Supplementary Information for: MAESTRO: Orchestrating Robotics Modules with Vision-Language Models for Zero-Shot Generalist Robots

I. Method Full Technical Details

A. Tabletop Manipulation Modules

We equip our tabletop manipulation agent with a suite of tools spanning perception, reasoning, control, pre-trained visuomotor policies, and image editing. The tools are provided in **bolded text** below.

Perception. For perception tools, we adopt a "coarse-to-fine" approach. At the fastest and simplest level, the agent can access **raw sensory inputs** (RGB images and robot proprioception). Because raw depth is often noisy, we use FoundationStereo [1] to estimate more reliable **depth maps** from RGB. For **object center point**, given a language description, we employ Grounded-SAM to produce a mask and its center point, while a Gemini-based pointing tool returns single 2D points corresponding to language queries. For tasks requiring higher precision—such as grasping towel corners—we provide a **salient task-relevant points** tool inspired by ReKep [2]: after generating a segmentation mask, we overlay a uniform point grid and ask GPT-o3 to select the most relevant points. This tool hierarchy balances granularity and runtime: raw images are coarse but fast, while salient points are slower but more precise. MAESTRO autonomously reasons about which tools to use and how to use them for each task, achieving a balance between execution speed and task performance.

Reasoning. In our initial experiments, we observed that even with advanced perception tools, MAESTRO struggled to perform task-relevant spatial reasoning. We identified the key limitation as the lack of spatial chain-of-thought reasoning in current VLMs. To address this, we equipped MAESTRO with a small suite of simple but targeted tools designed to spark spatial reasoning and break complex tasks into intermediate steps. These include **measuring distances**, **constructing vectors using points**, **computing relative rotations between vectors**, and **rotating a vector by specified angles**. Together, these tools dramatically improve MAESTRO's ability to reason about spatial relationships and act accordingly.

Control. For low-level manipulation, we provide a set of simple Cartesian end-effector control tools: **move gripper to**, **open gripper**, and **close gripper**. To ensure safe and reliable motion, we incorporate cuRobo [3] for point-cloud-based, collision-free motion planning, which greatly mitigates the risk of unintended contact with the environment and drastically improve object interaction performance.

Learned Visuomotor Policies. We equip MAESTRO with two types of learned visuomotor policies: a **grasp model** and a **VLA**. For grasping, we provide GraspGen [4], while for general-purpose low-level visuomotor control we provide the state-of-the-art $\pi_{0.5}$ model [5]. A key challenge when integrating end-to-end VLAs as tools is determining when to interrupt their execution, since they continue running inference until externally stopped. Because VLAs operate at high inference speeds, our framework requires an equally fast closed-loop monitor. To achieve this, we host Qwen-2.5-VL-72B-Instruct locally to check task completion at 2 Hz using a simple yes/no question, allowing rapid feedback and timely intervention.

Image Editing. Prior work [6, 7] shows that adding visual marks can improve the grounding and reasoning abilities of VLMs. Thus, we provide MAESTRO with image-editing tools that can **draw points** and **overlay 6D poses** on images, enabling clearer spatial references and richer visual reasoning during manipulation.

B. Additional Modules for Mobile Manipulation

To extend beyond tabletop tasks, we equip MAESTRO with additional tools that enable mobile manipulation.

Perception. For a mobile manipulator, obtaining the full **robot 6D state** requires more than raw proprioception. We therefore employ Faster-LIO [8], a lightweight LiDAR–Inertial Odometry method, to provide robust state estimation of the mobile base even under visually challenging conditions. This enhanced perception ensures reliable navigation and spatial reasoning across larger workspaces. In addition, mobile manipulation benefits from active perception to build a more complete understanding of the surroundings and gather task-relevant information on demand. To this end, we provide MAESTRO with active perception tools—**look left**, **look right**, **look to the ground**, and **view carry-on basket**, **remember object location**—which allow it to scan the environment, adjust its viewpoint, and access the basket contents as needed.

Locomotion. We supply two tools: (1) the **nudge** tool, which applies small velocity adjustments for precise positioning near a target location, and (2) the **navigation** tool, which leverages Nav2[9] to move the robot safely to a target pose on the map. This dual interface allows MAESTRO to fluidly switch between global navigation and fine-grained local control.

Manipulation. To support transport tasks, we provide a dedicated **put in basket** tool that enables the robot to

efficiently carry multiple objects during long-horizon mobile manipulation. For unloading, MAESTRO leverages the **view carry-on basket** perception tool to inspect the basket's contents and then automatically generates the code needed to move objects out of the basket.

II. Task Evaluation Rubric

We performed rigorous evaluation on all tasks following StarGen [10] to generate new trial for zero-shot generalization capabilities. 5 trials each task (1 initial setup following 4 generated by StarGen), and for each StarGen generated trial, vary all of the following:

- Object placement
- Object instance
- Scene / Lab setting
- Language instruction (paraphrase)

We designed a task completion tracking rubric to quantitatively evaluate the system performance, presenting results as a percentage of completion, where the maximum score is 100%. The overall task is decomposed into a sequential series of verifiable sub-steps. This metric moves beyond a simple binary success/failure and provides diagnostic detail on where the agent fails, which is critical for evaluating complex, long-horizon manipulation tasks.

The evaluation metrics along with StarGen variations for table-top manipulation tasks are detailed below:

1) Pick-place (*"Put item in bowl."*)
   [25%] Approach the item.
   [50%] Grasp the item.
   [60%] Lift up the item.
   [75%] Approach the target place.
   [100%] Place correctly.
2) Deformable object (*"Fold the four corners of the towel into the center."*)
   For this task, StarGen introduces significant geometric and interactive variability across trials (e.g., varying object shape and location), we could not rely on one static metric. Instead, the task completion rubric was dynamically fine-tuned for each generated trial, ensuring a precise and fair evaluation of progress completion against the specific, activated sub-goals of that unique scenario.
   a) *"Fold the four corners of the towel into the center."* (color of the towel is different between two trials)
      [25%] One corner folded in.
      [50%] Two corners folded in.
      [75%] Three corners folded in.
      [100%] All four corner folded in successfully.
   b) *"Fold the T-shirt into a rectangle."*
      [15%] One t-shirt sleeve folded in.
      [30%] Both t-shirt sleeves folded in.
      [60%] Bottom of t-shirt folded in.
      [100%] Fold entire t-shirt inward successfully.
   c) *"Place one corner of the towel to its diagonal corner."*
      [30%] Approach one corner.
      [60%] Grasp and lift up the corner.
      [100%] Place at the diagonal corner successfully.
   d) *"Unfold the towel into a square."*
      [30%] Accurately approach the folded corner.
      [60%] Grasp and lift up the corner.
      [100%] Place at the table to finish unfolding.
3) Articulated object (*"Open cabinet."*)
   [10%] Approach to the cabinet.
   [40%] Grasp the handle.
   [60%] Attempt to pull.
   [70%] Open the door slightly.
   [100%] Open the door completely.
4) Spatial reasoning (*"Rotate cube purple side up."*)
   [10%] Approach the cube.
   [30%] Grasp the cube.
   [60%] Rotate the cube.
   [100%] Purple side faces up.
5) Memory & long-horizon & semantic (*"Erase instructions on whiteboard, then follow instruction to stack cups."*)
   [20%] Pick up the erasor.
   [30%] Erase the white board partially.
   [50%] Erase the white board completely.
   [60%] Return the eraser.
   [80%] Stack the second cube on top of the first cube.
   [100%] Stack the third cube on top of the second cube.

The evaluation metrics for mobile manipulation tasks are detailed below:

1) Long-horizon manipulation (*"Collect all toys on table."*)
   [25%] Collect one toy.
   [50%] Collect two toys.
   [75%] Collect three toys.
   [100%] Collect all four toys.
2) Long-horizon loco-manipulation (*"Throw the ball into garbage can."*)
   [16.7%] Find the ball.
   [33.3%] Move to the ball.
   [50%] Pick up the ball.
   [66.7%] Find the trash can.
   [83.4%] Move to the trash can.
   [100%] Drop the ball into the trash can.
3) Active exploration (*"Search item and return when grasped."*)
   [20%] Explore around the area.
   [40%] See the object.
   [60%] Move to the object.
   [80%] Grasp the object.
   [100%] Return to the initial position.
4) Object affordance (*"Press button to open door."*)
   [33.3%] Identify the correct label.
   [66.6%] Approach to the correct button.
   [100%] Press button to open the door.

III. Extended Discussion on Prior Work

A. VLMs and LLMS in Robotics

Vision–Language Models (VLMs) and Large Language Models (LLMs) are among today's most capable and widely available "foundation" AI models. In robotics, they have

been applied primarily as modular components in manually designed pipelines in various ways, ranging from high-level planning [11, 12], reward and constraint design [2, 13–16], environment and hardware design [17–20], visual question answering [7, 21–23], in-context learning [24, 25], and task-progress evaluation [26–28]. In these settings, however, their role in the control loop remains limited. Our work focuses on a more direct use of VLMs as agentic policies—writing, executing, and adapting code to orchestrate robotic modules.

B. Large Models in Rigid Modular Robotic Systems

A wide range of prior work has explored the use of large models — Vision-Language Models (VLMs) and Large Language Models (LLMs) — to support specific components of robotic systems. In the common paradigm, these models are embedded into manually designed, typically rigid, modular pipelines, where large models take on one or several well-defined roles while the remaining pipeline is built through "good old-fashioned engineering" by humans. These roles span reward and environment design [13, 17–20], constraint-function design [2, 14–16], high-level planning [11, 12], self-reflection [29], visual question answering [7, 21–23], in-context learning [24, 25], and task-progress evaluation [26–28].

While effective within their intended scope, these approaches remain constrained by their rigid workflows. Because large models automate only a small portion of the pipeline, substantial manual effort is still required to design the rest. This rigidity also limits scalability: systems tuned and fixed for specific settings struggle to generalize to diverse, in-the-wild scenarios. As a result, these methods fall short of the requirements for scalable, general-purpose robotic manipulation.

We show that achieving this goal requires the opposite of prior practice: rather than increasing the complexity of the agentic system, especially its hand-engineered components, we reduce it. By stripping away rigid, manually defined workflows and instead scaling up the breadth and quality of the tools available to the agent, we create a more fluid and autonomous framework: one capable of dynamically deciding *how*, *when*, and *which* modules to employ.

C. Scaling up Data for Zero-Shot Robot Control

Data-driven approaches have recently become the dominant route toward general-purpose robotic manipulation. While some efforts have explored alternative data sources—such as simulation data [30–32] or human videos [33–35]—the most performant methods still rely on massive real-world teleoperation data [5, 36–39], which are costly and labor-intensive to collect.

In our experiments, we adopt the state-of-the-art $\pi_{0.5}$ model [5] as a strong baseline. We demonstrate that, beyond simply scaling robot data, scaling the *right set of tools* for a robotics agent can also yield general-purpose manipulation capabilities. Furthermore, we show that MAESTRO can strategically leverage VLAs as callable tools in addition to its original set of tools, thus providing coverage in scenarios where VLAs struggle or face out-of-distribution inputs, while still maintaining the efficiency and strengths of VLAs themselves.

Taken together, these results indicate that large-scale robot data is not the only viable path to generalist robotic manipulation. By appropriately scaling the toolset and autonomy of code-based agents, it is possible to achieve and even surpass the performance of data-heavy approaches in zero-shot settings.

REFERENCES

[1] B. Wen, M. Trepte, et al., "Foundationstereo: Zero-shot stereo matching," *arXiv*, 2025.

[2] W. Huang, C. Wang, et al., "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," *arXiv preprint arXiv:2409.01652*, 2024.

[3] B. Sundaralingam, S. K. S. Hari, et al., *Curobo: Parallelized collision-free minimum-jerk robot motion generation*, 2023. arXiv: 2310.17274 [cs.RO].

[4] A. Murali, B. Sundaralingam, et al., "Graspgen: A diffusion-based framework for 6-dof grasping with on-generator training," *arXiv preprint arXiv:2507.13097*, 2025. [Online]. Available: https://arxiv.org/abs/2507.13097.

[5] P. Intelligence, K. Black, et al., $\pi_{0.5}$*: A vision-language-action model with open-world generalization*, 2025. arXiv: 2504.16054 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2504.16054.

[6] J. Yang, H. Zhang, et al., "Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v," *arXiv preprint arXiv:2310.11441*, 2023.

[7] K. Fang, F. Liu, et al., "Moka: Open-world robotic manipulation through mark-based visual prompting," *Robotics: Science and Systems (RSS)*, 2024.

[8] C. Bai, T. Xiao, et al., "Faster-lio: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4861–4868, 2022.

[9] S. Macenski, F. Martín, et al., "The marathon 2: A navigation system," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 2718–2725.

[10] J. Gao, S. Belkhale, et al., "A taxonomy for evaluating generalist robot policies," 2025.

[11] M. Ahn, A. Brohan, et al., "Do as i can and not as i say: Grounding language in robotic affordances," in *arXiv preprint arXiv:2204.01691*, 2022.

[12] H. Ha, P. Florence, and S. Song, "Scaling up and distilling down: Language-guided robot skill acquisition," in *Proceedings of the 2023 Conference on Robot Learning*, 2023.

[13] Y. J. Ma, W. Liang, et al., "Eureka: Human-level reward design via coding large language models," *arXiv preprint arXiv: Arxiv-2310.12931*, 2023.

[14] W. Huang, C. Wang, et al., "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[15] H. Huang, F. Lin, et al., *Copa: General robotic manipulation through spatial constraints of parts with foundation models*, 2024. arXiv: 2403.08248 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2403.08248.

[16] S. Patel, X. Yin, et al., *A real-to-sim-to-real approach to robotic manipulation with vlm-generated iterative keypoint rewards*, 2025. arXiv: 2502.08643 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2502.08643.

[17] Y. J. Ma, W. Liang, et al., "Dreureka: Language model guided sim-to-real transfer," in *Robotics: Science and Systems (RSS)*, 2024.

[18] W. Liang, S. Wang, et al., "Environment curriculum generation via large language models," in *Conference on Robot Learning (CoRL)*, 2024.

[19] L. Le, J. Xie, et al., "Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model," *arXiv preprint arXiv:2410.13882*, 2024.

[20] G. J. Gao, T. Li, et al., "Vlmgineer: Vision language models as robotic toolsmiths," *arXiv preprint arXiv:2507.12644*, 2025.

[21] Y. Feng, J. Han, et al., *Reflective planning: Vision-language models for multi-stage long-horizon robotic manipulation*, 2025. arXiv: 2502.16707 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2502.16707.

[22] S. Nasiriany, F. Xia, et al., "Pivot: Iterative visual prompting elicits actionable knowledge for vlms," 2024. arXiv: 2402.07872 [cs.RO].

[23] W. Yuan, J. Duan, et al., *Robopoint: A vision-language model for spatial affordance prediction for robotics*, 2024. arXiv: 2406.10721 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2406.10721.

[24] N. Di Palo and E. Johns, "Keypoint action tokens enable in-context imitation learning in robotics," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[25] Y. Yin, Z. Wang, et al., "In-context learning enables robot action prediction in llms," in *ICRA*, 2025.

[26] Z. Liu, A. Bahety, and S. Song, "Reflect: Summarizing robot experiences for failure explanation and correction," *arXiv preprint arXiv:2306.15724*, 2023.

[27] J. Duan, W. Pumacay, et al., *Aha: A vision-language-model for detecting and reasoning over failures in robotic manipulation*, 2024. arXiv: 2410.00371 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2410.00371.

[28] Y. J. Ma, J. Hejna, et al., *Vision language models are in-context value learners*, 2024.

[29] W. Huang, F. Xia, et al., "Inner monologue: Embodied reasoning through planning with language models," in *arXiv preprint arXiv:2207.05608*, 2022.

[30] M. Dalal, M. Liu, et al., "Local policies enable zero-shot long-horizon manipulation," *International Conference of Robotics and Automation*, 2025.

[31] T. G. W. Lum, M. Matak, et al., "DextrAH-g: Pixels-to-action dexterous arm-hand grasping with geometric fabrics," in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=S2Jwb0i7HN.

[32] T. Lin, K. Sachdev, et al., "Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids," *arXiv:2502.20396*, 2025.

[33] J. Shi, Z. Zhao, et al., "Zeromimic: Distilling robotic manipulation skills from web videos," in *International Conference on Robotics and Automation (ICRA)*, 2025.

[34] R. Yang, Q. Yu, et al., *Egovla: Learning vision-language-action models from egocentric human videos*, 2025. arXiv: 2507.12440 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2507.12440.

[35] L. Y. Zhu, P. Kuppili, et al., *Emma: Scaling mobile manipulation via egocentric human data*, 2025. arXiv: 2509.04443 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2509.04443.

[36] K. Black, N. Brown, et al., "π_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[37] G. R. Team, S. Abeyruwan, et al., "Gemini robotics: Bringing ai into the physical world," *arXiv preprint arXiv:2503.20020*, 2025.

[38] J. Bjorck, F. Castañeda, et al., "Gr00t n1: An open foundation model for generalist humanoid robots," *arXiv preprint arXiv:2503.14734*, 2025.

[39] J. Lee, J. Duan, et al., *Molmoact: Action reasoning models that can reason in space*, 2025. arXiv: 2508.07917 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2508.07917.