MANIPULATION

Learning a thousand tasks in a day

Kamil Dreczkowski*†, Pietro Vitiello*†, Vitalis Vosylius, Edward Johns

Humans are remarkably efficient at learning tasks from demonstrations, but today's imitation learning methods for robot manipulation often require hundreds or thousands of demonstrations per task. We investigated two fundamental priors for improving learning efficiency: decomposing manipulation trajectories into sequential alignment and interaction phases and retrieval-based generalization. Through 3450 real-world rollouts, we systematically studied this decomposition. We compared different design choices for the alignment and interaction phases and examined generalization and scaling trends relative to today's dominant paradigm of behavioral cloning with a single-phase monolithic policy. In the few-demonstrations-per-task regime (<10 demonstrations), decomposition achieved an order of magnitude of improvement in data efficiency over single-phase learning, with retrieval consistently outperforming behavioral cloning for both alignment and interaction. Building on these insights, we developed Multi-Task Trajectory Transfer (MT3), an imitation learning method based on decomposition and retrieval. MT3 learns everyday manipulation tasks from as little as a single demonstration each while also generalizing to previously unseen object instances. This efficiency enabled us to teach a robot 1000 distinct everyday tasks in under 24 hours of human demonstrator time. Through 2200 additional real-world rollouts, we reveal MT3's capabilities and limitations across different task families.

INTRODUCTION

Humans and animals demonstrate remarkable learning efficiency through imitation. Infants learn manipulation skills substantially faster when guided by expert demonstrations (*1*, *2*), primates learn manipulation tasks from fewer than five demonstrations (*3*–*7*), and rodents acquire both behavior and navigation skills from fewer than 10 demonstrations (*8*). In stark contrast, robots lag far behind their biological counterparts, often requiring hundreds or thousands of demonstrations per task (*9*–*18*).

State-of-the-art imitation learning systems using behavioral cloning (BC) exemplify this inefficiency. Behavior cloning with zero-shot task generalization (BC-Z) required ~26,000 demonstrations for 100 tasks (*9*), Robotics Transformer 1 (RT-1) needed ~130,000 demonstrations across 744 tasks (*12*), and multitask action-chunking transformer (MT-ACT) collected 7500 demonstrations for 38 tasks (*15*)—all averaging 175 to 250 demonstrations per task. For complex bimanual manipulation, ALOHA (A Low-Cost Open-Source Hardware System for Bimanual Teleoperation) Unleashed (*19*) suggests the need for ~8000 demonstrations per task.

Although these methods can be effective at scale, scaling to thousands of tasks would require massive real-world datasets demanding enormous financial and human resources. Improving learning efficiency is thus crucial for reducing the data requirements for highly capable and general robotic systems. To this end, we studied two priors for more data-efficient imitation learning: trajectory decomposition and retrieval-based generalization.

The first prior decomposes manipulation trajectories into alignment and interaction phases. In contrast with standard BC approaches that learn manipulation with a single monolithic policy, decomposition-based methods deploy two independent policies sequentially. First, an alignment policy positions the robot's end effector, or a grasped object, relative to the target object. Second, an interaction policy performs the object manipulation. For alignment, past research explored using pose estimation (*20*, *21*) and visual servoing (*22*–*28*). For interaction, prior work primarily focused on reinforcement learning (*20*, *26*) and open-loop replay (*21*–*25*, *27*, *28*). Unlike past work that typically focused on single-task learning with one specific alignment and interaction policy combination, we systematically compared four different combinations (pose estimation versus BC for alignment; open-loop replay versus BC for interaction) when learning multiple tasks. The second prior uses retrieval-based generalization as an alternative to BC. Recent retrieval methods for manipulation include flow-guided data retrieval for few-shot imitation learning (FlowRetrieval) (*29*), SAILOR (skill-augmented imitation learning with prior retrieval) (*30*), and behavior retrieval (*31*), which leverage optical flow matching, latent skill spaces, and task-specific querying, respectively. Whereas these approaches primarily retrieved data before policy training, our methods retrieved demonstrations at test time.

After retrieving the most appropriate demonstration using language- and geometry-based matching, we performed alignment through pose estimation (*20*, *21*) and executed interaction via open-loop replay (*21*–*25*, *27*, *28*). Unlike VINN (visual imitation through nearest neighbors) (*32*), which used red, green, blue (RGB)–based retrieval to identify and obtain actions from image-action pairs throughout task execution, we used language and geometry to retrieve a complete trajectory before task execution. Through 3450 real-world rollouts across 70 objects, we systematically analyzed the effects of trajectory decomposition and retrieval-based generalization on learning efficiency. Although existing research typically focuses on learning with abundant per-task demonstrations, we focused on the more practical scenario where demonstrations were limited—a critical gap in the current literature. We studied all four combinations of BC and retrieval-based policies when used for alignment and interaction and compared them against a standard monolithic BC method that learned entire manipulation trajectories without decomposition.

Our results reveal key insights into the relationship between the number of demonstrations per task, the number of tasks and object instances being learned, and the task performance. In the

Robot Learning Lab at Imperial College London, London SW7 2AZ, UK.
*Corresponding author. Email: kamil-dreczkowski@outlook.com (K.D.); pv2017@ic.ac.uk (P.V.)
†These authors contributed equally to this work.

very-low-data-per-task regime (<10 demonstrations per task), decomposition yielded an order of magnitude of improvement in data efficiency compared with learning trajectories with a single monolithic policy. Furthermore, retrieval-based methods proved more effective for generalization, consistently outperforming BC alternatives across both alignment and interaction phases. However, as demonstrations became more abundant or task diversity increased (distributing a fixed demonstration budget across more tasks), monolithic BC exhibited better scaling trends.

Building on these insights, we developed Multi-Task Trajectory Transfer (MT3), a fully retrieval-based decomposition method that leverages retrieval for both alignment and interaction. Although MT3 demonstrated particularly strong performances in our controlled experiments, a key question remains: Is MT3 a viable approach for learning a very large number of tasks from minimal per-task data? To answer this question, we conducted a large-scale evaluation in terms of task and object diversity: teaching a robot 1000 distinct everyday tasks—involving interactions with more than 400 objects—from single demonstrations in less than 24 hours (Fig. 1 and movie S1). Through 2200 experimental rollouts, we found that MT3 is capable of scaling to very large numbers of tasks and gained insights into its limitations and failure modes.

In summary, our work makes three key contributions. First, we provide a systematic evaluation of multitask imitation learning in the few-demonstration-per-task regime, addressing a critical gap in the current literature. Second, we introduce MT3 and demonstrate that retrieval-based decomposition methods offer an attractive alternative to monolithic BC when demonstration data are limited. Third, we validate these findings at scale by learning 1000 distinct manipulation tasks from a single demonstration each, challenging the assumption that complex neural policies are necessary for large-scale robot learning while gaining insights into MT3's limitations and failure modes.

RESULTS

Experimental setup

We focused on teaching a robot multiple tasks, where each task involved a single interaction between the robot's end effector or a grasped object and a target object. For tasks involving grasped objects, we assumed that their pose in the gripper was the same during demonstrations and testing. This formulation covered most common manipulation tasks—from grasping to insertion to tool usage. Although we focused on single-interaction tasks, multistep behaviors such as pick-and-place operations could be achieved by chaining such tasks together using existing high-level planners (movies S2 to S7).

Our evaluation considered both seen tasks and unseen tasks where methods had to generalize to previously unseen object instances. For clarity, we defined three terms. A macro skill is a broad manipulation primitive defined by its core interaction type, for example, "open," "insert," or "fold." A micro skill is a macro skill specialized for a specific object category that required a distinct motion profile. Different motion profiles for the same object category constituted different micro skills, including "open oven door sideways" versus "open oven door downward." A task is a micro skill executed on a specific object instance, such as "unzip the round pink handbag." Our experimental hardware consisted of a Sawyer robot equipped with a 2F-85 Robotiq gripper (Fig. 2). For perception, we used a single RealSense D415 RGB-D (where D is depth) camera mounted on the robot's head. To ensure a fair comparison, we established a consistent system architecture across all methods. The robot received two inputs: a segmented point cloud of the target object and a language description of the task. A multitask policy processed these inputs to generate robot actions. In terms of policy design, we compared four decomposition-based methods against a monolithic BC baseline that learned entire trajectories.

Policy designs

Decomposition-based methods

Decomposition-based methods divide manipulation trajectories into two phases (Fig. 2A). The alignment phase involved moving the robot's end effector to a pose suitable for the subsequent manipulation, where only the final positioning matters, not the specific path taken. For example, positioning a plug in front of a socket can be achieved through many different trajectories. The interaction phase involved the actual manipulation, requiring precise trajectory execution. For example, the plug insertion motion must be carefully controlled to ensure proper connection.

All four decomposition-based methods used two specialized policies: one for alignment and one for interaction. We showed that this specialization led to efficiency gains compared with using a single policy when learning from limited per-task demonstrations. For each phase, we investigated two alternative approaches: BC and retrieval-based methods. BC is a prominent robot manipulation approach that trains neural networks to encode demonstrated behaviors into their weights. We therefore explored applying this technique within the decomposition framework. During training, BC used all demonstrations to learn a representation that enabled generalization across spatial configurations and object instances based on pose and geometric similarity.

For our BC implementation, we used a transformer-based backbone that used variational inference (*33*, *34*), because this architecture has demonstrated efficient learning of various manipulation tasks (*15*). Specifically, we adapted the MT-ACT architecture (*15*) to handle point cloud inputs and language descriptions. By training this architecture separately on alignment and interaction demonstrations, we obtained specialized BC policies for each phase ("BC implementation" section in Materials and Methods).

An alternative to BC is retrieval-based methods, which fundamentally differ from the former by using demonstrations at test time rather than during training. These policies were designed to replicate the behavior provided to them as context. Therefore, they stored all demonstrations in memory and, at inference, retrieved the most relevant one to use as guidance.

Both alignment and interaction retrieval-based policies share a common retrieval step that occurs before task execution. This identifies a demonstration of the same manipulation on an object with an appearance and pose similar to those of the test object (Fig. 2B). In our implementation, retrieval leveraged language processing of task descriptions combined with geometry similarity in a learned latent space. The geometry was extracted from an RGB-D image of the scene before execution. Therefore, when retrieval was used for both alignment and interaction, retrieval was performed once.

After retrieval, the alignment retrieval-based policy used pose estimation to map the demonstrated alignment pose to the test scene and reached this pose using motion planning (*21*). The interaction retrieval-based policy executed the retrieved trajectory by

Fig. 1. Learning a thousand tasks in a day. (**A**) Illustration of 1000 tasks taught in less than a day. The arrow represents the passing of time, whereas each image is a frame from a real-world rollout of one of the tasks. (**B**) Illustration of some information regarding the 1000-task dataset. We provide examples of some objects used and some of the skills evaluated.

replaying demonstrated end-effector velocities in the end-effector frame (*21*–*25*, *27*, *28*). See the "Retrieval-based alignment and interaction" section in Materials and Methods for more details.

For generalization, retrieval-based policies identified the closest demonstration object and proceeded as though the previously unseen object was identical to the training instance. For alignment, they positioned the robot relative to the previously unseen object as they would for the training one, whereas for interaction, they executed the precise demonstrated trajectory. This approach was effective because optimal trajectories often maintain similar structures across object instances within a category, with task tolerance accommodating geometric variations. For example, when grasping different mugs, although sizes and handle shapes vary, the core grasp motion remains consistent.

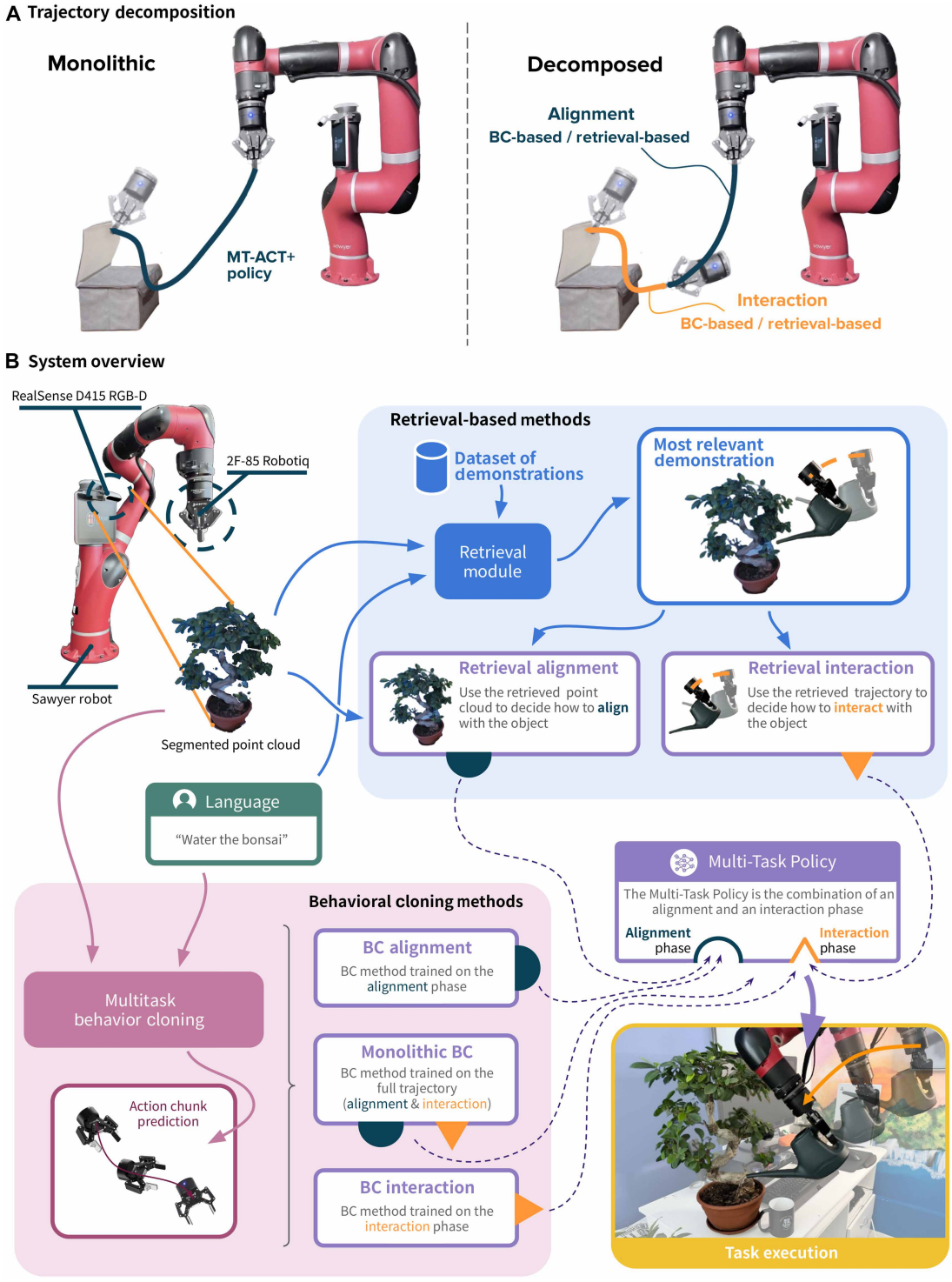


Fig. 2. Trajectory decomposition and overview of policy designs. (**A**) Trajectories are decomposed into alignment and interaction phases. Monolithic approaches use a single policy for entire trajectories. Decomposition-based approaches use two specialized policies: one for end-effector alignment with target objects and another for precise manipulations. We explored both BC and retrieval-based methods for each phase of this decomposition. (**B**) A multitask policy (purple) processes a segmented point cloud and task description as input and outputs robot actions. This can either be a monolithic policy or the combination of alignment and interaction policies. Retrieval-based policies (blue) use a retrieved demonstration as context to guide execution. BC policies (pink) directly predict actions through a neural network.

Combining BC and retrieval-based policies across both phases (alignment and interaction) created four distinct methods (Fig. 2B). BC-BC used BC for both alignment and interaction. BC-Ret combined BC alignment with retrieval-based interaction. Ret-BC used retrieval-based alignment followed by BC interaction. Last, Ret-Ret used retrieval-based policies for both alignment and interaction. Throughout this paper, we refer to Ret-Ret as MT3, because it can be seen as an extension of trajectory transfer (*21*, *35*) to the multitask learning setting.

Monolithic BC

To evaluate the benefits of decomposition, we compared all four methods against a monolithic BC baseline (MT-ACT+) that uses the same BC implementation as the alignment and interaction BC policies in BC-BC, BC-Ret, and Ret-BC. Instead of training separate policies for alignment and interaction, MT-ACT+ consists of a single policy trained to handle entire manipulation trajectories ("BC implementation" section in Materials and Methods).

Controlled experiments

To evaluate how performance scales with different data regimes, we designed two complementary experiments that independently varied dataset size (number of demonstrations per task) and diversity (number of tasks). In the first experiment, we studied how methods scale with more demonstrations on a fixed task set. We selected four diverse micro skills spanning articulated object manipulation, deformable object interaction, scooping, and insertion. For each micro skill, we included three seen and two unseen tasks, totaling 12 seen and 8 unseen tasks (Fig. 3A). We evaluated all methods by scaling from 1 to 50 demonstrations per task, where 50 demonstrations were shown to be sufficient for learning complex manipulation trajectories (*14*).

In the second experiment, we fixed the total demonstrations at 150 and studied performance as they were distributed across more tasks. This experiment assessed whether performance degrades with fewer demonstrations per task or whether methods could benefit

Fig. 3. Micro skills and objects considered in the scaling experiments. (**A**) Micro skills used to evaluate the methods' response to scaling the demonstrations per task. We also show the various seen and unseen objects used. (**B**) Micro skills used to evaluate the methods' response to scaling the number of tasks. These are in addition to those found in (A). (**C**) Objects used in the latter experiment.

from more object instances despite fewer demonstrations per task. We selected 10 micro skills (Fig. 3, A and B) and scaled from 10 tasks (15 demonstrations each) to 30 tasks (5 demonstrations each) and, lastly, 50 tasks (3 demonstrations each). For consistency, each diversity regime included two unseen tasks per micro skill (20 total). All objects are shown in Fig. 3C.

During both experiments, we conducted three evaluations per task and averaged the results across all micro skills. For each evaluation, we randomized the object's position within the 80 cm–by–45 cm task space and orientation within ±180° of the demonstration pose around the vertical axis.

Performance overview

Figure 4A shows the results from our dataset size and diversity experiments, revealing a performance hierarchy. MT3, the fully retrieval-based method, consistently demonstrated superior performance across all considered data regimes. This is particularly evident in the finding that for both seen and unseen tasks, MT3 with just three demonstrations per task outperformed all other methods, even when they were provided with 50 demonstrations per task. Moreover, the strong performance of MT3 on unseen tasks demonstrated that despite its simplicity, retrieval is a viable approach for tackling generalization to unseen object instances.

Decomposition also showed benefits, given that decomposition-based methods (Ret-BC, BC-Ret, and BC-BC) generally outperformed the monolithic baseline MT-ACT+. Notably, this benefit held even when the underlying method remained unchanged, given that BC-BC outperformed MT-ACT+ despite both using identical BC implementations. Figure 4B demonstrates the advantage of the decomposition prior over a single policy by comparing the average performance of all four decomposition-based methods [Ret-Ret (MT3), Ret-BC, BC-Ret, and BC-BC] against MT-ACT+. Decomposition-based methods consistently outperformed the monolithic baseline across all data regimes tested, with the largest performance gap observed when demonstrations per task were most limited.

Last, within the decomposition framework, Fig. 4C demonstrates that methods that used retrieval for alignment achieved higher average success rates than those that used BC (when averaged across interaction methods). Similarly, methods that used retrieval for interaction

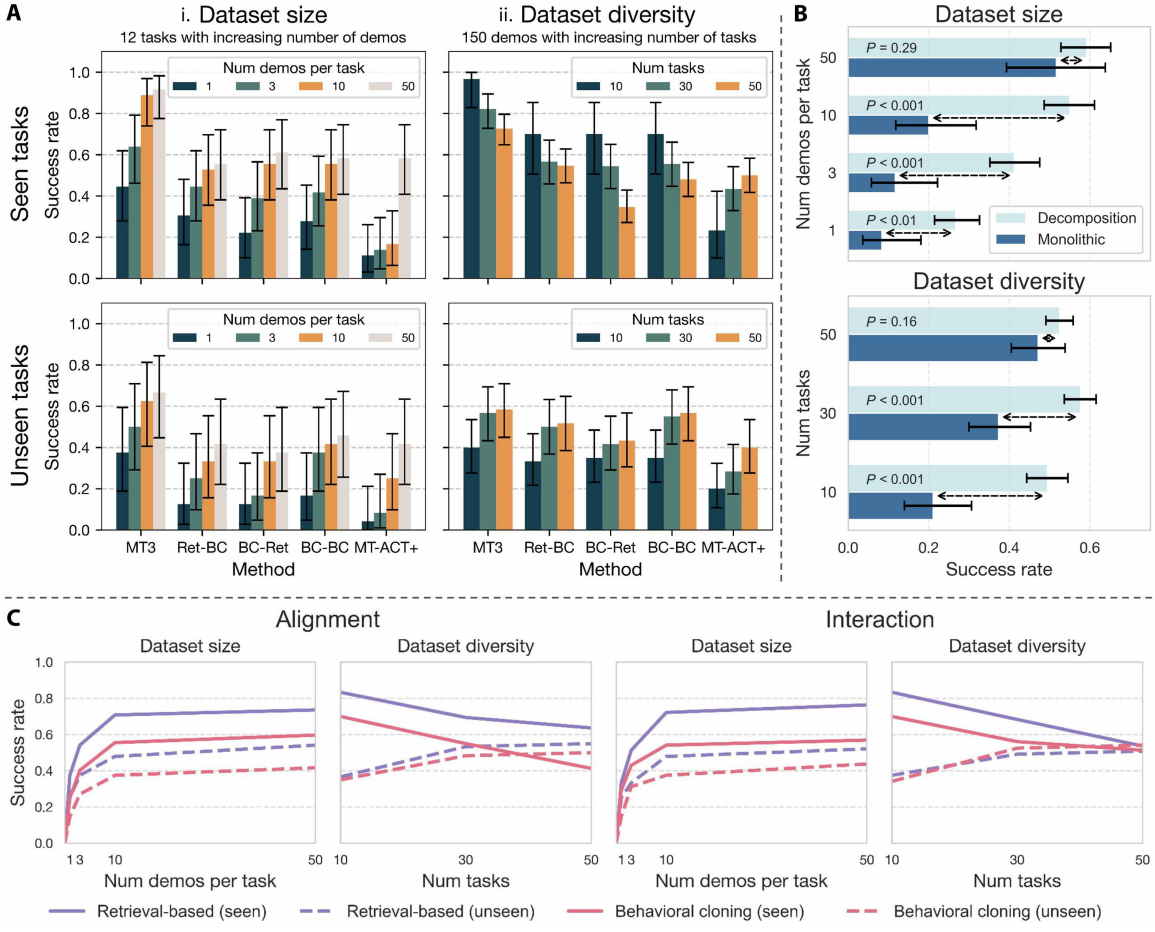


Fig. 4. Analysis of dataset size and diversity effects on task performance. (**A**) Performance comparison across all considered methods, with error bars showing 95% Wilson confidence intervals. For seen and unseen task sets, sample sizes were $n = 36$ and $n = 24$, respectively. (**B**) Comparison between decomposition-based approaches [aggregated results from Ret-Ret (MT3), Ret-BC, BC-Ret, and BC-BC] and monolithic learning (MT-ACT+), averaged across seen and unseen tasks, with error bars showing 95% Wilson confidence intervals. Sample sizes for each comparison are detailed in the "Statistical analysis" section in Materials and Methods. Statistical significance was assessed using the two-proportion Z test. (**C**) Analysis of alignment and interaction strategies: Alignment plots compare BC (BC-BC and BC-Ret) versus retrieval [Ret-BC and Ret-Ret (MT3)] for alignment, whereas interaction plots compare BC (BC-BC and Ret-BC) versus retrieval [BC-Ret and Ret-Ret (MT3)] for interaction. Success rates are shown as a function of dataset size (number of demonstrations per task) and diversity (number of tasks).

outperformed those that used BC for interaction (when averaged across alignment approaches).

Scaling dataset size

When increasing demonstrations per task (Fig. 4A.i), all methods showed improved performance on seen and unseen tasks. This improvement stemmed from different mechanisms for each approach: Retrieval-based methods benefited from more demonstrations to select from, increasing the likelihood of finding one well suited to the test instance and configuration. Conversely, BC methods could learn better representations and improve spatial generalization because of greater coverage of manipulation scenarios across demonstrations.

On the basis of the observed trends, we would expect MT-ACT+ to eventually outperform decomposition-based methods given sufficient data, although the exact crossover point would depend on the number of tasks and their similarity. Figure 4B reveals why this overtake is likely: Decomposition-based methods achieved rapid early gains with 1 to 10 demonstrations per task but plateaued near 50, whereas the monolithic baseline accelerated in the 10-to-50 range, narrowing the performance gap. We hypothesized that this occurred because decomposition-based methods leverage the built-in task structure, enabling strong performance with limited data but constraining their learning capacity. In contrast, the monolithic approach must learn the task structure from scratch—requiring more data initially—but lacks structural constraints, allowing it to continue improving and potentially surpass decomposition-based methods with abundant data.

Scaling dataset diversity

When we distributed a fixed demonstration budget across an increasing number of tasks (Fig. 4A.ii), we effectively added more object instances per micro skill. This diversity scaling revealed contrasting effects on retrieval-based methods: Unseen task performance improved as retrieval accessed more object instances, yielding closer matches to test objects, whereas seen task performance paradoxically degraded because of an inherent trade-off in the retrieval process.

This trade-off emerged because geometry-based retrieval had to balance two competing objectives when multiple object instances exist per micro skill: selecting demonstrations with a similar object pose [for successful trajectory transfer (*21*)] versus similar object geometry (for trajectory suitability for interacting with the test object). Our retrieval system considered both factors simultaneously but must make trade-offs. It may select a demonstration with better pose similarity on a different object instance, producing a trajectory less suited for the test instance. Alternatively, it may select a demonstration on the same object instance with a very different pose, making transfer harder because of large pose differences. This fundamental tension between trajectory transferability and suitability became more pronounced as object instances per micro skill increased.

The geometry-pose trade-off described above specifically affected scaling within micro skills (adding more object instances). However, retrieval-based approaches handled a different type of scaling effectively. When comparing across the two experimental paradigms while fixing demonstrations per task (light green lines in Fig. 4A.i versus orange lines in Fig. 4A.ii), retrieval-based methods showed either increased or stable performance despite increasing tasks from 12 to 50. This occurred because the first experiment considered four micro skills, whereas the second considered 10. The hierarchical retrieval design could explain this resilience: Initial language-based filtering isolates demonstrations for the target micro skill, preventing interference when adding previously unknown macro and micro skills.

In contrast, MT-ACT+ benefited from task diversity for both seen and unseen performances, learning more general latent representations by identifying patterns across different object instances. This benefit was substantial: Learning 50 tasks across 10 micro skills with 150 demonstrations (Fig. 4A.ii) achieved a performance comparable to learning 12 tasks across four micro skills with 600 demonstrations (Fig. 4A.i), effectively trading diversity for data efficiency.

However, decomposition prevented this beneficial representation sharing—BC-BC did not show the same diversity benefits as MT-ACT+ despite using identical BC implementations. We hypothesized that this occurred because effective BC learning requires diverse data with shared structural patterns. Monolithic BC learned from complete trajectories that naturally combined consistent alignment patterns with diverse interaction patterns, providing the optimal balance for representation learning. Decomposition disrupted this by isolating alignment learning on synthetic linear trajectories and interaction learning on sparse but diverse real demonstrations ("BC implementation" section in Materials and Methods), preventing the synergistic learning that monolithic approaches achieved. A further discussion of the effect of decomposition on the Pareto front of performance relative to dataset diversity and learning efficiency can be found in the Supplementary Materials.

Evaluating MT3 capabilities and failure modes across 1000 tasks

Although numerous studies have explored scaling monolithic BC across both tasks and demonstrations using large numbers of demonstrations per task (*12*, *13*, *15*, *19*), far less attention has been paid to scaling BC alternatives, especially in the minimal-demonstration-per-task regime. Our controlled experiments demonstrated that MT3 was highly effective when per-task demonstrations were limited, but a crucial question remained: What are the practical boundaries of retrieval-based decomposition methods when applied to diverse real-world manipulation tasks at scale?

To investigate this, we conducted a large-scale evaluation in which we taught a robot 1000 distinct manipulation tasks from a single demonstration each. This experiment served three analytical purposes. First, we identified the specific task characteristics where MT3 excels versus where it struggles. Second, we sought to understand the fundamental limitations of open-loop replay for interaction. Third, we aimed to characterize when global geometry–based retrieval and cross-instance pose estimation succeed versus fail for category-level generalization.

Performance overview

Our evaluation spanned 31 macro skills and 534 micro skills (detailed in table S1), involving 402 different objects. We deliberately included tasks spanning a spectrum of complexity: from tasks well suited to open-loop execution, including stacking, to those potentially requiring closed-loop control, such as manipulating deformable objects, and from tasks where success depends on global geometry alignment, for example, placing mugs on plates, to those requiring precise alignment of small geometric features, including inserting a coin into a slot of a piggy bank. To evaluate generalization, we tested an additional 100 unseen tasks spanning the same macro skills (detailed in table S2). All demonstrations were collected on a single robot over 17 hours.

Our evaluation consisted of 2200 total rollouts (two trials per task for both seen and unseen categories—see Fig. 5 for example rollouts) across challenging real-world conditions. These conditions included 5 to 20 distractor objects, varied lighting, and randomized placement with up to ±45° rotation—all designed to stress test the effectiveness of MT3. We compiled these real-world rollouts into a dataset and made them open source for the community.

Figure 6A shows MT3's performance across different macro skills, with the numbers below each macro skill name indicating the count of seen and unseen tasks evaluated. Because of practical constraints, the distribution of tasks across macro skills is not uniform, with some macro skills having smaller task counts, leading to more variability between seen and unseen task performance. MT3 achieved a 78.25% average success rate on seen tasks and 68% on unseen tasks. Beyond examining the aggregate averages, we analyzed the more meaningful patterns that emerged when comparing success rates in relation to task characteristics.

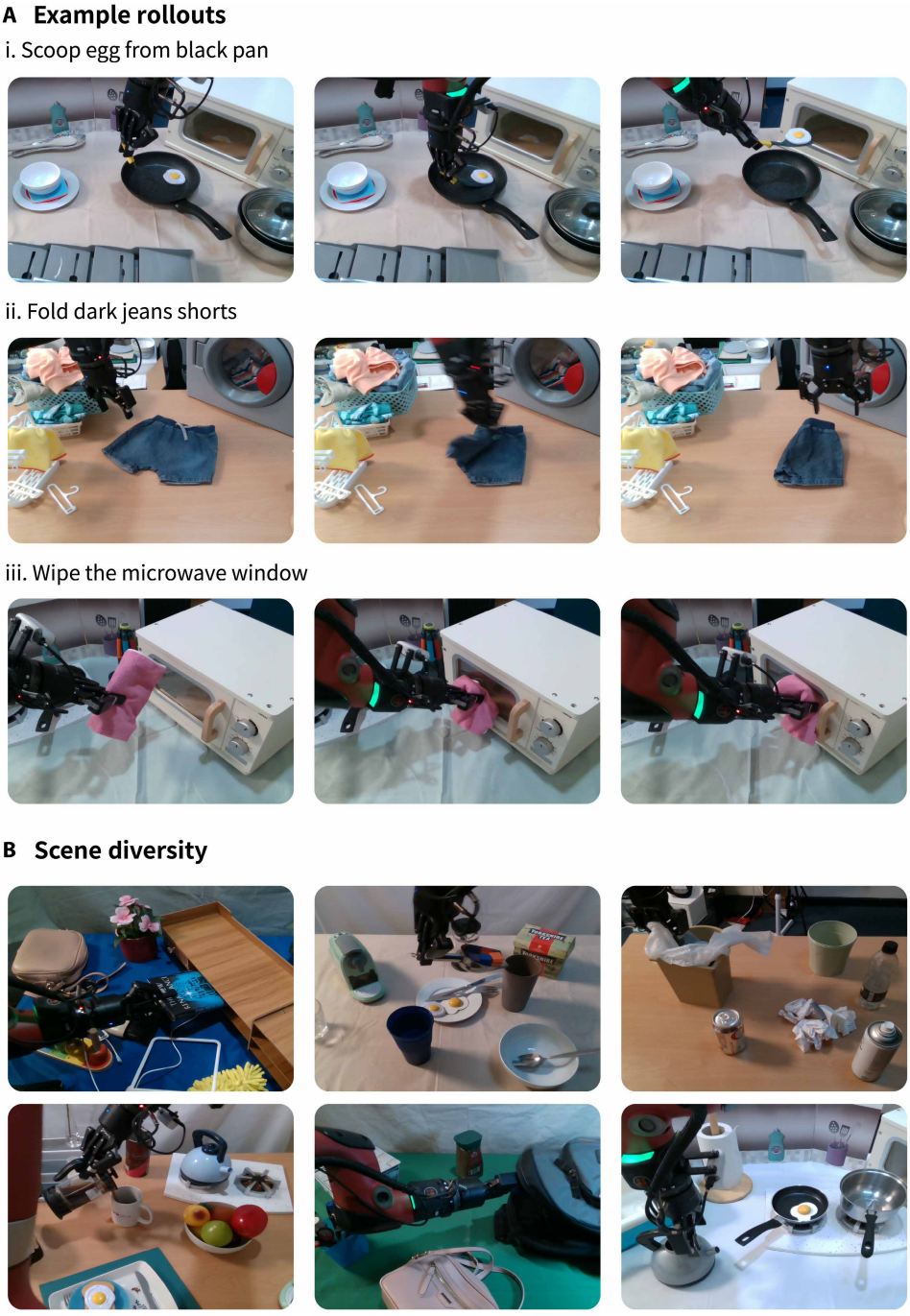


Fig. 5. Example rollouts and scene diversity from the 1000-task evaluation. (**A**) Examples of recorded rollouts from the 1000-task experiment. (**B**) Examples of the scene diversity to which MT3 was subject to during evaluation.

Spatial generalization

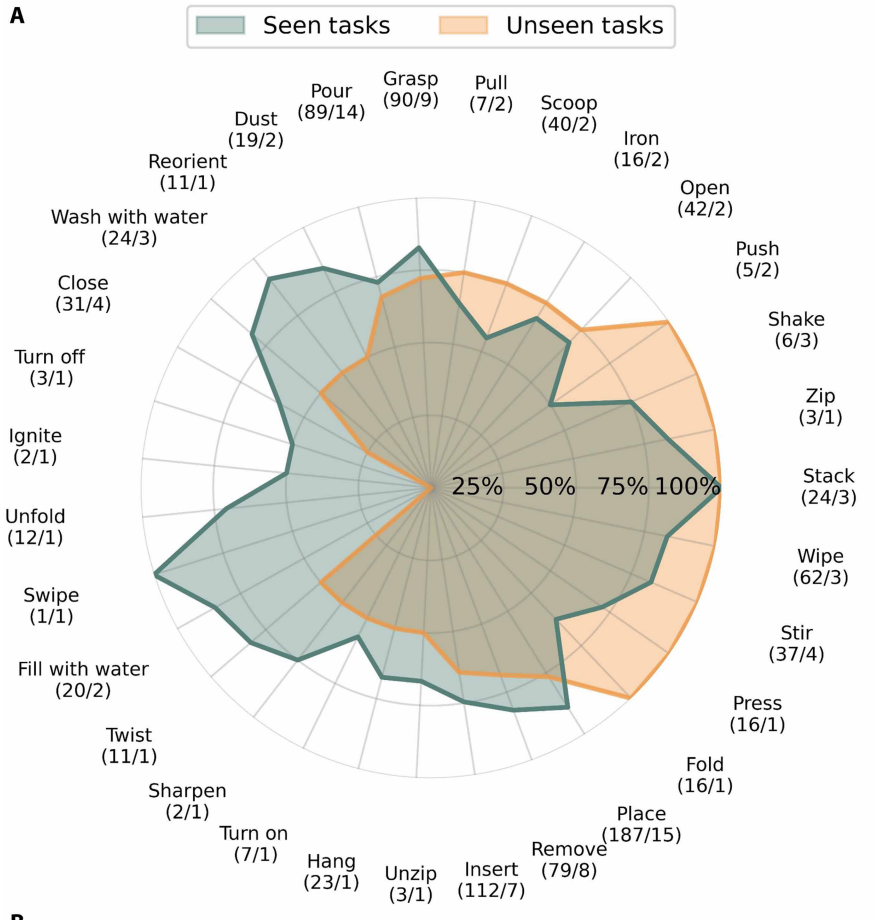
For tasks that permitted small deviations in approach angles and contact positions, such as wiping, stirring, placing, and grasping, the policy consistently achieved success rates exceeding 80%. Open-loop replay worked effectively for such tasks because the interaction phase can accommodate minor positioning inaccuracies while rarely affecting task outcomes.

However, even for some high-tolerance tasks, MT3 encountered difficulties when small geometric features broke object symmetry and were critical for task success. This limitation stemmed from our pose estimator, which predominantly focused on global geometry, occasionally misregistering these small yet task-critical features. For instance, when grasping a kettle, its spout, being small relative to the main body, can be overlooked, leading to a 180° orientation error and subsequent task failure.

Beyond tasks where small features broke symmetry, insertion and operations requiring high-precision alignment also proved challenging. For example, inserting a plug into a socket or hanging small keys requires millimeter-level precision alignment. Because the interaction is open loop, the interaction policy could not compensate for small alignment errors, making the system particularly vulnerable to pose estimation errors for high-precision tasks.

Category-level generalization

Tasks with interaction trajectories that remained consistent across object instances succeeded in generalization. Wiping motions transferred reliably between different surfaces, and object placement followed similar patterns regardless of minor geometric variations. For example, "grasp mug" succeeded because the handle location relative to the body was approximately consistent across different mug designs.



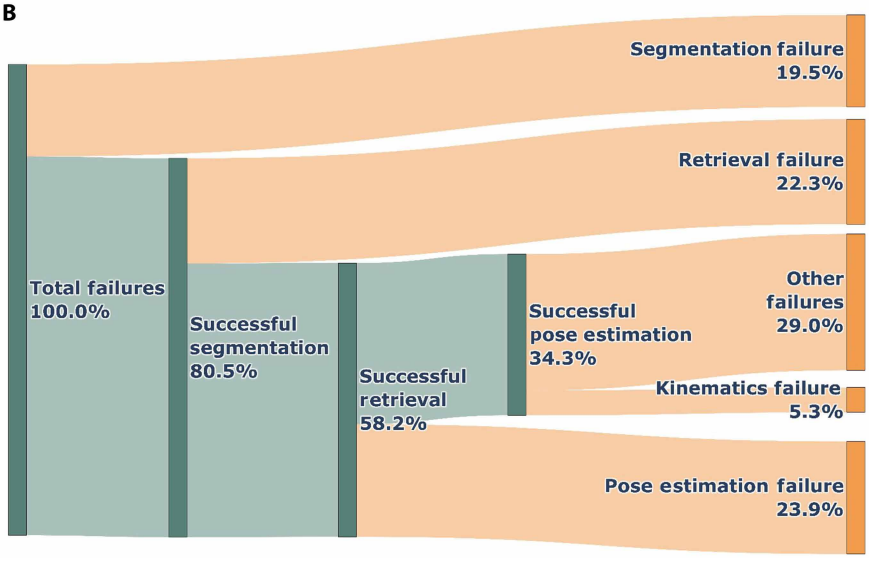


Fig. 6. Results on 1000 tasks. (**A**) Performance of MT3 on 1000 different seen tasks and 100 unseen ones. The results are aggregated by the macro skill. (**B**) Analysis of the different failure causes experienced throughout the 1000-task experiment.

However, failures occurred when changes in object instance geometry caused large variations in the required interaction trajectories. For example, pouring from a kettle required aligning the spout with the edge of a receptacle, and changing the receptacle's geometry might have required the robot to slightly adapt the pouring motion. Similarly, the swiping task failed because although MT3 could match the overall geometry of the cash register, it could not specifically align to the instance-variant card slot that was central to task success.

Another limiting factor arose from our retrieval approach, which fundamentally could not interpolate between demonstrated behaviors. In general, when the required trajectory lies between two demonstrated trajectories, retrieval will select one of the demonstrated ones rather than generating an appropriate intermediate solution. This binary selection process prevented adaptation to object instances that required trajectories not explicitly demonstrated.

Last, tasks involving deformable objects proved particularly challenging because visual similarity alone was insufficient to infer the required trajectory. Different instances of deformable objects have distinct dynamic properties—such as stiffness and elasticity—that are not apparent from visual observation but critically affect manipulation success. For example, inserting a book into a backpack often failed because it required lifting the backpack flap with the book, and this dynamic interaction varied substantially across different backpacks despite visual similarity.

Limitations of open-loop interaction

As demonstrated by our results, MT3 could proficiently tackle a huge variety of tasks. Nonetheless, our large-scale evaluation exposed fundamental limitations of open-loop trajectory replay. This approach often failed for tasks requiring online adaptation during execution. Once a trajectory begins, there is no mechanism for the policy to detect errors or adjust a course midexecution. Although it might be possible to detect task failure and retry, this approach is often suboptimal. For instance, tasks involving deformable objects, such as folding fabric, demanded continuous adjustments based on how the material responded to manipulation. More fundamentally, this open-loop approach cannot satisfy the requirement for reactive control. Operations like reorienting objects through contact or multistep pushing would require continuous feedback and adaptation—capabilities that open-loop replay inherently cannot provide.

Systematic failure analysis

To provide insight into the most common failure modes, we analyzed failure cases on seen tasks, with an expert evaluator assessing each rollout for correct segmentation, exact retrieval, pose estimation success, and motion execution (Fig. 6B). Retrieval emerged as the primary challenge (22.3% of failures), with failures occurring most frequently with partially occluded objects or when relevant geometric variations involved small object parts that our global matching approach could not reliably identify. Segmentation and pose estimation contributed 19.5

and 23.9% of failures, respectively. Although segmentation issues with transparent objects and cluttered scenes should diminish with advancing models, pose estimation challenges with marked pose changes revealed fundamental limitations when object asymmetries create substantially different partial point clouds. The remaining 29% of failures were predominantly from tasks with grasped objects (20.2%), where inconsistent grasp poses between demonstration and deployment highlighted another systematic limitation of the open-loop approach: Any deviation in initial conditions can propagate throughout trajectory execution without the possibility for correction. Despite its limitations, MT3 demonstrated effectiveness across many practical manipulation tasks. Our evaluation of 1000 diverse tasks showed that many everyday tasks—from grasping and placement to insertion and washing—can be learned efficiently from single demonstrations.

DISCUSSION

Learning from limited per-task demonstrations

Our controlled experiments demonstrated that decomposition and retrieval-based policies excelled in the low-demonstration-per-task regime, with MT3 consistently outperforming all alternatives. This effectiveness stemmed from specialized policies that exploited distinct inductive biases rather than learning complex mappings from limited data. For alignment, explicit pose estimation and motion planning addressed spatial generalization through analytical geometric reasoning. This bypassed the need to learn spatial relationships from limited data—a requirement that BC methods had. Similarly, for interaction, trajectory replay ensured sensible interaction execution by directly preserving demonstrated motion patterns, bypassing the challenging problem of learning complex and precise manipulation dynamics from sparse examples. These analytical biases enabled retrieval-based methods to achieve strong performance with minimal data while maintaining predictable behavior.

Beyond performance advantages in low-demonstration-per-task regimes, MT3 offered practical benefits through its inherent interpretability and streamlined task acquisition. For alignment, users could visualize pose estimation results by overlapping registered point clouds (Fig. 7A), enabling preemptive execution halting when misregistration was detected. For interaction, the system directly tracked demonstrated trajectories retrieved from the data buffer, ensuring that the behavior never deviated from what was shown during demonstrations (Fig. 7B). This interpretability provides distinct advantages over BC methods, where policy decisions remain opaque, abstracted away within neural network weights. Furthermore, incorporating previously unknown tasks required only appending demonstrations to the existing dataset, avoiding fine-tuning and retraining procedures that BC methods typically require. This difference makes retrieval-based approaches particularly suitable for applications demanding frequent acquisition of previously unknown tasks.

Data requirements and scaling properties of BC

BC's poor performance in low-demonstration-per-task regimes stemmed from the simultaneous learning challenges it faced when data were scarce. BC had to concurrently learn object geometry understanding, spatial reasoning, and control from limited data while conditioning on language descriptions to distinguish between tasks. In our experimental setting, BC policies had to generalize across spatial variations (±180° rotations and varied positions across a large workspace) and geometric differences between object instances within categories. This setting proved particularly challenging for BC when data were scarce, because the policy struggled to identify meaningful patterns across sparse examples and instead resorted to memorizing the few demonstrations provided, which severely limited generalization performance.

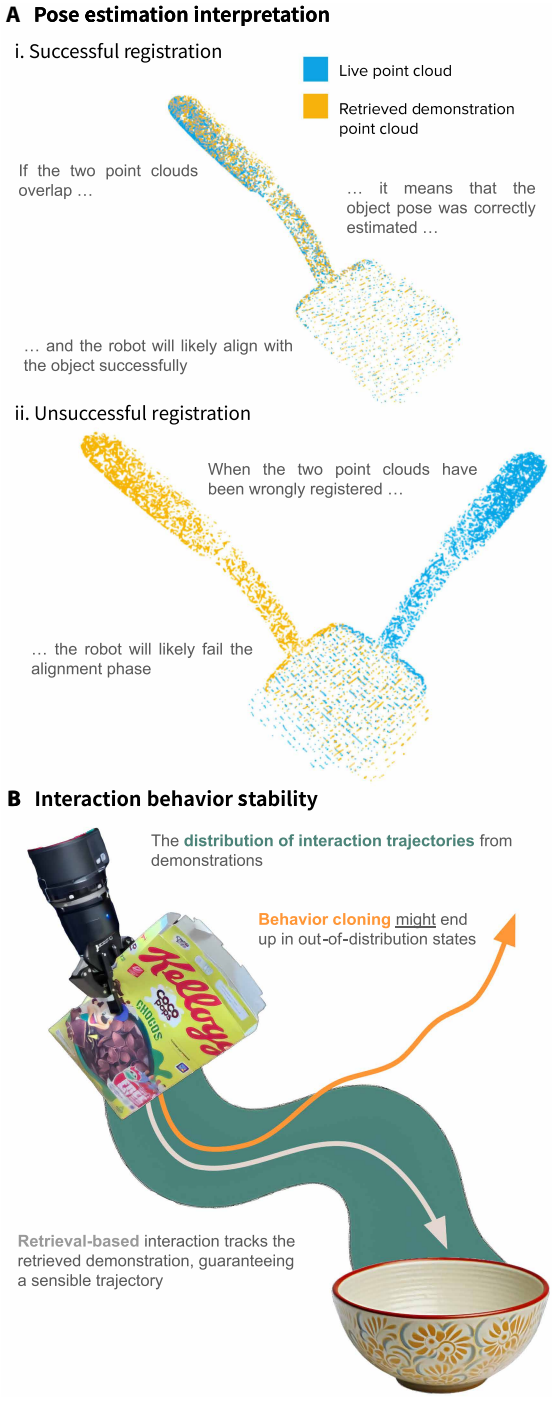


Fig. 7. Interpretability and stability of retrieval-based methods. (**A**) Demonstration of the easily interpretable pose estimation component used by the retrieval-based alignment. (**B**) Visualization of the stability analysis regarding using BC and retrieval for interaction.

Specifically, BC was forced to learn how to map point clouds to trajectory reorientations and control actions, tasks that MT3 handled analytically through a pose estimator and motion planning for alignment and trajectory replay for interaction. This analytical approach completely bypassed the need to learn these mappings from limited demonstrations.

Nevertheless, BC's scaling properties appeared more promising as demonstrations increased. Although MT3's inductive biases proved helpful in low-data regimes, they limited the scaling potential because retrieval methods select single demonstrations and prevent knowledge sharing across demonstrations and tasks. In contrast, monolithic BC could exploit shared patterns across complete trajectories and benefited from increased demonstration diversity, likely making it a more suitable approach than MT3 when data collection resources are unconstrained.

Limitations and scope of the current study

Our study focused on single-interaction, single-arm manipulation tasks with consistent grasped-object poses between demonstration and deployment. For all evaluated methods, varying grasp poses could be overcome without additional demonstrations using existing approaches (*36*), because all methods use segmented target object point clouds as the input. We did not evaluate environments where distractors interfered with demonstrated trajectories. Open-loop replay would have ignored obstacles and likely caused collisions, whereas BC policies would have encountered out-of-distribution scenarios requiring additional demonstration data. As such, these environmental constraints would have been expected to decrease success rates for methods reliant on retrieval-based interaction and to increase data requirements for BC methods and were deemed beyond the scope of the paper.

Our reliance on vision alone made it impossible to infer dynamic properties of deformable objects without tactile feedback. Although BC could adapt through closed-loop control, open-loop retrieval-based interaction often failed because it treated previously unseen objects exactly as training instances and committed to fixed trajectories regardless of material differences. Our approach also relied on accurate object segmentation, which can fail with transparent objects or cluttered scenes, although improving segmentation models would mitigate these issues over time. Extension to bimanual manipulation would require dual-arm coordination mechanisms, although the alignment-interaction decomposition has already been demonstrated for bimanual tasks (*28*), suggesting that MT3's principles could extend to dual-arm scenarios. For multistage tasks, our single-interaction primitives could be chained using high-level planning and skill chaining (*37*–*41*). Further discussion of MT3 limitations can be found in the Supplementary Materials.

Despite these limitations, our findings demonstrated that decomposition combined with retrieval offered a compelling alternative to monolithic BC when demonstration data were limited. The choice between approaches should depend on application constraints: Decomposition excels at rapid task acquisition from minimal data, whereas monolithic BC becomes preferable when data collection resources are less constrained and data are more diverse. Last, our large-scale evaluation identified key technical challenges for retrieval-based approaches. Addressing these challenges while preserving data efficiency advantages represents an important direction for future work.

MATERIALS AND METHODS

Demonstration data collection and processing

We denote a demonstration

$$\tau = \{o_i, e_i\}_{i=1}^{N} \tag{1}$$

as sequences of observations o and end-effector states e recorded at 30 Hz, where i indexes time steps and N is the sequence length. Each observation o_i is an RGB-D image from a calibrated head-mounted camera. The corresponding end-effector state e_i includes the six-dimensional pose of the end-effector frame E in the robot's base frame R, $\mathbf{T}_{RE} \in SE(3)$, and the binary gripper state that indicates whether the gripper is open or closed. Each demonstration was paired with a language description l to differentiate between tasks. This created a dataset D of M demonstrations and their corresponding descriptions

$$D = \{\tau_j, l_j\}_{j=1}^{M} \tag{2}$$

During data collection, we recorded only the interaction phase of each task. This approach was motivated by the distinct requirements of each phase. Alignment requires achieving a specific end-effector pose relative to the target object, whereas the exact trajectory path is less critical. In contrast, interaction requires precise trajectory execution that captures task-relevant manipulation dynamics.

In practice, the demonstrator began recording demonstrations when the end effector reached a pose suitable for initiating the intended interaction. The alignment target pose was extracted as the first pose of the recorded interaction trajectory. This selective recording strategy enabled synthetic generation of alignment trajectories when needed for BC training ("Simulating alignment trajectories for BC" section in Materials and Methods).

After data collection, we segmented all RGB-D images and converted them to target object point clouds. Segmentation enhanced the efficiency and robustness of all methods against background changes and distractors. For the initial RGB image of each demonstration, we used Grounding DINO (*42*) to segment the target object using the object name extracted from the task description l via template-based natural language parsing (*43*). More sophisticated approaches using large language models could be used for more complex task descriptions. For subsequent frames, we propagated the target object segmentation using XMem (*44*), which handled partial and full occlusions.

We converted segmented RGB-D images to target object point clouds using known camera parameters, with coordinate frame representation depending on the method. Retrieval-based methods used point clouds expressed in the robot frame to enable geometry- and pose-based retrieval and to support pose estimation for alignment ("Retrieval-based alignment and interaction" section in Materials and Methods). BC policies used point clouds expressed in the end-effector frame to improve learning efficiency and spatial generalization (*45*). For all demonstrations, we precomputed the geometry embedding of the target object point cloud from the first frame to enable efficient retrieval at test time.

Retrieval-based alignment and interaction

Hierarchical retrieval

Our retrieval system used a two-stage approach illustrated in Fig. 8A. First, we extracted the micro skill name from the task description l using template matching and identified all demonstrations for the

Fig. 8. Language- and geometry-based retrieval. (**A**) Hierarchical retrieval pipeline: Language-based retrieval identifies all demonstrations corresponding to the described micro skill. This is followed by geometry-based retrieval, which matches the object shape and pose to select the single most relevant demonstration. (**B**) A *t*-distributed stochastic neighbor embedding visualization of geometry encodings from the dataset size experiment with 50 demonstrations per object, showing clustering by object category (backpack, toaster, box, and pan). Each category exhibits subclusters corresponding to different object instances, with similar geometries positioned closer in the embedding space (box and toaster). Within subclusters, points from similar object poses have closer embeddings.

same micro skill, for example, "open bottle." This language-based filtering served a crucial architectural role by separating task specification from execution. This design off-loaded a portion of the task complexity from the policy, allowing it to focus solely on replicating the behavior of the specific demonstration provided as context. Although we used template-based extraction and matching for simplicity, more sophisticated approaches using large language models could be used for complex task descriptions ("Large language models for language-based retrieval" section in the Supplementary Materials).

Second, we selected the demonstration with the object most similar to the test object in terms of pose and geometry. Geometric

similarity ensured that objects requiring similar interactions were matched, whereas pose similarity minimized covariate shift for the pose estimator used by the alignment policy. To capture geometry and pose similarity, we used a PointNet++ (*46*)–based encoder trained to predict occupancy grids using the dataset from (*21*). The demonstration with the highest cosine similarity to the test object embedding was selected.

Figure 8B shows a *t*-distributed stochastic neighbor embedding plot of object embeddings from our controlled experiment with 50 demonstrations per task, revealing clustering by object category with subclusters for different instances. This hierarchical organization enabled effective generalization: Similar global geometries clustered together to match manipulation requirements, whereas pose clustering within subclusters enabled relevant demonstration selection for previously unseen object configurations.

Retrieval-based alignment

At inference, the retrieval-based alignment policy receives the target object point cloud and a demonstration of the desired task, and its goal is to align the end effector and the target object in the same way as shown at the beginning of the demonstration. To this end, the policy first used geometric reasoning to infer the required end-effector pose for the test scene and then reached this pose through motion planning.

In this work, we calculated the end-effector pose for the test scene that aligns the end effector and target object in the same way as shown at the beginning of the demonstration using trajectory transfer (*21*). The intuition behind trajectory transfer is that given the relative target object pose between the demonstration and test scene $\mathbf{T}_\delta$, we can map the end-effector pose at the beginning of the demonstration to the test scene using

$$\mathbf{T}_{WE}^{\text{Test}} = \mathbf{T}_\delta \mathbf{T}_{WE}^{\text{Demo}} \tag{3}$$

where $\mathbf{T}_{WE}^{\text{Test}}$ and $\mathbf{T}_{WE}^{\text{Demo}}$ are the end-effector poses expressed in the world frame W for the test and demonstration scenes, respectively, that correspond to the same end effector to target object pose. We estimated $\mathbf{T}_\delta$ by refining the output of the regression method proposed in (*21*) using the Open3D (*47*) implementation of Generalized ICP (*48*).

Retrieval-based interaction

Similar to the retrieval-based alignment policy, at inference, the interaction policy received a demonstration of the desired task. The demonstrated trajectory was then replicated in the test scene by executing the demonstrated end-effector velocities expressed in the end-effector frame.

BC implementation

We used the same network architecture and loss function to learn to align and interact with objects and to learn the single policy for the MT-ACT+ baseline. The only difference between these applications was the training data they relied on. Below, we describe our chosen backbone architecture, the loss function used, and the data that all these policies have been trained on.

Network architecture and design choices

Our policy architecture was required to address three key requirements across all applications. First, it had to process point cloud and language inputs to enable fair comparison with retrieval-based components. Second, it had to effectively handle multitask learning to support evaluation across diverse tasks. Third, it had to capture the multimodal nature of manipulation demonstrations, where multiple valid trajectories may exist for completing the same task or task phase.

To handle point cloud inputs, we used PointNet++ (*46*) that used the CLIP (Contrastive Language-Image Pretraining) (*49*) embedding of the task description l to adapt point cloud features for specific tasks using FiLM (Feature-Wise Linear Modulation) (*50*). To address the multimodal nature of demonstrations, we used variational inference, which enabled the policy to model the multimodal distribution of valid actions. Although diffusion models offer an alternative approach, we chose variational inference because it has demonstrated strong performance for multitask imitation learning from limited demonstrations (*15*).

We adapted the MT-ACT (*15*) architecture to meet these requirements, modifying it to handle point cloud inputs. Additional differences from MT-ACT include incorporating action history as input to help infer task progress; removing proprioception from the input, which our preliminary experiments showed improved spatial generalization; and adding a terminal action output to explicitly signal task completion. We refer to our backbone architecture as MT-ACT+. To ensure peak performance under all experimental conditions, we independently optimized per data regime the number of network parameters for each method that used a BC policy.

Loss function

Similar to the network architecture, the loss function used to train all BC policies was kept consistent. During training, all policies maximized the log likelihood of demonstration action chunks, that is

$$\min_\theta \Sigma_{o_i, a_i, l \sim D} \pi_\theta\left(a_{i:i+k} \mid o_i, l\right) \tag{4}$$

with the standard variational autoencoder objective, which has a reconstruction loss and a term that regularizes the encoder to a Gaussian prior. Here, o_i and $a_{i:i+k}$ are a sampled target object point cloud and an action chunk ("Additional demonstration processing" section in Materials and Methods), respectively, with i representing the time step, k being the action chunk horizon, and l being the corresponding task description. We further augmented this loss by using learned weighting with homoscedastic uncertainty (*51*) to automatically learn the weighting between different components of the reconstruction loss.

Common data augmentation steps

We applied common data augmentation steps during BC training regardless of whether the policy learned alignment, interaction, or both phases. To improve robustness to partial occlusions and varied object poses, we randomly masked portions of the target object point cloud using furthest point sampling followed by nearest neighbor clustering to create 10 clusters, of which we randomly masked 4. We also added Gaussian noise to both point clouds and action history labels to improve robustness to sensor noise.

For interaction policies specifically, we applied additional augmentation to improve robustness to covariate shift when learning from limited data. During training, we perturbed the end-effector pose within 0.9 cm and 5° of its original position and orientation and then updated the corresponding state and action labels to reflect this perturbation. This augmentation helped the policy handle small deviations from demonstrated trajectories that could have occurred during deployment and was enabled by using target object point clouds expressed in the end-effector frame. Additional details regarding data processing

specific to BC interaction policies can be found in the "Additional demonstration processing" section in the Supplementary Materials.

Simulating alignment trajectories for BC

To train alignment capabilities, both the alignment BC policies (used by BC-Ret and BC-BC) and the MT-ACT+ baseline required trajectories that reached the initial pose of each demonstration. We simulated 1000 alignment trajectories per demonstration by sampling starting poses within a 20 cm–by–80 cm–by–80 cm cuboid above the robot's task space and generating linear trajectories to the demonstrated initial end-effector pose.

Given that our BC policies used target object point clouds expressed in the end-effector frame, we could generate training data by reusing the same target object point cloud across different virtual end-effector poses. Specifically, we took the target object point cloud from the first demonstration frame and used it at every waypoint along simulated linear trajectories that end at the target alignment pose. This generated synthetic observation-action pairs where each waypoint had the same geometry of the point cloud input but different action outputs corresponding to the remaining trajectory to reach the target pose. We generated waypoints with 1-cm spacing along each linear path.

To improve alignment accuracy, we supplemented the training data with additional observation-action pairs near the final alignment pose. For each waypoint in a simulated alignment trajectory, we generated an additional observation-action pair by perturbing the end-effector pose within 1 mm to 1 cm and 0.5° to 5° of the target alignment pose.

Combining simulated alignment trajectories and demonstrations

Our monolithic baseline, MT-ACT+, required training data for both the alignment and interaction phases of demonstrated tasks. As such, we combined the simulated alignment trajectories with demonstrated trajectories to create a dataset of entire manipulation trajectories, adjusting the history and action labels at the boundary between alignment and interaction phases.

Statistical analysis

Experimental outcomes were obtained from n independent repeats for each considered task. These data are represented in Figs. 4 and 6 as mean values calculated across all associated tasks and repeats. Error bars in Fig. 4 (A and B) represent 95% Wilson confidence intervals. The two-proportion Z test was applied to assess the statistical significance of performance differences between all considered decomposition-based methods and the monolithic baseline, with P values shown in Fig. 4B.

For Fig. 4 (A and B), the number of evaluated trajectories depended on the experiment configuration. For the dataset size experiment in Fig. 4A, the sample sizes were $n = 36$ for seen tasks and $n = 24$ for unseen tasks. For the dataset diversity experiment in Fig. 4A, when learning 10 tasks, $n = 30$ (seen) and $n = 60$. When learning 30 tasks, $n = 90$ (seen) and $n = 60$ (unseen). Last, when learning 50 tasks, $n = 150$ (seen) and $n = 60$ (unseen).

For the dataset size experiment in Fig. 4B, $n = 240$ (decomposition) and $n = 60$ (monolithic). For the dataset diversity experiment, when learning 10 tasks, $n = 120$ (decomposition) and $n = 30$ (monolithic). When learning 30 tasks, $n = 200$ (decomposition) and $n = 50$ (monolithic). When learning 50 tasks, $n = 280$ (decomposition) and $n = 70$ (monolithic).

Supplementary Materials

The PDF file includes:

Methods and Discussion
Fig. S1
Tables S1 and S2
Legends for movies S1 to S7
References (*52*, *53*)

Other Supplementary Material for this manuscript includes the following:

Movies S1 to S7

REFERENCES AND NOTES

1. E. Somogyi, C. Ara, E. Gianni, L. Rat-Fischer, P. Fattori, J. K. O'Regan, J. Fagard, The roles of observation and manipulation in learning to use a tool. *Cogn. Dev.* **35**, 186–200 (2015).
2. J. Fagard, L. Rat-Fischer, R. Esseily, E. Somogyi, J. K. O'Regan, What does it take for an infant to learn how to use a tool by observation? *Front. Psychol.* **7**, 267 (2016).
3. K. J. Hayes, C. Hayes, Imitation in a home-raised chimpanzee. *J. Comp. Physiol. Psychol.* **45**, 450–459 (1952).
4. V. Horner, A. Whiten, Causal knowledge and imitation/emulation switching in chimpanzees (*Pan troglodytes*) and children (*Homo sapiens*). *Anim. Cogn.* **8**, 164–181 (2005).
5. J. Call, M. Carpenter, M. Tomasello, Copying results and copying actions in the process of social learning: Chimpanzees (*Pan troglodytes*) and human children (*Homo sapiens*). *Anim. Cogn.* **8**, 151–163 (2005).
6. M. M. Rigamonti, D. M. Custance, E. P. Previde, C. Spiezio, Testing for localized stimulus enhancement and object movement reenactment in pig-tailed macaques (*Macaca nemestrina*) and young children (*Homo sapiens*). *J. Comp. Psychol.* **119**, 257–272 (2005).
7. C. Tennie, J. Call, M. Tomasello, Push or pull: Imitation vs. emulation in great apes and human children. *Ethology* **112**, 1159–1169 (2006).
8. M. Meister, Learning, fast and slow. *Curr. Opin. Neurobiol.* **75**, 102555 (2022).
9. E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, C. Finn, "BC-Z: Zero-shot task generalization with robotic imitation learning" in *Proceedings of the 5th Conference on Robot Learning*, A. Faust, D. Hsu, G. Neumann, Eds., vol. 164 of *Proceedings of Machine Learning Research* (PMLR, 2022), pp. 991–1002.
10. Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, L. Fan, "VIMA: General robot manipulation with multimodal prompts" in *NeurIPS 2022 Foundation Models for Decision Making Workshop* (NeurIPS Foundation, 2022); https://openreview.net/forum?id=oU2DzdTI94.
11. N. M. Shafiullah, Z. Cui, A. A. Altanzaya, L. Pinto, "Behavior transformers: Cloning *k* modes with one stone" in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh, Eds. (Curran Associates, 2022), vol. 35, pp. 22955–22968.
12. A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. S. Ryoo, G. Salazar, P. R. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. H. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, B. Zitkovich, "RT-1: Robotics transformer for real-world control at scale" in *Robotics: Science and Systems XIX*, K. E. Bekris, K. Hauser, S. Herbert, J. Yu, Eds. (RSS Foundation, 2023); 10.15607/RSS.2023.XIX.025.
13. B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, Q. Vuong, V. Vanhoucke, H. Tran, R. Soricut, A. Singh, J. Singh, P. Sermanet, P. R. Sanketi, G. Salazar, M. S. Ryoo, K. Reymann, K. Rao, K. Pertsch, I. Mordatch, H. Michalewski, Y. Lu, S. Levine, L. Lee, T.-W. E. Lee, I. Leal, Y. Kuang, D. Kalashnikov, R. Julian, N. J. Joshi, A. Irpan, B. Ichter, J. Hsu, A. Herzog, K. Hausman, K. Gopalakrishnan, C. Fu, P. Florence, C. Finn, K. Avinava Dubey, D. Driess, T. Ding, K. Marcin Choromanski, X. Chen, Y. Chebotar, J. Carbajal, N. Brown, A. Brohan, M. Gonzalez Arenas, K. Han, "RT-2: Vision-language-action models transfer web knowledge to robotic control" in *Proceedings of the 7th Conference on Robot Learning*, J. Tan, M. Toussaint, K. Darvish, Eds., vol. 229 of *Proceedings of Machine Learning Research* (PMLR, 2023), pp. 2165–2183.
14. T. Z. Zhao, V. Kumar, S. Levine, C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware" in *Robotics: Science and Systems XIX*, K. E. Bekris, K. Hauser, S. Herbert, J. Yu, Eds. (RSS Foundation, 2023); 10.15607/RSS.2023.XIX.016.
15. H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, V. Kumar, "RoboAgent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking" in *2024 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2024), pp. 4788–4795; 10.1109/ICRA57147.2024.10611293.
16. M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, C. Finn, OpenVLA: An open-source vision-language-action model. arXiv:2406.09246 (2024).
17. K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, U. Zhilinsky, π0: A vision-language-action flow model for general robot control. arXiv:2410.24164 (2024).

18. D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, J. Luo, Y. L. Tan, L. Y. Chen, Q. Vuong, T. Xiao, P. R. Sanketi, D. Sadigh, C. Finn, S. Levine, "Octo: An open-source generalist robot policy" in *Proceedings of Robotics: Science and Systems XX* (RSS Foundation, 2024); 10.15607/RSS.2024.XX.090.
19. T. Z. Zhao, J. Tompson, D. Driess, P. Florence, S. K. S. Ghasemipour, C. Finn, A. Wahid, "ALOHA unleashed: A simple recipe for robot dexterity" in *Proceedings of the 8th Conference on Robot Learning*, P. Agrawal, O. Kroemer, W. Burgard, Eds., vol. 270 of *Proceedings of Machine Learning Research* (PMLR, 2025), pp. 1910–1924.
20. M. A. Lee, C. Florensa, J. Tremblay, N. Ratliff, A. Garg, F. Ramos, "Guided uncertainty-aware policy optimization: Combining learning and model-based strategies for sample-efficient policy learning" in *2020 IEEE International Conference on Robotics and Automation* (*ICRA*) (IEEE, 2020), pp. 7505–7512; 10.1109/ICRA40945.2020.9197125.
21. P. Vitiello, K. Dreczkowski, E. Johns, "One-shot imitation learning: A pose estimation perspective" in *Proceedings of the Conference on Robot Learning*, J. Tan, M. Toussaint, K. Darvish, Eds., vol. 229 of *Proceedings of Machine Learning Research* (PMLR, 2023), pp. 943–970.
22. E. Johns, "Coarse-to-fine imitation learning: Robot manipulation from a single demonstration" in *IEEE International Conference on Robotics and Automation* (*ICRA*) (IEEE, 2021), pp. 4613–4619.
23. E. Valassakis, N. Di Palo, E. Johns, "Coarse-to-fine for sim-to-real: Sub-millimetre precision across wide task spaces" in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS*) (IEEE, 2021), pp. 5989–5996.
24. N. Di Palo, E. Johns, "Learning multi-stage tasks with one demonstration via self-replay" in *Proceedings of the 5th Conference on Robot Learning*, A. Faust, D. Hsu, G. Neumann, Eds., vol. 164 of *Proceedings of Machine Learning Research* (PMLR, 2021), pp. 1180–1189.
25. E. Valassakis, G. Papagiannis, N. Di Palo, E. Johns, "Demonstrate once, imitate immediately (DOME): Learning visual servoing for one-shot imitation learning" in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS*) (IEEE, 2022), pp. 8614–8621.
26. J. Zhao, Z. Wang, L. Zhao, H. Liu, A learning-based two-stage method for submillimeter insertion tasks with only visual inputs. *IEEE Trans. Ind. Electron.* **71**, 7381–7390 (2024).
27. N. Di Palo, E. Johns, "DINOBot: Robot manipulation via retrieval and alignment with vision foundation models" in *IEEE International Conference on Robotics and Automation* (*ICRA*) (IEEE, 2024), pp. 2798–2805.
28. Y. Wang, E. Johns, "One-shot dual-arm imitation learning" in *IEEE International Conference on Robotics and Automation* (*ICRA*) (IEEE, 2024), pp. 5660–5668.
29. L.-H. Lin, Y. Cui, A. Xie, T. Hua, D. Sadigh, "FlowRetrieval: Flow-guided data retrieval for few-shot imitation learning" in *Proceedings of the 8th Annual Conference on Robot Learning*, J. Tan, M. Toussaint, K. Darvish, Eds., vol. 229 of *Proceedings of Machine Learning Research* (PMLR, 2024).
30. S. Nasiriany, T. Gao, A. Mandlekar, Y. Zhu, "Learning and retrieval from prior data for skill-based imitation learning" in *Proceedings of the 6th Conference on Robot Learning*, K. Liu, D. Kulic, J. Ichnowski, Eds., vol. 205 of *Proceedings of Machine Learning Research* (PMLR, 2023), pp. 2181–2204.
31. M. Du, S. Nair, D. Sadigh, C. Finn, "Behavior retrieval: Few-shot imitation learning by querying unlabeled datasets" in *Proceedings of Robotics: Science and Systems XIX*, K. E. Bekris, K. Hauser, S. Herbert, J. Yu, Eds. (RSS Foundation, 2023); 10.15607/RSS.2023.XIX.011.
32. J. Pari, N. M. Shafiullah, S. P. Arunachalam, L. Pinto, The surprising effectiveness of representation learning for visual imitation. arXiv:2112.01511 (2021).
33. T. Shankar, A. Gupta, Learning robot skills with temporal variational inference. arXiv:2006.16232 (2020).
34. A. Graves, "Practical variational inference for neural networks" in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K. Q. Weinberger, Eds. (Curran Associates, 2011), vol. 24; https://proceedings.neurips.cc/paper_files/paper/2011/file/7eb3c8be3d411e8ebfab08eba5f49632-Paper.pdf.
35. J. Schulman, J. Ho, C. Lee, P. Abbeel, "Learning from demonstrations through the use of non-rigid registration" in *Robotics Research: The 16th International Symposium ISRR*, M. Inaba, P. Corke, Eds., vol. 114 of *Springer Tracts in Advanced Robotics* (Springer International Publishing, 2016), pp. 339–354; 10.1007/978-3-319-28872-7_20.
36. G. Papagiannis, K. Dreczkowski, V. Vosylius, E. Johns, "Adapting skills to novel grasps: A self-supervised approach" in *Proceedings of the International Conference on Intelligent Robots and Systems* (*IROS*) (IEEE, 2024), pp. 10897–10904.
37. B. Ichter, A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, D. Kalashnikov, S. Levine, Y. Lu, C. Parada, K. Rao, P. Sermanet, A. T. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, M. Yan, N. Brown, M. Ahn, O. Cortes, N. Sievers, C. Tan, S. Xu, D. Reyes, J. Rettinghouse, J. Quiambao, P. Pastor, L. Luu, K.-H. Lee, Y. Kuang, S. Jesmonth, N. J. Joshi, K. Jeffrey, R. Jauregui Ruano, J. Hsu, K. Gopalakrishnan, B. David, A. Zeng, C. K. Fu, "Do as I can, not as I say: Grounding language in robotic affordances" in *Proceedings of the 6th Conference on Robot Learning*, K. Liu, D. Kulic, J. Ichnowski, Eds., vol. 205 of *Proceedings of Machine Learning Research* (PMLR, 2022), pp. 287–318.
38. S. H. Vemprala, R. Bonatti, A. Bucker, A. Kapoor, ChatGPT for robotics: Design principles and model abilities. *IEEE Access* **12**, 55682–55696 (2024).
39. L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, J. Wen, A survey on large language model based autonomous agents. *Front. Comp. Sci.* **18**, 186345 (2024).
40. J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, "Code as policies: Language model programs for embodied control" in *2023 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2023), pp. 9493–9500; 10.1109/ICRA48891.2023.10160591.
41. M. Argus, A. Nayak, M. Büchner, S. Galesso, A. Valada, T. Brox, "Compositional servoing by recombining demonstrations" in *2024 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2024), pp. 7339–7346.
42. S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, L. Zhang, "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection" in *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XLVII*, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, G. Varol, Eds., vol. 15105 of *Lecture Notes in Computer Science* (Springer Nature, 2025), pp. 38–55.
43. T. Gunawardena, M. Lokuhetti, N. Pathirana, R. Ragel, S. Deegalla, "An automatic answering system with template matching for natural language questions" in *2010 Fifth International Conference on Information and Automation for Sustainability* (IEEE, 2010), pp. 353–358; 10.1109/ICIAFS.2010.5715686.
44. H. K. Cheng, A. G. Schwing, "XMem: Long-term video object segmentation with an Atkinson-Shiffrin memory model" in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, T. Hassner, Eds., vol. 13688 of *Lecture Notes in Computer Science* (Springer Nature, 2022), pp. 640–658; 10.1007/978-3-031-19815-1_37.
45. M. Liu, X. Li, Z. Ling, Y. Li, H. Su, "Frame mining: A free lunch for learning robotic manipulation from 3D point clouds" in *Proceedings of the 6th Conference on Robot Learning*, K. Liu, D. Kulic, J. Ichnowski, Eds., vol. 205 of *Proceedings of Machine Learning Research* (PMLR, 2022), pp. 527–538.
46. C. R. Qi, L. Yi, H. Su, L. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space" in *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett, Eds. (Curran Associates, 2017), vol. 30; https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf.
47. Q.-Y. Zhou, J. Park, V. Koltun, Open3D: A modern library for 3D data processing. arXiv:1801.09847 (2018).
48. A. Segal, D. Haehnel, S. Thrun, "Generalized-ICP" in *Proceedings of Robotics: Science and Systems V*, J. Trinkle, Y. Matsuoka, J. A. Castellanos, Eds. (RSS Foundation, 2009); 10.15607/RSS.2009.V.021.
49. A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, "Learning transferable visual models from natural language supervision" in *Proceedings of the 38th International Conference on Machine Learning*, vol. 139 of *Proceedings of Machine Learning Research* (PMLR, 2021), pp. 8748–8763.
50. E. Perez, F. Strub, H. de Vries, V. Dumoulin, A. Courville, "FiLM: Visual reasoning with a general conditioning layer" in *Thirty-Second AAAI Conference on Artificial Intelligence* (AAAI, 2018), vol. 32; 10.1609/aaai.v32i1.11671.
51. A. Kendall, R. Cipolla, "Geometric loss functions for camera pose regression with deep learning" in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2017), pp. 6555–6564; 10.1109/CVPR.2017.694.
52. N. Heppert, M. Argus, T. Welschehold, T. Brox, A. Valada, "DITTO: Demonstration imitation by trajectory transformation" in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS*) (IEEE, 2024), pp. 7565–7572; 10.1109/IROS58592.2024.10801982.
53. S. Mirchandani, F. Xia, P. Florence, B. Ichter, D. Driess, M. Gonzalez Arenas, K. Rao, D. Sadigh, A. Zeng, "Large language models as general pattern machines" in *Proceedings of the 7th Annual Conference on Robot Learning*, J. Tan, M. Toussaint, K. Darvish, Eds., vol. 229 of *Proceedings of Machine Learning Research* (PMLR, 2023), pp. 2498–2518.

Acknowledgments: We thank the members of the Robot Learning Lab at Imperial College London for all the support. More specifically, we thank G. Papagiannis, I. Kapelyukh, Y. Ren, Y. Wang, R. Fan, A. Daniel, and N. Di Palo for the helpful discussions and feedback on the paper. **Funding:** This work was supported by the following: EPSRC DTP Reference Number 2297064 (to K.D.); EPSRC DTP Reference Number EP/W524323/1 (to P.V.); RAEng Research Fellowship RF/201617/16A7 (to E.J.).
Author contributions: Conceptualization: K.D., P.V., and E.J. Methodology: K.D., P.V., and E.J. Software: K.D., P.V., and V.V. Investigation and analysis: K.D. and P.V. Resources: E.J. Data curation: K.D. and P.V. Writing—original draft: K.D. and P.V. Writing—review and editing: K.D., P.V., and E.J. Visualization: K.D. and P.V. Supervision: E.J. Project administration: K.D., P.V., and E.J. Funding acquisition: E.J. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper or the Supplementary Materials. In addition, we shared the code used to train the behavior cloning models, as well as example deployment scripts, and observation-action pairs from MT3's evaluation rollouts. These have been deposited at the following Zenodo repository: https://doi.org/10.5281/zenodo.17334511.

Submitted 10 January 2025
Accepted 15 October 2025
Published 12 November 2025
10.1126/scirobotics.adv7594

Science Robotics

Learning a thousand tasks in a day

Kamil Dreczkowski, Pietro Vitiello, Vitalis Vosylius, and Edward Johns

Sci. Robot. **10** (108), eadv7594. DOI: 10.1126/scirobotics.adv7594

Science Robotics (ISSN 2470-9476) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Robotics* is a registered trademark of AAAS.