasks, with 50 evaluation episodes per task. Bars show success rates (%). Baseline results are taken from the full-data comparison reported in DIAL. RouteWAM achieves the highest success rate among the compared methods in both task categories.

Table 2: **Contribution of individual future experts.** Success rates (%) on RoboCasa pick-and-place (PnP) and articulated-object (Artic.) tasks.

Config.	Vis.	Geom.	Mot.	PnP ↑	Artic. ↑
1. Ours	R, D, S	M, G	T	**75.0±1.2**	**85.2±1.8**
2. w/o R	D, S	M, G	T	73.1±1.9	82.5±2.1
3. w/o D	R, S	M, G	T	72.4±1.5	82.3±2.1
4. w/o S	R, D	M, G	T	72.8±1.4	83.2±1.9
5. w/o M	R, D, S	G	T	71.7±1.7	81.3±2.2
6. w/o G	R, D, S	M	T	72.0±1.9	82.1±1.8
7. w/o T	R, D, S	M, G	–	71.9±1.6	82.5±2.1

Abbreviations. PnP: pick-and-place; Artic.: articulated-object; R: RGB; D: DINOv3; S: SAM3; M: MoGe-2; G: SAM3D; T: trajectory. Vis., Geom., and Mot. denote the visual, geometric, and motion groups.

Contribution of future representations. Table 2 removes each future expert while preserving the remaining group assignments. Every removal lowers mean success in both task categories. Removing MoGe-2 causes the largest drops of 3.3% and 3.9% on PnP and Artic., respectively, despite retaining SAM3D, suggesting complementary contributions within the geometric group. Removing the trajectory expert reduces success by 3.1% and 2.7%, indicating the value of motion prediction alongside visual and geometric cues. Removing any visual expert also incurs a 1.9–2.9% drop. Together, these results support our motivation to preserve specialized future representations, whose complementary perceptual cues contribute to action learning.

Routing design. As shown in Table 3, Flat Top-2 routing achieves 68.4% and 78.2% on PnP and Artic., respectively. Enforcing the cross-group constraint improves the results to 70.4% and 80.6%. Built on this structure, hierarchical routing without annealing further reaches 73.2% and 83.5%, supporting our motivating distinction between selecting among related representations within groups and combining complementary information across groups. Using the current context alone reduces performance to 71.6% and 80.0%, indicating that candidate future content provides additional evidence for expert selection. For annealing, scalar CTA improves the group-constrained flat router to 72.1% and 82.4%, while applying the same rule independently to the hierarchical gates reaches 73.7% and 83.6%. Finally, coordinating the within- and cross-group budgets through HRA yields the best results of 75.0% and 85.2%. These results support the content-aware hierarchical routing and coordinated annealing in RouteWAM.

Future-expert learning. As shown in Table 4, adding independent branches without future supervision improves the base policy by 1.8 and 2.2% on PnP and Artic., respectively. Adding future supervision yields another 8.7 and 11.0%, reaching 75.0% and 85.2%. A shared predictor deepened to match the combined parameter count of the independent predictors still trails by 4.0 and 4.8%, supporting the value of specialized prediction paths. Blocking direct action gradients lowers success by 5.1 and 6.4%, highlighting the benefit of adapting future representations to the action

Table 3: **Ablation of routing structure and annealing.** Candidate conditioning denotes the use of candidate future features by the router. CTA is VER-style scalar global-K annealing; Ind.-CTA applies CTA independently to each hierarchical gate; HRA jointly anneals the within- and cross-group budgets. None trains directly with the terminal sparse budget. All sparse variants use the same final two-leaf inference budget.

Routing variant	Candidate conditioning	Schedule	PnP ↑	Artic. ↑
1. Flat dense fusion	–	–	67.7±2.0	77.1±2.7
2. Flat Top-2 routing	✓	None	68.4±1.9	78.2±2.5
3. Group-constrained flat Top-2 routing	✓	None	70.4±1.6	80.6±2.1
4. Group-constrained flat Top-2 routing	✓	CTA	72.1±1.5	82.4±1.9
5. Fixed cross-group expert pair	–	–	69.2±1.8	79.4±2.4
6. Context-only hierarchical routing	✗	HRA	71.6±1.7	80.0±1.9
7. Hierarchical Top-2 routing	✓	None	73.2±1.5	83.5±2.0
8. Hierarchical Top-2 routing	✓	Ind.-CTA	73.7±0.8	83.6±1.1
9. Ours	✓	**HRA**	**75.0±1.2**	**85.2±1.8**

Table 4: **Ablation of future-expert learning.** Success rates (%; higher is better) on RoboCasa pick-and-place (PnP) and articulated-object (Artic.) tasks.

Variant	Pred.	Sup.	Grad.	PnP ↑	Artic. ↑
1. Base policy	–	–	–	64.5±1.8	72.0±2.7
2. w/o future supervision	Ind.	✗	✓	66.3±1.7	74.2±2.6
3. Shared future predictor	Shared	✓	✓	71.0±1.6	80.4±2.3
4. Stop-gradient to future predictors	Ind.	✓	✗	69.9±1.7	78.8±2.4
5. Ours	**Ind.**	✓	✓	**75.0±1.2**	**85.2±1.8**

Table 5: **Sensitivity to expert grouping.** RoboCasa success rates (%). Expert abbreviations follow Table 2. All six experts are retained, with T fixed in the motion group M. Group sizes are listed as A/B/M. Inference uses within-group Top-1 and cross-group Top-2 routing.

Variant	Group A	Group B	Sizes A/B/M	PnP ↑	Artic. ↑
1. Ours	R, D, S	M, G	3/2/1	**75.0±1.2**	**85.2±1.8**
2. G moved to A	R, D, S, G	M	4/1/1	73.5±1.6	83.4±2.2
3. Swap S and G	R, D, G	M, S	3/2/1	72.8±1.7	82.6±2.3

objective. Together, these results support learning separate future experts under joint predictive and action supervision to preserve specialized capabilities while enabling task-driven adaptation.

Sensitivity to expert grouping. Table 5 examines alternative group assignments while retaining all six experts and the singleton trajectory group. Moving SAM3D to the visual group lowers mean success by 1.5% and 1.8% on PnP and Artic., respectively. Swapping SAM3 and SAM3D causes larger drops of 2.2% and 2.6% despite preserving the original group sizes. This size-preserving comparison suggests that grouping by perceptual function matters beyond group size, supporting our motivation to select among related representations within groups and combine complementary information across groups.

5 Real-World Experiments

5.1 Setup

We evaluate real-world duck pick-and-place on an AgileX PiPER robot under an in-distribution condition and three shifts in duck color, holder occupancy, and illumination (Figs. 5 and 4). Each method is evaluated over 25 trials per condition; hardware, training data, and success criteria are detailed in Appendix B.1.

5.2 Evaluation Results

Table 6 reports successful trials under each condition. RouteWAM achieves the highest success rate among the compared methods in all four settings. It completes 24 of 25 trials (96.0%) in the in-distribution setting and 62 of 75 trials (82.7%) across the three shifted conditions. DreamVLA, the

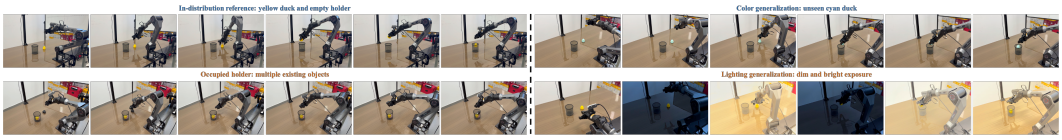


Figure 4: **Qualitative examples of real-world duck pick-and-place.** The four panels show the in-distribution reference with a yellow duck and an empty holder (top left), an unseen cyan duck (top right), an unseen black duck with a pre-occupied holder (bottom left), and dim and bright lighting conditions (bottom right).

Table 6: **Real-world duck pick-and-place evaluation.** Number of successful trials out of 25 under the in-distribution setting and three distribution shifts. The color-and-occupancy condition combines an unseen duck color with a pre-occupied holder. Shifted total aggregates the three shifted conditions (75 trials).

Method	In-distribution	Color shift	Color + holder occupancy	Illumination shift	Shifted total
DP	19/25	13/25	10/25	11/25	34/75
DIAL	21/25	16/25	14/25	15/25	45/75
DreamVLA	23/25	19/25	16/25	18/25	53/75
RouteWAM	**24/25**	**21/25**	**20/25**	**21/25**	**62/75**

strongest baseline, achieves 92.0% and 70.7%, respectively, corresponding to margins of 4.0% and 12.0%.

Across the color, occupancy, and lighting shifts, RouteWAM achieves success rates of 84.0%, 80.0%, and 84.0%, compared with 76.0%, 64.0%, and 72.0% for DreamVLA. These results establish improved empirical robustness on the evaluated real-world task and distribution shifts, while the preceding ablations isolate the contributions of hierarchical routing and future supervision.

6 More Information

Given the limited space, we provide additional details and analyses in the Appendix, including implementation details (Appendix A), simulation and real-world evaluation protocols (Appendix B), and visualizations of real-world executions and multimodal supervision sources (Appendix C). We further analyze how within-group expert selection and cross-group fusion evolve during task execution (Appendix C.3). Additional videos of real-world robot executions are provided in the supplementary material.

7 Conclusion

In this paper, we propose RouteWAM, a hierarchical mixture-of-experts policy for learning and utilizing heterogeneous future representations in robot manipulation. RouteWAM retains independent future prediction branches and organizes them into visual, geometric, and motion groups. Its hierarchical routing separates within-group expert selection from cross-group combination, using both the current task context and candidate future predictions to guide action generation. Joint future prediction and action imitation allow the experts, routers, and action policy to adapt together to robot demonstrations. Experiments on LIBERO, RoboCasa, and real-world pick-and-place, together with controlled ablations, support the value of structured, content-aware utilization of future knowledge for control. In the future, we will explore how to learn expert groupings automatically and enable conditional expert computation, while extending our evaluation to a broader range of real-world manipulation tasks.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and presentation. The authors reviewed and verified all AI-assisted edits and take full responsibility for the final content of the paper.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Tianxing Chen, Yao Mu, Zhixuan Liang, Zanxin Chen, Shijia Peng, Qiangyu Chen, Mingkun Xu, Ruizhen Hu, Hongyuan Zhang, Xuelong Li, et al. G3flow: Generative 3d semantic flow for pose-aware and generalizable object manipulation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1735–1744. IEEE, 2025.

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7220–7232, 2026a.

Yi Chen, Yuying Ge, Hui Zhou, Mingyu Ding, Yixiao Ge, and Xihui Liu. Dial: Decoupling intent and action via latent world modeling for end-to-end vla. *arXiv preprint arXiv:2603.29844*, 2026b.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Guangqi Jiang, Yifei Sun, Tao Huang, Huanyu Li, Yongyuan Liang, and Huazhe Xu. Robots pre-train robots: Manipulation-centric robotic representation from large-scale robot datasets. In *International Conference on Learning Representations*, volume 2025, pp. 81885–81905, 2025.

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labeling real videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–10. IEEE, 2025.

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

Yajie Li, Bozhou Zhang, Chun Gu, Zipei Ma, Jiahui Zhang, Jiankang Deng, Xiatian Zhu, and Li Zhang. From imagined futures to executable actions: Mixture of latent actions for robot manipulation. *arXiv preprint arXiv:2605.12167*, 2026.

Paul Pu Liang, Yun Cheng, Xiang Fan, Chun Kai Ling, Suzanne Nie, Richard Chen, Zihao Deng, Nicholas Allen, Randy Auerbach, Faisal Mahmood, et al. Quantifying & modeling multimodal interactions: An information decomposition framework. *Advances in Neural Information Processing Systems*, 36:27351–27393, 2023.

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.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *Advances in Neural Information Processing Systems*, 36:655–677, 2023.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. In *ArXiv Preprint*, March 2025.

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.

Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation model reduce all domains into one. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12490–12500. IEEE, 2024.

Jinghuan Shang, Karl Schmeckpeper, Brandon B May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. *arXiv preprint arXiv:2407.20179*, 2024.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on robot learning*, pp. 894–906. pmlr, 2022.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE international conference on robotics and automation (ICRA)*, pp. 13438–13444. IEEE, 2021.

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

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.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *Advances in Neural Information Processing Systems*, 38:35928–35959, 2026a.

Yixiao Wang, Mingxiao Huo, Zhixuan Liang, Yushi Du, Lingfeng Sun, Haotian Lin, Jinghuan Shang, Chensheng Peng, Mohit Bansal, Mingyu Ding, et al. Ver: Vision expert transformer for robot learning via foundation distillation and dynamic routing. In *International Conference on Learning Representations*, volume 2026, pp. 155122–155148, 2026b.

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.

Zhiyuan Xu, Kun Wu, Junjie Wen, Jinming Li, Ning Liu, Zhengping Che, and Jian Tang. A survey on robotics with foundation models: toward embodied ai. *arXiv preprint arXiv:2402.02385*, 2024.

Shanglin Yuan, Weiheng Zhao, Xin Shi, Haoyi Jiang, Xianda Guo, Liu Liu, Wenyu Liu, Wei Sui, and Xinggang Wang. Dreamwam: Beyond rgb future prediction for world action models. *arXiv preprint arXiv:2608.04996*, 2026.

Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3712–3722, 2018.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952. IEEE, 2023.

Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. Up-vla: A unified understanding and prediction model for embodied agent. *arXiv preprint arXiv:2501.18867*, 2025.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *Advances in Neural Information Processing Systems*, 38: 24195–24228, 2026.

Han Zhao, Jingbo Wang, Wenxuan Song, Shuai Chen, Yang Liu, Yan Wang, Haoang Li, and Donglin Wang. Frappe: Infusing world modeling into generalist policies via multiple future representation alignment. *arXiv preprint arXiv:2602.17259*, 2026a.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1702–1713. IEEE, 2025.

Weiheng Zhao, Haoyi Jiang, Xin Shi, Liu Liu, Fan Huang, Zhizhong Su, Wei Sui, and Xinggang Wang. Faster-wam: Efficient inference-time future conditioning for robust world action models. *arXiv preprint arXiv:2608.04404*, 2026b.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. Flare: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025.

Yijie Zhu, Zitong Yu, Wei Li, Hui Ma, Wen Li, Rui Shao, and Liqiang Nie. Learning to use imagination: Progress-conditioned future utilization for world action models. *arXiv preprint arXiv:2609.06578*, 2026.

Appendix Contents

A. Methodology and Implementation Details 13
A.1 Baseline and Expert Architecture 13
A.2 Future Supervision 13
A.3 Hierarchical Routing 13
A.4 End-to-End Optimization 14
B. Experimental Protocols 14
B.1 Real-World Experimental Protocol 15
C. Additional Visualizations 15
C.1 Real-World Task Executions 15
C.2 Processing Results of Pretrained Models 16
C.3 Routing Dynamics Analysis 16

A Methodology and Implementation Details

A.1 Baseline and Expert Architecture

We build on the DIAL policy architecture, which adopts Qwen2.5-VL-3B Bai et al. (2025) as backbone. We further augment the policy's conditioning tokens with routed future representations.

The current context contains 128 tokens with a feature dimension of 2048. Each of the six experts uses an independent two-layer Transformer predictor with eight attention heads and a hidden dimension of 256. We predict future representations at offsets of $\{4, 8, 12, 16\}$. An expert-specific resampler produces 16 evidence tokens per offset, yielding 64 tokens per expert. The alignment module maps these tokens to 2048-dimensional conditioning residuals and 256-dimensional routing features.

A.2 Future Supervision

Each expert predicts targets in its native representation space, as summarized in Table 7. Target shapes correspond to one future timestep, excluding the batch dimension. Future observations provide training targets and are not required during inference.

A.3 Hierarchical Routing

We divide the experts into three groups: visual $\{\mathrm{RGB}, \mathrm{DINO}, \mathrm{SAM3}\}$, geometry $\{\mathrm{MoGe}, \mathrm{SAM3D}\}$, and motion $\{\mathrm{Trajectory}\}$. Routing scores depend on the current context, predicted future content, and expert or group identity embeddings. The sparse routing procedure is summarized below.

Input: Current conditioning tokens.

1. Predict future representations with all six experts.
2. Resample and align each expert's predictions.
3. Select the Top-1 expert within each of the visual and geometry groups, and retain the single trajectory expert in the motion group.
4. Select the Top-2 group candidates.
5. Multiply within-group and cross-group weights, and fuse the corresponding residuals into the current conditioning tokens.

Output: Future-conditioned tokens for action generation.

The selected weights are renormalized at each routing level. During training, we use hard Top-K weights in the forward pass and dense-softmax gradients through a straight-through estimator. All six prediction and alignment branches remain active during inference, while native-space decoders are omitted.

Hierarchical Routing Annealing. Following VER's Curriculum Top-K Annealing (CTA), which anneals a scalar routing budget for a flat router, we extend the annealing variable to the structured

Table 7: Supervision targets for the six future experts.

Expert	Supervision target	Target shape
RGB	Visual features extracted by the policy's visual encoder	$16 \times 16 \times 1024$
DINO	DINOv3 patch features	$14 \times 14 \times 768$
SAM3	Semantic FPN features weighted by the object mask	$72 \times 72 \times 256$
MoGe	Intermediate geometry features	$50 \times 50 \times 512$
SAM3D	Stage-1 shape latents	$16 \times 16 \times 16 \times 8$
Trajectory	Motion descriptors from the current anchor to a future endpoint	$28 \times 28 \times 8$

budget of our two-level router. Let $s \in \{0, \ldots, T-1\}$ denote the optimizer-update index during joint training, and define

$$\mathbf{k}(s) = \big(k_{\mathrm{V}}(s), k_{\mathrm{G}}(s), k_b(s)\big),$$

where k_{V}, k_{G}, and k_b denote the visual-group, geometry-group, and cross-group budgets, respectively. For $T = 40{,}000$ optimizer updates, we use

$$\mathbf{k}(s) = \begin{cases} (3,2,3), & 0 \le s < 8{,}000, \\ (2,2,3), & 8{,}000 \le s < 16{,}000, \\ (1,1,2), & 16{,}000 \le s < 40{,}000. \end{cases}$$

The motion group contains a single trajectory expert and therefore uses $k_{\mathrm{M}}(s) = 1$ whenever it is non-empty. The schedule consequently reduces the number of fused leaf experts per token from six to five and finally to two.

For expert-removal and alternative-grouping ablations, the effective budgets are clipped as

$$\widehat{k}_g(s) = \min(k_g(s), |\mathcal{E}_g|)\,, \qquad \widehat{k}_b(s) = \min\big(k_b(s), |\mathcal{G}^+|\big)\,,$$

where $\mathcal{G}^+$ is the set of non-empty expert groups. The same schedule is used across tasks and random seeds.

The learning rate is linearly warmed up for the first $2{,}000$ optimizer updates (5% of training) and then follows a cosine schedule. HRA introduces no additional routing-specific warm-up; its initial dense phase serves as the exploration stage. At validation and inference time, the schedule is disabled and deterministic hard routing always uses the terminal budget $(1, 1, 2)$. All reported variants are evaluated using this same terminal budget.

A.4 End-to-End Optimization

During end-to-end joint training, we optimize the policy, future predictors, alignment modules, and routers together using the following objective:

$$\mathcal{L} = \lambda_{\mathrm{act}} \mathcal{L}_{\mathrm{act}} + \lambda_{\mathrm{future}} \mathcal{L}_{\mathrm{future}}, \tag{5}$$

where $\mathcal{L}_{\mathrm{act}}$ is the flow-matching action loss inherited from the baseline policy. $\mathcal{L}_{\mathrm{future}}$ aggregates the prediction losses of the six experts, with each loss normalized over valid targets. It includes Smooth-L1 regression and a cosine term for RGB, DINO, SAM3, and MoGe.

We set $\lambda_{\mathrm{act}} = 1$ and $\lambda_{\mathrm{future}} = 0.05$. We use AdamW with $\beta = (0.95, 0.999)$, weight decay 10^{-5}, a cosine learning-rate schedule, and BF16 precision. The global batch size is 32. Module-specific learning rates are listed in Table 8. The action prediction horizon is 16, and action generation uses four inference iterations.

B Experimental Protocols

LIBERO. We evaluate RouteWAM on LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long, with ten tasks per suite. We follow the demonstration datasets and task splits used by DreamVLA (Zhang et al., 2026). Observations comprise RGB images from workspace and

Table 8: Module-specific learning rates during end-to-end joint training.

Module	Learning rate
Backbone	1×10^{-6}
Conditioning bridge	2×10^{-6}
Action model	5×10^{-6}
Future predictors	1×10^{-5}
Native-space decoders	5×10^{-6}
Routers and conditioning adapters	3×10^{-5}

wrist cameras, a language instruction, and the robot's proprioceptive state. Actions use a seven-dimensional representation consisting of end-effector translation and rotation increments and a gripper command. Following DreamVLA's public evaluation implementation, each evaluation round contains 20 episodes per task using the benchmark-provided initial states. We report the mean task success rate for each suite. RouteWAM uses the 16-step action horizon specified in the implementation section.

RoboCasa. We follow DIAL's full-data robot-only setting on RoboCasa GR1 Tabletop (Chen et al., 2026b), which contains 18 pick-and-place tasks and six articulated-object tasks. The training set contains 1,000 demonstrations per task, totaling 24,000 trajectories. Inputs consist of RGB observations at 224×224 resolution, language instructions, and robot states. Following DIAL, states and actions use a 47-dimensional representation comprising 29 joint-space dimensions and 18 dimensions for the poses of both end effectors. Robot commands are executed in the 29-dimensional joint space. We predict 16-step action chunks and evaluate each task over 50 episodes. Success rates are averaged separately over the pick-and-place and articulated-object tasks.

Baseline results. The LIBERO baseline results are taken from the comparison reported in DreamVLA (Zhang et al., 2026). The RoboCasa baseline results are taken from the full-data comparison reported in DIAL (Chen et al., 2026b).

B.1 Real-World Experimental Protocol

We use a 6-DoF AgileX PiPER arm equipped with a 1-DoF gripper and RealSense D415 cameras. The task requires the robot to grasp a duck from the tabletop and place it inside a cylindrical holder. The training dataset contains 200 demonstrations collected using a yellow duck and an empty holder.

Each method is evaluated over 25 trials in four conditions: the in-distribution setting, an unseen duck color, a combined color and holder-occupancy shift, and altered illumination produced by changing the scene lighting. Non-yellow ducks and occupied holders do not appear in the demonstration dataset. A trial is successful if the robot completes the placement without human intervention and the duck remains inside the holder after release.

C Additional Visualizations

We provide additional visualizations of real-world task executions and the processing results of the pretrained models and tracking tools associated with our multimodal supervision.

C.1 Real-World Task Executions

Figures 6 and 7 show selected frames from real-world duck pick-and-place executions. The examples cover the reference setting, unseen duck colors, shifted initial placements, occupied holders, and lighting variations. Each sequence contains 12 selected frames illustrating the progression from approaching and grasping the duck to transporting and releasing it into the holder. Frames are ordered chronologically from left to right across the upper row and then the lower row, with timestamps shown beneath them. These sequences provide a closer view of policy execution across the illustrated scene configurations.

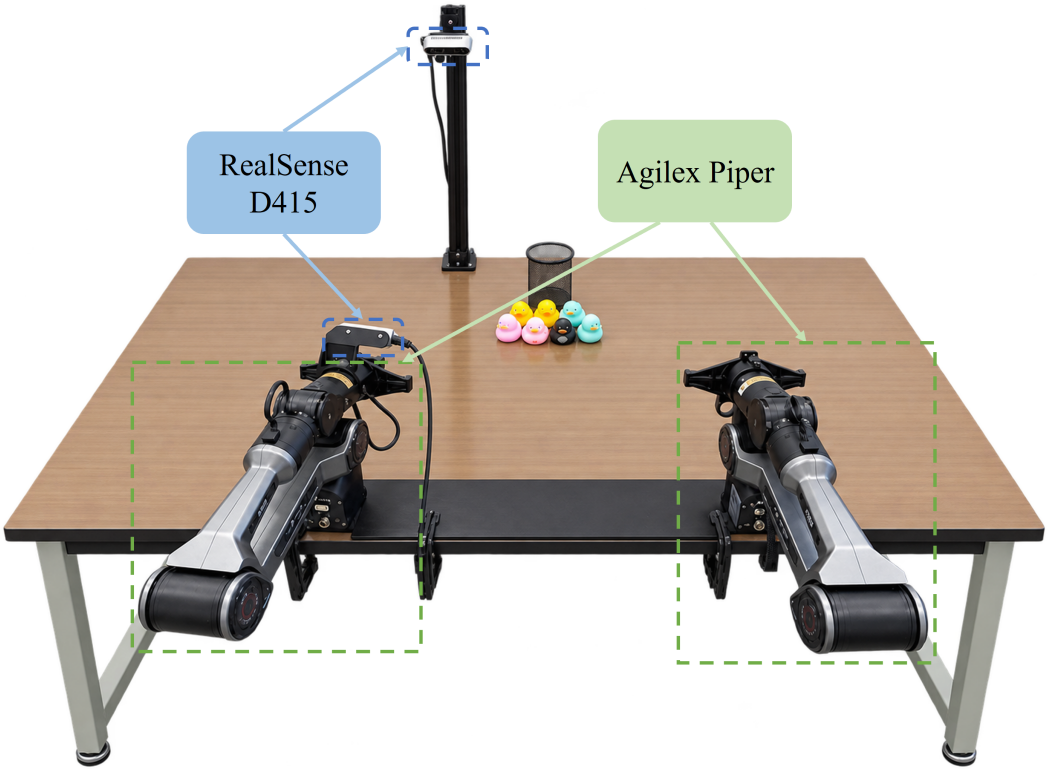


Figure 5: Real-world experiment setup.

C.2 Processing Results of Pretrained Models

Figure 8 visualizes the processing results of pretrained models and tracking tools on real-world observations. The first column shows the input RGB image. The remaining columns present SAM3 segmentation overlays, SAM3D shape visualizations, MoGe geometry maps, trajectory overlays, and DINOv3 feature visualizations. The rows cover reference, color-shift, placement, occupied-holder, and low-light scenes. These outputs illustrate different types of information extracted from the observations, including object regions, three dimensional shape, scene geometry, motion, and dense visual features. They provide an interpretable view of the heterogeneous knowledge sources used by RouteWAM. The native representations used for training are specified in Appendix A.

C.3 Routing Dynamics Analysis

Figures 9 and 10 report routing statistics aggregated over conditioning tokens during simulated and real-world task execution. In simulation, fusion gradually shifts from visual and geometric information toward geometric and motion information during placement and door closure. The real-world rollout emphasizes geometry and motion during arm raising, increases visual information during identification and grasping, and returns to geometry and motion during transport and release.

The candidate and retention panels distinguish within-group selection from final fusion. The trajectory expert is always the motion-group candidate, but contributes only to tokens that retain Motion. All six prediction and alignment branches run throughout; routing determines which two expert outputs enter each token's fusion. These stage-dependent patterns support our motivation to select among related representations within groups and adjust the combination of complementary information across groups.

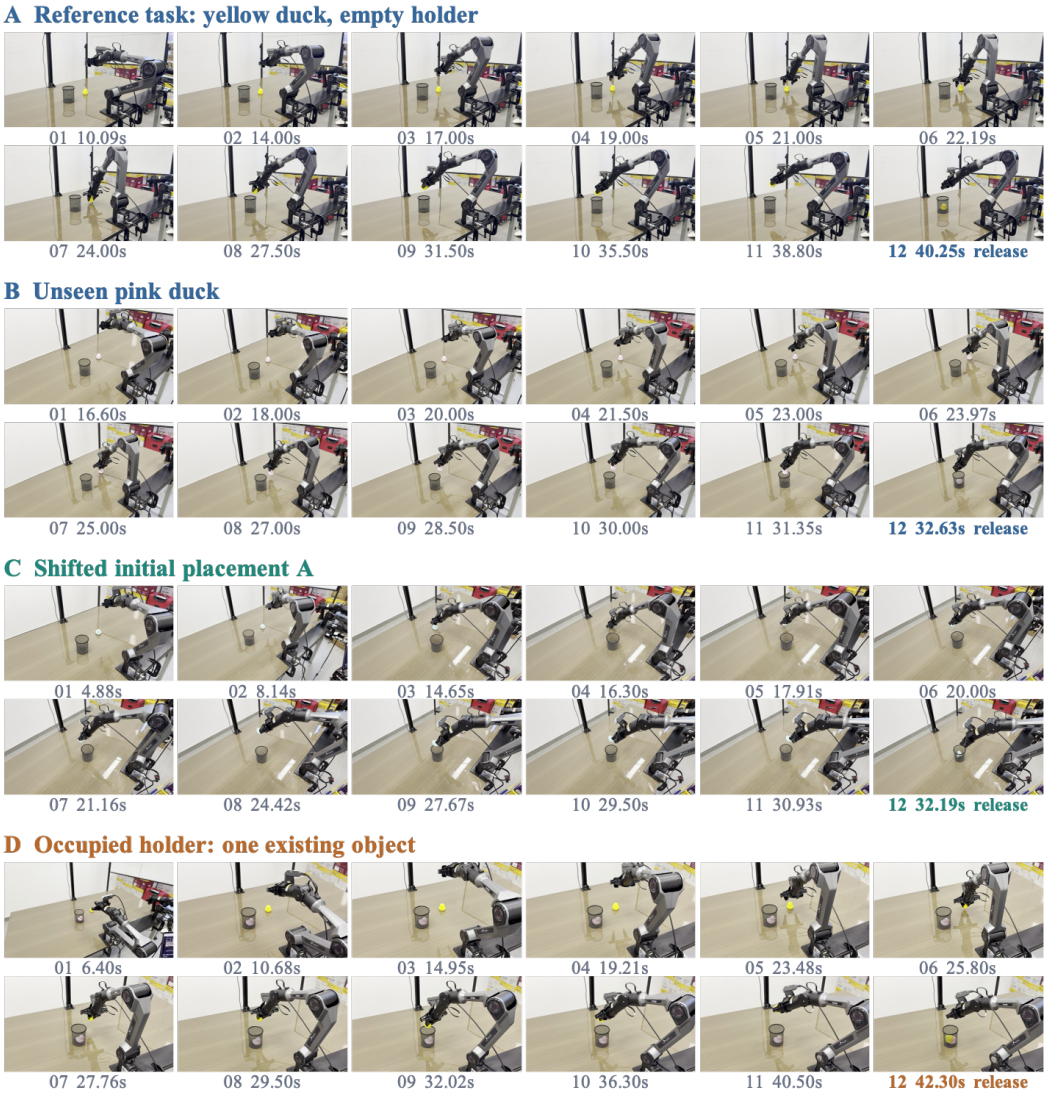


Figure 6: Real-world task executions under different scene configurations. Panels A–D show the reference setting with a yellow duck and an empty holder, an unseen pink duck, shifted initial placement A, and a holder containing one existing object, respectively. Each panel presents 12 selected frames, with the final frame marking release.

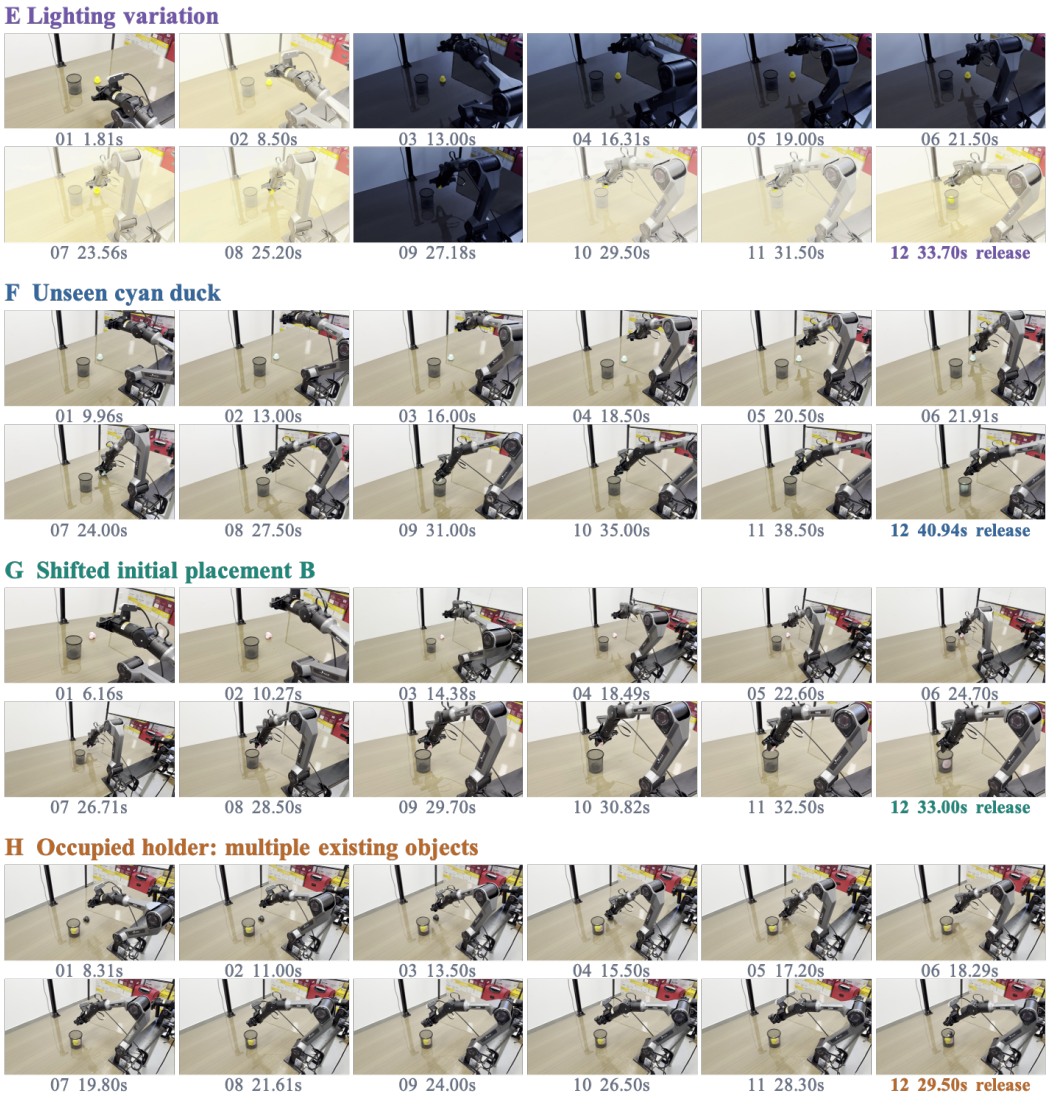


Figure 7: Additional real-world task executions. Panels E–H show lighting variation, an unseen cyan duck, shifted initial placement B, and a holder containing multiple existing objects, respectively. The frames depict the manipulation process through release, with timestamps provided below each image.

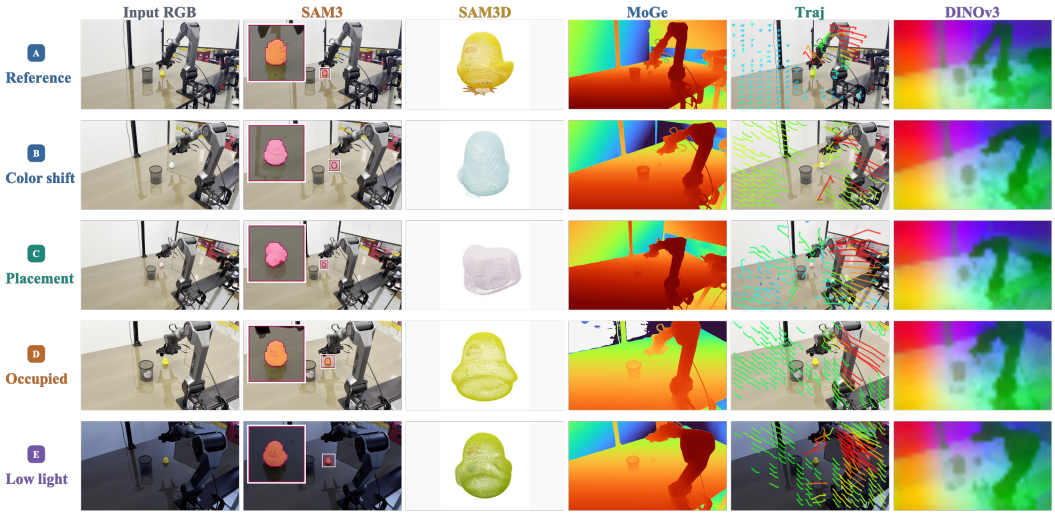


Figure 8: Processing results of the pretrained models and tracking tools. From left to right: input RGB, SAM3, SAM3D, MoGe, trajectory, and DINOv3 visualizations. Rows A–E correspond to the reference, color-shift, placement, occupied-holder, and low-light scenes.

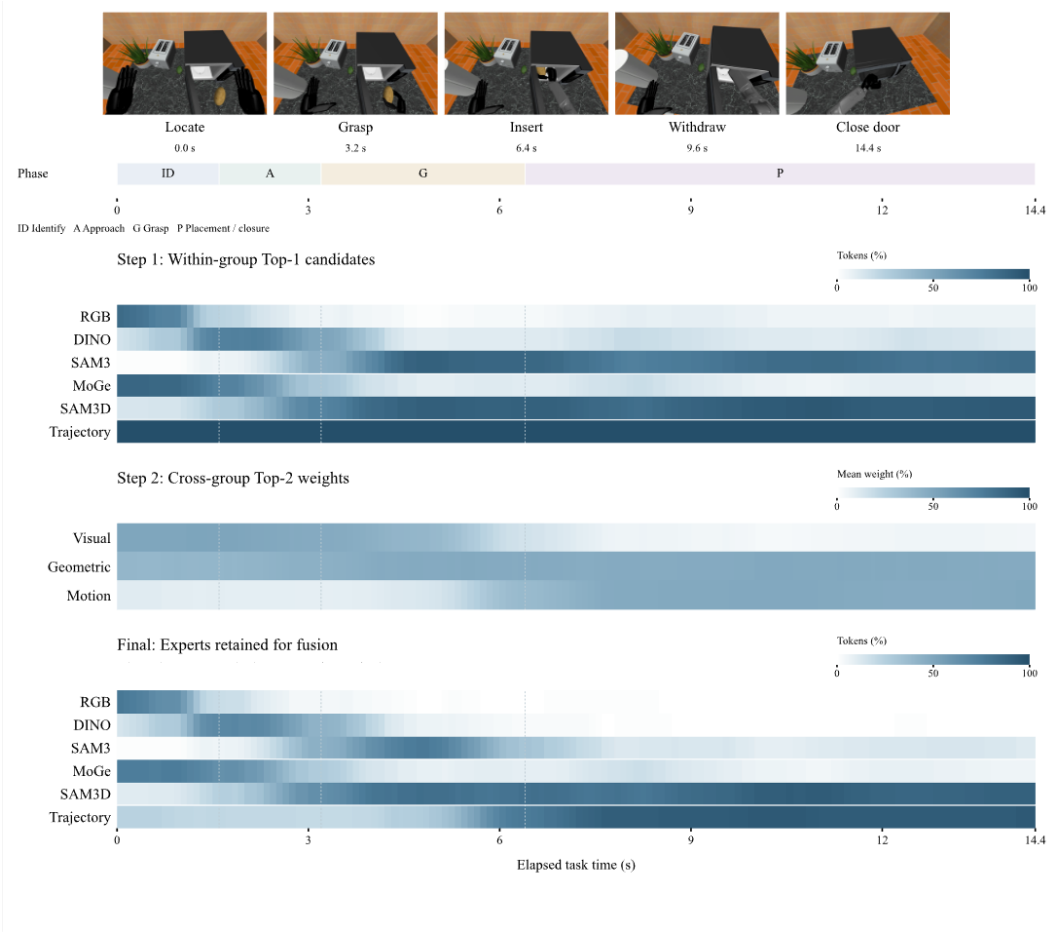


Figure 9: **Routing dynamics for simulated potato placement into a microwave.** Keyframes and the phase bar summarize execution stages. At each time, heatmaps aggregate conditioning tokens and show, from top to bottom: within-group Top-1 candidate frequencies (100% within each group); cross-group Top-2 mean weights (including zeros for unselected groups, totaling 100%); and final expert retention frequencies (totaling 200%, with two experts retained per token). Darker colors indicate larger values.

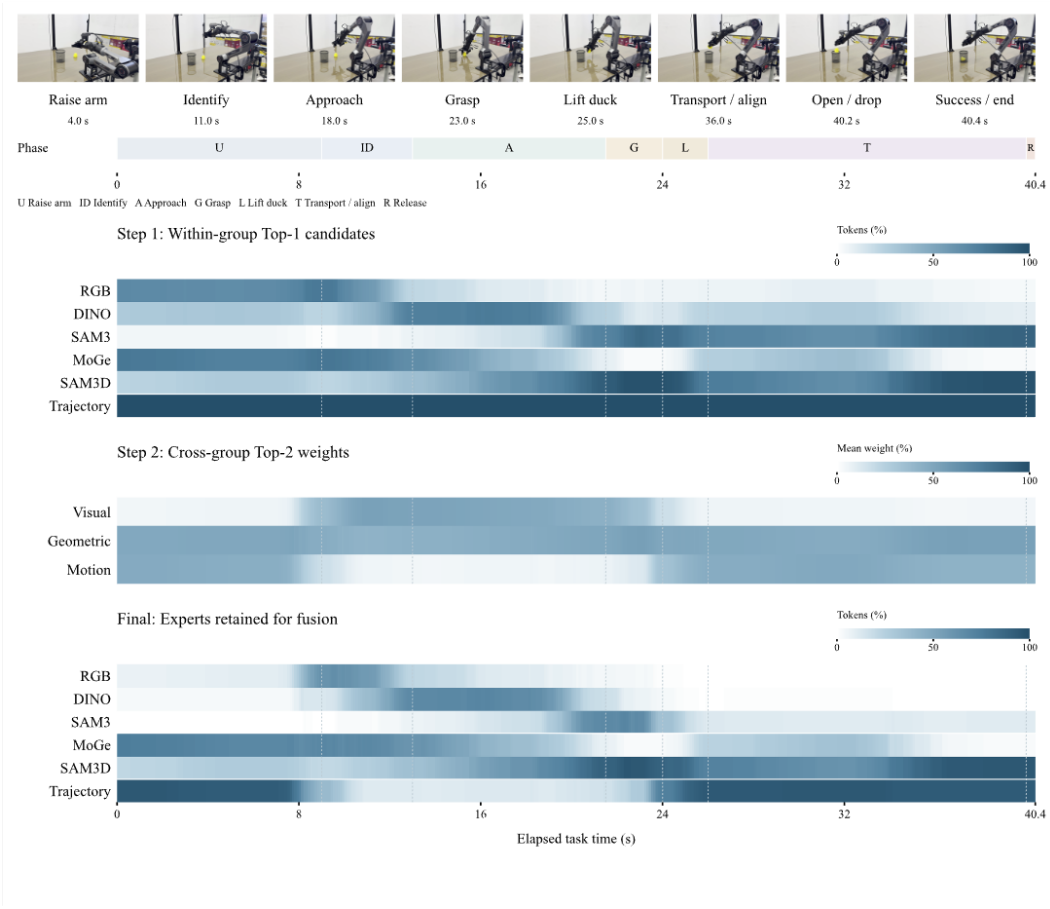


Figure 10: **Routing dynamics for real-world duck placement into a pen holder.** Keyframes span arm raising, identification, grasping, transport, and release. Panels follow Fig. 9: within-group candidate frequencies (100% per group), mean cross-group weights (100% across groups, including unselected-group zeros), and final expert retention frequencies (200% across experts). Retention denotes selection for fusion; all six prediction and alignment branches are computed.