SynthLA: Synthetic Language–Action Policies for Zero-shot Real-world Manipulation via Structured Perception

Marco Maccarini[1,2,*], Angelo Moroncelli[1,2,*], Asad Ali Shahid [1,3], Samuele Mara[1,2], Mirko Nava[1] and Loris Roveda[1,3]

***Abstract*— Synthetic data offers a scalable alternative to costly real-world data collection for robot learning, yet most language-conditioned manipulation systems still rely on large demonstration datasets or predefined primitive libraries. We present SynthLA, a modular framework that learns low-level language-action policies for robot manipulation entirely from synthetic data. Our key idea is to use structured text as an interface between perception and control. A vision model converts RGB-D observations into a textual scene representation, and a fine-tuned large language model predicts the next low-level robot action in closed loop. Training data are generated automatically by randomizing symbolic scenes and labeling them with a deterministic geometric oracle, producing millions of task-state-action triplets without robot demonstrations. We evaluate SynthLA on four real-world tabletop tasks under both in-distribution and out-of-distribution conditions, as well as static and dynamic scenes. Despite being trained exclusively on synthetic supervision, SynthLA achieves a 73% mean success rate, outperforming a fine-tuned OpenVLA baseline (60%). Our results demonstrate that structured representations can enable effective sim-to-real transfer, highlighting synthetic data as a practical and scalable catalyst for robust robot manipulation.**

I. Introduction

Synthetic data is emerging as a key driver of progress in robot learning, offering a scalable alternative to real-world data collection that is often expensive, slow, and difficult to control. Recent advances in simulation, rendering, and procedural generation have enabled large-scale synthetic datasets for perception and control, while transferring models to real-world robotic systems remains an open problem. Specifically for language-conditioned manipulation, sim-to-real transfer is challenging due to the entanglement of perception and control, where errors in visual grounding or language understanding can directly propagate to low-level actions under real-world variability. Existing approaches typically fall into two categories. *LLM-based planners* decompose instructions into sequences of predefined robot primitives [1]–[4], enabling flexible task reasoning but requiring reliable primitive libraries and accurate state estimation. In contrast, *vision-language-action* (VLA) models directly map images and instructions to low-level actions [5]–[8], but depend on large real-world datasets of task demonstrations, and suffer from limited robustness under distribution shift [9], [10].

In this work, we explore an alternative design enabled by synthetic data. Our key idea is to decouple perception and control through a structured textual interface. A perception module converts RGB-D observations into a compact symbolic scene description, and a language-action policy predicts low-level actions conditioned on this representation. This abstraction allows the policy to be trained entirely on synthetic data while remaining compatible with real-world execution; synthetic data is not only used for pretraining, but serves as the primary mechanism for policy learning, controlled evaluation, and sim-to-real transfer. By operating on structured state representations rather than raw pixels, our approach mitigates visual domain gaps and enables scalable data generation through procedural scene randomization. Our contributions are threefold:

- We introduce a synthetic data generation pipeline that produces large-scale supervision for language-conditioned manipulation without demonstrations.
- We propose a structured perception interface that enables effective transfer from synthetic training to real-world execution.
- We demonstrate on real robots that policies trained purely on synthetic data can achieve competitive performance and improved robustness compared to data-intensive baselines.

II. Method

Our goal is to learn a multi-task, language-conditioned policy $\pi : \mathcal{Q} \times \mathcal{S} \to \mathcal{A}$, where $\mathcal{Q}$ denotes the space of language instructions, $\mathcal{S}$ the state space, and $\mathcal{A}$ the action space. We define a state $s \in \mathcal{S}$ as $s = (s_{\text{robot}},\ s_{\text{env}})$, where the robot state is given by $s_{\text{robot}} = (\mathbf{p}_{ee},\ g)$, with $\mathbf{p}_{ee}$ the end-effector (EE) pose and $g \in \mathcal{G}$ the gripper state (e.g., open/closed). In our setting, we operate directly in EE pose space. The environment state is represented as $s_{\text{env}} = \{(c_i, \mathbf{p}_i, \theta_i)\}_{i=1}^{M}$, where c_i is the object class, $\mathbf{p}_i \in \mathbb{R}^3$ its position, and θ_i its orientation. The action $a \in \mathcal{A}$ corresponds to a desired change (or target) in EE pose space together with a gripper command. Figure 1 summarizes the proposed synthetic data-generation process. Given a task, we decompose it into an ordered sequence of primitive sub-tasks. Each primitive is implemented by a *geometric oracle*, defined as a deterministic expert policy

* equal contribution, [1] Marco Maccarini, Angelo Moroncelli, Asad Ali Shahid, Samuele Mara, Mirko Nava, Loris Roveda are with the Department of Innovative Technologies, University of Applied Science and Arts of Southern Switzerland (SUPSI), Istituto Dalle Molle di studi sull'intelligenza artificiale (IDSIA), Lugano, 6900, Switzerland {name.surname}@supsi.ch

[2] Marco Maccarini, Angelo Moroncelli and Samuele Mara are with Università della Svizzera Italiana, Faculty of Informatics, Lugano, 6900, Switzerland {name.surname}@usi.ch

[3] Asad Ali Shahid and Loris Roveda are with Politecnico di Milano, Mechanical Department, Milano, 20156, Italy{name.surname}@polimi.it

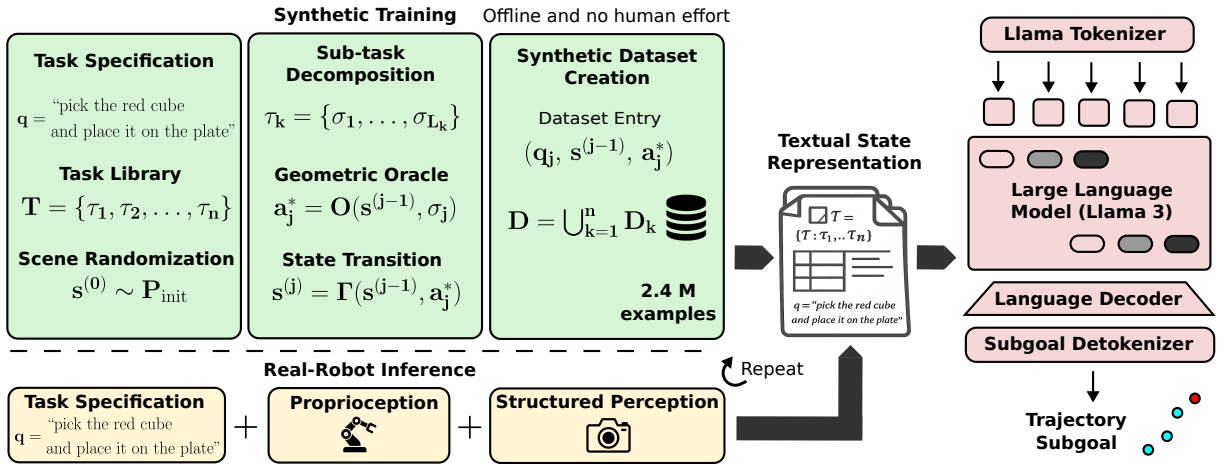

Fig. 1: SynthLA uses synthetic task-state-action triplets to train a low-level language-action policy. A randomized symbolic scene is paired with a user instruction and labeled by a deterministic geometric oracle. At deployment, a vision model converts RGB-D observations into the same structured textual state used during training.

that uses the current symbolic state and the current sub-task specification to produce the corresponding low-level action, e.g., an absolute Cartesian target for the end effector or an open/close gripper command. Data are generated in *symbolic environments*, namely procedurally generated state configurations in which the scene is represented by structured variables rather than pixels, including the robot pose and the object identities, positions, and orientations. This abstraction allows us to synthesize large numbers of supervision samples efficiently, while preserving the geometric constraints relevant to the manipulation task. After fine-tuning the LLM-based policy on the generated data, it is deployed in the real world, leveraging a computer vision model to detect and localize objects in the scene to compose the structured textual state representation used as the policy input. During training, the model receives the instruction and structured state as text, and predicts the next low-level action token sequence. Leveraging a textual representation and grounding the policy in object poses resolves the entanglement problem of perception and control of VLAs.

A. Synthetic data generation

For each task, we build a procedure that automatically creates many random scene configurations and assigns the correct next action to each one. Given a task, we manually define a sequence of control steps to progress the task toward completion, without the need to actuate any robot. Each generated episode starts from a feasible random configuration of the robot and of the objects in the scene. The task is then expressed as a sequence of simple stages such as reaching the target object, grasping it, lifting it, moving toward the destination, and releasing it. We use handcrafted rules that look at the current state and stage of the task, and return the next low-level action that should be executed. The rules encode the long-term plan given a specific task, i.e., they define what the robot should do next to move toward the current goal. As an example, the instruction *"pick the carrot and place it on the plate"* is converted into a sequence of intermediate geometric goals: move above the carrot, descend, close the gripper, lift, move above the plate, descend, and open the gripper. For each step, the corresponding target action is generated deterministically from the state consisting of the current robot pose and object poses, and represented in the same structured textual format used by the policy.

The generation process produces triplets (q, s, a^*), where q is the instruction, s is the current symbolic state, and a^* is the expert next action. Since labels are generated automatically from manually specified task rules, the dataset scales well to millions of triplets, and does so without human annotations or robot rollouts. This also gives direct control over the state distribution, task variations, and difficult corner cases that would be hard to cover consistently with real-world data collection.

B. Training and deployment

We adapt a pretrained LLM into a language-action policy using QLoRA [11], a balance between training cost and performance, using only our synthetic generate data. During deployment, we generate the structured state with a SoA vision model that processes RGB-D observations and estimates object classes and poses. These detections are combined with proprioceptive state estimates into the structured textual representation, matching the structure used during synthetic training, allowing the policy to operate in closed loop on the real robot. The policy predicts target poses and gripper commands at approximately 3Hz, while a low-level controller continuously tracks the latest target, enabling online correction when objects move.

III. Results

We evaluate SynthLA on four real-world tabletop tasks: *lift*, *pick&place*, *move*, and *close-the-pot*, where we measure the agent task success rate over multiple real-world robot trials. We consider combinations of environment settings:

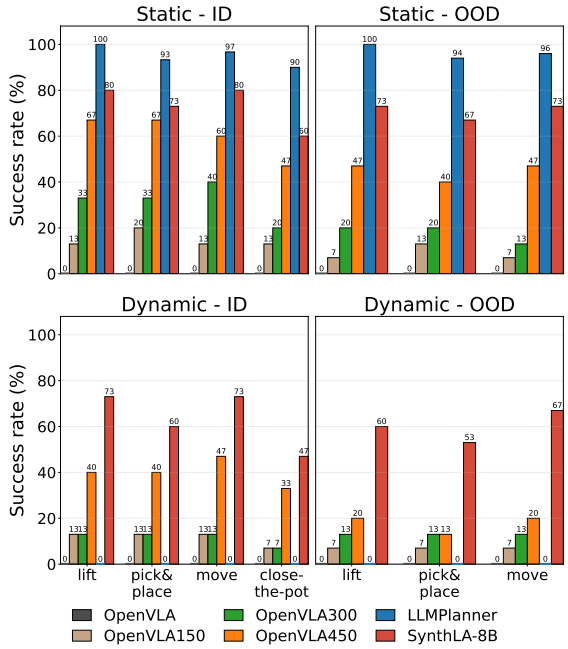

Fig. 2: Success rate comparison in static and dynamic scenes. SynthLA remains competitive in real-world execution despite being trained only on synthetic data, and is more robust than OpenVLA under scene perturbations.

static and *dynamic* scenes; manipulation of *in-distribution* (ID) and *out-of-distribution* (OOD) objects. Specifically in *static* scenes, object poses remain fixed, while in *dynamic* ones, objects are moves by an operator during task execution. For perception, we use a YOLO-v8 OBB detector fine-tuned on a custom RGB-D dataset featuring household objects. The language-action backbone is Llama 3.1 8B [12]. SynthLA is trained exclusively on synthetic textual triplets, totaling 2.4M triplets (600k per task) generated over 2 minutes with procedural generation. We compare against OpenVLA [7] and an LLM-based planner using predefined primitives.

SynthLA achieves a mean success rate of 73% in static scenes (ID: 73%, OOD: 71%), outperforming a fine-tuned OpenVLA baseline (60% ID, 45% OOD). In dynamic scenes, performance decreases moderately to 63% (ID) and 60% (OOD), indicating robustness to perturbations. These results highlight a key advantage of synthetic data: the ability to generate diverse and structured supervision at scale. By training on symbolic representations, SynthLA learns task-relevant geometric relationships rather than overfitting to visual appearance, improving generalization under distribution shift. The relatively small gap between ID and OOD performance suggests strong instance-level generalization, despite zero real-world training data. Compared to VLA models, SynthLA is more robust to scene changes, while retaining closed-loop adaptability absent in primitive-based planners.

IV. Discussion and Limitations

SynthLA is best suited for structured manipulation scenarios where tasks and object categories are known in advance. Its main advantage lies in scalable policy learning from synthetic data, enabling controlled evaluation across diverse conditions. However, several limitations remain. First, synthetic data generation requires task-specific design, including decomposition into sub-tasks and oracle definitions. Second, the perception module assumes a fixed set of detectable object categories. Third, the current evaluation focuses on short-horizon tasks. These limitations suggest directions for future work aligned with the workshop themes: richer synthetic environments, improved sim-to-real transfer for perception, extension to deformable objects, and standardized synthetic benchmarks for evaluation. More broadly, our results indicate that the effectiveness of synthetic data depends critically on representation design. By abstracting away low-level visual details, structured representations reduce the sim-to-real gap and enable more scalable learning.

V. Conclusion

We presented SynthLA, a modular framework for learning language-conditioned robot control entirely from synthetic data. By using structured text as the interface between perception and action, our approach enables scalable supervision, eliminates the need for demonstrations, and supports zero-shot real-world deployment. Experiments on real robots show that SynthLA achieves competitive performance and strong robustness under distribution shift and dynamic conditions. These findings reinforce the central premise of this workshop: synthetic data can act as a practical catalyst for robot learning, enabling new training paradigms that improve scalability, robustness, and reliability.

References

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

[2] R. Hazra et al. Saycanpay: Heuristic planning with large language models using learnable domain knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[3] I. Singh et al. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11523–11530. IEEE, 2023.

[4] J. Liang et al. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation*, pp. 9493–9500, 2023.

[5] A. Brohan et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[6] Octo Model Team et al. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024.

[7] M. J. Kim et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[8] K. Black et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[9] X. Li et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

[10] J. Zhou et al. Exploring the limits of vision-language-action manipulations in cross-task generalization. *arXiv preprint arXiv:2505.15660*, 2025.

[11] T. Dettmers et al. Qlora: Efficient finetuning of quantized llms. In A. Oh et al., editors, *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115, 2023.

[12] A. Dubey et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.