GENERATIVE ACTION SYNTHESIS: INFERRING ROBOT ACTIONS FROM KINEMATIC DEMONSTRATIONS

Anonymous authors
Paper under double-blind review

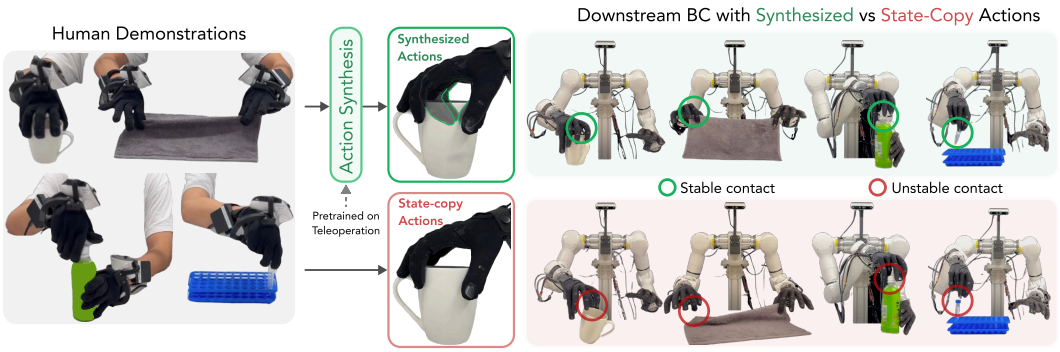

Figure 1: Generative Action Synthesis (GAS) labels kinematic-only human demonstrations with synthesized robot actions. While glove demonstrations provide hand kinematics without the corresponding joint commands, GAS synthesizes commands that sustain contact (green), rather than directly using retargeted joint states as actions (red). Policies trained on GAS-labeled demonstrations maintain stable contact, whereas State-copy policies lose contact or form loose grasps.

ABSTRACT

For dexterous hands, kinematic demonstrations from motion-capture gloves and DexUMI-style exoskeletons are easier and faster to collect than demonstrations collected via robot teleoperation, but they record only observed hand motion, not the corresponding joint commands. This missing information is crucial during contact, when applying forces to objects or the environment requires joint commands to differ from the hand's observed configuration. Robot teleoperation data directly capture this missing information through the residual between commanded joint targets and observed hand states. In this work, we propose Generative Action Synthesis (GAS), which trains a conditional generative model of these residuals from teleoperation data and uses it to synthesize action labels for kinematic-only demonstrations. For each frame, the model uses past and future visual observations together with observed joint states to sample plausible command residuals for relabeling kinematic-only demonstrations. GAS synthesizes useful actions zero-shot across diverse manipulation tasks beyond its teleoperation training distribution, including mug grasping, towel folding, cap twisting, test tube insertion, USB unplugging, and Rubik's cube manipulation. With these synthesized actions, we train policies for mug grasping, test tube insertion, cap twisting, and towel folding that achieve high success rates and outperform both naive state-as-action baselines and constant heuristic offsets. These results suggest that teleoperation data can provide supervision for converting action-free kinematic demonstrations into useful training data for robot policies. More videos are available at https://generativeactionsynthesis.github.io.

1 INTRODUCTION

Learning dexterous manipulation from demonstrations requires data that capture not only hand motion, but also the control commands that maintain effective physical interaction. Human-centric data

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

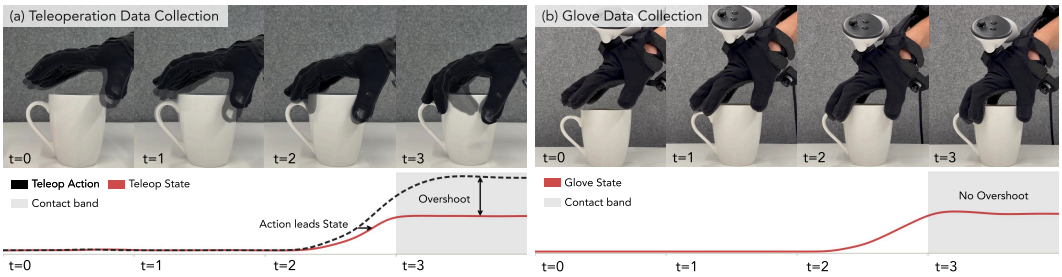

Figure 2: **Joint states and commanded actions differ during contact.** (a) Teleoperation records both measured joint states and commanded joint targets, visualized as solid and transparent hands, respectively. Commands lead the measured states during motion and specify further closure during contact. (b) Glove demonstrations record hand motion without the corresponding robot commands.

collection interfaces, such as motion-capture gloves and UMI-style systems, provide an appealing way to collect manipulation data at scale because they do not require operating the target robot during data collection (Chi et al., 2024; Xu et al., 2025; Lin & Zhao, 2026). However, a limitation of these systems is that they only capture hand and end-effector kinematics, rather than the robot joint-position commands required for control.

This distinction is less apparent in free space, where commanded and observed joint positions are near-identical. During contact, however, they diverge (Fig. 2). As a finger makes contact, its joint angle is constrained by the contact geometry, while the commanded joint angle can correspond to a configuration that would penetrate the object's surface. The difference between commanded and observed configurations therefore provides information about the commands needed to make, sustain, and regulate contact. A common approach is to use observed or retargeted joint states directly as action targets (Wang et al., 2024). However, a policy trained on these labels will simply reproduce the geometry of the demonstrated hand motion without reproducing the contact forces needed to perform manipulation.

One solution to this problem is to collect action data directly via teleoperation. During teleoperation, a human user sends closed-loop joint commands that account for contact and interaction forces, providing both the robot's observed motion and the joint commands that produced it. However, collecting large-scale teleoperation data is substantially slower and more cumbersome than using more natural interfaces such as motion-capture gloves. This raises the question: *rather than using teleoperation data to learn how to perform particular tasks, can we use it to learn the general relationship between observed kinematics and robot actions?* In particular, in this work we explore whether general, large-scale teleoperation data can be used to synthesize action labels for independently collected kinematic demonstrations.

We introduce GAS, a generative action-synthesis framework for learning dexterous manipulation from human demonstrations. We first train a conditional flow-matching model (Lipman et al., 2023) on a multitask teleoperation corpus to predict joint-position command residuals from RGB observations and temporal joint-state context. By synthesizing actions offline, our model can condition on both past and future observations, providing context about the demonstrated interaction. We then use our model to relabel glove demonstrations of novel tasks not seen during synthesis-model training. Finally, we use the resulting action-labeled demonstrations to train visuomotor policies through behavior cloning.

We evaluate GAS on tasks excluded from action-synthesis training, using open-loop replay across seven tasks and downstream policy learning across five tasks. Retargeted hand states can reproduce the demonstrated kinematics but often fail to maintain contact with objects. By synthesizing the missing action labels, GAS substantially improves both replay success and downstream policy performance, with particularly large gains on contact-sensitive tasks such as mug grasping and test tube insertion. Under approximately matched data-collection time, glove-based collection yields more demonstrations than robot teleoperation, and policies trained on these demonstrations with synthesized action labels achieve higher average success. Together, these results suggest that robot teleoperation data can be used to learn a transferable mapping from observed hand kinematics to

robot actions. This mapping synthesizes actions for independently collected human demonstrations of unseen tasks, making them a scalable source of training data for dexterous manipulation policies.

2 Related Work

2.1 Human-centric data collection for dexterous manipulation

Human-centric interfaces enable manipulation demonstrations to be collected without operating the target robot. UMI-style handheld systems (Chi et al., 2024; Choi et al., 2026; Liu et al., 2025), motion-capture pipelines (Wang et al., 2024), and glove or exoskeleton interfaces (Xu et al., 2025; Lin & Zhao, 2026; Yin et al., 2025; Zhang et al., 2025) capture human hand and end-effector motion that can be retargeted to robot embodiments. These systems greatly improve the scalability of data collection, but their supervision is fundamentally kinematic: the control commands that would produce the demonstrated interaction on the target robot are not directly observed.

Prior systems address this gap through hardware or simple heuristics. For parallel-jaw grippers, UMI records only gripper width and a fixed closing offset or close-until-contact rule is often sufficient for a single degree of freedom (Chi et al., 2024). Dexterous hands, in contrast, must coordinate many joints across multiple, changing contacts with diverse geometry. DexUMI instead mounts the tactile sensors on its exoskeleton as on the robot hand (Xu et al., 2025). Tactile sensing improves contact awareness (Lin et al., 2025b; Liu et al., 2025; Pan et al., 2026), but tactile systems vary substantially in sensor layout, calibration, and signal characteristics, and many glove or UMI-style interfaces do not provide a common tactile representation. We therefore consider tactile-free kinematic demonstrations and learn to synthesize these commands from robot teleoperation data.

2.2 Human-to-Robot Retargeting for Dexterous Manipulation

Human-to-robot retargeting maps demonstrated human motion to robot embodiments by matching hand poses, fingertip configurations, or end-effector trajectories. Purely kinematic retargeting is effective for transferring motion across embodiments, but does not explicitly account for the physical interaction that gives rise to the demonstrated behavior.

Recent work therefore incorporates interaction constraints directly into the retargeting process. Demonstrated hand–object spatial and contact relations can be explicitly preserved during retargeting and subsequently realized through residual control (Feng et al., 2026; Li et al., 2025). Other approaches use simulation to enforce interaction feasibility, combining motion and contact guidance with virtual object controllers (Mandi et al., 2025) or searching for dynamically feasible control sequences under virtual contact guidance (Pan et al., 2025). Reconstructed interaction models can similarly provide task-level constraints for obtaining robot behaviors consistent with demonstrated objectives (Zhu et al., 2026).

These methods improve the physical consistency of transferred trajectories, but primarily optimize the retargeted motion or interaction outcome. A complementary line of work infers missing action labels from observed state transitions or videos before using the relabeled demonstrations for behavioral cloning (Torabi et al., 2018; Baker et al., 2022). In our setting, the demonstrations are collected from humans rather than the target robot, and during contact, multiple joint commands can be consistent with the same retargeted hand state. We therefore model the distribution over robot commands with a generative model conditioned on visual observations and past and future context to synthesize robot hand commands.

2.3 Contact-Rich Dexterous Hand Manipulation

Contact-rich manipulation with multifingered hands requires coordinating high-dimensional joint actions across multiple, evolving contacts. Prior work has addressed this challenge through improved sensing, physical modeling, and interaction-aware policy learning. Sim-to-real reinforcement learning has enabled dexterous grasping and bimanual manipulation through careful contact modeling, reward design, and robust perception (Qin et al., 2023; Lin et al., 2025a; Hsieh et al., 2026). Real-world imitation learning instead leverages robot teleoperation to acquire complex dexterous behaviors, with tactile and force sensing further improving contact awareness and force reg-

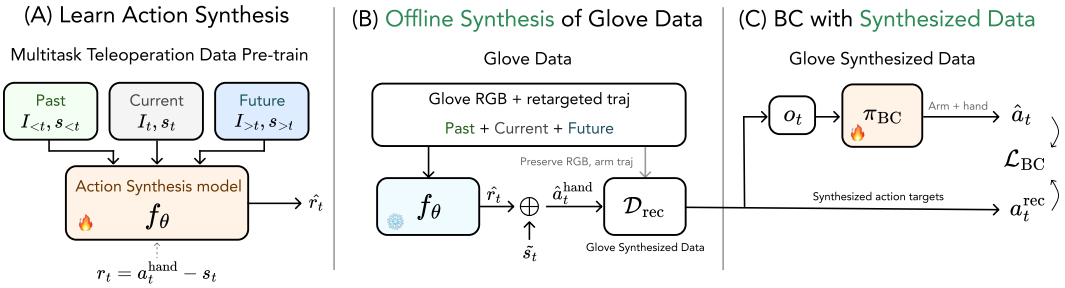

Figure 3: **An overview of our approach.** We first train a multimodal action-synthesis model on a broad set of teleoperation pretraining data to learn the relationship between measured hand states and commanded joint targets. We then apply the model to synthesize contact-aware hand actions for task-disjoint glove demonstrations. Finally, we train visuomotor behavior-cloning policies on the synthesized demonstrations to obtain coordinated arm–hand manipulation.

ulation (Lin et al., 2025b; Heng et al., 2026). Together, these works demonstrate the importance of modeling physical interaction beyond purely kinematic motion for dexterous manipulation.

Existing approaches typically obtain executable action supervision either from simulation followed by sim-to-real transfer or directly from robot teleoperation. Our setting instead separates the source of control supervision from the source of downstream task demonstrations. Robot teleoperation provides supervision for a reusable action-synthesis model, which is applied zero-shot to human demonstrations of tasks excluded from its training corpus. The downstream visuomotor policies are then trained exclusively on human demonstrations labeled by GAS, without task-specific robot teleoperation demonstrations. Moreover, these demonstrations are collected with a lightweight soft electromagnetic-field (EMF) sensing glove rather than an embodiment-specific exoskeleton, decoupling downstream data collection from both the target robot and robot-specific wearable hardware.

3 Action Synthesis for Human Kinematic Demonstrations

An overview of our method is shown in Figure 3. It consists of three stages. First, we train an action-synthesis model on multitask teleoperation data to learn command–state residuals from visual observations and temporal joint-state context (Sec. 3.1). We then apply the frozen model to synthesize action labels for human glove demonstrations of unseen tasks (Sec. 3.2). Finally, we use the labeled demonstrations to fine-tune visuomotor policies through behavior cloning (Sec. 3.3).

3.1 Learning Action Synthesis from Teleoperation

Robot teleoperation provides direct supervision for the discrepancy between measured hand configurations and commanded joint targets during physical interaction. Let s_t denote the measured joint positions of both hands and a_t the corresponding joint-position commands. Each demonstration also contains synchronized multi-view RGB images $\mathcal{I}_t$. We group these with the joint states into an observation window $\mathbf{o}_t = \left(s_{t-H+1:t+F}, \mathcal{I}_{t-H+1:t+F}\right)$, where H and F are the history and future horizons. RGB observations are sampled from this window at the temporal offsets specified in Appendix C. We train the action-synthesis model on diverse multitask teleoperation data and evaluate its transfer to downstream tasks unseen during training.

Residual action representation. Instead of predicting the absolute command a_t, we parameterize the synthesized action as a residual relative to the measured state s_t,

$$r_t = a_t - s_t. \tag{1}$$

This representation isolates the measured configuration from the learning target and emphasize modeling on the command–state discrepancy, including discrepancies arising from contact and actuator dynamics. We model chunks of C residuals, $\mathbf{R}_t = r_{t:t+C-1}$, conditioned on the observation window:

$$p_\theta(\mathbf{R}_t \mid \mathbf{o}_t)\,. \tag{2}$$

Since action synthesis is performed offline, the model can be conditioned on both past and future observations. Future observations provide additional information about the physical interaction that followed a command, which is particularly useful when the commanded closure is not directly observable from the current joint configuration alone. Moreover, multiple commands can produce similar observed interactions, particularly during contact. We therefore model a distribution over residual commands rather than a single deterministic prediction.

Generative residual modeling. We model the conditional distribution of residual-action chunks using conditional flow matching (Lipman et al., 2023), following prior work on generative action-chunk modeling for robot manipulation (Chi et al., 2023; Zhao et al., 2023). Let x_1 denote a standardized residual-action chunk and c_t the conditioning representation of the corresponding proprioceptive and multi-view visual context. We sample $x_0 \sim \mathcal{N}(0, I)$ and $\lambda \sim \mathcal{U}[0, 1]$ and define the linear interpolation

$$x_\lambda = (1 - \lambda)x_0 + \lambda x_1. \tag{3}$$

We train a conditional velocity field $v_\theta(x_\lambda, \lambda, c_t)$ to predict the velocity of this path,

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{x_0, x_1, \lambda}\left[\|v_\theta(x_\lambda, \lambda, c_t) - (x_1 - x_0)\|_2^2\right]. \tag{4}$$

At inference, we sample Gaussian noise and integrate the learned velocity field from $\lambda = 0$ to 1 to generate residual-action chunks. We draw K samples per window and use their mean as the residual prediction. The synthesized residuals are then added to the retargeted hand states to obtain joint-position commands.

3.2 Offline Action Synthesis for Human Demonstrations

After training, we freeze the action-synthesis model and apply it to task-disjoint human demonstrations without target-task adaptation or additional robot interaction. Let $\tilde{s}_t$ denote the retargeted hand configuration in a glove demonstration. During action synthesis, $\tilde{s}_t$ serves as the state input in place of the measured robot state used during teleoperation training, while the recorded glove RGB observations provide the visual context. We use the same observation preprocessing and normalization statistics as during action-synthesis training.

Residual synthesis. For each query time t, the frozen model samples K residual-action chunks conditioned on the surrounding state and visual context. Because the demonstrations are available offline, action synthesis can use both past and future observations. Following ACT (Zhao et al., 2023), we query the model at every frame and ensemble predictions from overlapping chunks. We first average the K samples within each chunk and then uniformly average predictions from all chunks covering frame t. The resulting residual is converted back to joint-angle units and added to the retargeted hand configuration to obtain the synthesized command.

The synthesized commands replace the kinematic hand-action labels in the human demonstrations, while the recorded RGB observations and retargeted end-effector trajectories remain unchanged. The resulting dataset therefore preserves the demonstrated target-task behavior while augmenting it with hand commands synthesized from the control structure learned on robot teleoperation.

3.3 Behavior Cloning with Synthesized Data

The dataset augmented with synthesized action $\mathcal{D}_{\mathrm{rec}}$ provides complete visuomotor demonstrations for downstream policy learning. Each demonstration retains the recorded visual observations and retargeted end-effector trajectories, while its hand-action targets are replaced by the synthesis action in Sec. 3.2. Importantly, the downstream policy is trained exclusively on these human demonstrations with synthesized-action; the robot teleoperation data used to train the action-synthesis model are not used for behavior cloning.

Visuomotor policy. We train a flow-matching behavior-cloning policy π_{BC} that maps multi-view RGB observations, end-effector poses, hand-joint states, and a task instruction to arm–hand action chunks. A DiT-style action model jointly predicts future end-effector targets and synthesized hand joint-position commands, enabling coordinated bimanual arm–hand control. The policy is trained with the same flow-matching objective as the synthesis model (Eq. 4), initialized from a pretrained manipulation checkpoint, and fine-tuned on the synthesized demonstrations. Architecture, optimization, and inference details are provided in Appendix A.

4 EXPERIMENTS

In this section, we evaluate whether GAS can turn human demonstrations into effective training data for dexterous manipulation policies. We study two main questions: (1) Can action-synthesis model, GAS, learned from a separate multitask teleoperation corpus synthesize useful action labels for diverse unseen tasks? (2) Can policies trained on human demonstrations with synthesized actions attain comparable task performance to policies trained on task-specific robot teleoperation demonstrations? Together, these experiments evaluate whether action synthesis produces useful labels for unseen tasks and whether the labeled human demonstrations can reduce data collection effort while maintaining downstream policy performance.

4.1 EXPERIMENTAL SETUP

Robot Setup. Our system consists of a pair of Tianji robot arms, each equipped with a 20-DoF Wuji Hand v1. The hands are controlled through joint-position targets at 30 Hz. Visual observations are captured at 30 Hz using an Intel RealSense D455 for the egocentric view and two wrist-mounted RGB cameras. Visual observations, measured joint states, and commanded joint targets are time-synchronized during robot deployment.

Human demonstration collection. We collect human demonstrations using the interface shown in Figure 4. We use Meta Quest 3 to capture arm positions and orientations, and Wuji Glove 1.0 to capture fingertip positions and orientations. An Intel RealSense D455 provides the egocentric view, and two wrist-mounted RGB cameras provide the wrist views.

Metrics.

We evaluate each task and condition over 20 real-world trials with matched initial object configurations. For downstream policies, we report progress ratio, the fraction of ordered task stages completed, and success rate, the fraction of trials completing all stages. For action replay, we report success rate. Violin plots show Bernoulli success-rate posteriors under a uniform Beta prior, with empirical rates marked as point estimates. Letters indicate pairwise statistical comparisons; methods sharing a letter do not differ significantly at the specified threshold. Scoring criteria, statistical analysis, and initialization details are provided in Appendix B.

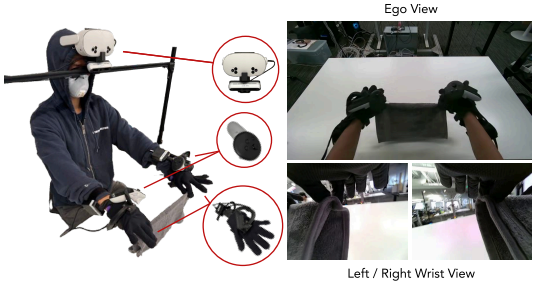

Figure 4: **Human Data Collection Interface.** Left: wearable motion-capture setup. Right: RGB observations from the egocentric and left/right wrist cameras.

4.2 DO SYNTHESIZED ACTIONS TRANSFER BEYOND THE TRAINING CORPUS?

We first ask whether an action-synthesis model learned from a separate multitask teleoperation corpus can synthesize useful action labels for human demonstrations beyond its training distribution. We evaluate this in two ways: (1) directly replaying synthesized actions on the robot, and (2) training behavior-cloning (BC) policies on human demonstrations labeled by GAS. None of the evaluation tasks appear in the teleoperation corpus used to train GAS, and the action-synthesis model is applied zero-shot, without target-task adaptation.

Action synthesis replay. Given a human glove demonstration of an unseen task, we replay it on the robot using retargeted arm commands together with hand commands produced by each labeling method. Every replay condition uses the same arm commands, differing only in hand commands. We evaluate seven tasks: mug-rim grasping, mug-top grasping, towel folding, cap twisting, test tube insertion, USB unplugging, and Rubik's-cube manipulation. These tasks differ in grasp configuration, contact regime, and finger-motion pattern. Replay directly evaluates the synthesized actions independently of downstream policy learning.

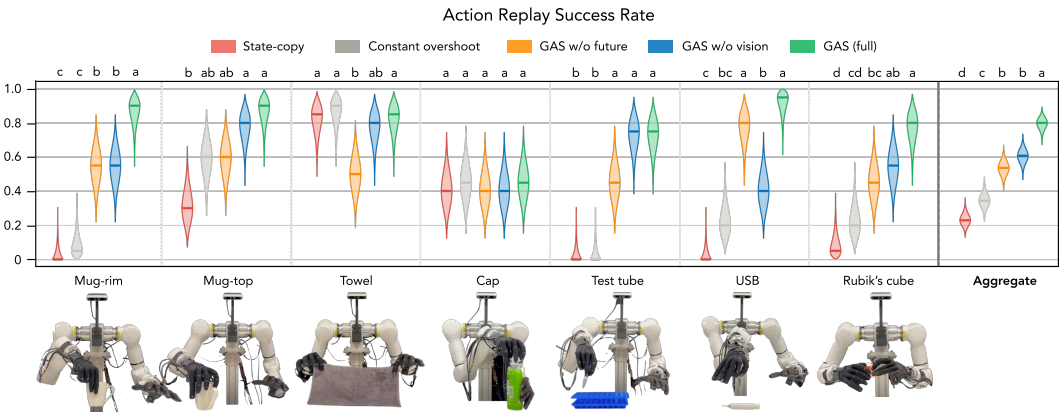

Figure 5: **Zero-shot action replay on seven unseen tasks.** Glove demonstrations are replayed open-loop with identical retargeted arm commands; conditions differ only in hand commands. The violin plots show Beta posteriors over success rate from 20 trials per task and condition. Letters denote compact letter displays from pairwise Fisher's exact tests ($\alpha = 0.05$); shared letters indicate no significant difference. GAS (full) achieves the highest aggregate success, with the largest gains on contact-sensitive tasks.

We compare three labeling strategies. *State-copy* uses the retargeted hand states directly as action targets. *Constant overshoot* adds a fixed offset to each retargeted joint state, computed as that joint's mean command–state residual $a_t - s_t$ in Eq. 1 over all frames in the GAS training corpus. *GAS (full)* synthesizes interaction-dependent commands from visual and temporal kinematic context. We also ablate two sources of context. *GAS w/o vision* removes RGB observations while retaining the same temporal hand-state context. *GAS w/o future* removes future observations while retaining past and current observations, testing whether offline access to subsequent observations improves synthesis. All action-synthesis variants are trained on the same source teleoperation corpus. Figure 6 illustrates the hand commands produced by these methods on teleoperation and glove demonstrations, with recorded teleoperation commands shown as a reference in the left panel.

Replay results. GAS improves average replay success over State-copy by 57.1 percentage points (Fig. 5). Constant overshoot improves performance on some tasks but does not consistently match the full model, indicating that the required command–state correction cannot be captured by a single task-independent offset. The largest gains occur on tasks requiring sustained or precise contact, such as mug grasping and test tube insertion, where reproducing the observed hand configuration alone is often insufficient to maintain interaction forces.

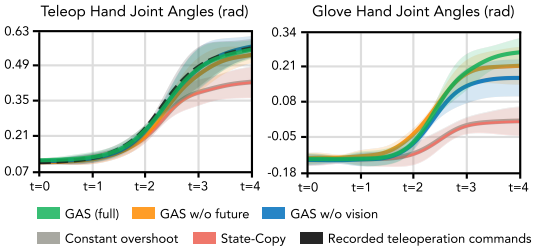

Figure 6: **Action synthesis on teleoperation (left) and glove demonstrations (right).** Joint commands during mug-top grasping. Synthesized commands reproduce teleoperation overshoot and introduce overshoot relative to State-copy on glove demonstrations. Curves show mean and standard deviation over five randomly selected evaluation episodes aligned at grasp onset.

Context ablation. Removing RGB observations reduces performance on tasks such as USB unplugging, indicating that hand states alone can provide insufficient information about the hand–object interaction. Removing future observations has a larger effect than removing vision on mug-top grasping, towel folding, and Rubik's cube manipulation, suggesting that subsequent motion provides additional cues for synthesizing suitable commands. Combining visual observations with past and future joint states yields the highest average replay success.

Downstream policy learning. We then fine-tune BC policies on human glove demonstrations for five real-world dexterous manipulation tasks: mug-rim grasping, mug-top grasping, towel folding,

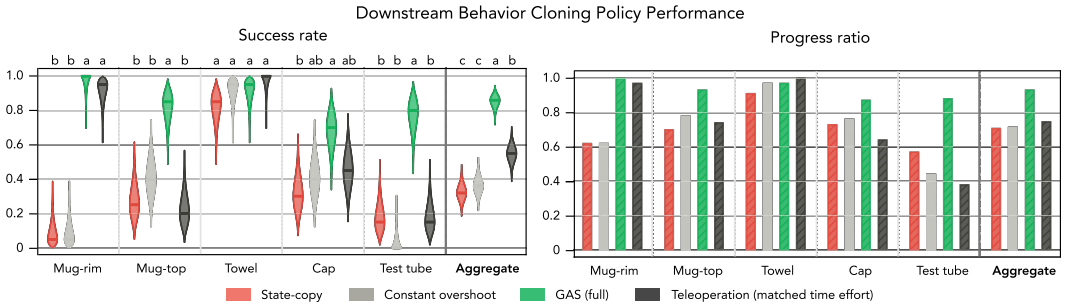

Figure 7: **Downstream policy performance on five tasks unseen by GAS.** BC policies are trained on the same glove demonstrations with different hand-action labels, and compared with policies trained on task-specific teleoperation demonstrations collected under approximately matched collection time (Sec. 4.3).

cap twisting, and test tube insertion, using the setup in Sec. 3.3. We compare policies trained with State-copy, Constant overshoot, and GAS (full) labels. These conditions use the same glove demonstrations and differ only in their hand-action labels. Figure 7 also includes a teleoperation baseline, whose collection protocol and comparison are described in Sec. 4.3. All conditions use the same pretrained initialization, policy architecture, observation modalities, and training protocol.

Policy results. Policies trained with GAS labels improve average task success over State-copy by 54 percentage points (Fig. 7). The largest improvements occur on mug-rim grasping, mug-top grasping, and test tube insertion, with a smaller gain on cap twisting and little difference on towel folding.

Qualitative behavior. State-copy policies often reach the object and establish contact but fail to lift it (Fig. 8), indicating that reproducing grasp geometry does not guarantee a sustained grasp. During cap twisting, State-copy policies reproduce the twisting motion but require more cycles and frequently time out. Policies trained with synthesized labels instead use smaller twisting sweeps, allowing more frequent finger contact with the cap. The smaller improvement on towel folding may be due to towel deformability, which reduces sensitivity to precise joint-position commands during grasping.

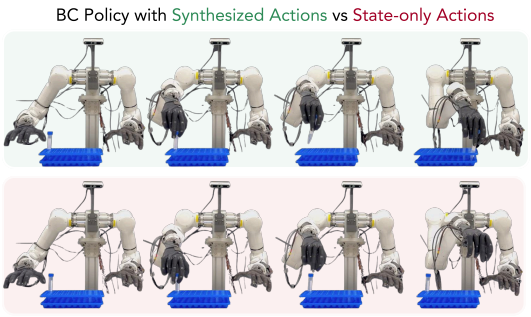

Figure 8: **Top**: The policy trained with synthesized action labels grasps the test tube, lifts it, and inserts it into the target slot. **Bottom**: The policy trained with State-copy labels approaches the test tube but fails to maintain a stable grasp, leaving the tube outside the target slot.

4.3 Can Glove Demonstrations Match Teleoperation for Policy Learning?

We next ask whether policies trained on glove demonstrations labeled by GAS can match policies trained on task-specific robot teleoperation demonstrations. We evaluate this in two ways: (1) comparing the number of demonstrations each interface yields under approximately matched collection time, and (2) comparing the downstream performance of BC policies trained on each data source.

Data collection and policy training. For each of the five tasks, we collect robot teleoperation and glove demonstrations under approximately matched wall-clock collection budgets. These budgets include task familiarization, time spent waiting for control responses, and failed collection attempts; we exclude the fixed overhead of teleoperation system calibration and setup. Only successful episodes are retained for policy training.

Using these demonstrations, we fine-tune one behavior cloning policy on task-specific teleoperation demonstrations and another on glove demonstrations labeled by GAS (full). Both policies use the architecture, pretrained initialization, observation modalities, and training protocol in Sec. 3.3; the source and number of training demonstrations differ.

Results. Under approximately matched collection time, glove collection yields 2.3–3.6 times as many successful demonstrations as teleoperation (Table 4, Appendix D). This trend is consistent with prior reports of higher collection throughput for wearable and handheld interfaces than for robot teleoperation (Wang et al., 2024; Chi et al., 2024; Tao et al., 2025).

Policies trained on labeled glove demonstrations achieve higher average success than the teleoperation baseline. Across the five tasks, policies trained with GAS labels achieve an average success rate of 86%, compared with 55% for teleoperation policies (Fig. 7). The smaller teleoperation datasets may provide less coverage of task-level behavior, although this comparison does not isolate dataset size from other differences between the demonstration sources. In our teleoperation setup, operators receive no direct haptic feedback and must coordinate the robot fingers through the control interface, which can make demonstration collection more difficult. Glove collection allows operators to manipulate objects directly, but does not record the corresponding robot commands. GAS supplies these missing action labels, allowing the larger glove datasets to support downstream policy learning.

5 Limitations and Future Work

While GAS synthesizes useful actions across a range of unseen tasks, a few limitations remain.

First, synthesized-action quality depends on coverage of the teleoperation corpus, and commands outside the training distribution may be difficult to synthesize. Scaling the source corpus to more diverse dexterous interactions may improve robustness and coverage. Additionally, although our generative formulation models potentially multimodal command distributions, our current inference averages multiple samples, and we do not isolate its benefit from conditional residual prediction itself. Future work should compare deterministic predictors and investigate how best to exploit multiple action hypotheses.

Second, GAS synthesizes joint-position commands rather than explicit force targets. Although the command–state residual implicitly captures how firmly the fingers press, the resulting force depends on the hand's low-level controller, and similar motions may require different contact forces. Future works could incorporate tactile or force sensing, either as additional conditioning or as supervision in the source data, to synthesize commands that meet specific force requirements.

Finally, in this work we did not extend GAS across robot embodiments and controllers. A particularly exciting future follow-up could be to apply it to hand poses estimated from egocentric human video, which would remove the need for wearable hardware and enable learning from even larger sources of human demonstrations.

6 Conclusion

We presented GAS, a framework for synthesizing robot hand commands for human kinematic demonstrations. GAS learns the residual between commanded and measured joint positions from multitask teleoperation data, using past and future RGB images and joint states as context. We apply the trained model zero-shot to glove demonstrations of tasks outside of training corpus. Across seven action-replay tasks and five downstream policy-learning tasks excluded from action-synthesis training, synthesized actions support robot execution and improve average policy success from 32% to 86% compared with State-copy labels, while achieving higher average success than policies trained on task-specific teleoperation demonstrations under approximately matched collection time. These findings highlight the value of teleoperation as supervision for a transferable state-to-command relationship, enabling human demonstrations to support contact-rich manipulation without target-task teleoperation.

AI USE STATEMENT

We used Claude Code, ChatGPT, and Gemini to assist with manuscript polishing, literature review, code development and debugging, and improving the clarity and presentation of the paper. All AI-generated outputs were reviewed and verified by the authors.

REPRODUCIBILITY STATEMENT

The model training and retargeting procedures are fully specified in Sec. 3.1–3.3 and Appendices A, C, and E, including architectures, hyperparameters, random seeds, and data splits. The evaluation protocol, stage rubrics, and statistical analysis are described in Appendix B, with dataset collection and effort details provided in Appendix D.

REFERENCES

Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24639–24654, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9c7008aff45b5d8f0973b23e1a22ada0-Abstract-Conference.html.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023.

Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.045. URL https://www.roboticsproceedings.org/rss20/p045.html.

Hojung Choi, Yifan Hou, Chuer Pan, Seongheon Hong, Austin Patel, Xiaomeng Xu, Mark R. Cutkosky, and Shuran Song. In-the-Wild Compliant Manipulation with UMI-FT, 2026. URL https://arxiv.org/abs/2601.09988.

Yunhai Feng, Natalie Leung, Jiaxuan Wang, Lujie Yang, Haozhi Qi, and Preston Culbertson. REGRIND: A minimalist retargeting-guided reinforcement learning recipe for dexterous manipulation. In *Proceedings of the Conference on Robot Learning*, 2026.

Liang Heng, Haoran Geng, Kaifeng Zhang, Pieter Abbeel, and Jitendra Malik. ViTacFormer: Learning cross-modal representation for visuo-tactile dexterous manipulation. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.129.

Elvis Hsieh, Wen-Han Hsieh, Yen-Jen Wang, Toru Lin, Jitendra Malik, Koushil Sreenath, and Haozhi Qi. Learning dexterous manipulation skills from imperfect simulations. In *International Conference on Robotics & Automation (ICRA)*, 2026.

Kailin Li, Puhao Li, Tengyu Liu, Yuyang Li, and Siyuan Huang. ManipTrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://maniptrans.github.io/.

Changyi Lin and Ding Zhao. ART-Glove: Articulated Tactile Glove for Contact-Grounded Dexterous Interaction Capture, 2026. URL https://arxiv.org/abs/2606.16370.

Toru Lin, Zhao-Heng Yin, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Twisting lids off with two hands. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 5220–5235. PMLR, 2025a. URL https://proceedings.mlr.press/v270/lin25c.html.

Toru Lin, Yu Zhang, Qiyang Li, Haozhi Qi, Brent Yi, Sergey Levine, and Jitendra Malik. Learning visuotactile skills with two multifingered hands. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5637–5643, 2025b. doi: 10.1109/ICRA55743.2025.11128180. URL https://ieeexplore.ieee.org/document/11128180.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

Fangchen Liu, Chuanyu Li, Yihua Qin, Jing Xu, Pieter Abbeel, and Rui Chen. ViTaMIn: Learning Contact-Rich Tasks Through Robot-Free Visuo-Tactile Manipulation Interface, 2025. URL https://arxiv.org/abs/2504.06156.

Zhao Mandi, Yifan Hou, Dieter Fox, Yashraj Narang, Ajay Mandlekar, and Shuran Song. DexMachina: Functional Retargeting for Bimanual Dexterous Manipulation, 2025. URL https://arxiv.org/abs/2505.24853.

Chaoyi Pan, Changhao Wang, Haozhi Qi, Zixi Liu, Homanga Bharadhwaj, Akash Sharma, Tingfan Wu, Guanya Shi, Jitendra Malik, and Francois Hogan. SPIDER: Scalable physics-informed dexterous retargeting, 2025. URL https://arxiv.org/abs/2511.09484.

Jiahe Pan, Stelian Coros, Jitendra Malik, and Toru Lin. Beyond binary: Sim-to-real dexterous manipulation with physics-grounded contact representation. *arXiv preprint arXiv:2605.28812*, 2026.

Yuzhe Qin, Binghao Huang, Zhao-Heng Yin, Hao Su, and Xiaolong Wang. DexPoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 594–605. PMLR, 2023. URL https://proceedings.mlr.press/v205/qin23a.html.

Tony Tao, Mohan Kumar Srirama, Jason Jingzhou Liu, Kenneth Shaw, and Deepak Pathak. DexWild: Dexterous human interactions for in-the-wild robot policies. In *Robotics: Science and Systems*, 2025.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, pp. 4950–4957, 2018. doi: 10.24963/ijcai.2018/687. URL https://doi.org/10.24963/ijcai.2018/687.

TRI LBM Team, Jose Barreiros, Andrew Beaulieu, Aditya Bhat, Rick Cory, Eric Cousineau, Hongkai Dai, Ching-Hsin Fang, Kunimatsu Hashimoto, Muhammad Zubair Irshad, Masha Itkina, Naveen Kuppuswamy, Kuan-Hui Lee, Katherine Liu, Dale McConachie, Ian McMahon, Haruki Nishimura, Calder Phillips-Grafflin, Charles Richter, Paarth Shah, Krishnan Srinivasan, Blake Wulfe, Chen Xu, Mengchao Zhang, Alex Alspach, Maya Angeles, Kushal Arora, Vitor Campagnolo Guizilini, Alejandro Castro, Dian Chen, Ting-Sheng Chu, Sam Creasey, Sean Curtis, Richard Denitto, Emma Dixon, Eric Dusel, Matthew Ferreira, Aimee Goncalves, Grant Gould, Damrong Guoy, Swati Gupta, Xuchen Han, Kyle Hatch, Brendan Hathaway, Allison Henry, Hillel Hochsztein, Phoebe Horgan, Shun Iwase, Donovon Jackson, Siddharth Karamcheti, Sedrick Keh, Joseph Masterjohn, Jean Mercat, Patrick Miller, Paul Mitiguy, Tony Nguyen, Jeremy Nimmer, Yuki Noguchi, Reko Ong, Aykut Onol, Owen Pfannenstiehl, Richard Poyner, Leticia Priebe Mendes Rocha, Gordon Richardson, Christopher Rodriguez, Derick Seale, Michael Sherman, Mariah Smith-Jones, David Tago, Pavel Tokmakov, Matthew Tran, Basile Van Hoorick, Igor Vasiljevic, Sergey Zakharov, Mark Zolotas, Rares Ambrus, Kerri Fetzer-Borelli, Benjamin Burchfiel, Hadas Kress-Gazit, Siyuan Feng, Stacie Ford, and Russ Tedrake. A careful examination of large behavior models for multitask dexterous manipulation, 2025. URL https://arxiv.org/abs/2507.05331.

Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C. Karen Liu. DexCap: Scalable and portable mocap data collection system for dexterous manipulation. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.043. URL https://www.roboticsproceedings.org/rss20/p043.html.

Mengda Xu, Han Zhang, Yifan Hou, Zhenjia Xu, Linxi Fan, Manuela Veloso, and Shuran Song. DexUMI: Using human hand as the universal manipulation interface for dexterous manipulation. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 437–459. PMLR, 2025. URL https://proceedings.mlr.press/v305/xu25b.html.

Jessica Yin, Haozhi Qi, Youngsun Wi, Sayantan Kundu, Mike Lambeta, William Yang, Changhao Wang, Tingfan Wu, Jitendra Malik, and Tess Hellebrekers. OSMO: Open-Source Tactile Glove for Human-to-Robot Skill Transfer, 2025. URL https://arxiv.org/abs/2512.08920.

Han Zhang, Songbo Hu, Zhecheng Yuan, and Huazhe Xu. DOGlove: Dexterous Manipulation with a Low-Cost Open-Source Haptic Force Feedback Glove, 2025. URL https://arxiv.org/abs/2502.07730.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.016.

Xinghao Zhu, Zixi Liu, Shalin Jain, Chenran Li, Milad Noori, Michael Andres Lin, Huihua Zhao, John Welsh, Mrinal Verghese, Wei Liu, Tingwu Wang, Xingye Da, Zhengyi Luo, Vishal Kulkarni, Naema Bhatti, Yuke Zhu, Linxi Fan, Bowen Wen, Danfei Xu, Soha Pouya, and Yan Chang. Learning Dexterous Manipulation Using Contact Wrench Guidance From Human Demonstration, 2026. URL https://arxiv.org/abs/2607.00033.

A Behavior Cloning Policy Training Details

Policy architecture. The policy receives three current-frame RGB images at 256×256 resolution, the robot state, and a task instruction. The 52-dimensional state contains both end-effector positions (6 dimensions), axis-angle orientations (6 dimensions), and hand joint angles (40 dimensions). Actions specify absolute targets in the same representation, predicted in 32-step chunks at a demonstration frequency of 30 Hz. States and actions are normalized per dimension to $[-1, 1]$ using training-set statistics.

We use a DINOv3 ViT-B/16 vision encoder and a conditional flow-matching transformer with 8 blocks, hidden dimension 1536, and 32 attention heads. Conditioning combines visual features, proprioception, and a 384-dimensional task embedding from a frozen `all-MiniLM-L6-v2` sentence encoder. The task instruction is fixed within each single-task dataset. The vision encoder and action model are fine-tuned jointly.

Initialization and fine-tuning. All policies are initialized from the same policy pretrained on 3,381 teleoperation episodes spanning primarily pick-and-place tasks, with additional manipulation behaviors including folding and cap removal. We fine-tune each policy for 25,000 steps and evaluate the final exponential moving average (EMA) checkpoint, without validation-based checkpoint selection. Each downstream dataset reserves 50 episodes for validation using split seed 42. Label variants of the same glove dataset share identical splits; teleoperation baselines use separate splits of their own datasets. All comparisons use the same initialization and training settings.

Training objective and optimization. We train with conditional flow matching over action chunks. Per-dimension losses for end-effector positions, orientations, and hand joints are weighted by 1.0, 1.0, and 0.15, respectively, giving the three groups equal total dimensional weight. Optimization settings are listed in Table 1. Each run takes approximately 10–12 hours on eight NVIDIA H100 GPUs.

Inference and deployment. We generate action chunks from standard Gaussian noise using 10 forward Euler steps with $\Delta t = 0.1$, clipping to $[-1, 1]$ after each update before converting the final predictions back to absolute commands. The policy generates chunks asynchronously at 2 Hz while commands execute at 30 Hz. New chunks are blended into the executing trajectory to smooth transitions.

Table 1: Behavior-cloning fine-tuning settings.

Parameter	Value
Optimizer	AdamW
Learning rate	10^{-4}
AdamW (β_1, β_2)	$(0.95, 0.999)$
Weight decay	10^{-4}
Learning-rate schedule	Cosine decay
Warmup steps	500
Minimum learning rate	10^{-5}
Gradient clipping	Maximum norm 1.0
Global batch size	512 (64 per GPU)
Training seed	2
EMA decay ramp power	0.75
Maximum EMA decay	0.9999

B Real-World Evaluation Protocol

Metrics and scoring. For downstream policy evaluation, each rollout receives an ordered stage score $s_i \in \{0, \dots, K\}$ with $K = 5$, where a stage counts only if all preceding stages are completed. Over $N = 20$ trials,

$$\text{Progress} = \frac{1}{NK} \sum_{i=1}^{N} s_i, \qquad \text{Success} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}[s_i = K]. \tag{5}$$

Table 2 defines the task-specific stages.

For action replay, we report task success. Shared tasks use the same stage-5 criterion as downstream evaluation. Rubik's-cube rotation succeeds if the robot picks up the cube, rotates it, and places it back on the table; USB unplugging succeeds if the USB connector is fully removed from its port.

Statistical analysis. To account for uncertainty from the finite number of real-world rollouts, we model binary task success as a Bernoulli random variable with unknown success probability p. Following TRI LBM Team et al. (2025), we place a uniform prior over p:

$$p \sim \text{Beta}(1, 1). \tag{6}$$

After observing S successful and F failed rollouts, the posterior is

$$p \mid S, F \sim \text{Beta}(S + 1, F + 1). \tag{7}$$

The violin plots in Figures 7 and 5 show these posterior densities. The reported point estimate is the empirical success rate $S/(S + F)$.

We compare methods using two-sided Fisher's exact tests on pairwise success/failure counts, using per-task counts for individual tasks and pooled counts for the aggregate comparison. Tests use $\alpha = 0.05$ without multiple-comparison correction. Although evaluations use matched initial configurations, these tests treat the samples as independent and do not account for pairing. Results are summarized with a compact letter display (CLD): methods sharing a letter have no statistically significant difference under these tests, while methods with no letter in common differ significantly. A nonsignificant comparison does not establish equivalent performance.

Initialization protocol. For each task, we randomly select 20 held-out glove trajectories that are excluded from BC training. Each trial begins from the robot initialization pose recorded in the first frame of the corresponding held-out trajectory, and this pose is identical across all conditions for that trajectory. We overlay the first egocentric-camera frame of the reference trajectory on the live camera view and manually align the object with its reference position. All conditions use the same 20 reference trajectories and object-alignment procedure.

Table 2: Ordered stages for downstream policy evaluation. Each trial receives the highest stage whose preceding stages are all complete. A score of 0 indicates that the first stage was not reached; success requires stage 5.

Stage	Test tube Insertion	Cap Twisting
1	Test tube reached	Bottle reached
2	Grasp pose reached	Bottle grasped with the left hand
3	Test tube lifted	Initial rotations executed
4	Test tube transported to the block	Rotation sustained
5	Test tube inserted into the block	All rotations completed
Stage	**Towel Folding**	**Mug-top / Mug-rim Grasping**
1	Towel reached	Mug reached
2	One edge grasped	Grasp pose reached
3	Both edges grasped	Contact established
4	Folding motion executed	Mug lifted off the table
5	Fold completed	Lift sustained

C Generative Action-Synthesis Training Details

Observations and targets. The model jointly processes the 40 joint angles of both hands. For each query frame t, it uses 16 state frames ending at t and 4 future state frames, together with RGB observations at $t-4$, t, and $t+4$ from the egocentric and two wrist cameras. The target is a four-frame command–state residual chunk, corresponding to approximately 0.13 s at 30 Hz. We train on standardized residuals using the flow-matching objective in Eq. 4, with equal weighting across residual dimensions.

Model architecture. Images are encoded at 224×224 resolution using a DINOv3 ViT-S/16 backbone shared across cameras and fine-tuned jointly with the model. Image features are pooled to 16 tokens per image, and each joint-state frame is represented by one token. All tokens have dimension 768 and receive learned modality and relative-time embeddings; image tokens additionally receive camera and patch-position embeddings. An 8-layer Transformer encoder with 12 attention heads, feed-forward dimension 3072, and dropout 0.1 processes the tokens. Its output is mean-pooled into a 768-dimensional conditioning vector.

The flow-matching velocity model is an MLP with three hidden layers of width 1536 and SiLU activations. It receives the noisy residual chunk, a flow-time embedding, and the conditioning vector, and predicts a 160-dimensional velocity. The complete model contains 86.8 million parameters.

Training data and optimization. GAS is trained on a multitask bimanual teleoperation corpus collected across multiple robot stations sharing the same hardware configuration as our evaluation setup. Each episode contains egocentric and wrist cameras, measured joint states and commanded joint targets for both hands, and arm end-effector poses at 30 Hz. The corpus contains 11,564 episodes (77.1 h) across 31 tasks in six families: pick-and-place (24% of hours), hanging objects on racks (23%), insertion and transfer (21%), pouring (16%), multi-object sorting (11%), and button or key pressing (5%). About half of the episodes are bimanual. We train on this natural mixture without task re-weighting. The corpus is task-disjoint from all evaluation tasks.

We train for 10 epochs on eight NVIDIA H100 GPUs, taking approximately 15.4 hours. The vision backbone uses a learning rate 100 times lower than the rest of the model. All reported synthesis results use the final exponential moving average (EMA) checkpoint, without validation-based checkpoint selection. Table 3 summarizes the optimization settings.

Offline synthesis. At each frame, we sample four residual chunks from Gaussian noise using 25 forward Euler steps with $\Delta\lambda = 0.04$. We average these samples within each chunk, then uniformly average predictions from overlapping chunks for each frame. The resulting residual is converted

Table 3: Action-synthesis model training settings.

Parameter	Value
Optimizer	AdamW
Learning rate (head / vision backbone)	10^{-4} / 10^{-6}
AdamW (β_1, β_2)	$(0.9, 0.999)$
Weight decay	10^{-4}
Learning-rate schedule	Linear warmup + cosine decay
Warmup steps	2,000
Minimum learning rate	$0.1\times$ base rate
Gradient clipping	Maximum norm 1.0
Global batch size	256 (32 per GPU)
Training steps	306,850 (10 epochs)
Mixed precision	bfloat16
Training seed	0
EMA decay	0.9995, from step 0

Table 4: Datasets collected under approximately matched wall-clock collection time per task. Episodes denote successful demonstrations; minutes denote their total recorded duration, not elapsed collection time.

	Teleoperation		Glove	
Task	Episodes	Minutes	Episodes	Minutes
Mug-rim grasping	150	10.3	450	17.8
Mug-top grasping	301	25.0	691	44.1
Towel folding	302	23.6	1,028	89.3
Cap twisting	300	105.0	1,081	249.7
Test tube insertion	300	41.1	981	123.7

back to joint-angle units and added to the retargeted hand state. Missing history or future context at episode boundaries is filled by repeating the first or last frame.

D Teleoperation and Glove Data Collection

Table 4 summarizes the collected datasets. Under approximately matched collection time, glove collection yields 2.3–3.6 times as many successful demonstrations as teleoperation. The collection advantage is larger for towel folding, cap twisting, and test tube insertion than for mug grasping. These tasks require bimanual coordination or precise finger control, which may be more difficult through our teleoperation interface, where operators control the robot hands using MANUS motion-capture gloves without direct haptic feedback.

E Glove-to-Robot Hand Retargeting

The glove records five fingertip poses per hand, each consisting of a position $p_{t,f} \in \mathbb{R}^3$ and an orientation $R_{t,f} \in SO(3)$ in the palm frame. We obtain the retargeted hand state $\tilde{s}_t \in \mathbb{R}^{20}$ by solving inverse kinematics (IK) on the robot hand model. All glove demonstrations use the same retargeter for action synthesis, behavior cloning, and open-loop replay. Arm trajectories use the recorded wrist poses.

Fingertip IK. For each frame and hand, we solve

$$\tilde{s}_t = \arg\min_{q_{\min} \le q \le q_{\max}} w_p\, \mathcal{L}_{\mathrm{pos}}(q) + w_r\, \mathcal{L}_{\mathrm{rot}}(q) + w_u\, \mathcal{L}_{\mathrm{joint}}(q) + w_\tau \|q - \tilde{s}_{t-1}\|_2^2, \tag{8}$$

with

$$\mathcal{L}_{\mathrm{pos}}(q) = \textstyle\sum_f \|\mathrm{FK}_f^p(q) - \hat{p}_{t,f}\|_2^2, \qquad \mathcal{L}_{\mathrm{rot}}(q) = \textstyle\sum_f \theta\big(\mathrm{FK}_f^R(q), \hat{R}_{t,f}\big)^2,$$

$$\mathcal{L}_{\mathrm{joint}}(q) = \textstyle\sum_f \big(q_{f,\mathrm{pip}} - q_{f,\mathrm{dip}}\big)^2, \qquad \mathcal{L}_{\mathrm{smooth}}(q) = \|q - \tilde{s}_{t-1}\|_2^2.$$

Here, FK_f^p and FK_f^R denote robot fingertip position and orientation, θ is the geodesic angle between rotations, and $(\hat{p}_{t,f}, \hat{R}_{t,f})$ are the target fingertip poses derived from the glove measurements. The joint loss couples each finger's PIP and DIP joints, while the smoothness loss penalizes changes between consecutive configurations. We use $w_p = 1$, $w_r = 10^{-5}$, $w_u = 10^{-2}$, and $w_\tau = 10^{-4}$. We solve with IPOPT, warm-starting each frame from the previous solution. The little-finger target is shifted by 2 cm along its local axis to account for the robot's longer little finger.

We assign a small orientation weight because human fingertip positions and orientations may not be simultaneously reachable by the robot hand. In our setup, increasing w_r to 10^{-3} shifts the thumb and index fingertips by approximately 8–9 mm and loosens precision grasps. We therefore use $w_r = 10^{-5}$ to prioritize fingertip positions while retaining orientation guidance.