RGMP: Recurrent Geometric-prior Multimodal Policy for Generalizable Humanoid Robot Manipulation

Xuetao Li[1*], Wenke Huang[1*], Nengyuan Pan[2], Kaiyan Zhao[1], Songhua Yang[1], Yiming Wang[3], Mengde Li[4], Mang Ye[1], Jifeng Xuan[1], Miao Li[1,4,5†]

[1]School of Computer Science, Wuhan University
[2]Faculty of Artificial Intelligence, Hubei University
[3]State Key Laboratory of Internet of Things for Smart City, University of Macau
[4]Institute of Technological Sciences, Wuhan University
[5]School of Robotics, Wuhan University
{xtli312, wenkehuang, yemang, jxuan, miao.li}@whu.edu.cn

Abstract

Humanoid robots exhibit significant potential in executing diverse human-level skills. However, current research predominantly relies on data-driven approaches that necessitate extensive training datasets to achieve robust multimodal decision-making capabilities and generalizable visuomotor control. These methods raise concerns due to the neglect of geometric reasoning in unseen scenarios and the inefficient modeling of robot-target relationships within the training data, resulting in significant waste of training resources. To address these limitations, we present the **R**ecurrent **G**eometric-prior **M**ultimodal **P**olicy (**RGMP**), an end-to-end framework that unifies geometric-semantic skill reasoning with data-efficient visuomotor control. For perception capabilities, we propose the Geometric-prior Skill Selector, which infuses geometric inductive biases into a vision language model, producing adaptive skill sequences for unseen scenes with minimal spatial common sense tuning. To achieve data-efficient robotic motion synthesis, we introduce the Adaptive Recursive Gaussian Network, which parameterizes robot–object interactions as a compact hierarchy of Gaussian processes that recursively encode multi-scale spatial relationships, yielding dexterous, data-efficient motion synthesis even from sparse demonstrations. Evaluated on both our humanoid robot and desktop dual-arm robot, the RGMP framework achieves 87% task success in generalization tests and exhibits 5× greater data efficiency than the state-of-the-art model. This performance underscores its superior cross-domain generalization, enabled by geometric-semantic reasoning and recursive-Gaussion adaptation.

Code — https://github.com/xtli12/RGMP.git

1 Introduction

Humanoid robots demonstrate substantial potential in performing diverse human-level tasks, ranging from adaptive decision-making to complex manipulation (Tong, Liu, and Zhang 2024; Li et al. 2024a). However, current research predominantly relies on data-driven approaches, which require extensive training datasets to achieve robust multimodal decision-making and generalizable visuomotor control (Zitkovich et al. 2023; Liu et al. 2024c; Intelligence et al. 2025). While these methods show promise in task-specific applications, they often overlook geometric reasoning and spatial awareness, limiting the ability of robot to perceive contextually under unseen environments. Consequently, there is a pressing need for approaches that combine data-efficient techniques with reasoning capabilities to enable more context-aware, and adaptable humanoid systems for real-world applications (Abu-Jassar et al. 2025; Skubis and Wodarski 2023).

Traditional Vision-Language Model (VLM) such as PaLM-E (Driess et al. 2023a) and InstructBLIP (Liu et al. 2024a) demonstrate remarkable capabilities in parsing semantic intent from language-vision inputs. These models leverage large-scale pretraining to generate task plans con-

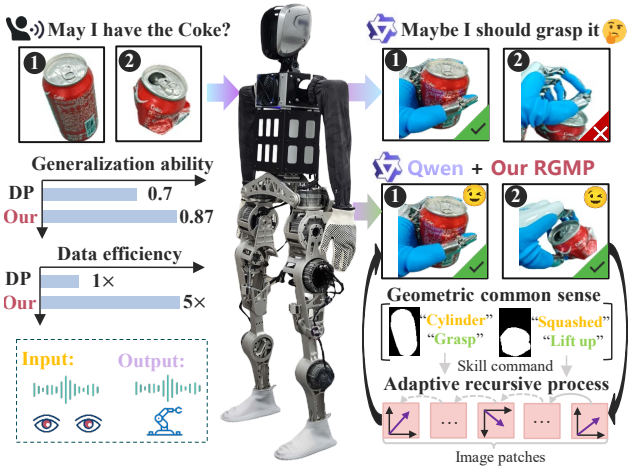


Figure 1: **Overview of our framework.** By applying semantic cues from human instructions with common sense information derived from visual perception, our RGMP formulates the robot-targets spatial relationships for tasks. RGMP achieves an 8% performance improvement and exhibits 5× greater data efficiency than Diffusion Policy.

*These authors contributed equally.
†Corresponding author

ditioned on visual observations, yet their ability to associate abstract instructions with contextually appropriate robotic skills remains constrained (Team et al. 2024). For instance, those models struggle to resolve ambiguities in skill selection (e.g., grasping vs. pinching) when confronted with targets of varying shapes under unseen scenes. This limitation stems from insufficient integration of spatial object geometry (e.g., bounding box, shape) with semantic task specifications, which is a gap exacerbated in dynamic environments where skill feasibility depends on generalized spatial reasoning (Rothert et al. 2024). Given this context, a fundamental question that arises is: **I)** *How can robots leverage spatial-geometric reasoning to enable feasible skill selection?*

Secondly, learning precise action policies from limited demonstrations remains an open challenge. While diffusion models (Chi et al. 2023) and transformer-based architectures (Vaswani et al. 2017) have shown promise in trajectory generation, their reliance on extensive training data (10k+ trajectories) and computational complexity (1–5 Hz inference rates (Zitkovich et al. 2023)) limits practical deployment. Imitation learning methods (Zhang et al. 2018) partially mitigate this by leveraging human priors, but they often overfit to demonstration-specific features, achieving merely 40–60% success rates on unseen objects (Liang et al. 2023). The crux lies in disentangling task-invariant visual features (e.g., context-based features) from task-specific motion patterns. Therefore, an additional intriguing question is: **II)** *How can inherent mechanisms of robot learn the generalized ability with limited demonstrations?*

To bridge these critical gaps, we introduce the **RGMP** (**R**ecurrent **G**eometric-prior **M**ultimodal **P**olicy), an end-to-end architecture that synergizes multimodal spatial-geometric reasoning with data-efficient visuomotor control. Regarding the issue of spatial-geometric reasoning discussed in **I)**, we present Geometric-prior Skill Selector (GSS): **the first framework to explicitly bridge geometric reasoning with semantic task planning** through a novel geometric-object decomposition mechanism. By incorporating geometric inductive biases into a VLM with minimal common-sense tuning, the GSS introduces a human-like decision-making process that mirrors how humans combine visual geometry and task semantics to select appropriate skills. Our geometric priors are plug-and-play, modular, and minimal (e.g., basic shape/affordance heuristics), requiring only 20 rule-based constraints for robust performance. Real-world deployment experiments on humanoid platforms confirm the ability of GSS to manipulate objects of diverse shapes in unseen scenarios via geometric consistency checks, demonstrating its practical utility beyond theoretical constructs.

Regarding the challenge of data efficiency discussed in **II)**, we propose Adaptive Recursive Gaussian Network (ARGN): a framework that dynamically models **spatial dependencies between robots and targets by adaptively reconstructing spatial memory**. In robotics, the high cost and labor intensity of data collection often result in limited dataset sizes, which can lead to overfitting if the visual processing network lacks careful architectural design to uncover latent data relationships. To this end, our ARGN employs Rotary Position Embedding (RoPE) to establish an implicit association between each observed image patch and the final executed action. We then introduce recursive computation in the Spatial Mixing Block to progressively model global spatial relationships from the first to the last visual patch. This recursive global connection forms the **spatial memory** of observed images for the robot, enabling it to identify end-effector positions most relevant to task execution. However, recursive computation is prone to vanishing gradients, which increases training difficulty and requires substantial data to mitigate this issue. To address this, we propose an Adaptive Decay Mechanism that dynamically controls the decay rate of historical memory, preventing the loss of key spatial memories and adaptively amplifying the weights of task-critical patches. Furthermore, we utilize Gaussian Mixture Models (GMM) to fit six Gaussian distributions, approximating a series of motions controlled by distinct joints of a six-degree-of-freedom robotic arm. Our contributions are threefold:

❶ ***A geometric-prior skill selector.*** We propose GSS to fuses a VLM with low-rank geometric adapters to dispatch parameterized manipulation skills from a pretrained library conditioned on common sense. Injecting shape-level commonsense biases permits GSS to select viable skills by their likelihood of satisfying latent geometric constraints, which is aligned with human reasoning without task-specific fine-tuning.

❷ ***A plug-and-play data-efficient visuomotor.*** We propose ARGN to modulate latent representations via adaptive decay mechanisms and rotary embedding to capture directional spatial dependencies in a temporally-consistent latent space. A hierarchical fusion block retains multi-scale visual cues and feeds them into a Gaussian Mixture encoder that factorizes 6-DoF trajectories into a compact mixture of full-covariates, enabling explicit goal-conditional density modeling under severe data scarcity.

❸ ***Comprehensive real-robot evaluation.*** Our RGMP undergoes rigorous evaluation on two physical robotic platforms, exhibiting robust performance by jointly coupling geometric-semantic reasoning with recursive Gaussian feature re-weighting. Compared to Diffusion Policy, RGMP achieves 87% success rate in generalization tests and exhibits 5× greater data efficiency.

2 Related Work

2.1 Vision-Language Model

Large Language Models (LLMs) have become pivotal in robotic task planning, offering capabilities such as common sense reasoning, and language comprehension, response generation (Ahn et al. 2022). Palm-E (Driess et al. 2023b), for instance, integrates visual, linguistic, and robot state data to facilitate dynamic task execution. LLMs can autonomously generate control code incorporating loops, conditionals, and subroutines, making them well-suited for complex perception-control integration tasks (Liang et al. 2023). Despite these advancements, current models still struggle to meet the diverse demands of real-world

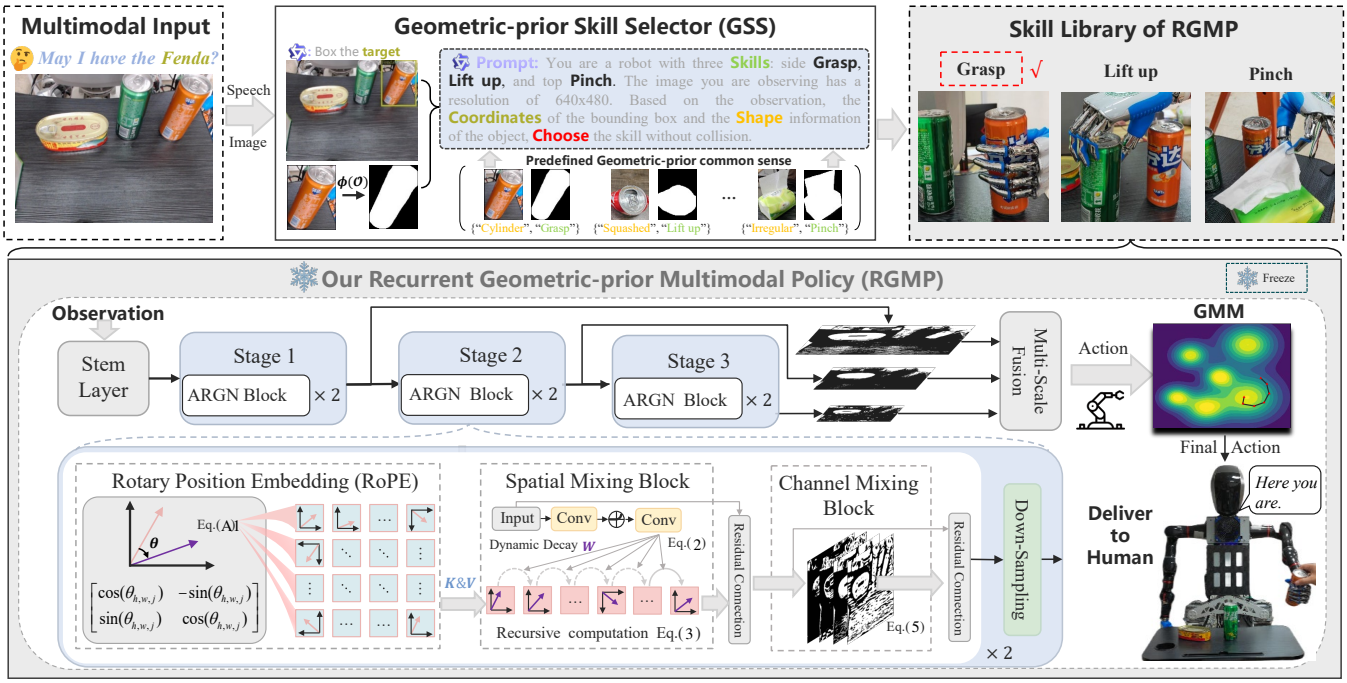


Figure 2: **Pipeline of RGMP**. Upon receiving a speech command, the robot utilizes GSS to identify and localize the target object. By integrating object coordinates, shape cues (from Yolov8n-seg (Yaseen 2024) model $\phi()$), and geometric-prior knowledge, the robot selects an appropriate skill from the skill library, each associated with a pretrained RGMP model. The selected RGMP model then executes the task precisely through adaptive recursive feature extraction and GMM-based refinement.

robotics. Recent progress in Vision-Language Model (VLM) has significantly advanced vision-and-language integration tasks. Models like InstructBLIP (Dai et al. 2023), InstructGPT (Ouyang et al. 2022), LLaVA (Liu et al. 2024b, 2023), and PALM (Chowdhery et al. 2022) leverage instruction tuning to improve image-text integration, setting new state-of-the-art benchmarks. However, their use in robotics applications faces considerable challenges due to real-world variability, platform heterogeneity, and the necessity for reliable action control (Driess et al. 2023b; Shridhar, Manuelli, and Fox 2023; Team et al. 2024; Huang et al. 2025a). These challenges often result in suboptimal performance in highly dynamic environments. To this end, we introduce the GSS to fuses the Qwen-vl (Yang et al. 2024) with low-rank geometric adapters to dispatch parameterized manipulation skills from a pre-computed library using common-sense priors. Compared to previous models, GSS effectively manages uncertainties and dynamic requirements, making it a stable solution for multimodal robotic tasks.

2.2 Learning-Based Action Generation

Recent advances in robot motion planning have increasingly prioritized learning-driven methodologies, particularly for developing sophisticated manipulation systems (Team et al. 2024; Liu et al. 2024c; Huang et al. 2025b,c). But those methods encounter three primary constraints: systematic dependence on predefined motion primitives (Cruciani et al. 2018, 2019), inadequate cross-domain adaptability (Liang et al. 2021), and inherent complexities in reward formulation (Kim et al. 2023; Zeng et al. 2018; Xu et al. 2021). Imitation learning (IL) provides an alternative paradigm (Zhang et al. 2018; Haldar et al. 2023; Li et al. 2023; Bogdanovic, Khadiv, and Righetti 2020) and has proven effective in physical deployments, yet its performance remains inextricably tied to demonstration fidelity and exhibits scalability limitations (Chen et al. 2021; Xu et al. 2022). Emerging diffusion-based generative frameworks have shown potential for robotic decision-making, utilizing multi-stage probabilistic optimization for trajectory synthesis (Zhang, Rao, and Agrawala 2023; Yoneda et al. 2023; Huang et al. 2023; Li et al. 2024b). However, their practical adoption faces constraints arising from suboptimal inference speeds, predominantly caused by the temporal latency of sequential reverse diffusion processes, rendering them unsuitable for time-sensitive applications (Dong et al. 2024). To overcome these challenges, we introduce RGMP that synergizes objective-aware action synthesis with statistical motion modeling. Unlike conventional diffusion architectures, our vision-guided planner implements computationally efficient multimodal action inference, eliminating resource-intensive iterative denoising while maintaining decision robustness. The proposed hierarchical architecture achieves seamless visuomotor integration through distilled action primitives and adaptive probability distributions, effectively unifying strategic task decomposition with dynamic actuator coordination.

3 Methodology

Our RGMP consists of two key components: the GSS and the ARGN. The GSS translates verbal commands and visual cues into executable skill sequences using geometric-prior common sense, while the ARGN leverages a pretrained skill model and processes the RGB image from the ego-centric

camera to predict the joint angles required for robotic manipulation. We train the policy to decode implicit geometric information from RGB by associating 3D spatial cues with robotic action labels and commonsense reasoning, avoiding costly explicit 3D reconstruction in favor of an efficient implicit representation. The whole pseudocode for our pipeline is presented in Algorithm 1.

3.1 Geometric-prior Skill Selector

Motivation. A key challenge in robotics is fine-grained skill selection (e.g., grasping vs. pinching) for diverse-shaped targets or in unseen scenes. Traditional VLMs, despite enabling object recognition and localization, fail to map semantic observations to accurate actions due to overlooking **geometric priors** in vision-action mapping. This motivates our pioneering GSS framework, which bridges geometric reasoning and semantic task planning via a novel geometric-object decomposition mechanism.

The GSS comprises two stages. In the first stage, VLM (Bai et al. 2023) is utilized to interpret human commands, enabling the robot to identify and localize the target object within the observed image. In the second stage, based on the bounding box obtained from the first stage, the system analyzes the target object's common sense information, including its relative position and its shape information. Subsequently, the system selects the pretrained skill model from an skill library accordwing to the output of the GSS. The planning function operates through:

$$\mathcal{P} = plan(\mathcal{I},\ \mathcal{O}\,|\,\mathcal{C}\ ), \tag{1}$$

where $\mathcal{P}$ is the generated action plan, $\mathcal{I}$ denotes the current user instruction, $\mathcal{O}$ is a current visual observation, and $\mathcal{C}$ represents a predefined context (instruction, prompt, and common sense) that consists of n examples $\{(\mathcal{I}_i, \mathcal{O}_i, \mathcal{P}_i)\}_{i=1}^{n}$, enabling in-context learning.

Specifically, the observation $\mathcal{O}$ is an RGB image annotated with a bounding box by the VLM. Subsequently, the VLM generates an executable skill based on the predefined context C and the geometric-prior common sense, which includes relative position and the shape information. For example, when the instruction is *"I want Fanta"*, our pipeline adheres to the context *"Please box the target object in the instruction"* to identify the *"Fanta"* can among various other items and apply Yolov8n-seg to get the shape information of Fanta. The VLM subsequently synthesizes operational directives by integrating its established contextual framework C with geometric-based prior reasoning (e.g., "cylindrical" $\rightarrow$ "grasp"; "squashed" $\rightarrow$ "lift up"). Our GSS are plug-and-play, modular, and minimal (e.g., basic shape/affordance heuristics), requiring only 20 rule-based constraints for robust performance (please refer to the Appendix A in supplements for implement details of GSS and skill library).

3.2 Adaptive Recursive Gaussian Network

Motivation. In robotic tasks, understanding spatial relationships in the vision of robot is essential. The robot must identify which parts of the scene correspond to its end-effector position. Previous methods often struggle to uncover the underlying relationships between different image regions in unseen scenes due to inherent limitations in visuomotor representation learning, which hinders the generalized ability. To address this issue, we propose the ARGN framework, which is designed to adaptively model comprehensive **spatial dependencies** between the robot and target objects in unseen environments, while mitigating overfitting in scenarios with limited training data.

In our framework, we apply recursive operation to get the global connection, which establishes the **spatial memory** of observed images. This memory mechanism enables the identification of end-effector positions most relevant to task execution. However, recursive computation inherently suffers from vanishing gradients, increasing training difficulty and demanding substantial data to mitigate this limitation. To address this, we propose an Adaptive Decay Mechanism (ADM) to dynamically **controls historical memory decay rates** to prevent the vanishing of key spatial memories, and adaptively amplifies weights for task-critical patches. In Stage 1, the input F_0 is processed by the Spatial Mixing Block, where ADM generates content-adaptive decay factors $\mathcal{W}$ to regulate memory retention.

$$\mathcal{W} = \sigma(\mathcal{C}_{1\times1}(SReLU(\mathcal{C}_{3\times3}(F_0))))), \tag{2}$$

where $\mathcal{C}_{1\times1}(\cdot)$ represents a convolutional operation with 1×1 kernel that enable re-calibrate channels, $\sigma(\cdot)$ denotes the Sigmoid activation function. RoPE is then applied to encode positional information through rotational transformations, enhancing sensitivity to relative spatial offsets with-

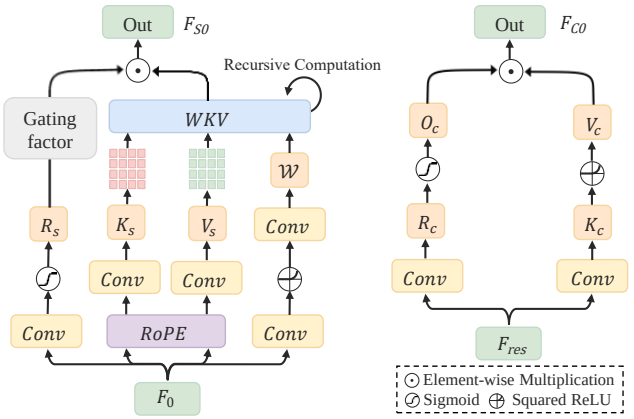
Algorithm 1: The RGMP Framework

Input: Training epochs E, conversation round T, human speech $\mathcal{I}$, human demonstrations $\mathcal{D}$ with capacity M, VLM model Q, RGMP $\mathcal{G}_m$
Output: Actions of robot a^*

for $i = 1, 2, ..., M$ **do**
  $d_i \leftarrow (\mathcal{O}_i, \mathcal{J}_i)$ through Eq. (12)
  $\mathcal{D} \leftarrow d_i$
return $\mathcal{D}$

`/* RGMP Training pipeline: */`
for $e = 1, 2, ..., E$ **do**
  $F_0 \leftarrow Stem(\mathcal{O}_i)$
  $\mathcal{W}, K_s, V_s \leftarrow \mathcal{A}(F_0), \mathcal{R}(F_0)$ by Eq. (2))
  $F_1, F_2, F_3 \leftarrow \mathcal{S}(K_s, V_s, \mathcal{W})_{\times3}$
  $a_{in} \leftarrow \mathcal{M}(F_1, F_2, F_3)$ by Eq. (7)
  $\mathcal{L} \leftarrow (a_{in}, a_{ground})$ through Eq. (8)
  $\mathcal{G}_m \leftarrow \mathcal{G}_m - \eta\nabla\mathcal{L}$
  $\Theta \leftarrow \mathcal{J}_i$ in $\mathcal{D}$ through Eq. (9)
return $\mathcal{G}_m, \Theta$

`/* Inferencing pipeline: */`
for $t = 1, 2, ..., T$ **do**
  $Box(x_1, y_1, x_2, y_2) \leftarrow \mathcal{Q}(\mathcal{I}, \mathcal{O}|\mathcal{C})$ by Eq. (1)
  $\mathcal{O}_s \leftarrow (x_1, y_1, x_2, y_2),\ \phi_a(\mathcal{O}, Box)$
  $\mathcal{P} \leftarrow \mathcal{Q}(\mathcal{I}, \mathcal{O}_s|\mathcal{C})$
  Voice $\leftarrow$ response in $\mathcal{P}$
  if 'Skill' in $\mathcal{P}$ **then**
    $a^* \leftarrow \mathcal{G}_m(\mathcal{O})$ through Eq. (11)
  return a^*



(a) Spatial Mixing Block (b) Channel Mixing Block

Figure 3: **Structure of (a) Spatial Mixing Block and (b) Channel Mixing Block.** (a) The Spatial Mixing Block integrates an ADM module to get Dynamic Decay $\mathcal{W}$ during recursive operations, and employs RoPE to introduce directional awareness relative to spatial positions. These mechanisms collectively enhance the ability of block to integrate spatial information effectively. (b) The Channel Mixing Block reallocates channel-wise feature responses by integrating correlations between channels.

out learnable position parameters. After applying RoPE, we slice K_s and V_s into 16×16 image patches (as illustrated in Fig. 3), and then employ recursive computation in the Spatial Mixing Block to progressively model global spatial relationships from the first visual patch to the last:

$$\begin{aligned} WKV_i &= \frac{n_i + e^u \odot k_i \odot v_i}{d_i + e^u \odot k_i}, \\ \underbrace{n_i = n_{i-1} \odot e^{-\mathcal{W}} + k_i \odot v_i}_{\text{Cumulative memory of } k_i \odot v_i}&, \underbrace{d_i = d_{i-1} \odot e^{-\mathcal{W}} + k_i}_{\text{Cumulative memory of } k_i}, \end{aligned} \tag{3}$$

where $i \in [0, (H \times W)/(16 \times 16))$, k_i and v_i represent the patches of K_s and V_s, respectively. The initial values n_0 and d_0 are copied from k_0. The parameter $u \in (0, 1))$ denotes the learnable position compensation, which enhances the sensitivity of model to local positions. The term $\mathcal{W}$ represents content-adaptive decay factors that control the decay rate of historical memory (as shown in Equation 2). Finally, a dynamic weight is generated through the gating factor R_s to modulate the contribution of the output from the Spatial Mixing Block to the current state:

$$F_{S0} = \sigma(\mathcal{C}_{1\times1}(F_0)) \odot WKV. \tag{4}$$

Then, F_{S0} is residually connected with F_0 to obtain F_{res} (as shown in Fig.3). We apply the Channel Mixing Block to reallocate channel-wise feature weights for feature extraction:

$$F_{C0} = \sigma(\mathcal{C}_{1\times1}(F_{res})) \odot (SReLU(\mathcal{C}_{3\times3}(F_{res}))) \tag{5}$$

F_1 is obtained after down-sampling the output of two ARGN blocks using a 3×3 convolutional operation. Subsequent stages (Stage 2–3) repeat this process, and multi-scale features F_1, F_2, F_3 are fused via learnable weights:

$$F_f = \alpha_1(\mathcal{C}_{1\times1}(F_1)) + \alpha_2(\mathcal{C}_{1\times1}(Up(F_2))) + \alpha_3(Up(F_3)), \tag{6}$$

where F_i denotes the feature map processed by the $Stage_i$ $(i = 1, 2, 3)$, and a_i are the learnable parameters that assign weights to feature maps of different levels during the feature fusion process. We then generate the initial predicted action a_{in} based on the fused feature map F_f as follows:

$$a_{in} = Linear(\mathcal{C}_{3\times3}(F_f))), \tag{7}$$

To minimize the mean-squared error (MSE) between the predicted action and the ground truth action, we use the following loss function:

$$\mathcal{L} = \text{MSE}(a_{in}, a_{groud}), \tag{8}$$

where $\mathcal{L}$ represents loss function, a_{groud} is the ground truth action from human demonstrations (detailed formulations of ARGN please refer to Appendix B in supplements).

Why Gaussian Mixture Model? When using a single Gaussian (Chi et al. 2023), the model tends to regress to the mean, suppressing distinct action modes and leading to suboptimal control accuracy. In contrast, a Gaussian Mixture Model (GMM) enables the modeling of separate action clusters, each with its own mean and covariance, allowing for more accurate representation of the action distribution. Let $\mathbf{x} \in \mathbb{R}^n$ denote ground-truth joint configurations. The GMM uses $K = 6$ components with prior α_k, mean μ_k, and covariance Σ_k, with probability density:

$$P(\mathbf{x} \mid \Theta) = \sum_{k=1}^{K} \alpha_k \mathcal{N}\left(\mathbf{x} \mid \mu_k, \Sigma_k\right), \tag{9}$$

where $\mathcal{N}$ is a multivariate Gaussian. Parameters $\{\alpha_k, \mu_k, \Sigma_k\}$ are estimated via the EM algorithm to maximize data likelihood, capturing latent joint space structures. The initial prediction a_{in} is compared to GMM clusters using Mahalanobis distance:

$$l_k = \sqrt{(a_{\text{in}} - \mu_k^{\omega})^T (\Sigma_k^{\omega,\omega})^{-1} (a_{\text{in}} - \mu_k^{\omega})}, \tag{10}$$

where l_k measures distance to the k-th component. The final action a^* is the closest cluster center:

$$a^* = \arg\min_{\mu\omega^k} l_k, \tag{11}$$

where a^* is the final predicted action (The detailed derivations please refer to the Appendix B and C in supplements).

4 Experiments

In this section, we evaluate the effectiveness and generalized ability of the proposed RGMP framework. Our experiments are designed to assess the performance of each core component, including the GSS and ARGN. We also compare our approach against state-of-the-art baselines in various manipulation tasks. The evaluation metrics, experimental setup, implementation details, and comprehensive results are detailed as follows.

4.1 Hardware Setup

We conduct experiments on two robotic platforms: a humanoid robot, with evaluations focused on the upper limb (please see Appendix D in supplements for details), and a desktop dual-arm robot, developed to test cross-embodied generalized ability. The desktop robot is equipped with an RGB camera and two 6-DoF arms for manipulation tasks.

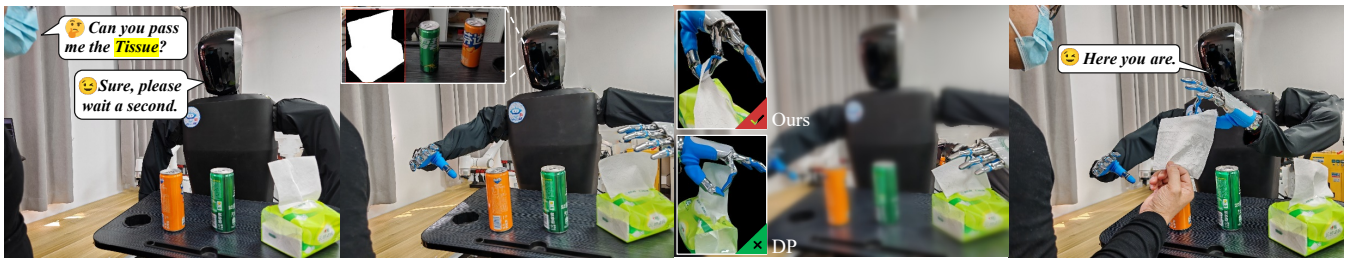


Figure 4: **Pipeline of human-robot interactions**. We validate models on the task of "passing me the tissue", training dataset comprising only 40 instances of pinching a tissue up. Our RGMP can generate more precise actions than DP (Diffusion Policy).

Methods	Fanta			Sprite			Tissue			Squashed Coke			Human Hand		
	Acc_s	Acc_t	Acc ↑	Acc_s	Acc_t	Acc ↑	Acc_s	Acc_t	Acc ↑	Acc_s	Acc_t	Acc ↑	Acc_s	Acc_t	Acc ↑
with *ResNet50*															
Qwen-vl	0.65	0.54	0.35	0.60	0.42	0.25	0.65	0.46	0.30	0.65	0.46	0.30	0.70	0.57	0.40
GSS	0.85	0.53	0.45	0.75	0.46	0.35	0.85	0.47	0.40	0.85	0.47	0.40	0.85	0.56	0.48
with *Transformer*															
Qwen-vl	0.60	0.58	0.35	0.65	0.54	0.30	0.70	0.50	0.35	0.60	0.58	0.35	0.65	0.62	0.40
GSS	0.80	0.56	0.45	0.75	0.53	0.40	0.85	0.53	0.45	0.85	0.53	0.45	0.85	0.64	0.54
with *Maniskill2-1st*															
Qwen-vl	0.70	0.57	0.40	0.65	0.69	0.45	0.65	0.53	0.34	0.65	0.54	0.35	0.65	0.62	0.40
GSS	0.85	0.53	0.45	0.80	0.68	0.54	0.85	0.53	0.45	0.80	0.56	0.45	0.85	0.70	0.60
with *Diffusion Policy*															
Qwen-vl	0.65	0.76	0.49	0.65	0.75	0.50	0.65	0.68	0.44	0.65	0.62	0.40	0.70	0.71	0.50
GSS	**0.85**	**0.76**	**0.65**	**0.80**	**0.77**	**0.62**	**0.85**	**0.69**	**0.59**	**0.85**	**0.65**	**0.55**	**0.90**	**0.83**	**0.74**

Table 1: **Ablation study of GSS and Qwen-vl**. Experiments use the scenes with Fanta, Sprite, and tissue paper (objects repositioned randomly per trial). Flattened Coke cans and human hands were tested separately. Each skill category included 40 training demonstrations, with test results from 20 random repositioning trials.

Methods	Fanta↑	Coke ↑	Spray ↑	Hand ↑	Average ↑
Maniskill2-1st	0.70	0.60	0.63	0.62	0.64
Octo	0.65	0.55	0.58	0.62	0.60
OpenVLA	0.68	0.58	0.61	0.60	0.62
RDT-1b	0.70	0.61	0.60	0.62	0.64
Diffusion Policy	0.75	0.65	0.68	0.72	0.70
Dex-VLA	0.87	0.66	0.71	0.84	0.77
RGMP(ours)	**0.98**	**0.78**	**0.81**	**0.90**	**0.87**

Table 2: **Evaluation results of generalized manipulation capability.** Models are only trained on 40 Fanta can grasping demonstrations. Metrics for Fanta, Coke, Spray, and Hand represent grasping accuracy across these objects.

4.2 Dataset and Evaluation Criteria

To validate the effectiveness of the RGMP, we collected 120 trajectories for the skill library. Each trajectory corresponds to an execution path associated with an RGB image captured prior to the robotic arm performing an action. Specifically, each RGB image is linked to a target action, and each trajectory represents the movement of the robotic arm from its initial position to the target spatial location:

$$d_i = (\mathcal{J}, \mathcal{O}), \tag{12}$$

where $\mathcal{J}$ denote the joint space of the robotic arm, where each trajectory specifies the motion of the arm from its initial configuration to the target spatial location and pose of the end effector. In real-world evaluations, the model performance is assessed using two complementary metrics. The skill success rate, denoted as Acc_s, is recorded when the robot correctly identifies and selects the appropriate skill for the task. Additionally, the execution accuracy Acc_a quantifies the precision with which the robot executes the selected skill to retrieve the target object. Consequently, the final success rate Acc is defined as the product of these two metrics:

$$Acc = Acc_s \times Acc_t, \tag{13}$$

The detailed criteria for ManiSkill2 manipulation tasks can be referred to in Appendix E in supplements.

4.3 Performance Comparison and Ablation study

To evaluate RGMP, we conducted real-world comparative experiments against ResNet50 (He et al. 2016), Transformer (Vaswani et al. 2017), the first-place entry (Gao et al. 2023) in the Maniskill2 challenge, Octo (Team et al. 2024), OpenVLA (Kim et al. 2024), RDT-1b (Liu et al. 2024c), Dex-VLA (Wen et al. 2025), and Diffusion Policy. Experiments involved random target object placement, with success defined as accurate instruction understanding, correct manipulation execution (object delivery to humans), and collision avoidance with surrounding objects. To assess generalization and cross-domain transferability, we deployed the trained model on a desktop dual-arm platform, using a low-data setup: 40 interaction samples for Fanta can grasping as the exclusive training data, with evaluation on three unseen categories (human hands, spray bottles, Coke cans) at random workspace positions. Tables 1 and 2 show RGMP outperforms baselines across tasks, with top Acc, Acc_s, Acc_t for Fanta, Sprite, tissues, squashed Coke cans, human hands, validating its effectiveness on regular/irregular objects. As Table 1 demonstrates, our GSS yields a 15-

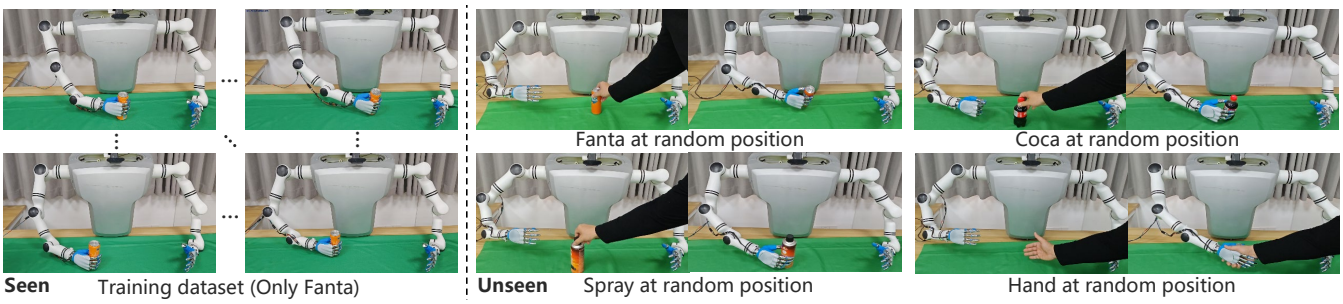


Figure 5: **Generalized manipulation ability of RGMP.** We test RGMP in grasping various unseen objects at random positions. Notably, our RGMP was trained on a dataset comprising only 40 demonstrations of grasping a Fanta can. RGMP exhibited proficient performance in grasping the Fanta can from any position. Furthermore, it displayed significant generalization capabilities to other unseen objects, including a Coke, a Spray, and a Hand, underscoring its versatility and adaptability in manipulation.

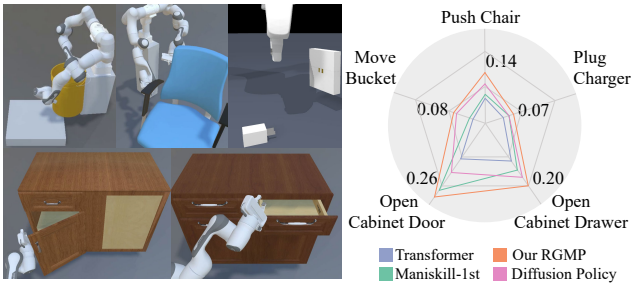


Figure 6: **Performance on Maniskill2 simulator.** We validate the effectiveness of RGMP and SOTA models in each of the five manipulation tasks on the simulation platform.

25% accuracy improvement in skill selection compared to Qwen-vl. Ablation studies (Table 3) confirm that integrating GMM with ARGN enhances performance: for Diffusion Policy, GSS+GMM yields a 0.55 Acc versus 0.49 for without GMM, while ARGN with GSS+GMM achieves a 0.69 Acc in picking squashed Coke, demonstrating the effectiveness of GMM in refining predictions. Additionally, Table 4 validates the contributions of RoPE, Spatial Mixing Blocks (SMB), and Channel Mixing Blocks (CMB): their combined use yields the highest accuracy across all objects (0.98 for Fanta, 0.78 for Coke, 0.81 for spray, 0.90 for hands). Beyond three primitives (grasp/lift-up/pinch), we evaluate five non-grasp ManiSkill2 tasks, complex tasks like plugging chargers (pinch) and opening cabinets (grasp) are dynamically composed from our atomic primitives. As shown in Figure 6, our RGMP achieves the highest performance across all tasks, demonstrating its transferability and generalization capability. Furthermore, as shown in Table 5, RGMP achieves a score of 0.98 with 40 training samples, which is 5× fewer than the 200 samples required by DP.

5 Conclusion and Future Work

This work addresses semantic-spatial skill alignment and visuomotor overfitting in humanoid robotics via RGMP, an end-to-end framework integrating GSS and ARGN. By dynamically associating contextual skills and decomposing 6-DoF trajectories into probabilistically regularized Gaussian components, RGMP achieves 87% generalization success and 5× greater data efficiency than Diffusion Policy in human-robot interaction. Results show explicit neuro-symbolic coordination enables robust generalization across unseen objects/scenes, advancing collaboration with a scalable adaptive manipulation foundation. Future work will explore functional generalization: demonstrating one primary object function allows automatic inference of trajectories for others, eliminating exhaustive teaching and enhancing efficiency in dynamic environments.

Method	GMM	Tissue			Squashed Coke		
		Acc_s	Acc_t	Acc	Acc_s	Acc_t	Acc
Diffusion policy	-	0.85	0.58	0.50	0.80	0.61	0.49
	✓	0.80	0.68	0.56	0.85	0.65	0.55
ARGN(ours)	-	0.80	0.69	0.55	0.85	0.71	0.60
	✓	**0.85**	**0.71**	**0.60**	**0.90**	**0.77**	**0.69**

Table 3: **Ablation study of ARGN, and GMM**. We validate models on the task of passing tissue and squashed Coke.

RoPE	SMB	CMB	Fanta ↑	Coke ↑	Spray ↑	Hand ↑
-	✓	✓	0.86	0.69	0.71	0.77
✓	-	✓	0.83	0.75	0.76	0.82
✓	✓	-	0.91	0.66	0.65	0.74
✓	✓	✓	**0.98**	**0.78**	**0.81**	**0.90**

Table 4: **Ablation study of the componet of ARGN.** We evaluate RoPE, Spatial Mixing Block (SMB), and Channel Mixing Block (CMB) in grasping tasks.

Methods	40	80	120	160	200
Diffusion Policy	0.81	0.89	0.94	0.95	0.98
RGMP(ours)	**0.98**	**0.98**	**0.99**	**0.99**	**0.99**

Table 5: **Data efficiency comparision of RGMP and Diffusion Policy.** RGMP achieves 0.98 with 40 train samples of grasping fanta (5× fewer than 200 of DP).

6 Acknowledgement

This work is supported by National Natural Science Foundation of China under Grant (62361166629, 62225113, 623B2080), the Major Project of Science and Technology Innovation of Hubei Province (2024BCA003, 2025BEA002), the Innovative Research Group Project of Hubei Province under Grants 2024AFA017, and the Key Research Project of Wuhan City 2024060788020073. The supercomputing system at the Supercomputing Center and the Learning Algorithms & Soft Manipulation Laboratory of Wuhan University supported the numerical calculations and the robot platforms in this paper.

References

Abu-Jassar, A. T.; Attar, H.; Amer, A.; Lyashenko, V.; Yevsieiev, V.; and Solyman, A. 2025. Development and Investigation of Vision System for a Small-Sized Mobile Humanoid Robot in a Smart Environment. *International Journal of Crowd Science*, 9(1): 29–43.

Ahn, M.; Brohan, A.; Brown, N.; Chebotar, Y.; Cortes, O.; David, B.; Finn, C.; Fu, C.; Gopalakrishnan, K.; Hausman, K.; et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Bogdanovic, M.; Khadiv, M.; and Righetti, L. 2020. Learning variable impedance control for contact sensitive tasks. *IEEE Robotics and Automation Letters*, 5(4): 6129–6136.

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34: 15084–15097.

Chi, C.; Xu, Z.; Feng, S.; Cousineau, E.; Du, Y.; Burchfiel, B.; Tedrake, R.; and Song, S. 2023. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 02783649241273668.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; Schuh, P.; Shi, K.; Tsvyashchenko, S.; Maynez, J.; Rao, A.; Barnes, P.; Tay, Y.; Shazeer, N. M.; Prabhakaran, V.; Reif, E.; Du, N.; Hutchinson, B. C.; Pope, R.; Bradbury, J.; Austin, J.; Isard, M.; Gur-Ari, G.; Yin, P.; Duke, T.; Levskaya, A.; Ghemawat, S.; Dev, S.; Michalewski, H.; García, X.; Misra, V.; Robinson, K.; Fedus, L.; Zhou, D.; Ippolito, D.; Luan, D.; Lim, H.; Zoph, B.; Spiridonov, A.; Sepassi, R.; Dohan, D.; Agrawal, S.; Omernick, M.; Dai, A. M.; Pillai, T. S.; Pellat, M.; Lewkowycz, A.; Moreira, E.; Child, R.; Polozov, O.; Lee, K.; Zhou, Z.; Wang, X.; Saeta, B.; Díaz, M.; Firat, O.; Catasta, M.; Wei, J.; Meier-Hellstern, K. S.; Eck, D.; Dean, J.; Petrov, S.; and Fiedel, N. 2022. PaLM: Scaling Language Modeling with Pathways. *J. Mach. Learn. Res.*, 24: 240:1–240:113.

Cruciani, S.; Hang, K.; Smith, C.; and Kragic, D. 2019. Dual-Arm In-Hand Manipulation and Regrasping Using Dexterous Manipulation Graphs. arXiv:1904.11382.

Cruciani, S.; Smith, C.; Kragic, D.; and Hang, K. 2018. Dexterous manipulation graphs. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2040–2047. IEEE.

Dai, W.; Li, J.; Li, D.; Tiong, A. M. H.; Zhao, J.; Wang, W.; Li, B.; Fung, P.; and Hoi, S. 2023. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. arXiv:2305.06500.

Dong, Z.; Hao, J.; Yuan, Y.; Ni, F.; Wang, Y.; Li, P.; and Zheng, Y. 2024. DiffuserLite: Towards Real-time Diffusion Planning. *arXiv preprint arXiv:2401.15443*.

Driess, D.; Xia, F.; Sajjadi, M. S.; Lynch, C.; Chowdhery, A.; Wahid, A.; Tompson, J.; Vuong, Q.; Yu, T.; Huang, W.; et al. 2023a. Palm-e: An embodied multimodal language model.

Driess, D.; Xia, F.; Sajjadi, M. S. M.; Lynch, C.; Chowdhery, A.; Ichter, B.; Wahid, A.; Tompson, J.; Vuong, Q. H.; Yu, T.; Huang, W.; Chebotar, Y.; Sermanet, P.; Duckworth, D.; Levine, S.; Vanhoucke, V.; Hausman, K.; Toussaint, M.; Greff, K.; Zeng, A.; Mordatch, I.; and Florence, P. R. 2023b. PaLM-E: An Embodied Multimodal Language Model. In *International Conference on Machine Learning*.

Gao, F.; Li, X.; Yu, J.; and Shaung, F. 2023. A Two-stage Fine-tuning Strategy for Generalizable Manipulation Skill of Embodied AI. *arXiv preprint arXiv:2307.11343*.

Haldar, S.; Pari, J.; Rai, A.; and Pinto, L. 2023. Teach a Robot to FISH: Versatile Imitation from One Minute of Demonstrations. In *Proceedings of Robotics: Science and Systems*. Daegu, Republic of Korea.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, S.; Wang, Z.; Li, P.; Jia, B.; Liu, T.; Zhu, Y.; Liang, W.; and Zhu, S.-C. 2023. Diffusion-based generation, optimization, and planning in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16750–16761.

Huang, W.; Liang, J.; Guo, X.; Fang, Y.; Wan, G.; Rong, X.; Wen, C.; Shi, Z.; Li, Q.; Zhu, D.; et al. 2025a. Keeping yourself is important in downstream tuning multimodal large language model. *arXiv preprint arXiv:2503.04543*.

Huang, W.; Liang, J.; Shi, Z.; Zhu, D.; Wan, G.; Li, H.; Du, B.; Tao, D.; and Ye, M. 2025b. Learn from Downstream and Be Yourself in Multimodal Large Language Model Fine-Tuning. In *ICML*.

Huang, W.; Liang, J.; Wan, G.; Zhu, D.; Li, H.; Shao, J.; Ye, M.; Du, B.; and Tao, D. 2025c. Be Confident: Uncovering Overfitting in MLLM Multi-Task Tuning. In *ICML*.

Intelligence, P.; Black, K.; Brown, N.; Darpinian, J.; Dhabalia, K.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; et al. 2025. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054*.

Kim, M.; Han, J.; Kim, J.; and Kim, B. 2023. Pre-and post-contact policy decomposition for non-prehensile manipulation with zero-shot sim-to-real transfer. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 10644–10651. IEEE.

Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.; Lam, G.; Sanketi, P.; et al. 2024. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*.

Li, J.; Zhu, Y.; Xie, Y.; Jiang, Z.; Seo, M.; Pavlakos, G.; and Zhu, Y. 2024a. Okami: Teaching humanoid robots manipulation skills through single video imitation. *arXiv preprint arXiv:2410.11792*.

Li, R.; Li, R.; Guo, S.; and Zhang, L. 2024b. Source Prompt Disentangled Inversion for Boosting Image Editability with Diffusion Models. *arXiv preprint arXiv:2403.11105*.

Li, S.; Keipour, A.; Jamieson, K.; Hudson, N.; Swan, C.; and Bekris, K. 2023. Demonstrating Large-Scale Package Manipulation via Learned Metrics of Pick Success. In *Proceedings of Robotics: Science and Systems*. Daegu, Republic of Korea.

Liang, H.; Lou, X.; Yang, Y.; and Choi, C. 2021. Learning visual affordances with target-orientated deep q-network to grasp objects by harnessing environmental fixtures. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2562–2568. IEEE.

Liang, J.; Huang, W.; Xia, F.; Xu, P.; Hausman, K.; Ichter, B.; Florence, P.; and Zeng, A. 2023. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 9493–9500. IEEE.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2023. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26296–26306.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2024b. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Liu, S.; Wu, L.; Li, B.; Tan, H.; Chen, H.; Wang, Z.; Xu, K.; Su, H.; and Zhu, J. 2024c. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Rothert, J. J.; Lang, S.; Seidel, M.; and Hanses, M. 2024. Sim-to-Real Transfer for a Robotics Task: Challenges and Lessons Learned. In *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1–8. IEEE.

Shridhar, M.; Manuelli, L.; and Fox, D. 2023. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, 785–799. PMLR.

Skubis, I.; and Wodarski, K. 2023. HUMANOID ROBOTS IN MANAGERIAL POSITIONS-DECISION-MAKING PROCESS AND HUMAN OVERSIGHT. *Scientific Papers of Silesian University of Technology. Organization & Management/Zeszyty Naukowe Politechniki Slaskiej. Seria Organizacji i Zarzadzanie*, (189).

Team, O. M.; Ghosh, D.; Walke, H.; Pertsch, K.; Black, K.; Mees, O.; Dasari, S.; Hejna, J.; Kreiman, T.; Xu, C.; et al. 2024. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*.

Tong, Y.; Liu, H.; and Zhang, Z. 2024. Advancements in humanoid robots: A comprehensive review and future prospects. *IEEE/CAA Journal of Automatica Sinica*, 11(2): 301–328.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wen, J.; Zhu, Y.; Li, J.; Tang, Z.; Shen, C.; and Feng, F. 2025. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*.

Xu, K.; Yu, H.; Lai, Q.; Wang, Y.; and Xiong, R. 2021. Efficient learning of goal-oriented push-grasping synergy in clutter. *IEEE Robotics and Automation Letters*, 6(4): 6337–6344.

Xu, M.; Shen, Y.; Zhang, S.; Lu, Y.; Zhao, D.; Tenenbaum, J.; and Gan, C. 2022. Prompting decision transformer for few-shot policy generalization. In *international conference on machine learning*, 24631–24645. PMLR.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yaseen, M. 2024. What is YOLOv9: An in-depth exploration of the internal features of the next-generation object detector. *arXiv preprint arXiv:2409.07813*.

Yoneda, T.; Sun, L.; Yang, G.; Stadie, B. C.; and Walter, M. R. 2023. To the Noise and Back: Diffusion for Shared Autonomy. In *Proceedings of Robotics: Science and Systems*. Daegu, Republic of Korea.

Zeng, A.; Song, S.; Welker, S.; Lee, J.; Rodriguez, A.; and Funkhouser, T. 2018. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4238–4245. IEEE.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3836–3847.

Zhang, T.; McCarthy, Z.; Jow, O.; Lee, D.; Chen, X.; Goldberg, K.; and Abbeel, P. 2018. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 5628–5635. IEEE.

Zitkovich, B.; Yu, T.; Xu, S.; Xu, P.; Xiao, T.; Xia, F.; Wu, J.; Wohlhart, P.; Welker, S.; Wahid, A.; et al. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, 2165–2183. PMLR.

RGMP: Recurrent Geometric-prior Multimodal Policy for Generalizable Humanoid Robot Manipulation *(Technical Appendix)*

A Implementation Details of GSS

A.1 GSS Architecture Overview

The GSS framework is implemented as a modular pipeline with three core components: (1) a Visual-Language Interpretation Module, (2) a Semantic Segmentation Module, and (3) a Prompt-Based Skill Selector. These components work in tandem to realize the translation from user instructions and visual observations to skill selection, with a total inference latency of 105 ms on an NVIDIA 4090 GPU.

Visual-Language Interpretation Module: This module leverages the Qwen-vl API (instead of a locally deployed model) for visual-language understanding. Prior to deployment, a critical step involves loading historical knowledge into Qwen-vl through structured prompt engineering. This process entails feeding the API a curated set of training examples, each consisting of: (1) an RGB image of an object with its bounding box, (2) the corresponding semantic segmentation result of the object, (3) the manually annotated shape category (e.g., "cylindrical", "crushed"), and (4) the optimal manipulation skill. For instance, one example includes the RGB image of a Fanta can with its bounding box, the segmentation result highlighting the contours of can, the label "cylindrical shape", and the associated skill "side Grasp". Another example comprises the RGB image of a crushed cola can with its bounding box, the segmentation result showing the flattened structure, the label "crushed shape", and the skill "Lift up". A total of 20 such examples covering common object categories are used to prime the model, enabling it to learn the mapping between visual features, shape attributes, and manipulation skills. The input prompt template for the API after knowledge loading is structured as: "*Instruction: Identify target object in image and output bounding box [x1, y1, x2, y2]* ". The API achieves 93.1% object localization accuracy on our custom dataset, with an average response time of 45 ms per query.

Semantic Segmentation Module: After obtaining the bounding box coordinates from the Qwen-vl API, the corresponding object is cropped out and input into the YOLOv8-seg model for semantic segmentation. This model, pretrained on a large-scale dataset and fine-tuned on our custom dataset, extracts the shape information of the target object. It achieves a mean Intersection over Union (mIoU) of 97.6% on our custom dataset, ensuring accurate segmentation results for shape analysis.

Prompt-Based Skill Selector: This component takes the bounding box coordinates and shape information of the target object as inputs and uses a specific prompt to make skill selections via the Qwen-vl API. The prompt is structured as: "*You are a robot with three Skills: side Grasp, Lift up, and top Pinch. The image you are observing has a resolution of 640x480. Based on the observation, the Coordinates of the bounding box and the Shape information of the object, Choose the skill without collision.*"

The selector utilizes the Qwen-vl API, which has learned from the prompt engineering examples, to map the input information to the appropriate skill. It resolves potential ambiguities by prioritizing skills that minimize collision risks based on the position of object and surrounding environment, with the API returning a confidence score for each candidate skill.

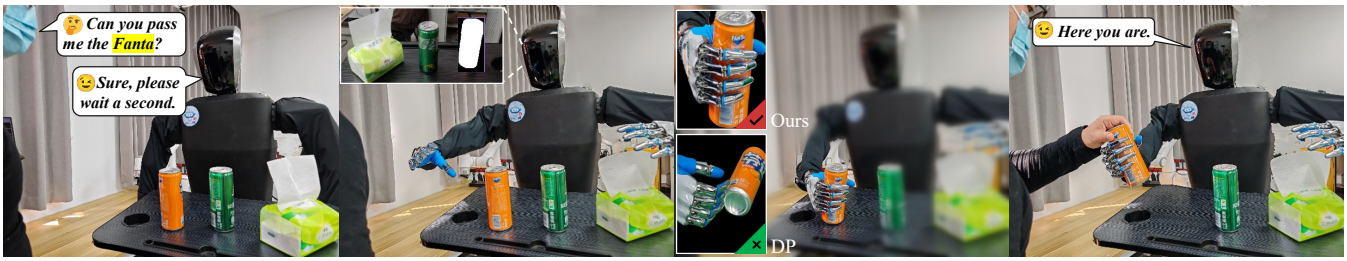
A.2 Skill Library

The skill library is defined based on various combinations of finger grip bending angles, enabling dexterous hands to execute different grasping techniques: side grasping, picking up, and pinching. The side grasping skill is applicable in scenarios where the robot arm can grasp objects from the side without any obstacles obstructing its path. The lifting up skill is used when an object must be lifted from above, typically in the presence of blocking obstacles. The pinching skill is intended for situations where the target object is small or thin, such as napkins, charging cables, and similar items. The skills within the library specifically define the grasping posture of the dexterous hand, without imposing strict constraints on the nature of the objects being grasped. The GSS selects the appropriate skill from the library based on both geometric-prior common sense and real-time RGB image observations, determining the feasible dexterous hand posture for the current scenario. Each skill is associated with distinct model weight parameters, and the action trajectory is dynamically generated in real-time by the RGMP.

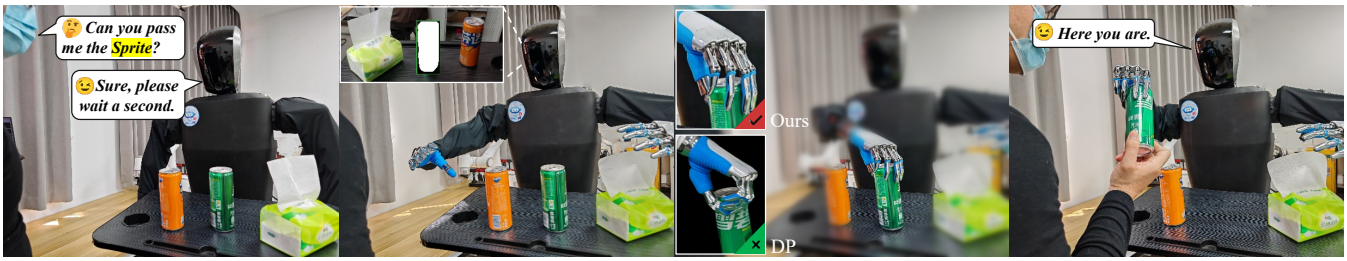
B Details of ARGN

Particularly, the ARGN architecture comprises three principal components: a Stem layer for initial feature extraction, three hierarchical processing stages (each with two ARGN blocks), and a multi-scale feature fusion module integrated with GMM. This modular design enables ARGN to progressively refine visual features while maintaining computational efficiency.

For an input RGB image $\mathcal{O} \in \mathbb{R}^{H\times W\times 3}$, the Stem layer performs spatial down-sampling and feature embedding to produce F_0 with a resolution of $\frac{H}{4} \times \frac{W}{4}$. This transformation is achieved through a series of convolutional operations, including a 3×3 convolution, batch normalization (BN), Squared ReLU (SReLU) activation, and max-pooling. The Stem layer serves two key purposes: it reduces the spatial dimensionality to alleviate computational burdens in subsequent stages, and it preserves essential low-frequency visual information (e.g., object contours and global scene structure) that is vital for robust action generation. The three hierarchical processing stages (Stage 1 to Stage 3) each consist of two ARGN blocks, which are designed to iteratively enhance feature representations through spatial and channel-



(a) The pipeline of grasping



(b) The pipeline of lifting up

Figure A1: **Pipeline of human-robot interactions**. We validate models on the task of "passing me the Fanta" and "passing me the Sprite". Our RGMP can generate more precise actions than DP (Diffusion Policy).

wise modeling. Each ARGN block comprises a Spatial Mixing Block and a Channel Mixing Block, working in tandem to capture both local spatial correlations and cross-channel dependencies. Through recursive operations within these blocks, ARGN builds a dynamic memory of visual patterns, allowing it to adaptively prioritize task-relevant regions (e.g., target objects or end-effector positions) in unseen environments. In order to reconstruct the spatial relationship, we encode the image patches with the Rotary Position Embedding (RoPE), which establishes an implicit association between each observed image patch and the final executed action by encoding positional information through geometric transformations. Formally, RoPE is defined as:

$$RoPE(x) = \begin{bmatrix} \cos(\theta_{h,w,j}) & -\sin(\theta_{h,w,j}) \\ \sin(\theta_{h,w,j}) & \cos(\theta_{h,w,j}) \end{bmatrix} \begin{bmatrix} x_h \\ x_w \end{bmatrix}, \tag{A1}$$

where x is a pixel from the initial feature map F_0, x_h and x_w denote the spatial indices (height and width) of x, and $\theta_{h,w,j}$ represents the two-dimensional position code. This position code is formulated as:

$$\theta_{h,w,j} = \theta_j \cdot h + \theta_j \cdot w, \quad \theta_j = \frac{1}{10000^{\frac{2j}{C}}}, \tag{A2}$$

Here, the frequency parameter θ governs rotation speeds across channel dimensions, with $j \in [0, C/2)$ denoting the channel index. The base constant 10000 and exponential decay term jointly establish a multi-scale spectral distribution mechanism: low-frequency components (corresponding to small j) prioritize modeling global structural relationships, while high-frequency components (larger j) capture fine-grained local positional variations. This design allows RoPE to inherently encode relative spatial offsets without relying on learnable position-specific parameters, thereby reducing overfitting risks and enhancing generalization to novel scenes. By combining the feature initialization of Stem layer, iterative refinement of hierarchical ARGN blocks, and positional encoding of RoPE, ARGN effectively models the spatial dependencies between visual inputs and robotic actions, laying the foundation for accurate and adaptive action generation in complex environments.

C Detailed Derivation of GMM for Action Refinement

In our framework, the Gaussian Mixture Model (GMM) is trained on demonstration data to model the distribution of ground-truth joint angles, featuring 6 components that correspond to the 6 robotic joints. The model parameters $\Theta = \{\alpha_k, \mu_k, \Sigma_k\}_{k=1}^{6}$ are estimated through the Expectation-Maximization (EM) algorithm, which iteratively maximizes the log-likelihood of the observed data. The key steps are elaborated as follows:

E-Step (Expectation). For each data point $\mathbf{x}_i$ (i.e., a sample of ground-truth joint angles) and each component k, the posterior probability (responsibility) that $\mathbf{x}_i$ belongs to component k is calculated as:

$$\gamma_{ik} = P(z_{ik} = 1 \mid \mathbf{x}_i, \Theta) = \frac{\alpha_k \mathcal{N}(\mathbf{x}i \mid \mu_k, \Sigma_k)}{\sum j = 1^6 \alpha_j \mathcal{N}(\mathbf{x}_i \mid \mu_j, \Sigma_j)}, \tag{A3}$$

where z_{ik} represents a latent variable indicating whether $\mathbf{x}_i$ is generated by component k.

M-Step (Maximization). The parameters are updated to maximize the expected log-likelihood, which is computed using the responsibilities obtained from the E-step. The weight of component k is determined as the average responsibility of that component across all data points:

$$\alpha_k = \frac{1}{N} \sum_{i=1}^{N} \gamma_{ik}, \tag{A4}$$

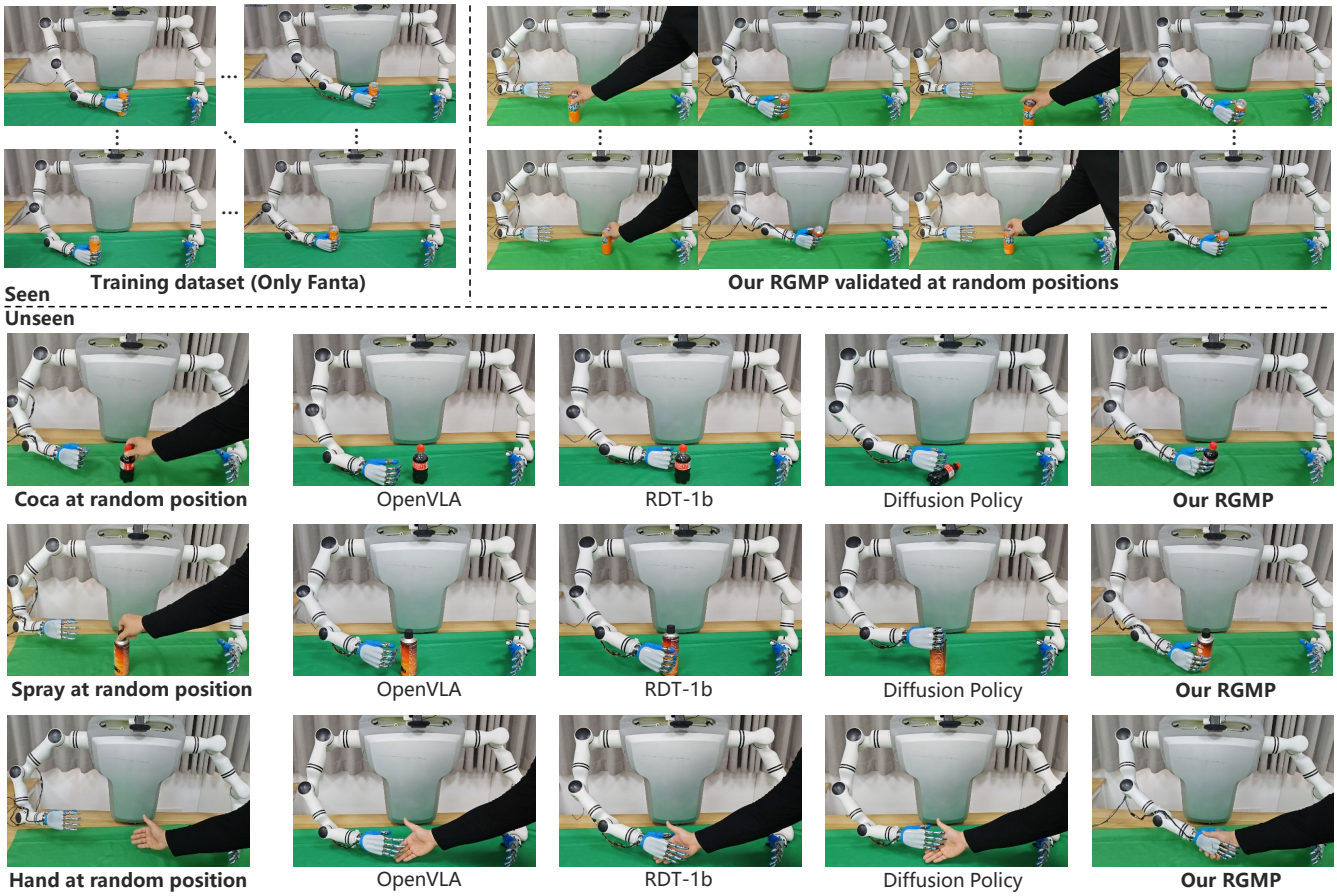


Figure A2: **Generalized manipulation capability comparison.** We validate RGMP and SOTA models in grasping various unseen objects at random positions.

with N denoting the number of demonstration samples. The mean of component k is the weighted average of the data points, with the weights being γ_{ik}:

$$\mu_k = \frac{\sum_{i=1}^{N} \gamma_{ik} \mathbf{x}i}{\sum i = 1^{N} \gamma_{ik}}. \tag{A5}$$

The covariance of component k is the weighted covariance of the data points around μ_k:

$$\Sigma_k = \frac{\sum_{i=1}^{N} \gamma_{ik} (\mathbf{x}i - \mu_k)(\mathbf{x}i - \mu_k)^\top}{\sum i = 1^{N} \gamma ik}. \tag{A6}$$

The E-step and M-step are repeated until the log-likelihood converges, i.e., the change in Θ drops below a predefined threshold (e.g., 10^{-6}). The GMM regression process refines the initial action a_{in} (generated by the GSA model) by aligning it with the most consistent Gaussian component. This involves deriving the conditional distribution of joint angles ω given a_{in}, along with the aggregated mean and covariance.

Conditional Distribution for Single Gaussian Component. For the k-th Gaussian component, the joint distribution of a_{in} and ω is modeled as a multivariate normal distribution $\mathcal{N}\left(\begin{bmatrix}\mu_k^{a_{\text{in}}} \\ \mu_k^{\omega}\end{bmatrix}, \begin{bmatrix}\Sigma_k^{a_{\text{in}},a_{\text{in}}} & \Sigma_k^{a_{\text{in}},\omega} \\ \Sigma_k^{\omega,a_{\text{in}}} & \Sigma_k^{\omega,\omega}\end{bmatrix}\right)$, where $\mu_k^{a_{\text{in}}}$ and μ_k^{ω} are sub-means, and $\Sigma_k^{a_{\text{in}},a_{\text{in}}}, \Sigma_k^{\omega,\omega}$ are sub-covariances. According to the properties of multivariate Gaussians, the conditional distribution $P(\omega \mid a_{\text{in}})$ for component k is also a normal distribution:

$$P(\omega \mid a_{\text{in}}, k) \sim \mathcal{N}\left(\mu_k^{\omega} + \Sigma_k^{\omega,a_{\text{in}}} (\Sigma_k^{a_{\text{in}},a_{\text{in}}})^{-1} (a_{\text{in}} - \mu_k^{a_{\text{in}}}), \Sigma_k^{\omega,\omega} - \Sigma_k^{\omega,a_{\text{in}}} (\Sigma_k^{a_{\text{in}},a_{\text{in}}})^{-1} \Sigma_k^{a_{\text{in}},\omega}\right) \tag{A7}$$

In our setup, due to the high correlation between a_{in} and ω (both representing joint angle predictions), we simplify by assuming a_{in} is a noisy observation of $\mu_k^{a_{\text{in}}}$. This leads to the conditional mean $\hat{\mu}_k^{\omega} \approx \mu_k^{\omega}$ and covariance $\hat{\Sigma}_k^{\omega,\omega} \approx \Sigma_k^{\omega,\omega}$ (as utilized in the main text).

Aggregated Mean and Covariance Across Components. The overall conditional distribution $P(\omega \mid a_{\text{in}})$ is a weighted sum of the component-wise distributions, with weights α_k (estimated from the EM algorithm). The aggregated mean:

$$\hat{\mu}^{\omega} = \sum_{k=1}^{6} \alpha_k \hat{\mu} k^{\omega}, \tag{A8}$$

where α_k is re-normalized based on the consistency of a_{in} with μ_k (measured via Mahalanobis distance). The aggre-

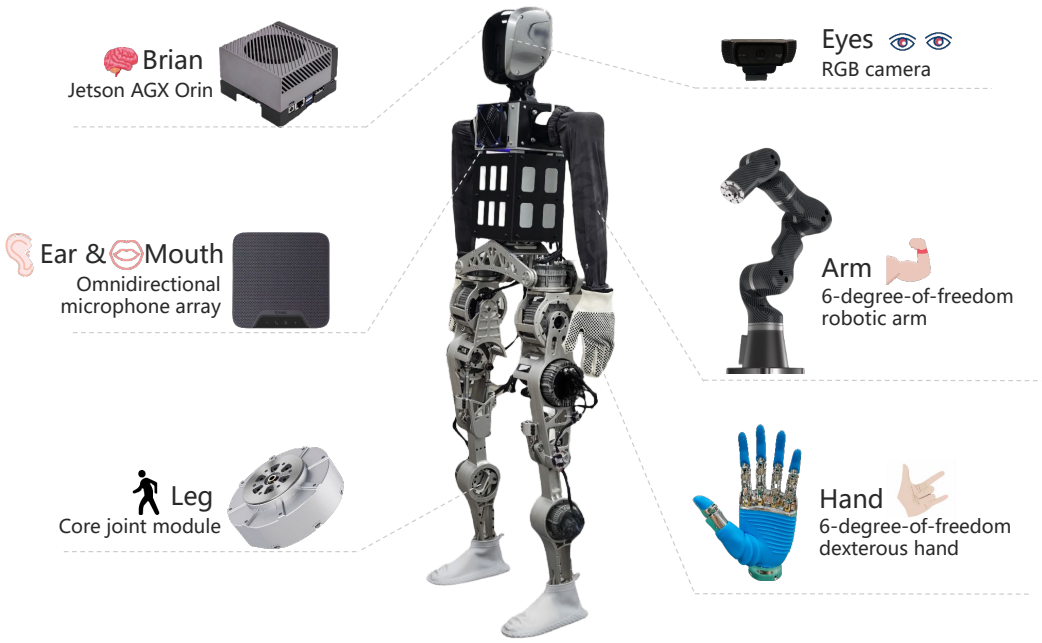


Figure A3: **Hardware setup of humanoid robot.** Core components and their positions are shown.

Methods	Push Chair↑	Move Bucket ↑	Plug Charger ↑	Cabinet Door ↑	Cabinet Drawer ↑	Average Score ↑
RDT-1b	0.04 ± 0.01	0.01 ± 0.00	0.01 ± 0.00	0.13 ± 0.03	0.10 ± 0.02	0.06 ± 0.01
Dex-VLA	0.02 ± 0.00	0.03 ± 0.01	0.01 ± 0.01	0.18 ± 0.02	0.08 ± 0.02	0.06 ± 0.01
Maniskill2-1st	0.08 ± 0.02	0.04 ± 0.01	0.03 ± 0.01	0.24 ± 0.03	0.12 ± 0.02	0.10 ± 0.02
Octo	0.05 ± 0.01	0.02 ± 0.01	0.03 ± 0.01	0.20 ± 0.03	0.10 ± 0.01	0.08 ± 0.01
OpenVLA	0.06 ± 0.01	0.04 ± 0.01	0.04 ± 0.02	0.24 ± 0.03	0.15 ± 0.01	0.10 ± 0.02
Diffusion Policy	0.04 ± 0.01	0.06 ± 0.02	0.06 ± 0.02	0.15 ± 0.01	0.17 ± 0.02	0.10 ± 0.01
RGMP(ours)	$\mathbf{0.14 \pm 0.02}$	$\mathbf{0.08 \pm 0.01}$	$\mathbf{0.07 \pm 0.01}$	$\mathbf{0.26 \pm 0.03}$	$\mathbf{0.20 \pm 0.03}$	$\mathbf{0.15 \pm 0.02}$

Table A1: **Performace on Maniskill2 simulator.** Average testing results of different seed for RGMP and other baselines on the five manipulation tasks on the simulation platform.

gated covariance, which accounts for component weights:

$$\hat{\Sigma}^{\omega,\omega} = \sum_{k=1}^{6} \alpha_k^2 \hat{\Sigma}_k^{\omega,\omega}, \tag{A9}$$

where squaring α_k ensures that components with lower weights contribute less to the overall uncertainty. The Mahalanobis distance between a_{in} and μ_k is preferred over the Euclidean distance for two main reasons: 1) it accounts for the covariance structure, and 2) it normalizes for units. Unlike the Euclidean distance, which treats all dimensions equally, the Mahalanobis distance incorporates the covariance matrix Σ_k, scaling distances by the variability of each joint angle. This is crucial for robotic joints, as some angles (e.g., those controlling fine movements) may have smaller variances and require stricter consistency checks. Additionally, joint angles can have different units or ranges (e.g., radians for rotation vs. meters for translation). The Mahalanobis distance normalizes these differences, enabling a meaningful comparison across dimensions. Formally, the distance l_k quantifies how "atypical" a_{in} is relative to the distribution of the k-th component. A smaller l_k indicates higher consistency, making the component a better candidate for refining the action prediction.

D Detailed Hardware Setup

The upper limb hardware comprises four components: visual input, voice input/output, behavioral decision-making, and action execution. The visual input consists of a self-centered RGB camera installed on the head of robot, while the voice input/output hardware employs the iFLYTEK S0Y22F omnidirectional microphone and speaker, also mounted on the chest of robot. The behavioral decision-making hardware utilizes an NVIDIA Orin chip installed in the head of robot. The motion execution hardware includes two 6-degree-of-freedom (6-DoF) robotic arms and two 6-DoF dexterous hands. The entire robot is self-powered and communicates via a local area network. The robot can be started and shut down without the external power or communication cables (as shown in Figure A3).

E ManiSkill2 Criteria

Open Cabinet Drawer A single-arm mobile robot opens a specified cabinet drawer (with randomized joint friction or damping); success requires the target drawer to be opened

Symbol	Type	Description
Training and Interaction		
E	Scalar	Total training epochs for RGMP parameter optimization
T	Integer	Maximum human-robot conversation rounds
M	Integer	Capacity of human demonstration collection
Inputs and Perception		
$\mathcal{I}$	Text	Transcribed human speech instruction
$\mathcal{O}$	Image	Ego-centric RGB camera frame of workspace
$\mathcal{J}$	Vector	6-DoF robotic arm joint angles (radians)
Dataset and Models		
$\mathcal{D}$	Collection	Demonstration tuples $(\mathcal{O}_i, \mathcal{J}_i)$ for training
Q	Model	Fine-tuned Qwen-vl vision-language model
$\mathcal{G}_m$	Policy	Pre-trained RGMP model for skill m
Features and Mechanisms		
F_0	Tensor	Initial visual feature map from stem network
$\mathcal{W}$	Scalar	Adaptive decay factor in recursive memory
K_s, V_s	Tensors	Key/value matrices for spatial attention mixing
F_1–F_3	Tensors	Multi-scale feature maps from ARGN stages
Action Predictions		
a_{in}	Vector	Initial action from ARGN network
a_{ground}	Vector	Human-demonstrated joint angles (ground truth)
$\mathcal{L}$	Scalar	MSE loss between predictions and ground truth
Gaussian Mixture Model		
K	Integer	Number of Gaussian components (6, matching arm DoFs)
$\alpha_k, \mu_k, \Sigma_k$	Scalar/Vector/Matrix	Prior, mean, and covariance of k-th component
Θ	Set	GMM parameters $\{\alpha_k, \mu_k, \Sigma_k\}_{k=1}^{6}$
$\mathbf{x}_i$	Vector	i-th ground-truth joint angle sample
γ_{ik}	Scalar	Responsibility of $\mathbf{x}_i$ to component k
N	Integer	Total demonstration samples
l_k	Scalar	Mahalanobis distance (a_{in} to μ_k)
Evaluation Metrics		
Acc_s	Scalar	Skill selection accuracy (0-1)
Acc_t	Scalar	Task execution accuracy (0-1)
Acc	Scalar	Overall success rate ($Acc_s \times Acc_t$)

Table A2: **Symbol definitions used in the RGMP framework.** Notation for variables, models, and evaluation metrics.

to $\geq$90% of its max range and remain static. The training set includes 25 cabinets (multiple drawers) with 300 trajectories per drawer; evaluation uses 10 unseen test cabinets in 2 stages (250 trajectories each: 125 from 5 unseen cabinets, 125 from training set). The target drawer is specified by its initial center of mass.

Open Cabinet Door A single-arm mobile robot opens a specified cabinet door (with randomized joint friction or damping); success requires the target door to be opened to $\geq$90% of its max range and remain static. The training set includes 42 cabinets (multiple doors) with 300 trajectories per door; evaluation uses 10 unseen test cabinets following the same protocol as Open Cabinet Drawer. The target door is specified by a segmentation mask.

Push Chair A dual-arm mobile robot pushes a swivel chair to a red hemisphere target without tipping it (with randomized joint parameters); success requires the chair to be within 15cm of the target, remain static, and stay upright. The training set includes 26 chairs with 300 trajectories per chair; evaluation uses 10 unseen chairs following the same protocol as Open Cabinet Drawer. No task-specific observations.

Move Bucket A dual-arm mobile robot moves a ball-containing bucket and lifts it onto a platform; success requires the bucket to be upright on/above the platform, remain static, and retain the ball. The training set includes 29 buckets with 300 trajectories per bucket; evaluation uses 10 unseen buckets following the same protocol as Open Cabinet Drawer. No task-specific observations.

Plug Charger A robot plugs a charger into a wall receptacle; success requires full insertion. The training set has 1,000 successful trajectories; evaluation has 2 stages (100 episodes each) with varied robot joint positions and charger poses. No task-specific observations.