CoDex: Learning Compositional Dexterous Functional Manipulation without Demonstrations

Anonymous Author(s)
Affiliation
Address
email

Abstract: Functional Object Manipulation (FOM) tasks require interacting with an object to activate its intended function. When the object features internal mechanisms such as triggers or buttons, success requires coordinated control over both the object's internal and external degrees of freedom, e.g., when aiming and operating a spray bottle or a glue gun. Tasks like these pose significant challenges for robots, requiring the integration of semantic understanding (of the object's function, actuation mode, and spatial goal) with intricate physical dexterity (to manage grasp stability, movement trajectory, and actuation) to control a high-DoF dexterous hand and arm. We introduce **CoDex**, a zero-demonstration framework leveraging VLM guidance for online dexterous grasp synthesis and policy learning in simulation, with direct transfer to the real world. **CoDex** leverages VLM semantic knowledge to generate analytical candidate functional grasps, optimized to activate the object's functionality. The generated grasps serve as initialization for a Reinforcement Learning procedure guided by VLM-provided objectives that optimizes a parameterized primitive with VLM-generated goals towards high success in a composed grasp-move-actuate sequence. We evaluate **CoDex** on a physical robot performing 6 FOM tasks involving previously unseen objects with internal mechanisms (e.g. spray bottles, hot glue gun, air duster, etc.) and their application on various unseen target objects, showcasing its ability to autonomously discover and execute complex, physically viable dexterous behaviors without human demonstrations. More information at codex-2025.github.io.

Keywords: Functional Object Manipulation, Dexterous Manipulation, Vision-Language Models

1 Introduction

Imagine a robot tasked with spraying a plant. It needs to grasp the bottle stably, move and aim it correctly, and squeeze the trigger – coordinating these actions simultaneously. This kind of task, where a robot uses an object for its specific purpose, is a special case of **Functional Object Manipulation (FOM)** [1–5] requiring the actuation of the object's mechanism like triggers, buttons, or levers. Manipulating these objects effectively requires the robot to precisely coordinate operating the tool's internal degrees of freedom (DoF) while controlling its overall position and stability [6, 7]. Achieving this level of integrated dexterity for versatile tool use remains an open challenge in the robotics.

Fundamentally, the difficulty of FOM lies in bridging the gap between **high-level semantic understanding** and **intricate physical dexterity**. The robot must interpret the task context – understanding the tool's function, identifying how and where to actuate it (*local* semantics), and reasoning about the desired outcome relative to other objects (*global* semantics). Simultaneously, it must execute the task physically: achieving a stable yet functional grasp, coordinating complex hand-arm

Submitted to the 9th Conference on Robot Learning (CoRL 2025). Do not distribute.

motions, and applying precise forces, all using a high-DoF dexterous hand. Effectively composing semantics and physical skill is crucial for success.

Learning from Demonstration methods acquire the semantic understanding from human teachers and have demonstrated advanced control capabilities even for contact-rich tasks with simple hands [8], but the difficulties of teleoperating complex multi-fingered hands [9–11] limit scalability and generalization in this domain [12, 7], requiring task-specific demonstrations for every new object [7, 13]. While imitating human videos eliminates the need for labor-intensive teleoperation, it requires learning object-specific strategies to overcome the significant morphological differences, which are hard to generalize [7, 14].

Alternatively, approaches like Reinforcement Learning [15, 16] and analytical grasp synthesis [17–21] bypass demonstrations but shift the burden to engineering detailed, object-specific guidance. Manually designing effective reward functions for RL or precise optimization targets for synthesis for every new tool severely limits the ability to scale to the vast diversity of real-world functional manipulation tasks.

Conversely, broad generalization can be achieved by integrating large-scale pre-trained models, such as *Vision-Language Models (VLMs)*, into robotic solutions [22, 23]. VLMs provide high-level semantic understanding of the manipulation tasks without task-specific demonstrations, thus enabling a zero-shot generalization. However, VLMs' limitations in geometric and embodied understanding restrict their guidance to a coarse, abstract level, lacking the capabilities to specify the intricate, coordinated hand-and-arm motions required for dexterous functional tasks [24–26].

We introduce **CoDex**, a zero-demonstration framework designed to compose semantic understanding and physical dexterity for these complex FOM tasks. **CoDex** integrates VLM guidance with an online pipeline where **grasp candidates generated via constrained optimization seed holistic policy learning** using simulation-based RL. A VLM provides semantic guidance at both a local level (e.g., identifying functional points like triggers/nozzles) and a global level (e.g., determining target object poses via iterative refinement), informing both the grasp optimization and the RL process to discover the complete composed grasp-move-actuate policy.

In a nutshell, with CoDex we advance the state-of-the-art in robot learning with:

- **Grounding VLM Guidance for Physical Dexterity:** Translating high-level VLM semantic interpretation (local/global task understanding) into concrete objectives, constraints, and targets suitable for driving online grasp candidate generation and policy learning.
- **Holistic Policy Learning for Composed Actions:** An online pipeline that generates functional grasp candidates via constrained optimization and discovers physically viable grasp-move-actuate policies via simulation-based policy learning (RL), optimizing for task success.
- **Autonomous Functional Object Manipulation without Demonstrations:** Demonstrating the practical viability and generalization of the framework by enabling a physical robot to autonomously solve 4 out of 6 diverse, highly complex tool-use tasks requiring coordinated dexterity, entirely without human demonstrations.

2 Related Work

CoDex proposes a novel method for *(1) compositional dexterous manipulation that simultaneously addresses in-hand and extrinsic actions*, integrating a new mechanism for *(2) functional grasping for tool use with physics-based refinement* that makes use of a *(3) VLM guidance for robot manipulation*. In the following, we review prior work in these areas.

Composed Dexterous Manipulation. Often, successful tool use demands composing the control of both in-hand adjustments and whole-arm extrinsic motions [7, 13, 6, 27], but most existing works on dexterous manipulation focus on one or the other. For instance, significant research addresses in-hand manipulation, focusing on fine finger coordination for tasks like object reorientation [5, 28, 13, 12], or rotating caps [2, 29, 30], without consideration for full arm motion to control the

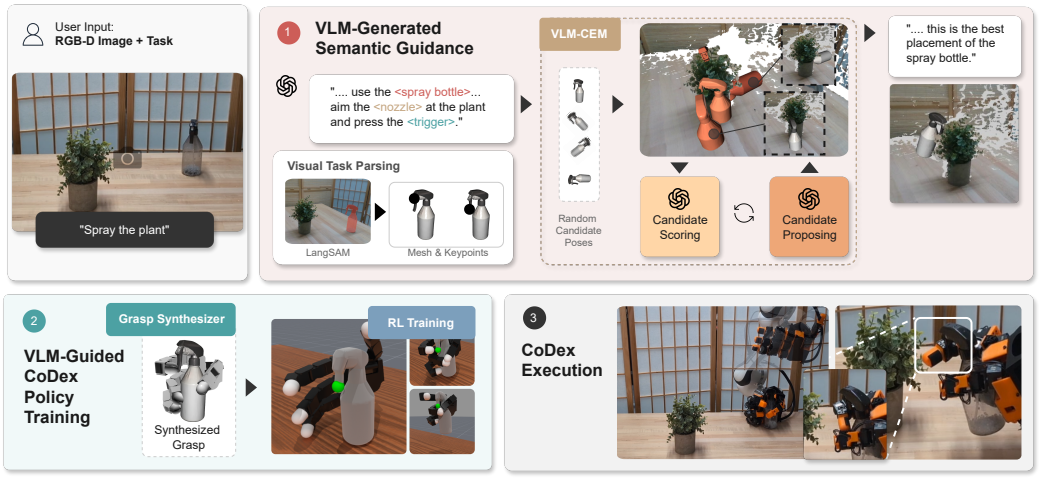

Figure 1: Overview of the CoDex zero-demonstration pipeline, structured in three main stages. (1) **VLM-Generated Semantic Guidance:** This stage processes the initial inputs (image, text command). It includes *Visual Task Parsing* (segmentation, object mesh reconstruction, and keypoints projection) and employs *VLM-CEM* (iteratively scoring/proposing candidates) to determine the target External Goal Pose for the object. (2) **VLM-Guided CoDex Policy Training:** Leveraging semantic information from Stage 1 (e.g., keypoints defining optimization targets), an analytical *Grasp Synthesizer* generates initial grasp candidates. These candidates then seed an online *RL policy training* process in simulation to produce a manipulation policy. (3) **CoDex Execution:** The trained policy from Stage 2, incorporating the External Goal Pose from Stage 1, guides the physical robot to perform the complete grasp-move-actuate sequence for the functional task.

object's overall trajectory for a task. Other works tackle extrinsic manipulation, where a grasped tool interacts with the environment, such as hammering or shoveling [5, 31, 32] on objects without internal degrees of freedom. Closer to ours, Agarwal et al. [5] provide a composed solution for arm and multi-fingered hand motion, but their method requires human demonstrations and focuses on optimizing the grasp stability to resist the forces resulting from subsequent arm motion, failing to actuate the object's internal degrees of freedom. In contrast, CoDex holistically addresses the entire grasp-move-and-actuate problem, generating a composed solution that actuates both the object's internal and external degrees of freedom.

Functional Object Grasping for Tool Use. In task-oriented grasping, the aim is to select a grasp that not only stabilizes the object but also facilitates the intended function [5, 4, 33–37]. Several methods focus on optimizing contacts to achieve stable grasps, often predicting force-closure metrics like the Ferrari-Canny [38, 35–37], but cannot be applied to grasps that enable the actuation of objects' internal degrees of freedom. Recent analytical strategies [4, 34, 33] consider internal degrees of freedom but they do not compose them with post-grasping trajectories to actuate the object at the right location to achieve a task. All these methods aim to provide a grasp synthesis solution that generates a successful grasp from images to be executed by a predefined controller, which can lead to failures. Recently, some methods have integrated a simulator with a model of the specific object into the loop for online improvement of grasping strategies using reinforcement learning or exploiting the simulator's differentiability for optimization [39, 40, 2, 37, 28]. While demonstrating better performance, this strategy requires manual annotation of the rewards and has yet to be extended to complex objects with internal degrees of freedom (DoF) and post-grasping motion. Moreover, all previous methods improve grasp stability and/or functionality, but treat grasping as an isolated problem, missing the opportunity to reason about it in conjunction with subsequent motion to enhance dynamics and stability during actuation. By tightly coupling high-level semantic cues with low-level physical simulation and optimizing the holistic grasp-move-actuate task online, our approach robustly bridges the gap between grasp synthesis and action generation for newly seen objects—addressing limitations inherent in previous frameworks.

Semantic Grounding via Vision-Language Models. Large Vision-Language Models (VLMs) enable zero-shot semantic understanding for robotics [22, 23], interpreting high-level goals from language and vision. Approaches like ReKep [22] or PIVOT [23] excel at identifying interaction points or refining coarse actions based on semantics. However, VLMs typically provide abstract guidance and often lack the detailed physical grounding needed to generate the precise, coordinated motions required for dexterous functional tasks [24–26]. CoDex utilizes VLM-derived semantic guidance (providing both local and global cues) specifically to inform and structure its physics-based optimization and policy learning process, effectively grounding abstract VLM knowledge into viable, dexterous actions.

3 CoDex: VLM-Guided Compositional Dexterous Functional Manipulation

Our CoDex framework integrates high-level semantic understanding derived from Vision-Language Models (VLMs) with online physical synthesis and learning to achieve zero-demonstration functional object manipulation. As illustrated in Fig. 1, the pipeline consists of three main stages executed sequentially: VLM-Generated Semantic Guidance, VLM-Guided CoDex Policy Training, and CoDex Execution. Below, we detail each stage.

3.1 VLM-Generated Semantic Guidance

This initial stage processes the primary inputs – a language task description and an RGB-D scene observation – using Vision-Language Models (VLMs). Its goal is to interpret the task and scene context, ultimately outputting crucial semantic and geometric guidance for the subsequent policy training stage. This guidance includes both **local** information (e.g., key object points for interaction, 3D object mesh) and **global** information (e.g., the target 6D pose of the functional object).

First, *Visual Task Parsing* processes the inputs (text description, RGB-D image). Using VLM queries and open-vocabulary segmentation, CoDex identifies the functional and target objects, even if unseen. The segmented functional object's mesh is then reconstructed using Tripo [41].

Next, CoDex queries the VLM with the object's image or mesh to extract **local semantic guidance**: identifying critical interaction points such as the **Activation Point** (e.g., trigger center, where force is applied) and the **Effector Point** (e.g., nozzle, where the function is applied). This guidance informs subsequent grasp planning. We typically assume the required actuation force is applied along the surface normal at the activation point. (See Appendix Fig. 4 for examples).

Finally, CoDex determines the **global semantic guidance** – the target 6D pose for the functional object relative to the target object (e.g., aiming the nozzle at the plant). This is achieved via our VLM-guided Cross-Entropy Method (VLM-CEM) process, drawing inspiration from [23]. As illustrated conceptually in Fig. 1 (Stage 1) and detailed in Appendix, VLM-CEM iteratively refines candidate poses. It uses a VLM to score synthetic renderings of the object at candidate poses, ensuring the final pose is both semantically aligned with the task (based on VLM scoring) and geometrically grounded in the scene.

3.2 VLM-Guided CoDex Policy Training

Given the functional object's geometry and the target actuation point identified by the VLM, CoDex generates a diverse set of initial grasp candidates optimized for the task's functional requirements and uses them as initialization for an online reinforcement learning process with action primitives in simulation.

First, a set of initial **grasp candidates** is generated via **constrained optimization**. Adapting prior work on grasp optimization [21], we formulate an objective (Eq. 1) that minimizes grasp effort, $\text{Effort}(q)$, for a hand configuration q. This minimization is subject to standard physical constraints ensuring grasp stability (e.g., force closure [38]), surface contact , and collision avoidance, as well as crucial **functional constraints derived from VLM guidance**. Specifically, these functional con-

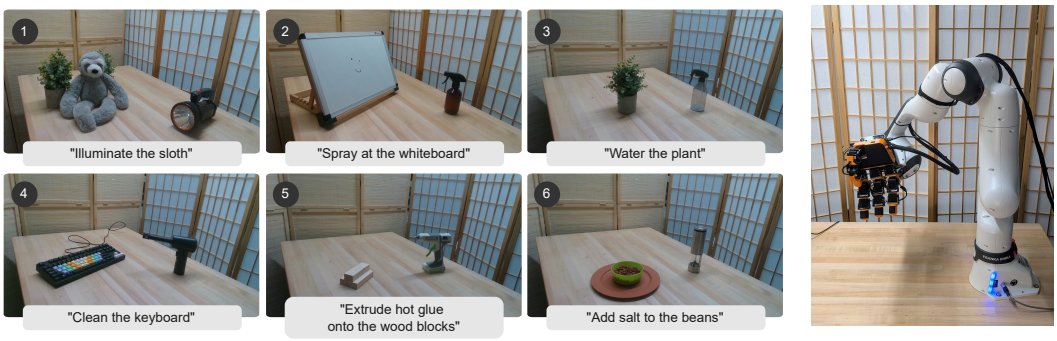

Figure 2: (*Left*) Six functional object manipulation tasks in our experiments. They require combining local manipulation of functional objects with internal DoF (flashlight, board spray, water spray, air blower, hotglue gun, and salt grinder) with their global motion in the scene. (*Right*) Our robot arm and multi-fingered hand. CoDex learns to use a Franka Emika Panda arm with 7 DoF and a LEAP hand with 23 DoF attached as end-effector for the tasks without any demonstrations.

straints enforce that at least one designated finger achieves appropriate proximity and alignment relative to the VLM-identified actuation point p_{act} and direction d_{act} (see Appendix for full formulation). The optimization structure is:

$$\begin{aligned} \min_{q \in \mathcal{Q}} \quad & \text{Effort}(q) \\ \text{s.t.} \quad & \text{IsStable}(q) \\ & \text{IsCollisionFree}(q) \\ & \text{IsFunctional}(q, p_{act}, d_{act}) \end{aligned} \tag{1}$$

where $\mathcal{Q}$ represents the valid joint space, and the constraint functions enforce stability, collision-free configurations, and the VLM-guided functional requirements, respectively.

This optimization process yields a diverse set of grasp candidates biased towards stability and potential functionality. However, this static optimization does not guarantee dynamic stability during movement or successful actuation under load, nor does it account for the approach trajectory or environment interactions. Therefore, these grasp candidates serve as initialization seeds for the subsequent holistic policy learning stage.

To generate physically plausible compositional FOM policies, CoDex uses the grasp candidates to initialize a reinforcement learning policy in simulation. CoDex makes use of the VLM guidance to optimize for the compositional success of the manipulation. Beyond a regular stability success (whether the object remains securely grasped and moves congruently with the hand), CoDex rewards for functional success: applying sufficient contact force at the activation point along the actuation direction. This unified reward function applies across different functional object manipulation tasks and objects, avoiding the need for the task-specific reward engineering often required in prior work [33].

To accelerate policy training and enable online RL of new functional objects, CoDex implements a grasp-move-actuate primitive that generates a trajectory to a pregrasping and grasping states (hand and finger configuration, initialized with one of the grasping candidates of the previous step), followed by the application of force by closing the functional finger. With this primitive action space, the policy training optimizes an action space that controls the pregrasping and grasping states, giving the policy control over the motion of the hand and fingers during the dynamic approach to the functional object. The resulting RL process optimizes the policy parameters online, effectively refining the initial grasp candidate towards a dynamically stable FOM policy for the grasp-move-actuate compositional motion.

Table 1: CoDex performance across six Compositional Dexterous Functional Manipulation Tasks (Successes / 5 Trials)

Metric	Illuminate toy	Spray whiteboard	Spray plant	Clean keyboard	Glue blocks	Grind salt
Stable Grasp	4/5	5/5	5/5	4/5	5/5	4/5
Successful Movement	1/5	5/5	5/5	3/5	5/5	3/5
Correct Actuation	0/5	5/5	5/5	4/5	5/5	0/5
Holistic Success	0/5	5/5	5/5	3/5	5/5	0/5

3.3 CoDex Execution in Real World

The result of the previous step is a full policy that generates a compositional grasp-move-actuate motion for the FOM task. In the final step, the RL policy is executed on the real robot, using a Franka arm with a LEAP hand.

4 Experimental Evaluation

In our experiments, we evaluate whether the proposed **CoDex** framework successfully bridges the gap between high–level vision–language understanding and low-level, physics-grounded execution. To that end, our experiments aim to answer the following three research questions:

Q1 *How well does CoDex perform in compositional grasp–move–actuate dexterous functional manipulation tasks in the real world?*

Q2 *Does the VLM-CEM procedure propose external goals that are both semantically **and** physically appropriate for the task?*

Q3 *How much does the **compositional policy learning** improve success compared to attempting directly the candidate grasps from constrained optimization?*

Experimental Setup: We test CoDex on a 7-DoF FRANKA Emika Panda arm, with a 16-DoF LEAP Hand mounted as end-effector, as shown in Fig 2. RL Policies are trained in MANISKILL3 using 2,048 parallel worlds and directly deployed on the real robot. We test on the six functional manipulation tasks introduced in §3. At the start of every episode, the object is placed at a random pose on the table before starting the trial. Each trial is evaluated with four binary criteria: (i) Stable grasp–the object does not slip from the hand, (ii) Successful movement–the object is moved to the task-required pose, (iii) Successful actuation–the mechanism is triggered, and (iv) Holistic success–if all the criteria are fulfilled and there are no other failures.

Q1 — Performance on Compositional Dexterous Functional Manipulation Tasks

Table 1 summarizes our real-world results (30 trials): CoDex achieved **60%** holistic task success and **90%** grasp stability during execution. Failures, concentrated primarily on the challenging `illuminate toy` and `grind salt` tasks, highlight remaining complexities in integrating semantic goals with physical execution for these demanding FOM tasks.

One category of failures involved grasp-pose conflicts. In some trials, the hand orientation required by the functionally necessary grasp proved incompatible with achieving the VLM-generated target object pose without collision. For instance, to effectively `illuminate the toy`, the flashlight might need to be aimed such that the hand, having grasped the handle appropriately for button access, would collide with the surface supporting the toy during placement. This underscores the challenge of satisfying both local grasp function and global placement goals using only rigid motions within the fixed-base robot's constraints, suggesting potential benefits of incorporating capabilities like in-hand reorientation.

Failures also occurred during in-hand actuation, revealing sensitivity to precise contact physics and sim2real gaps, particularly evident in the two most challenging tasks. The `illuminate toy` task demand sub-centimeter precision on its small ($< 1cm$), soft button; slight contact misalignments

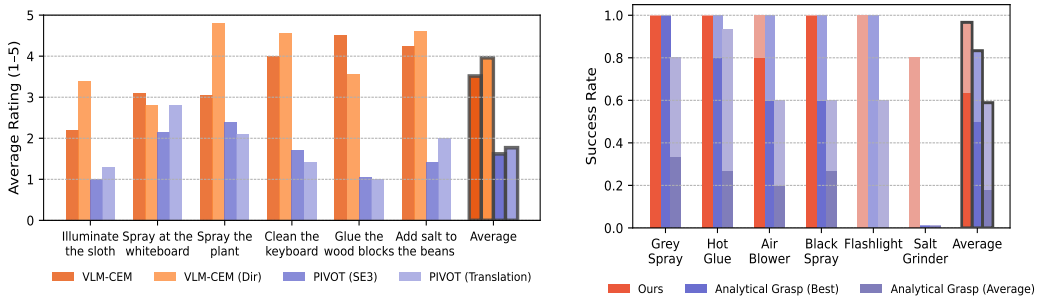

Figure 3: (*Left*) Human study ratings of generated global goals. We request human feedback on the global goals generated by our VLM-CEM procedure and baselines (VLM-CEM without rotation changes, PIVOT with rotation and without rotations). CoDex's procedure is ranked higher on average and in most tasks. (*Right*) Performance gains of CoDex compositional policy training compared to the direct execution of the 3 and the best original candidates from CoDex's constrained optimization. Total bar height indicates the success rate of achieving a stable grasp through lifting. The bottom segment (darker shade) represents the success rate of achieving both a stable grasp *and* successful actuation. By training with RL on a physical simulator using VLM-guided reward computation for the full task, CoDex significantly improves stability and actuation of the objects.

frequently caused the finger to slip off before successful actuation. The `grind salt` task suffered from a significant sim2real friction gap; the low friction of the physical metallic grinder led to grasp instability or slippage specifically during the forceful clicking motion, a behavior that requires **aggressive** friction randomization to be fully captured.

These challenging cases indicate that while our method significantly advances zero-shot FOM, robustly handling scenarios demanding extreme precision (like `illuminate toy`) or subject to substantial sim2real variations in contact physics (like `grind salt`) remains an important direction for future work, potentially involving richer object modeling and adaptive learning techniques.

Q2 — Quality of VLM-CEM Generated Goal Proposals

To evaluate the quality of our novel VLM-CEM procedure for generating goals, we asked seven participants to rate the rendered images generated based on the resulting goal poses from the VLM-CEM and three additional baselines:

- **VLM–CEM**: our keypoint-anchored sampler that generates candidate goal poses around detected interaction points on the object.
- **VLM–CEM (Dir.)**: a variant that restricts sampling to translations along the object's functional axis (e.g., nozzle direction); see Appendix for details.
- **PIVOT (SE3)** [23]: an adaptation of PIVOT that perturbs full 6-DoF poses in image-space without explicit keypoint anchoring, often resulting in misalignment in depth or lateral offset.
- **PIVOT (Trans.)**: akin to the original PIVOT method, searching only in 2D image-space translations based on coarse visual alignment.

We generated three multi-view images per task per method and requested human ratings on a five-point scale (1 = unreasonable, 5 = perfect), where a human rating of ($\geq$ 3) is considered as semantically acceptable in our analysis.

Fig. 3, left, depicts the result of our human ratings. We observe that our VLM-CEM with directional exploration, along with the variant without directional exploration, is ranked the highest by humans in all tasks. Surprisingly, for the *spray the plant* task, our VLM-CEM with directional exploration ablation scores significantly higher than the VLM-CEM with full translational exploration. We hypothesize this is because the directional constraint, while reducing the exploration space, effectively filters out candidate poses that might appear plausible in the 2D rendered images used for VLM scoring but are functionally misaligned in 3D (e.g., aiming near the plant but slightly off-axis). By

enforcing alignment along the functionally critical nozzle direction, the directional variant ensures better geometric task relevance for the highest-scoring poses in this specific spraying task.

By manually inspecting the lowest-scoring cases, we observe that most of them correspond to PIVOT's results and they are caused because they rely purely on image-space alignment and thus often propose poses that appear correct in 2D yet place the object off the interaction line in 3D (e.g., a flashlight "aiming" at the toy but missing it laterally). In contrast, VLM–CEM samples poses anchored at detected interaction keypoints, yielding goals that are both visually sensible and metrically sound.

Q3 — Benefits of Compositional Policy Learning

In this set of experiments, we evaluate the improvements gained from our compositional policy learning stage by comparing its performance against executing the initial **grasp candidates** generated via VLM-guided constrained optimization (adapted from Li et al. [21]) without refinement. We compare our learned policy (CoDex) against the direct execution of these initial candidates (evaluating 3 candidates per object, each repeated 5 times with random object initialization). We report comparisons against both the average performance across all candidates per object and the *oracle best performance* – that is, the success rate achieved by the single most successful candidate grasp for each object, selected post-hoc after evaluating all candidates. Fig. 3, right, depicts the results.

We observe that our compositional policy learning significantly improves grasp stability, exceeding the average performance of initial candidates by over 38% and surpassing the *oracle best performance* (the maximum potential success achievable without refinement) by over 13%. The benefit of policy learning is even more pronounced for functional actuation: the learned policy achieves 46% higher actuation success than the average candidate and crucially, over 13% higher success than the best possible outcome using only the initial candidates (oracle).

Interestingly, none of the initial candidates achieved stable grasping success (let alone actuation) for the challenging salt grinder, likely due to its slippery surface and geometry. However, CoDex's policy learning stage successfully discovers a stable grasp, although actuation remains challenging for this specific object due to the high forces required, highlighting the method's ability to improve even on difficult cases. This clearly demonstrates the critical importance of the holistic, simulation-based policy learning stage. It allows CoDex to refine statically plausible grasp candidates into dynamically robust policies that significantly enhance functional viability compared to executing the initial candidates directly, even when considering the best possible initial candidate.

5 Conclusion

We addressed the challenge of zero-demonstration functional object manipulation requiring composed grasping, movement, and actuation with dexterous hands. We introduced CoDex, a framework synergizing VLM semantic guidance with an online pipeline combining grasp candidate generation via constrained optimization and holistic policy learning via simulation-based RL. By leveraging VLM-derived local and global targets to guide this physics-grounded learning process, CoDex optimizes the complete composed action sequence for holistic task success. Our experiments demonstrated that this tight integration of semantic reasoning and physics-based policy learning is crucial, enabling CoDex to autonomously discover and execute complex FOM strategies on a physical robot for diverse tasks using unseen objects, significantly outperforming approaches that neglect holistic, physics-grounded optimization. This work represents a step towards more versatile autonomous tool manipulation. Future directions include extending the framework to more complex object mechanisms, incorporating tactile feedback, and exploring richer VLM interactions.

6 Limitations

While CoDex demonstrates success in multiple compositional functional object manipulation tasks, it presents several limitations. **First**, as observed in our experiments, some FOM tasks require ex-

tremely precise contact accuracy, such as pressing small buttons. Given the size of the robot fingers and the inaccuracies in the low-cost motors on the hand, achieving this accuracy is challenging, even if the strategy from CoDex is correct. Nimbler end-effectors and closer loop controllers would be required to overcome this limitation **Second**, the other main source of failure in our experiments stems from inaccurate material property simulation. Since CoDexreconstructs meshes from RGB images, the object material is not known and wrongly estimated. Additional efforts to either estimate the material property or a more sophisticated sim2real pipeline could alleviate this limitation in CoDex.

Third, while our method runs online, it still requires too long a time ($\tilde{1}$h, see apendix) to execute all the steps: grasp synthesis, RL training, and VLM-CEM. However, the continuous improvements in computational hardware may render this limitation negligible in a few years. **Fourth**, some functional object manipulation tasks require continuous arm and finger motions instead CoDex's approach of reaching a global goal. This includes complex tasks such as actuating scissors while sliding along a piece of paper to cut it. We plan to address this exciting challenge in the future. And **fifth**, very thin objects such as pens may require more sophisticated and dexterous hand and grasp strategy, and may be an issue for the open-loop policy in CoDex. Training a reactive closed-loop policy will overcome this problem at the cost of longer training.

References

[1] D. Paulius, Y. Huang, R. Milton, W. D. Buchanan, J. Sam, and Y. Sun. Functional object-oriented network for manipulation learning. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2655–2662. IEEE, 2016.

[2] K. Srinivasan, E. Heiden, I. Ng, J. Bohg, and A. Garg. Dexmots: Learning contact-rich dexterous manipulation in an object-centric task space with differentiable simulation. In *International Symposium on Robotics Research (ISRR)*, 2024. URL https://dexmots.github.io/.

[3] L. Huang, H. Zhang, Z. Wu, S. Christen, and J. Song. Fungrasp: Functional grasping for diverse dexterous hands. *IEEE Robotics and Automation Letters*, 2025.

[4] M. Aburub, K. Higashi, W. Wan, and K. Harada. Functional eigen-grasping using approach heatmaps. *arXiv preprint arXiv:2401.11681*, 2024.

[5] A. Agarwal, S. Uppal, K. Shaw, and D. Pathak. Dexterous functional grasping. In *Conference on Robot Learning (CoRL)*, 2023. URL https://arxiv.org/abs/2312.02975.

[6] M. Li, Z. Chen, C. Yang, and Q. Zhu. Dexterous manipulation with multi-fingered robotic hands: A review. *Frontiers in Neurorobotics*, 16:861825, 2022. doi:10.3389/fnbot.2022.861825.

[7] S. An, Z. Meng, C. Tang, Y. Zhou, T. Liu, F. Ding, S. Zhang, Y. Mu, R. Song, W. Zhang, Z.-G. Hou, and H. Zhang. Dexterous manipulation through imitation learning: A survey. *arXiv preprint arXiv:2504.03515*, 2025.

[8] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[9] A. Handa, K. Van Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. Ratliff, and D. Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170. IEEE, 2020.

[10] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024.

[11] A. Iyer, Z. Peng, Y. Dai, I. Guzey, S. Haldar, S. Chintala, and L. Pinto. Open teach: A versatile teleoperation system for robotic manipulation. *arXiv preprint arXiv:2403.07870*, 2024.

[12] V. Kumar, A. Gupta, E. Todorov, and S. Levine. Learning dexterous manipulation policies from experience and imitation. *arXiv preprint arXiv:1611.05095*, 2016.

[13] H. Charlesworth and G. Montana. Solving challenging dexterous manipulation tasks with trajectory optimisation and reinforcement learning. In *Proceedings of the 3rd Workshop on Machine Learning for Autonomous Driving, PMLR*, volume 139, 2021. URL http://proceedings.mlr.press/v139/charlesworth21a.html.

[14] A. Bahety, P. Mandikal, B. Abbatematteo, and R. Martín-Martín. Screwmimic: Bimanual imitation from human videos with screw space projection. In *Robotics: Science and Systems*, 2024.

[15] R. S. Sutton, A. G. Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[16] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[17] K. B. Shimoga. Robot grasp synthesis algorithms: A survey. *The International Journal of Robotics Research*, 15(3):230–266, 1996.

[18] A. T. Miller and P. K. Allen. Graspit!: A versatile simulator for grasp analysis. In *ASME International Mechanical Engineering Congress and Exposition*, volume 26652, pages 1251–1258. American Society of Mechanical Engineers, 2000.

[19] D. Berenson and S. S. Srinivasa. Grasp synthesis in cluttered environments for dexterous hands. In *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*, pages 189–196. IEEE, 2008.

[20] D. Morrison, P. Corke, and J. Leitner. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. *arXiv preprint arXiv:1804.05172*, 2018.

[21] A. H. Li, P. Culbertson, J. W. Burdick, and A. D. Ames. Frogger: Fast robust grasp generation via the min-weight metric. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6809–6816. IEEE, 2023.

[22] W. Huang, C. Wang, Y. Li, R. Zhang, and F.-F. Li. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2024. URL https://arxiv.org/abs/2409.01652.

[23] S. Nasiriany, F. Xia, W. Yu, T. Xiao, J. Liang, I. Dasgupta, A. Xie, D. Driess, A. Wahid, Z. Xu, Q. Vuong, T. Zhang, T.-W. E. Lee, K.-H. Lee, P. Xu, S. Kirmani, Y. Zhu, A. Zeng, K. Hausman, N. Heess, C. Finn, S. Levine, and B. Ichter. Pivot: Iterative visual prompting elicits actionable knowledge for vlms, 2024. URL https://arxiv.org/abs/2402.07872.

[24] Y. Ma, Z. Song, Y. Zhuang, J. Hao, and I. King. A survey on vision-language-action models for embodied ai. *arXiv preprint arXiv:2405.14093*, 2024.

[25] J. Wen, Y. Zhu, J. Li, Z. Tang, C. Shen, and F. Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*, 2024.

[26] J. Liu, M. Liu, Z. Wang, P. An, X. Li, K. Zhou, S. Yang, R. Zhang, Y. Guo, and S. Zhang. Robomamba: Efficient vision-language-action model for robotic reasoning and manipulation. *arXiv preprint arXiv:2406.04339*, 2024.

[27] B. Zhou, H. Yuan, Y. Fu, and Z. Lu. Learning diverse bimanual dexterous manipulation skills from human demonstrations. *arXiv preprint arXiv:2410.02477*, 2024.

[28] B. Sundaralingam, A. Lambert, C. Wang, Y. Li, F.-F. Li, and R. Zhang. Multi-finger manipulation via trajectory optimization with differentiable rolling and geometric constraints. *arXiv preprint arXiv:2408.13229*, 2024.

[29] P. Koczy, M. C. Welle, and D. Kragic. Learning dexterous in-hand manipulation with multifingered hands via visuomotor diffusion. *arXiv preprint arXiv:2503.02587*, 2025.

[30] C. Wang, R. Yang, J. Ichnowski, M. Danielczuk, Z. Xian, C. Gonzalez, R. H. Taylor, K. Goldberg, P. Abbeel, C. H. Rycroft, and Y. Ma. Kinesoft: Learning proprioceptive manipulation policies with soft robot hands. *arXiv preprint arXiv:2503.01078*, 2025.

[31] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik. Learning visuotactile skills with two multifingered hands. *arXiv preprint arXiv:2404.16823*, 2024.

[32] K. Fang, Y. Zhu, A. Garg, A. Kurenkov, V. Mehta, L. Fei-Fei, and S. Savarese. Learning task-oriented grasping for tool manipulation from simulated self-supervision. *The International Journal of Robotics Research*, 39(2-3):202–216, 2020.

[33] J. Zhang, W. Xu, Z. Yu, P. Xie, T. Tang, and C. Lu. DexTOG: Learning Task-Oriented Dexterous Grasp with Language Condition. *IEEE Robotics and Automation Letters*, 10(2):995–1002, 2025. doi:10.1109/LRA.2024.3518116.

[34] Z. Li, J. Liu, Z. Li, Z. Dong, T. Teng, Y. Ou, D. Caldwell, and F. Chen. Language-guided dexterous functional grasping by llm generated grasp functionality and synergy for humanoid manipulation. *IEEE Transactions on Automation Science and Engineering*, 22:10506–10519, 2025. doi:10.1109/TASE.2024.3524426.

[35] S. Brahmbhatt, A. Handa, J. Hays, and D. Fox. Contactgrasp: Functional multi-finger grasp synthesis from contact. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6396–6403, 2019.

[36] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4269–4276, 2021.

[37] S. Chen, J. Bohg, and C. K. Liu. Springgrasp: Synthesizing compliant, dexterous grasps under shape uncertainty. *arXiv preprint arXiv:2404.13532*, 2024.

[38] C. Ferrari and J. F. Canny. Planning optimal grasps. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 2290–2295. IEEE, 1992.

[39] L. Wang, Y. Xiang, and D. Fox. Manipulation trajectory optimization with online grasp synthesis and selection. In *Robotics: Science and Systems (RSS)*, 2020. URL `https://www.roboticsproceedings.org/rss16/p066.pdf`.

[40] T. Weng, D. Held, F. Meier, and M. Mukadam. Neural grasp distance fields for robot manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[41] D. Tochilkin, D. Pankratz, Z. Liu, Z. Huang, A. Letts, Y. Li, D. Liang, C. Laforte, V. Jampani, and Y.-P. Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024.

A Appendix

In this section, we provide further information about our VLM guidance (including prompts), grasp candidate generation, VLM-CEM method and its variants, reinforcement learning setup, and further experimental results for CoDex.

A.1 Visual Task Parsing, Activation Point and Effector Point Extraction

The initial part of CoDex's VLM guidance stage processes the raw inputs (text command and one RGB-D image) to extract geometric and semantic information necessary for downstream planning and learning.

Segmentation and Reconstruction: Given the input text command (e.g., *"Spray the plant"*) and an RGB-D image of the scene, we first identify the functional object using the VLM query:

```
What object should I use to {task}?  Express in <>.
```

We employ LangSAM (https://github.com/luca-medeiros/lang-segment-anything), an opensource implementation for open-vocabulary language-driven image segmentation that uses the VLM-identified object name to detect and segment the functional object, allowing CoDex to handle previously unseen objects, without any demonstrations. The segmented mask of the functional object is used to reconstruct a 3D mesh using TripoSR [41], an opensource partial-view shape reconstruction solution for everyday objects.

Activation & Effect Point Extraction: To identify critical local interaction points, CoDex integrates two VLMs with complementary capabilities. First, CoDex uses `GPT-o4-mini` to provide text descriptions of the concrete Activation and Effect Points for the detected functional object. GPT-o4-mini is lightweight and fast, and provides accurate text description about the relevant points for unseen functional objects. We use the following prompt to obtain the text description of the **Activation Point** ($p_{activation}$):

```
An actuation contact point is the location on the object
where a human finger interacts with.  The interaction
should lead to *internal* motion required for the task (e.g.
pressing a <pen button>, pulling a <trigger>, closing the two
<handles>).


What are the actuation contact points to TASK? Express in <>.
```

and the following prompt to obtain the text description of the **Effect Point** (p_{eff}):

```
What is the key object-object interaction point for the
<OBJECT> in the task of <TASK>?
Only describe what this point does in simple words (e.g.,
<where spray comes out>).
Do not consider humans in the response.  Express your answer
in <>.
```

These text descriptions are then grounded onto the object's image using Molmo (https://github.com/allenai/molmo), a VLM with better capabilities to detect concrete locations in images, when prompted with an accurate text description. Molmo obtains 2D pixel coordinates (u, v) corresponding the the Activation and Effect Points, which are subsequently "unprojected" onto the 3D space using the depth channel to find the points on the reconstructed 3D mesh corresponding to $p_{activation}$ and p_{eff}.

The 2D-3D correspondence requires aligning the reconstructed mesh to the RGB-D image. To that end, CoDex uses an opensource implementation of Foundation Pose (https://github.com/

Figure 4: Examples of reconstructed shapes, VLM-identified activation points (starting points of the blue arrows) and estimated actuation directions (blue arrows) on different functional objects. Better seen in color.

NVlabs/FoundationPose) to estimate the 3D pose of the mesh of the functional pose in the camera frame. Fig. 4 illustrates example outputs of the reconstruction and point detection procedure. We assume the required actuation force direction, $d_{activation}$, corresponds to the negative surface normal at $p_{activation}$. CoDex uses this VLM-generated information for grasp generation and reward shaping for parameterized policy training in subsequent steps.

A.2 Grasp Candidate Generation: Constrained Optimization Details

The initial grasp candidates that serve as initialization for CoDex's policy learning procedure are generated through a constrained optimization informed by VLM guidance. In the following, we include details about this process.

Initial Sampling for Optimization: Similar to Li et al. [21], we perform an iterative optimization procedure starting from different initial samples of the hand palm pose and finger joint configurations around the reconstructed mesh of the functional object. For sampling, we find palm and finger configurations using using inverse kinematics, imposing only that the activation finger (one of the fingers chosen randomly among the four fingers of the robot hand) starts near the 3D Activation Point, $p_{activation}$, provided by the VLM, with its fingerpad (a surface region around the finger tip) oriented approximately along the desired actuation direction, $-d_{activation}$. The result of this process are initial samples for constrained optimization, q_0.

Mathematical Formulation of the Iterative Constrained Optimization Process: CoDex's candidates for policy training result from optimizing the initial samples described above. Following Li et al. [21], we refine the initial candidates to generate optimized candidates by maximizing the min-weight force closure metric $l^*(q)$ (a proxy for grasp robustness) for a hand configuration, q, subject to physical and VLM-guided functional constraints:

$$
\begin{aligned}
\max_{q \in \mathcal{Q}} \quad & l^*(q) \\
\text{s.t.} \quad & l^*(q) \geq l_{\min} && \text{(Min Force Closure)} \\
& s(FK_i(q)) = 0, && \forall i \in \{1, ..., n_c\} \quad \text{(Surface Contact)} \\
& \sigma_j(q) \geq d_j, && \forall \text{ collision pairs } j \quad \text{(Collision Avoidance)} \\
& \exists i_{act} \in F_{act} \text{ s.t.} \\
& \quad \|FK_{tip}(q, i_{act}) - p_{activation}\|_2 \leq \delta_{dist} && \text{(Activation Pt Proximity)} \\
& \quad n_{pad}(q, i_{act}) \cdot (-d_{activation}) \geq \cos(\delta_{angle}) && \text{(Activation Alignment)}
\end{aligned}
\tag{2}
$$

where q is the hand configuration (within joint limits $\mathcal{Q}$), $l^*(q)$ is the min-weight force closure metric [21] with required minimum $l_{\min}$, $s(\cdot)$ is the object signed distance function (SDF), $FK_i(\cdot)$ is the contact point location computed using forward kinematics and a collision detector based on the SDFs of the hand and object, $\sigma_j(\cdot)$ is a detector of penetration distance at the contact points, which is allowed to be a small negative value (larger than d_j), F_{act} defines a potential activation finger that should be optimized to be generated at the Activation Point when the activation finger is closed, $p_{activation}$ and $d_{activation}$ are VLM-derived Activation Point and the Activation Direction, $FK_{tip}(\cdot)$ is fingertip location provided by forward kinematics, $n_{pad}(\cdot)$ is the fingerpad normal, and

Figure 5: Examples of diverse initial functional grasp candidates synthesized via constrained optimization. (a) Human-like grasps. (b) Non-human-like robot-specific functional grasps.

$\delta_{dist}, \delta_{angle}$ are tolerances enabling some small deviations in the location and contact actuation of the activation finger.

Starting from a sampled initial joint configuration, q_0, we solve the above constrained optimization problem using nlopt (`https://github.com/stevengj/nlopt`). Intuitively, the process maximizes the grasp robustness, while trying to (a) ensure the grasp remains always above a certain robustness threshold, (b) avoid penetrations, (c) ensure the fingers are all contacting the objects, and (d) is capable of performing the functional activation (apply force at the Activation Point, in the direction of actuation force direction). If the iterative optimization procedure fails, we resample q_0 and restart; the loop terminates at the first feasible solution.

Grasp Examples and Diversity: Fig. 5 depicts several examples of diverse grasp candidates generated by this optimization. We want to remark that, different from methods based on human grasps [], CoDex's process generates both human-like grasps (Fig. 5, *a*) and non-human-like grasps that functionally also are valid (Fig. 5, *b*). This fully exploits the capabilities of the robot hand morphology without restricting it to human grasp priors and limitations (e.g., less force in the smaller fingers).

A.3 VLM-CEM Goal Pose Generation Details

The VLM-guided Cross-Entropy Method (VLM-CEM) determines the target 6D object pose of the functional object in the entire scene to arrive before actuating it. The VLM-CEM procedure iteratively refines pose candidates using a VLM scoring of rendered scenes (Fig. 6).

Compared Variants (Ablations/Baselines from Sec. 4):

- **CoDex VLM-CEM (Ours):** CoDex first centers the functional object at the target location by applying a translation to the functional object such that the object's Effect Point, p_{eff}, and the Target Location, p_{target}, overlap in 3D space. Then, it generates candidates around this pose, T_{cand}, by applying sampled rotations, R_{cand}, and 3D translations, t_{cand}. The candidate poses are used to render new images with the moved functional object, which will be scored but the VLM to obtain sample weights. It iteratively refines sampling distributions following a CEM procedure with the elite samples of the previous iteration.
- **VLM-CEM (Directional):** Similar to CoDex's VLM-CEM full procedure, but the sampled translations, t_{cand}, are 1D along a pre-annotated object functional axis. Requires annotation.
- **PIVOT (SE3)** [23]: Our PIVOT-based baseline generates SE(3) transformations, T_{delta}, relative to the object's *current* pose, $T_{current}$, yielding goal candidates, $T_{cand} = T_{current} \circ T_{delta}$. The candidate poses are used to render new images that are scored by the VLM.

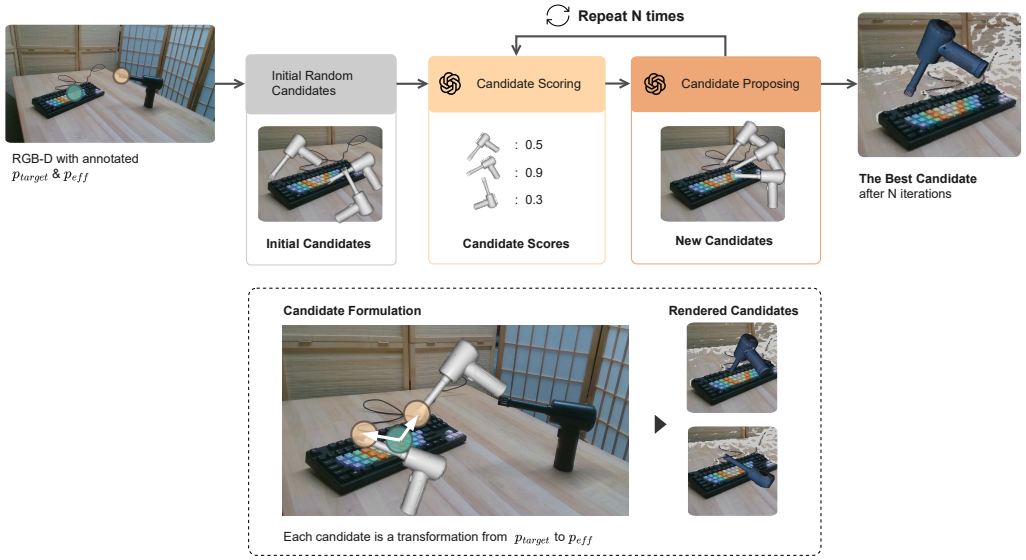

Figure 6: VLM-CEM process overview. The VLM provides initial information about the Effect Point where the functional object performs its function, p_{eff}, and the Target Location, p_{target}, where that function should be applied to achieve the manipulation task (e.g., *"blow the keyboard"*). With this information, an initial set of random candidate poses for the functional object is generated, constraining the effect point to be close to the target point. These candidates are used to render images that are scored by the VLM based on their expected utility to perform the task. The scores are used to reweight the sample distribution and resample based on the elite samples. The VLM-CEM procedure iterates N times ($N = 6$) with K ($K = 10$) candidates during each iteration to generate a good goal pose for the functional object in the compositional manipulation.

- **PIVOT (Translation)** [23]: Similar to the PIVOT baseline above but it samples only 3D translations, t_{delta}, relative to current pose, $T_{current}$. The object's orientation remains unchanged.

Fig. 7 compares output from different methods on the task *clean the keyboard*, and Fig. 8 depicts VLM-CEM (Ours) applied to other tasks in our experiments.

VLM Prompts for CEM: The iterative VLM-CEM process involves proposing new candidate poses and evaluating existing ones.

The prompt used to ask the VLM (e.g., `gpt-4o-mini`) to propose new candidates, given a history of previous attempts, is structured as follows:

```
For the task {TASK_DESCRIPTION}, we have a history of candidate
placements (parameters, scores) of {OBJECT_NAME} as below:
{CANDIDATE_HISTORY}

The accompanying images correspond to the latest candidates.

Please propose **exactly {NUM_CANDIDATES}** new candidates, each with
3-D rotation and a local offset (tx, ty, tz in meters) so that the
{OBJECT_NAME} is appropriately placed for the task.

You should try to maximize the scores, while avoiding any kind of
penetrations.  The history scores were only relative within each
batch; their descriptions may also be unreliable.  Feel free to
explore.

You have max {MAX_NUM_TRY} tries to propose new candidates.  The
current try is {TRY_INDEX}.
```

```
Return a single JSON object with keys "1"{"{NUM_CANDIDATES}". Each
value must contain:
- "yaw": <float>
- "pitch": <float>
- "roll": <float>
- "tx": <float> (metres along local +X)
- "ty": <float> (metres along local +Y)
- "tz": <float> (metres along local +Z)
- "reasoning": <short explanation>

Return **only** the JSON object.
```

The prompt used to ask the VLM to evaluate (score) rendered images of candidate poses is:

```
We want to evaluate how well each candidate image positions the
{OBJECT_NAME} for the task '{TASK_DESCRIPTION}'.
For each candidate image (numbered at the lower right corner),
please assess the {OBJECT_NAME}'s placement for how well it meets
the task requirements.

Scoring:
• Provide a numeric score between 0 and 1, where 1 means \perfect
alignment and angled," and 0 means \completely unsuitable."
• You can use intermediate values (e.g., 0.22, 0.5, 0.85) if the
placement is partially correct.
• Be strict with scoring: think about the physical feasibility of
each candidate (avoid penetrating the environment) and avoid being
overoptimistic.
• If you do not see the whole/partial object, it is probably due to
either occlusion or penetration (for example, object placed beneath
the table surface). Or the object is completely out of the camera
view (which is undesired).

*Having no penetration* (with tables/other objects) is extremely
important for the candidate feasibility.

Below we provide the table penetration of each candidate (0 means no
penetration). Note this is only penetration with the *table*, not
with the other objects. For penetration with other objects, please
refer to the images.
Penetration Info (meters):
{PENETRATION_TABLE}

Format Requirements:
1. Return your answer as a single, valid JSON object and
nothing else. Do not include extra text, commentary, or markdown
formatting.
2. The JSON object should have candidate IDs (as strings) as keys.
For each key, the value should be another JSON object containing
exactly two fields: "score" and "description".
3. For example, if there are 3 candidates, a valid response should
look like:
{
  "1": {"score": 0.73, "description": "The object is mostly above
the target and slightly tilted."},
  "2": {"score": 0.02, "description": "Thermos is upright and
nowhere near the mug."},
  "3": {"score": 1.0, "description": "Perfect angle and spout is
centered over the mug."}
}
```

Figure 7: Example visualizations of different goal-pose-generation methods on the task *clearn the keyboard*. The first row is the top-down view, and the second row is the wrist-camera view. Both variants of VLM-CEM generate **both semantically and physically valid goal poses** zero-shot, while the baseline methods perform poorly on the task.

Figure 8: Example visualizations of VLM-CEM's generated goals on more tasks. The first row is the top-down view, and the second row is the wrist-camera view. VLM-CEM generates high-quality goal poses across diverse unseen tasks.

```
Return only the JSON object, with no extra formatting or commentary.
```

Note: {TASK_DESCRIPTION}, {OBJECT_NAME}, {CANDIDATE_HISTORY} etc., are placeholders. They are filled programmatically.

A.4 Reinforcement Learning Setup

This section details the holistic policy learning stage using RL.

Policy Input and Output: The learned policy π takes the selected initial grasp candidate as input observation (relative hand pose 6D, candidate finger joints 16D, activation finger index 1D). The policy outputs an action $a \in \mathbb{R}^{38}$ that parameterizes the motion primitive targets relative to the input candidate $q_{cand} = (p_{palm,cand}, j_{fingers,cand})$. The action vector defines:

- Target grasp joint offsets $\Delta j_{grasp} \in \mathbb{R}^{16}$ (added to $j_{fingers,cand}$, scaled by 0.5).
- Target pre-grasp joint percentages $p_{pregrasp} \in \mathbb{R}^{16}$ ($j_{pregrasp} = j_{fingers,cand} \times p_{pregrasp}$).
- Residual hand pose offset $\Delta T_{hand} \in \mathbb{R}^{6}$ (applied to $p_{palm,cand}$, with internal scaling).

Parameterized Primitive: We use a grasp-move-actuate primitive (Fig. 9). The policy action a determines target pre-grasp/grasp states relative to q_{cand}. The primitive executes: (1) Teleport to pre-pre-contact pose. (2) Move to target pre-grasp state. (3) Move to target grasp pose. (4) Close fingers to target grasp joints. (5) Lift. (6) Move towards VLM-CEM global goal (if specified). (7) Attempt activation.

Reward Function: The reward function R guides the holistic policy learning. It is a normalized weighted sum ($R = (\sum w_k R_k + \sum w'_k S_k + w'_{success} S_{success})/15.0$) of shaped rewards ($R_k$) and binary success flags (S_k), plus a bonus for holistic success ($S_{success}$). A penalty resets reward to 0 if significant grasp instability occurs ($\Delta dist > 0.15$). Key components are described in Table 2.

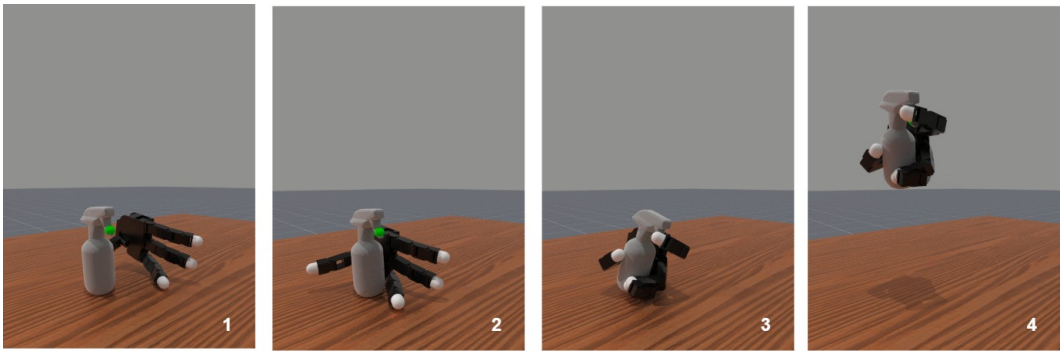

Figure 9: Visualization of the parameterized motion's key stages. (1) The palm is moved to the precontact pose, with the fingers set to precontact joint configuration. (2) The palm is moved to the grasp pose, with the fingers unchanged. (3) The fingers are set to grasp joint configuration. (4) The object is moved & activated.

Table 2: Reward Function: Key Weights and Thresholds.

Component	Parameter / Description	Value
Stability Flag (S_{stable})	Threshold ($\Delta dist_{max}$)	0.04
	Flag Weight (w'_{stable})	1.0
Lift Flag (S_{lift})	Threshold ($height_{min}$, meters)	0.1
	Flag Weight (w'_{lift})	1.0
Uprightness ($R_{upright}$)	Shaped Reward Weight ($w_{upright}$)	1.0
Align Flag (S_{align})	Dist Threshold (δ_{align_dist})	0.03
	Dir Threshold (δ_{align_dir})	0.01
	Flag Weight (w'_{align})	1.0
Align Reward (R_{align})	Shaped Reward Weight (w_{align})	2.0
Force Reward (R_{force})	Shaped Reward Weight (w_{force})	1.0
Force Flag (S_{force})	Threshold (F_{thresh}, N?)	0.6
Collision Flag ($S_{collision}$)	Flag Weight ($w'_{collision}$)	1.0
Holistic Bonus ($S_{success}$)	Bonus Weight ($w'_{success}$)	5.0
Normalization	Denominator (Norm)	15.0
Instability Penalty	Reset Threshold ($\Delta dist_{max}$)	0.15

RL Algorithm Details: We use the Proximal Policy Optimization implementation from Stable Baselines3 (https://github.com/DLR-RM/stable-baselines3) for RL training. Hyperparameters are listed in Table 3. Online training is performed in ManiSkill3 (https://github.com/haosulab/ManiSkill) and converges within approximately one hour on a single NVIDIA RTX 4090 GPU. We expect it could be greatly paralellized with multiple GPUs, bringing this training time down to a few minutes.

Table 3: PPO Hyperparameters for Online Policy Learning.

Parameter	Value
Algorithm	PPO
Policy Network	MlpPolicy
Total Timesteps	200,000
Discount factor (γ)	0.8
GAE Lambda (λ)	0.9
Entropy Coeff (ent_coef)	0.01
Steps per Env per Update (n_steps)	1
Batch Size	2048
Number of Epochs (n_epochs)	8
Parallel Environments (N_{env})	2048
Learning Rate	3e-4

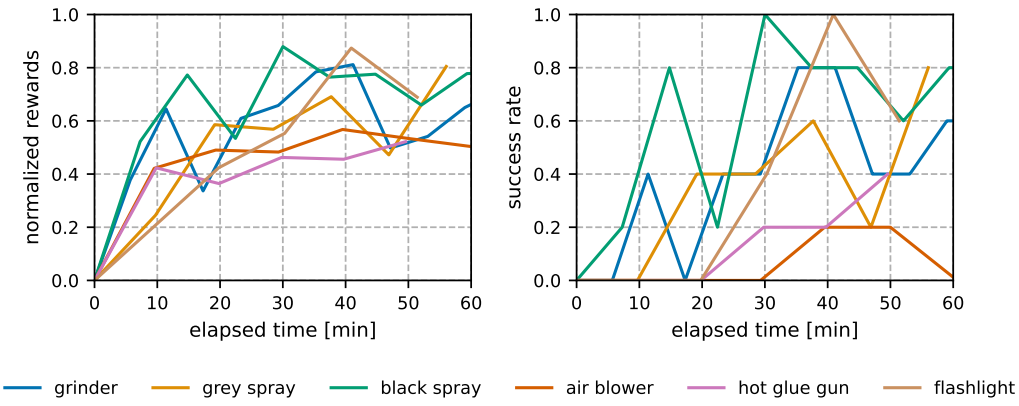

Figure 10: Training Curves of CoDex's Policy Training with Reinforcement Learning over wall time. *Left* is the eval normalized rewards; *right* is the eval success rate. CoDex "bootstraps" the policy with the optimized candidate grasps, greatly reducing the training difficulties–most of the trainings on the above dexterous tasks converge within **60 minutes**. Note that we perform domain randomization and use a strict success criteria, causing the eval success rates to be potentially lower than the checkpoints' actual success rates in real world.

A.5 Extended Experimental Results

Detailed Quantitative Experimental Results (Q2): Table 4 presents the numerical data for the average human ratings of VLM-generated goal poses (corresponding to Fig. 3, *left*), comparing the different VLM-CEM and PIVOT variants evaluated in Q2. Ratings were on a 1-5 scale (5=best). Method abbreviations: V-CEM (Ours), V-CEM(D) (Directional), P(SE3) (PIVOT SE3), P(T) (PIVOT Translation).

Table 4: Average Human Ratings (1-5 scale) for VLM-Generated Goal Poses (Q2).

Task	V-CEM (Ours)	V-CEM(D)	P(SE3)	P(T)
Illuminate the sloth	2.2	3.4	1.0	1.3
Spray at the whiteboard	3.1	2.8	2.1	2.8
Spray the plant	3.0	4.8	2.4	2.1
Clean the keyboard	4.0	4.6	1.7	1.4
Glue the wood blocks	4.5	3.6	1.0	1.0
Add salt to the beans	4.2	4.6	1.4	2.0
Average	**3.5**	**4.0**	**1.6**	**1.8**

Detailed Quantitative Experimental Results (Q3): Table 5 presents the numerical data for the policy learning evaluation (corresponding to Fig. 3, *right*), comparing the performance of CoDex against the initial grasp candidate baselines (average and oracle best) on the grasp-lift-activate task. Results show successes out of five trials per object. The final row shows the average success counts across all six objects.

Qualitative Real-World Rollouts: Figure 11 displays image sequences from successful real-world executions by CoDex.

Table 5: Quantitative results for policy learning evaluation (Q3, Fig. 3 Right). Format: Stable Grasp / Activation Success (Successes / 5 Trials).

Object	CoDex (Ours)	Initial Grasp (Best)	Initial Grasp (Average)
Grey Spray	5 / 5	5 / 5	4 / 2
Hot Glue	5 / 5	5 / 4	5 / 1
Air Blower	5 / 4	5 / 3	3 / 1
Black Spray	5 / 5	5 / 3	3 / 1
Flashlight	5 / 0	5 / 0	3 / 0
Salt Grinder	4 / 0	0 / 0	0 / 0
Average	**4.8 / 3.2**	**4.2 / 2.5**	**3.0 / 0.8**

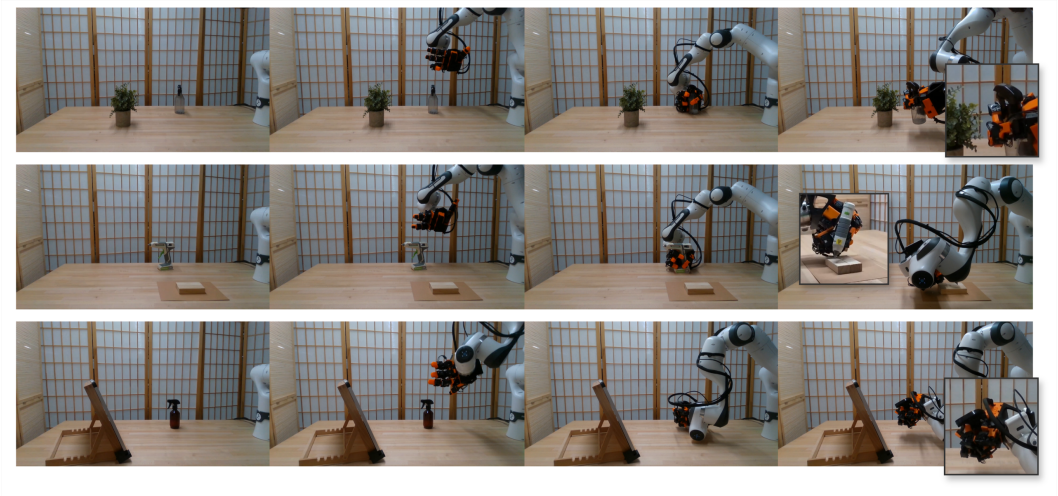

Figure 11: Example real-world execution rollouts for CoDex. Each row shows a different task (e.g., top: Spray Plant). Columns depict stages: (1) Initial scene setup, (2) Approaching/Pre-contact phase, (3) Object grasped, (4) Object at target pose during or after activation (inset shows close-up of hand-object interaction). For more rollouts/videos, please see https://codex-2025.github.io/.