

Learning External Force Sensing for Commodity Robot Arms Improves Policy Learning

Steven Oh^{*1,2} Jason Jingzhou Liu^{*†1} Tony Tao^{*1}
 Philip Han¹ Kenneth Shaw¹ Satoshi Funabashi²
 Ruslan Salakhutdinov¹ Deepak Pathak¹
¹Carnegie Mellon University ²Waseda University
 * Equal contribution

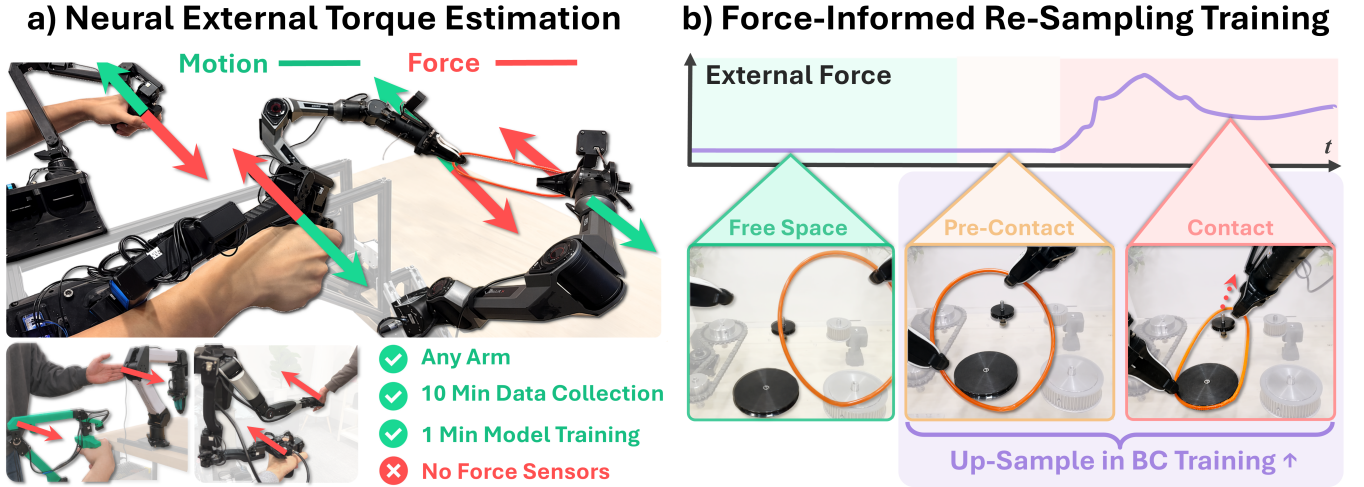


Fig. 1. **Overview.** (a) Neural External Torque Estimation (NEXT) learns external joint torque from 10 minutes of free-space data without dedicated force sensors, enabling force-feedback teleoperation on commodity arms. (b) Force-Informed Re-Sampling Training (FIRST) uses the learned signal to identify free-space, pre-contact, and contact phases and up-samples contact-relevant data during behavior cloning.

Abstract—Contact-rich manipulation requires force sensitivity, yet many commodity robot arms lack dedicated force sensors. We present Neural External Torque Estimation (NEXT), which learns a free-space inverse-dynamics model from only 10 minutes of contact-free motion and estimates external joint torque as the residual between measured and predicted motor torque. NEXT trains in approximately 1 minute, matches dedicated torque sensing on a Franka arm, and enables force-feedback teleoperation on low-cost arms. We further introduce Force-Informed Re-Sampling Training (FIRST), which uses NEXT to identify pre-contact and contact segments and sample them more frequently during behavior cloning. Across five long-horizon contact-rich tasks, FIRST improves average task progress by more than 17% over prior force-aware baselines. Together, NEXT and FIRST provide a practical path to force-aware data collection and policy learning without additional sensing hardware.

I. INTRODUCTION

Force feedback is central to insertion, fastening, and deformable-object manipulation, but high-quality force-torque sensing is commonly restricted to expensive robot platforms. Retrofitting an arm with an end-effector sensor adds cost, bulk, and integration complexity, while low-cost tactile alternatives are often designed for smaller local contact loads rather than full-arm interaction forces [3, 5, 9, 4]. As a result, many

commodity arms expose motor current but provide no reliable external-force estimate.

We introduce Neural External Torque Estimation (NEXT), which learns the torque required for contact-free motion and subtracts it from measured motor torque at deployment. Unlike analytical inverse dynamics or disturbance observers [12, 29], NEXT requires neither dedicated force supervision nor robot-specific system identification. The resulting signal supports force-feedback teleoperation and serves as a compact contact representation for policy learning.

We then propose Force-Informed Re-Sampling Training (FIRST). Failures in contact-rich imitation learning often concentrate in short pre-contact intervals, where precise alignment is required, and during contact, where small corrective actions determine success. FIRST uses NEXT to segment demonstrations into free-space, pre-contact, and contact phases, then modifies the batch-sampling distribution to emphasize the latter two. This differs from methods that only concatenate force or current to policy observations [6, 46, 21] or predict force targets at inference time [26, 41].

Our main findings are: (i) NEXT reduces Franka contact-estimation error to 0.547 Nm and enables force feedback comparable to dedicated sensing; (ii) the same estimator

transfers to low-cost arms without force sensors; and (iii) FIRST produces the best task progress across five long-horizon tasks, with pre-contact samples providing the largest benefit. Appendices A–B provide the extended literature review and analytical background. NEXT and FIRST implementation details are organized in Appendices A and I, with additional FIRST results in Appendix Q.

II. RELATED WORK

Sensorless force estimation. Classical approaches infer contact from the residual between measured motor torque and analytical rigid-body dynamics. These methods can work well with accurate inertial and friction parameters, but model mismatch, actuator nonlinearities, backlash, torque ripple, and unobserved transmission effects contaminate the residual [20, 13, 24, 22]. Momentum and disturbance observers avoid direct acceleration estimation and improve smoothness [29], while methods such as FILIC use identified inverse dynamics [12]. Both still depend on robot-specific modeling. Learning-based alternatives have been used for contact detection, anomaly detection, or torque prediction [31, 32, 23, 43, 48], but often require contact labels, privileged force supervision, or specialized hardware. NEXT instead learns only from ordinary contact-free motion and produces a continuous external-torque signal at deployment.

Force-aware imitation learning. Prior policies condition on joint effort, joint torque, or end-effector force [46, 6, 21], use force reconstruction as an auxiliary objective [46, 14], or predict targets for hybrid force–position and impedance control [26, 12, 41]. FACTR uses force during training to encourage generalization [25], and ForceVLA introduces modality-specific experts [44]. These methods establish that force information can improve contact-rich behavior, but generally assume a force-related signal is already available and use each demonstration sample with the original frequency. FIRST addresses both issues: NEXT makes external torque available on commodity arms, and FIRST uses it to reshape which parts of demonstrations are emphasized during optimization.

Dataset reweighting. Advantage-weighted reinforcement-learning methods prioritize high-value actions [33, 38, 30]. Recent imitation-learning methods select or reweight demonstrations using group optimization, data-model estimates, mutual information, learned progress, reward models, rollout success, or influence functions [40, 15, 10, 16, 45, 8, 7, 1]. In contrast, FIRST requires neither rewards nor policy rollouts. It uses a physically grounded signal already produced by NEXT to identify short, difficult transitions within otherwise successful demonstrations. A more complete discussion and analytical background are provided in Appendices A–B.

III. METHOD

A. Neural External Torque Estimation

Let $\tau_m = KI_m$ denote measured motor torque obtained from motor current I_m and torque constant K . NEXT learns

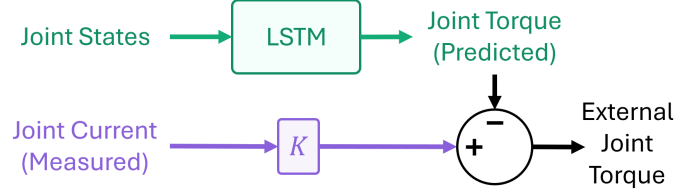


Fig. 2. NEXT deployment pipeline. Measured current is converted to motor torque, an LSTM predicts the free-space component, and subtraction yields external joint torque.

a free-space inverse-dynamics model f_θ and estimates external torque by

$$\hat{\tau}_f = f_\theta(\mathbf{x}), \quad \hat{\tau}_{\text{ext}} = \tau_m - \hat{\tau}_f. \quad (1)$$

The input $\mathbf{x}$ is a fixed window of joint position, joint velocity, and commanded-position tracking error, $\mathbf{x}_i = [\mathbf{q}_{i-H:i}, \dot{\mathbf{q}}_{i-H:i}, \Delta\mathbf{q}_{d,i-H:i}]$. Because training trajectories contain no external contact, measured motor torque directly supervises $\hat{\tau}_f$. We instantiate f_θ as a stateless sliding-window LSTM, which captures friction, hysteresis, actuator dynamics, and controller-dependent effects without accumulating hidden-state drift across long trajectories. We use $H = 50$ at 100 Hz. The full formulation, architecture, runtime, data-collection guidance, and model/input ablations are provided in Appendix A; the deployment pipeline is shown in Fig. 2.

B. Force-Informed Re-Sampling Training

FIRST first converts NEXT estimates into a scalar contact score $f_t = \|\hat{\tau}_{\text{ext},t}\|_1$. Hysteresis thresholding produces a binary contact indicator c_t : contact begins when $f_t \geq T_{\text{high}}$, ends when $f_t \leq T_{\text{low}}$, and otherwise retains its previous state. We label the one-second interval before each contact onset as Pre-Contact, active interaction as Contact, and all remaining samples as Free Space.

For phase label $s_t \in \{\text{F}, \text{PC}, \text{C}\}$, FIRST changes only the probability with which training samples are drawn:

$$p_t = \frac{w(s_t)}{\sum_{j=1}^N w(s_j)}, \quad w_{\text{PC}}, w_{\text{C}} > w_{\text{F}}. \quad (2)$$

The policy objective and architecture remain unchanged. Our primary policy is a flow-matching transformer conditioned on multi-view RGB images, robot proprioception, and NEXT torque estimates; we also evaluate ACT [47]. The full objective, segmentation definition, evaluation protocol, and policy architecture are provided in Appendix I; selected sampling ratios and additional ablations appear in Appendix Q.

IV. EXPERIMENTS AND RESULTS

A. External Torque Estimation and Teleoperation

We evaluate NEXT on held-out Franka trajectories. Contact data contain human-applied forces and use the Franka factory external-torque estimate as ground truth; free-space data use zero external torque as ground truth. We compare against FILIC [12], a momentum-based disturbance observer [29], and the Franka sensor itself in free space. Each condition contains

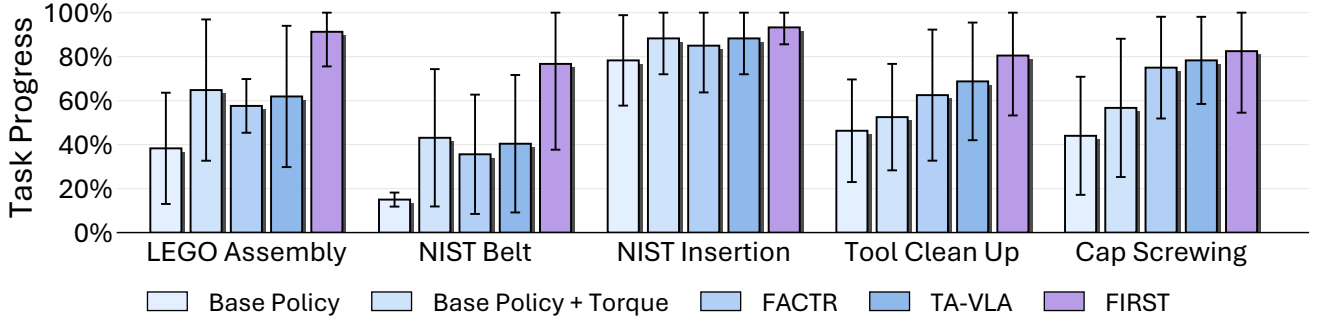


Fig. 3. Task progress across five contact-rich manipulation tasks using the flow-matching policy. FIRST achieves the strongest overall performance.

TABLE I
AVERAGE JOINT L_1 EXTERNAL-TORQUE ERROR ON FRANKA (Nm).

Motion	Method	L_1 Error
Contact	External Sensor (GT)	0.000 ± 0.000
	FILIC	4.395 ± 1.531
	Disturbance Observer	1.471 ± 0.761
	NEXT (Ours)	0.547 ± 0.348
Free Space	Free Space (GT)	0.000 ± 0.000
	FILIC	2.460 ± 1.358
	Disturbance Observer	2.429 ± 1.298
	External Sensor	0.449 ± 0.208
	NEXT (Ours)	0.414 ± 0.278

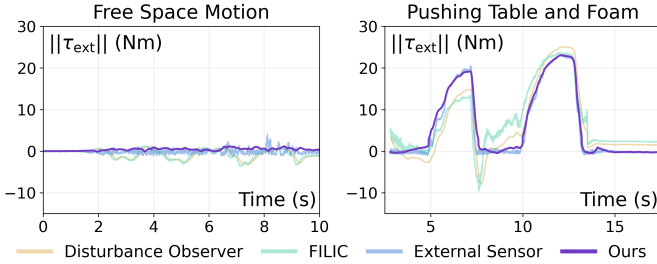


Fig. 4. External-torque traces. **Left:** NEXT remains near zero during free motion. **Right:** NEXT tracks the Franka external-torque signal during contact more closely than FILIC and the disturbance observer.

five minutes of data. Baseline implementations and additional Piper results are provided within Appendix A.

Table I shows that NEXT lowers contact error by 87.6% relative to FILIC and 62.8% relative to the disturbance observer. In free space, NEXT is also less noisy than the factory external-torque estimate. Figure 4 shows that NEXT remains near zero in free motion and follows the Franka signal through contact transitions more closely than the model-based baselines.

We next conduct a 20-participant wiping study with no feedback, disturbance-observer feedback, position–position feedback [17], FACTR Teleop using the Franka sensor [25], and the same controller using NEXT. Participants rate NEXT feedback as easier to use than the non-sensor baselines, while applied joint torque is comparable to sensor-based FACTR

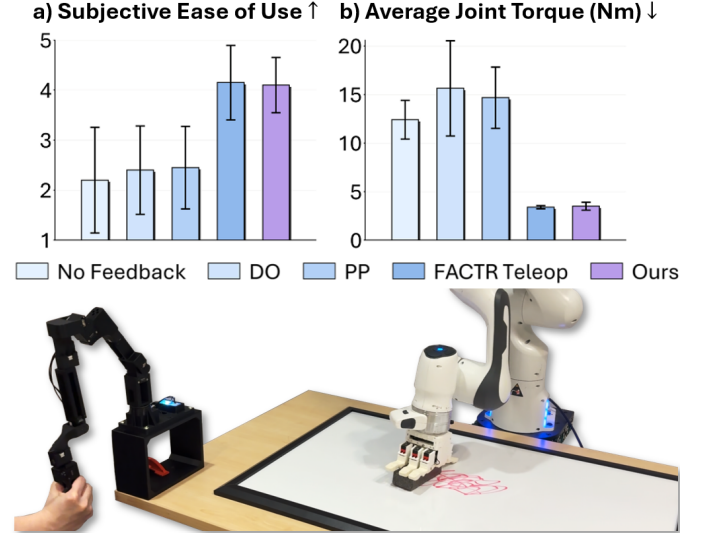


Fig. 5. Franka user study. NEXT-based force feedback improves ease of use over non-sensor baselines and requires similar applied torque to FACTR Teleop using dedicated torque sensing.

Teleop (Fig. 5). On the sensorless Piper, NEXT achieves 0.018 Nm free-space error and again improves user ratings; full protocols and results are provided within Appendix A.

B. Force-Informed Policy Learning

We evaluate five long-horizon tasks (Fig. 6) using 250 demonstrations per task on a bimanual Piper setup with two wrist and two overhead RGB cameras. Each policy is evaluated for 20 rollouts. We compare a vision/proprioception base policy, the same policy with NEXT torque, FACTR [25], TA-VLA [46], and FIRST. Task definitions, progress metrics, policy details, and baseline implementations are provided in Appendix I.

FIRST achieves the highest task progress on all five tasks (Fig. 3), with the largest gains on long-horizon tasks that require repeated transitions between alignment and sustained contact. Phase-wise validation loss is highest near contact under uniform sampling; FIRST reduces this gap by repeatedly exposing the policy to these difficult transitions (Fig. 7).

Table II shows that pre-contact up-sampling obtains the

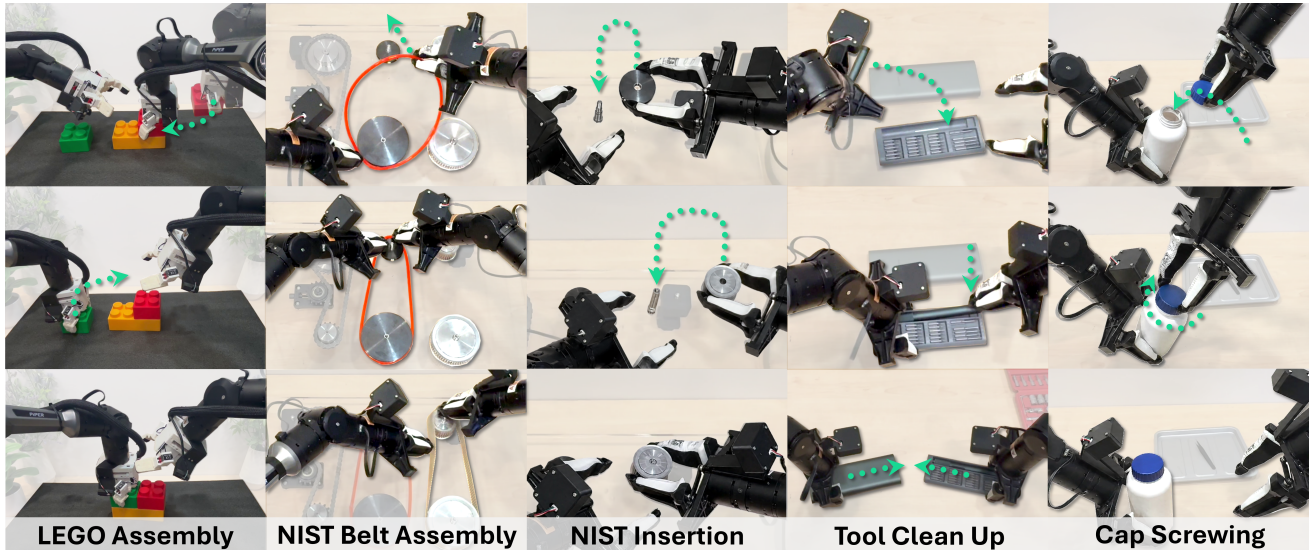


Fig. 6. Five long-horizon contact-rich evaluation tasks. Each contains multiple alignment, contact, and recovery stages.

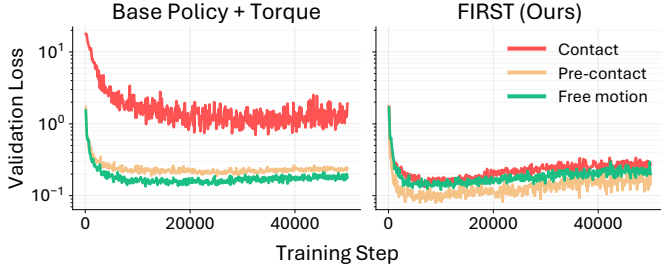


Fig. 7. Uniform sampling produces higher validation loss on pre-contact and contact phases. FIRST reduces the loss on these difficult phases.

TABLE II
PHASE UP-SAMPLING ABLATION. C: CONTACT; PC: PRE-CONTACT.

Task	C	PC	PC+C
LEGO Assembly	0.697	0.913	0.875
NIST Belt	0.494	0.767	0.649
NIST Insertion	0.800	0.933	0.933
Tool Cleanup	0.650	0.805	0.775
Cap Screwing	0.707	0.673	0.825
Average	0.670	0.818	0.811

highest average progress (0.818), narrowly exceeding combined pre-contact and contact up-sampling (0.811); contact-only up-sampling reaches 0.670. This indicates that reaching the correct pose before interaction is often more important than simply emphasizing samples after contact has begun. Sampling-magnitude and architecture ablations are reported in Appendix Q.

Qualitatively, FIRST improves the setup and recovery behaviors surrounding contact rather than merely making the policy react more strongly after contact occurs. Across insertion, belt assembly, and cap screwing, the policy more often pauses, realigns, and retries before committing to interaction,

TABLE III
POLICY PROGRESS WITH RAW CURRENT VERSUS NEXT TORQUE.

Method	Insertion	Cap Screw
ACT	0.220	0.440
ACT w/ current	0.431	0.440
ACT w/ τ_{ext}	0.567	0.525

reducing failures caused by small pose errors that would otherwise propagate through later stages. This behavior is consistent with the segmentation visualization in Appendix Fig. 10: NEXT identifies a short but information-dense interval immediately before contact, and FIRST increases how often the policy observes the corresponding fine corrective actions during training. The result suggests that learned force signals are useful not only as instantaneous feedback, but also as an automatic means of identifying where limited demonstration data should receive additional optimization effort.

The learned external-torque representation also matters. As shown in Table III, raw current improves NIST Insertion but not Cap Screwing, whereas NEXT torque performs best on both. Appendix Q provides the attention analysis, policies without force input, and the weighted-regression comparison.

V. CONCLUSION AND LIMITATIONS

We present NEXT, a data-driven method for estimating external joint torque without dedicated force sensors. With only 10 minutes of free-space data and 1 minute of training, NEXT produces estimates comparable to dedicated sensor measurements and enables force-feedback teleoperation on low-cost arms. We further introduce FIRST, which uses NEXT torque estimates to up-sample pre-contact and contact segments during behavior cloning, improving long-horizon contact-rich manipulation by emphasizing the phases most critical for alignment and interaction. NEXT has two main

limitations: its absolute torque scale depends on the accuracy of the motor torque constant K , so inaccurate manufacturer parameters may require calibration with an external force reference, and the model is currently robot-specific, requiring retraining across arms with different hardware. Future work will investigate cross-robot adaptation and online recalibration to reduce platform-specific data collection.

REFERENCES

- [1] Christopher Agia, Rohan Sinha, Jingyun Yang, Rika Antonova, Marco Pavone, Haruki Nishimura, Masha Itkina, and Jeannette Bohg. Cupid: Curating data your robot loves with influence functions. In Joseph Lim, Shuran Song, and Hae-Won Park, editors, *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 2907–2932. PMLR, 27–30 Sep 2025. URL <https://proceedings.mlr.press/v305/agia25a.html>.
- [2] AgileX Robotics. agx_arm_urdf: Agilex robot arm urdf models. https://github.com/agilexrobotics/agx_arm_urdf, 2026. Accessed: 2026-05-11.
- [3] ATI Industrial Automation. Gamma Force/Torque sensor. https://www.ati-ia.com/products/ft/ft_models.aspx?id=gamma, 2026. Accessed: April 28, 2026.
- [4] Raunaq Bhirangi, Tess Hellebrekers, Carmel Majidi, and Abhinav Gupta. Reskin: versatile, replaceable, lasting tactile skins. In *CoRL*, 2021.
- [5] Bota Systems. 6-Axis Force Torque Sensors for Robotics. <https://www.botasys.com/force-torque-sensors>, 2024. Accessed: April 28, 2026.
- [6] Thanpimon Buamanee, Masato Kobayashi, Yuki Uraishi, and Haruo Takemura. Bi-act: Bilateral control-based imitation learning via action chunking with transformer. In *2024 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 410–415, 2024. doi: 10.1109/AIM55361.2024.10637173.
- [7] Annie S. Chen, Alec M. Lessing, Yuejiang Liu, and Chelsea Finn. Curating demonstrations using online experience. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.071.
- [8] Qianzhong Chen, Justin Yu, Mac Schwager, Pieter Abbeel, Fred Shentu, and Philipp Wu. SARM: Stage-aware reward modeling for long horizon robot manipulation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=aemqAxScI9>.
- [9] Hojung Choi, Jun En Low, Tae Myung Huh, Seongheon Hong, Gabriela A Uribe, Kenneth AW Hoffmann, Julia Di, Tony G Chen, Andrew A Stanley, and Mark R Cutkosky. Coinft: A coin-sized, capacitive 6-axis force torque sensor for robotic applications. *arXiv preprint arXiv:2503.19225*, 2025.
- [10] Shivin Dass, Alaa Khaddaj, Logan Engstrom, Aleksander Madry, Andrew Ilyas, and Roberto Martín-Martín. DataMIL: Selecting data for robot imitation learning with datamodels. In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=AcTsKgIDdh>.
- [11] Claudio Gaz, Marco Cognetti, Alexander Oliva, Paolo Robuffo Giordano, and Alessandro De Luca. Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization. *IEEE Robotics and Automation Letters*, 4(4): 4147–4154, 2019. doi: 10.1109/LRA.2019.2931248.
- [12] Haizhou Ge, Yufei Jia, Zheng Li, Yue Li, Zhixing Chen, Ruqi Huang, and Guyue Zhou. Filic: Dual-loop force-guided imitation learning with impedance torque control for contact-rich manipulation tasks, 2025. URL <https://arxiv.org/abs/2509.17053>.
- [13] Sami Haddadin, Alessandro De Luca, and Alin Albu-Schäffer. Robot collisions: A survey on detection, isolation, and identification. *IEEE Transactions on Robotics*, 33(6):1292–1312, 2017.
- [14] Zihao He, Hongjie Fang, Jingjing Chen, Hao-Shu Fang, and Cewu Lu. Foar: Force-aware reactive policy for contact-rich robotic manipulation. *IEEE Robotics and Automation Letters*, 10(6):5625–5632, 2025. doi: 10.1109/LRA.2025.3560871.
- [15] Joey Hejna, Chethan Anand Bhateja, Yichen Jiang, Karl Pertsch, and Dorsa Sadigh. Remix: Optimizing data mixtures for large scale imitation learning. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 145–164. PMLR, 06–09 Nov 2025. URL <https://proceedings.mlr.press/v270/hejna25a.html>.
- [16] Joey Hejna, Suvir Mirchandani, Ashwin Balakrishna, Annie Xie, Ayzaan Wahid, Jonathan Tompson, Pannag Sanketi, Dhruv Shah, Coline Devin, and Dorsa Sadigh. Robot data curation with mutual information estimators. In *Robotics: Science and Systems (RSS)*, 2025.
- [17] Peter F. Hokayem and Mark W. Spong. Bilateral teleoperation: An historical survey. *Automatica*, 42(12):2035–2057, 2006. ISSN 0005-1098. doi: <https://doi.org/10.1016/j.automatica.2006.06.027>. URL <https://www.sciencedirect.com/science/article/pii/S0005109806002871>.
- [18] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding predictions with data and data with predictions. In *ICML*, 2022.
- [19] Koki Inami, Koki Yamane, and Sho Sakaino. Loss function considering dead zone for neural networks, 2024. URL <https://arxiv.org/abs/2402.00393>.
- [20] A. Jubien, M. Gautier, and A. Janot. Dynamic identification of the kuka lwr robot using motor torques and joint torque sensors data. In *19th IFAC World Congress*, 2014.
- [21] Tatsuya Kamijo, Cristian C. Beltran-Hernandez, and Masashi Hamaya. Learning variable compliance control from a few demonstrations for bimanual robot with haptic feedback teleoperation system. In *IEEE/RSJ Interna-*

- tional Conference on Intelligent Robots and Systems (IROS)*, 2024.
- [22] Nenad M. Kircanski and Andrew A. Goldenberg. An experimental study of nonlinear stiffness, hysteresis, and friction effects in robot joints with harmonic drives and torque sensors. *The International Journal of Robotics Research*, 16(2):214–239, 1997.
 - [23] Jacky Liang and Oliver Kroemer. Contact localization for robot arms in motion without torque sensing. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6322–6328. IEEE, 2021.
 - [24] Magnus Linderöth, Andreas Stolt, Anders Robertsson, and Rolf Johansson. Robotic force estimation using motor torques and modeling of low velocity friction disturbances. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3550–3556, 2013.
 - [25] Jason Jingzhou Liu, Yulong Li, Kenneth Shaw, Tony Tao, Russ Salakhutdinov, and Deepak Pathak. FACTR: Force-Attending Curriculum Training for Contact-Rich Policy Learning. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.079.
 - [26] Wenhai Liu, Junbo Wang, Yiming Wang, Weiming Wang, and Cewu Lu. Forcemimic: Force-centric imitation learning with force-motion capture system for contact-rich manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.
 - [27] Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=XVjTT1nw5z>.
 - [28] Yecheng Jason Ma, Joey Hejna, Chuyuan Fu, Dhruv Shah, Jacky Liang, Zhuo Xu, Sean Kirmani, Peng Xu, Danny Driess, Ted Xiao, et al. Vision language models are in-context value learners. In *The Thirteenth International Conference on Learning Representations*, 2024.
 - [29] Shamil Mamedov and Stanislav Mikhel. Practical aspects of model-based collision detection. *Frontiers in Robotics and AI*, 7:571574, 2020.
 - [30] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.
 - [31] Kyu Min Park, Jihwan Kim, Jinhyuk Park, and Frank C. Park. Learning-based real-time detection of robot collisions without joint torque sensors. *IEEE Robotics and Automation Letters*, 6(1):103–110, 2021. doi: 10.1109/LRA.2020.3033269.
 - [32] Kyu Min Park, Younghyo Park, Sangwoong Yoon, and Frank C. Park. Collision detection for robot manipulators using unsupervised anomaly detection algorithms. *IEEE/ASME Transactions on Mechatronics*, 27(5):2841–2851, 2022. doi: 10.1109/TMECH.2021.3119057.
 - [33] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.
 - [34] Moritz Reuss, Niels van Duijkeren, Robert Krug, Philipp Becker, Vaisakh Shaj, and Gerhard Neumann. End-to-end learning of hybrid inverse dynamics models for precise and compliant impedance control. In *Robotics: Science and Systems*, 2022.
 - [35] Kenneth Shaw and Deepak Pathak. Leap hand v2: Dexterous, low-cost anthropomorphic hybrid rigid soft hand for robot learning. In *2nd Workshop on Dexterous Manipulation: Design, Perception and Control (RSS)*.
 - [36] Haochen Shi, Songbo Hu, Yifan Hou, Weizhuo Wang, Karen Liu, and Shuran Song. Minimalist compliance control, 2026. URL <https://arxiv.org/abs/2603.00913>.
 - [37] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.
 - [38] Ziyu Wang, Alexander Novikov, Konrad Zolna, Jost Tobias Springenberg, Scott Reed, Bobak Shahriari, Noah Siegel, Josh Merel, Caglar Gulcehre, Nicolas Heess, and Nando de Freitas. Critic regularized regression. *Advances in Neural Information Processing Systems*, 33: 7768–7778, 2020.
 - [39] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. GELLO: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12156–12163. IEEE, 2024. doi: 10.1109/IROS58592.2024.10801581.
 - [40] Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36: 69798–69818, 2023.
 - [41] Han Xue, Jieji Ren, Wendi Chen, Gu Zhang, Yuan Fang, Guoying Gu, Huazhe Xu, and Cewu Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.
 - [42] Koki Yamane, Yunhan Li, Masashi Konosu, Koki Inami, Junji Oaki, Toshiaki Tsuji, and Sho Sakaino. Design and experimental validation of sensorless 4-channel bilateral teleoperation for low-cost manipulators, 2026. URL <https://arxiv.org/abs/2507.06174>.
 - [43] Nural Yilmaz, Jie Ying Wu, Peter Kazanzides, and Ugur Tumerdem. Neural network based inverse dynamics identification and external force estimation on the da vinci research kit. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1387–1393. IEEE, 2020.
 - [44] Jiawen Yu, Hairuo Liu, Qiaojun Yu, Jieji Ren, Ce Hao, Haitong Ding, Guangyu Huang, Guofan Huang, Yan Song, Panpan Cai, Wenqiang Zhang, and Cewu Lu.

ForceVLA: Enhancing VLA models with a force-aware moe for contact-rich manipulation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL <https://openreview.net/forum?id=2845H8Ua5D>.

- [45] Yu Zhang, Yuqi Xie, Huihan Liu, Rutav Shah, Michael Wan, Linxi Fan, and Yuke Zhu. Scizor: Self-supervised data curation for large-scale imitation learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.
- [46] Zongzheng Zhang, Haobo Xu, Zhuo Yang, Chenghao Yue, Zehao Lin, Huan-ang Gao, Ziwei Wang, and Hao Zhao. Elucidating the design space of torque-aware vision-language-action models. In Joseph Lim, Shuran Song, and Hae-Won Park, editors, *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 4019–4037. PMLR, 27–30 Sep 2025. URL <https://proceedings.mlr.press/v305/zhang25k.html>.
- [47] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.016.
- [48] Alvin Zhu, Yusuke Tanaka, Fadi Rafeedi, and Dennis Hong. Cycloidal quasi-direct drive actuator designs with learning-based torque estimation for legged robotics. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2025.

APPENDIX

A. Related Work on Sensorless External Force Estimation

Estimating external forces without dedicated sensors usually involves modeling robot dynamics and interpreting residuals between predicted and observed signals as external contact. Traditional analytical and system-identification approaches [6, 42, 36] can be effective, but require substantial engineering and often struggle with actuator nonlinearities such as dead zones [19] and torque ripple [48]. Supervised learning methods use paired proprioceptive and real-world [31] or simulation [23] contact data, but require labels or privileged sensing. Unsupervised anomaly-detection methods [32] identify deviations from free motion but primarily provide contact detection rather than continuous force estimates. Inverse-dynamics learning has been studied for individual actuators [48] and specialized systems such as the *da Vinci* robot [43], but has not been broadly demonstrated on commodity arms.

B. Analytical External-Torque Estimation

External joint torque τ_{ext} is the joint-space torque induced by physical interaction, excluding torque required for free-space motion. With motor torque $\tau_m = KI_m$, analytical estimation uses

$$\tau_{\text{ext}} = \tau_m - \tau_f, \quad (3)$$

$$\begin{aligned} \tau_f = & \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \\ & + \mathbf{g}(\mathbf{q}) + \tau_{\text{other}}. \end{aligned} \quad (4)$$

Accurate identification of inertial, gravitational, and friction parameters is difficult, especially from motor-side measurements [20, 13]. The residual also contains stiction, backlash, hysteresis, temperature-dependent drive behavior, sensing noise, torque ripple, dead zones, and saturation [24, 22, 34]. Many of these effects are history dependent, motivating NEXT's learned temporal model.

C. Full NEXT Formulation

NEXT is trained on contact-free trajectories $\mathcal{D}_{\text{free}} = \{(\mathbf{x}_i, \tau_{m,i})\}_{i=1}^N$, where

$$\mathbf{x}_i = [\mathbf{q}_{i-H:i}, \dot{\mathbf{q}}_{i-H:i}, \Delta \mathbf{q}_{d,i-H:i}]. \quad (5)$$

Because all data are collected without external contact, $\tau_{m,i}$ supervises the free-space torque. We train with

$$\mathcal{L}_{\text{NEXT}}(\theta) = \frac{1}{N} \sum_{i=1}^N \|f_{\theta}(\mathbf{x}_i) - \tau_{m,i}\|_2^2. \quad (6)$$

At deployment, a rolling history buffer is evaluated independently at each timestep and external torque is computed from the residual in Eq. 1.

D. Free-Space Inverse Dynamics Model Architecture

We use an LSTM-based sequence model for free-space inverse dynamics prediction. Rather than maintaining a persistent hidden state across the entire trajectory, we adopt a stateless sliding-window formulation. The network therefore implicitly models temporal dynamics within the look-back

TABLE IV
NETWORK ARCHITECTURE AND TRAINING HYPERPARAMETERS FOR THE FREE-SPACE INVERSE DYNAMICS MODEL USED IN NEXT.

Hyperparameter	Value
Recurrent backbone	2-layer LSTM with hidden size 128
Prediction head	2-layer MLP with hidden size 256
Dropout	0.1
Loss function	Mean squared error (MSE)
Optimizer	AdamW
Base learning rate	1×10^{-3}
Weight decay	1×10^{-6}
Gradient clipping	1.0
Early stopping	Validation loss; patience 20; warmup 10; $\Delta_{\min} = 10^{-5}$
Hardware	NVIDIA RTX 3090, 24 GB
Training time	Approximately 1 minute with early stopping
Data frequency	100 Hz

window, while the hidden state is reset for each sequence. This formulation provides recent motion context without accumulating hidden-state drift over long trajectories. The local history allows the model to capture velocity-dependent, actuator-dependent, and unmodeled friction effects without explicitly estimating acceleration. For deployment, the estimator can be evaluated using a rolling buffer of the most recent N observations. External joint torque is then estimated as the residual between the measured joint torque and the predicted free-space torque. The network architecture and training hyperparameters are listed in Table IV.

On a single thread of an Intel Core i9-10920X CPU, the free-space inverse dynamics model required approximately 1.76 ms per forward pass, corresponding to an effective update rate of 568 Hz. This enables high-frequency force feedback teleoperation. In practice, we run the model at 100 Hz.

E. Free-Space Inverse Dynamics Network Ablations

To select the free-space inverse dynamics model used in NEXT, we ablate model architecture, input modality, and history length using external torque estimation error in both free-space and contact settings. Prior work has used MLPs for torque estimation on specialized *da Vinci* systems [43] and GRUs for single-actuator external torque estimation [48]. We therefore compare MLP, GRU, and LSTM architectures.

Table V reports the architecture ablation results, with values corresponding to the average joint L_1 validation error between predicted free-space torque and ground-truth free-space torque on the held-out set. MLPs perform worse than recurrent models, showing the importance of temporal history for free-space torque estimation. GRUs perform well in free space, but perform worse during contact. LSTMs achieve the best free-space and contact performance, where longer-term actuator, frictional, and controller-dependent effects may play a larger role.

We further ablate input modalities and history length. Specifically, we compare histories of joint positions q , joint positions and velocities $(q, \dot{q})$, and joint positions, velocities, and tracking error $(q, \dot{q}, \Delta q_d)$, with history lengths of 10, 25,

and 50 steps. Using $(q, \dot{q}, \Delta q_d)$ consistently outperforms the other inputs, suggesting that tracking error provides useful information about controller effort and actuator response. Short histories of 10 steps generally yield higher error, while increasing the history to 50 steps improves performance, especially for LSTM models. Based on these results, we use an LSTM with history length $H = 50$ and input $(q, \dot{q}, \Delta q_d)$ as the final NEXT free-space inverse dynamics model.

F. Free-Space Data Collection Guidelines for NEXT

Figure 8 shows the distribution of the 10-minute free-motion dataset collected on the Piper arms for training the free-space inverse dynamics LSTM for NEXT. During data collection, we follow a simple coverage-oriented procedure: first, each joint is moved independently across its range of motion; next, the end effector is moved through Cartesian-like motions involving multiple joints; finally, both slow and fast motions are repeated to capture velocity-dependent dynamics. Although these trajectories can be generated automatically through motion planning, in practice they can also be recorded once and replayed on new robot setups to quickly collect training data.

G. External Joint Torque Estimation Baseline Implementation

We implement two model-based external joint torque estimation baselines for NEXT: FILIC [12] and a disturbance observer [29]. We evaluate both methods on the Franka Panda and AgileX Piper arms. For the Franka, we use the dynamics model from [11], whose parameters were obtained through optimization-based system identification. For the AgileX Piper, we use the dynamics model derived from the manufacturer-provided URDF [2].

FILIC [12]. For FILIC, we use the standard residual-based estimate introduced in Appendix B. Given the measured motor torque τ_m , external torque is estimated by subtracting the nominal free-space torque computed using MuJoCo inverse dynamics:

$$\hat{\tau}_{\text{ext}}^{\text{FILIC}} = \tau_m - \tau_f, \quad (7)$$

$$\tau_f = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}). \quad (8)$$

This method directly attributes the residual between measured motor torque and modeled free-space dynamics to external interaction.

Disturbance Observer [29]. As a second baseline, we implement a momentum-based disturbance observer. Instead of directly computing acceleration-dependent inverse dynamics, the observer estimates external torque from the residual in generalized momentum,

$$\mathbf{p} = \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}. \quad (9)$$

It maintains an internal momentum estimate $\hat{\mathbf{p}}$ with dynamics

$$\begin{aligned} \dot{\hat{\mathbf{p}}} &= \tau_m - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) \\ &\quad + \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}} + \hat{\tau}_{\text{ext}}^{\text{MDO}}, \end{aligned} \quad (10)$$

$$\hat{\tau}_{\text{ext}}^{\text{MDO}} = \mathbf{K}_o(\mathbf{p} - \hat{\mathbf{p}}). \quad (11)$$

By avoiding direct computation of $\ddot{\mathbf{q}}$, the momentum observer is less sensitive to encoder noise and produces smoother estimates than FILIC. As shown in Figure 4, it is less noisy than FILIC but remains less accurate than NEXT.

H. Force-Feedback Teleoperation Baseline Implementation

For the force-feedback teleoperation user study, we implement five feedback conditions on the Franka Panda:

- *No Feedback*: A GELLO leader-follower system [39] without force feedback.
- *Disturbance Observer (DO)*: External joint torque is estimated on the follower arm using the disturbance observer described in Appendix G, then relayed to the leader arm as force feedback.
- *Position-Position Feedback (PP)*: Feedback torque is computed by scaling the joint position difference between the leader and follower arms, following [17].
- *FACTR Teleop*: The FACTR teleoperation system [25], which feeds back external joint torque from dedicated joint torque sensors, along with gravity compensation, friction compensation, and null-space regulation.
- *Ours*: The same teleoperation framework as FACTR Teleop, but using NEXT-predicted external joint torque instead of dedicated joint torque sensor measurements.

We repeat the same study on the AgileX Piper arm, excluding FACTR Teleop because the Piper does not provide dedicated joint torque sensing.

I. AgileX Piper NEXT Evaluations

Since the AgileX Piper does not provide dedicated external joint torque sensing, we evaluate torque estimation only in the free-space setting, where the ground-truth external torque is known to be zero. As shown in Table VI, NEXT achieves the lowest error, with an average L_1 error of only 0.018 Nm, outperforming both FILIC and the disturbance observer. These results show that NEXT can produce accurate external torque estimates even on a low-cost arm without dedicated force sensors.

To evaluate NEXT in contact settings, we perform a force-feedback teleoperation user study. As shown in Figure 9, participants rated teleoperation with NEXT-based force feedback as easier to use than the baseline feedback methods.

J. Related Work on Imitation Learning with Force Information

Force-aware imitation learning methods use effort, torque, or force as policy inputs [46, 6, 21, 44], auxiliary prediction objectives [46, 14], or targets for hybrid force-position and impedance control [26, 12, 41]. FACTR [25] uses force information during training to improve generalization. FIRST instead uses estimated external torque both as a policy observation and to infer contact phases that reshape the training distribution. Our ablations show that pre-contact samples are often more valuable than in-contact samples.

TABLE V

ABLATION OF MODEL FAMILY, INPUT MODALITY, AND TEMPORAL HISTORY LENGTH FOR EXTERNAL TORQUE ESTIMATION. VALUES ARE THE AVERAGE JOINT L_1 ERROR BETWEEN PREDICTED AND GROUND-TRUTH FREE-SPACE TORQUE ON THE HELD-OUT SET, REPORTED IN NM.

Input	H	MLP		GRU		LSTM (Ours)	
		Free Motion ↓	Contact ↓	Free Motion ↓	Contact ↓	Free Motion ↓	Contact ↓
q	10	2.35 ± 2.34	1.78 ± 1.64	2.25 ± 1.79	1.12 ± 1.09	2.53 ± 2.52	2.61 ± 2.29
	25	2.02 ± 2.01	1.34 ± 1.32	1.68 ± 1.30	0.665 ± 0.478	2.21 ± 2.20	1.45 ± 1.39
	50	1.76 ± 1.76	1.13 ± 1.06	1.62 ± 1.24	0.843 ± 0.835	2.15 ± 2.15	1.09 ± 0.908
$q, \dot{q}$	10	1.04 ± 1.03	0.681 ± 0.647	0.863 ± 0.862	0.580 ± 0.535	1.15 ± 1.15	0.715 ± 0.553
	25	0.980 ± 0.976	0.908 ± 0.802	0.814 ± 0.814	0.596 ± 0.506	1.11 ± 1.11	0.551 ± 0.511
	50	1.02 ± 1.02	0.905 ± 0.892	0.795 ± 0.794	0.653 ± 0.549	1.12 ± 1.12	0.578 ± 0.526
$q, \dot{q}, \Delta q_d$	10	0.706 ± 0.702	1.89 ± 1.71	0.423 ± 0.422	1.94 ± 1.75	0.504 ± 0.503	0.876 ± 0.244
	25	0.755 ± 0.747	1.56 ± 1.45	0.473 ± 0.472	1.47 ± 1.29	0.490 ± 0.489	0.789 ± 0.453
	50	0.813 ± 0.811	1.47 ± 1.33	0.429 ± 0.429	1.62 ± 1.45	0.414 ± 0.278	0.547 ± 0.348

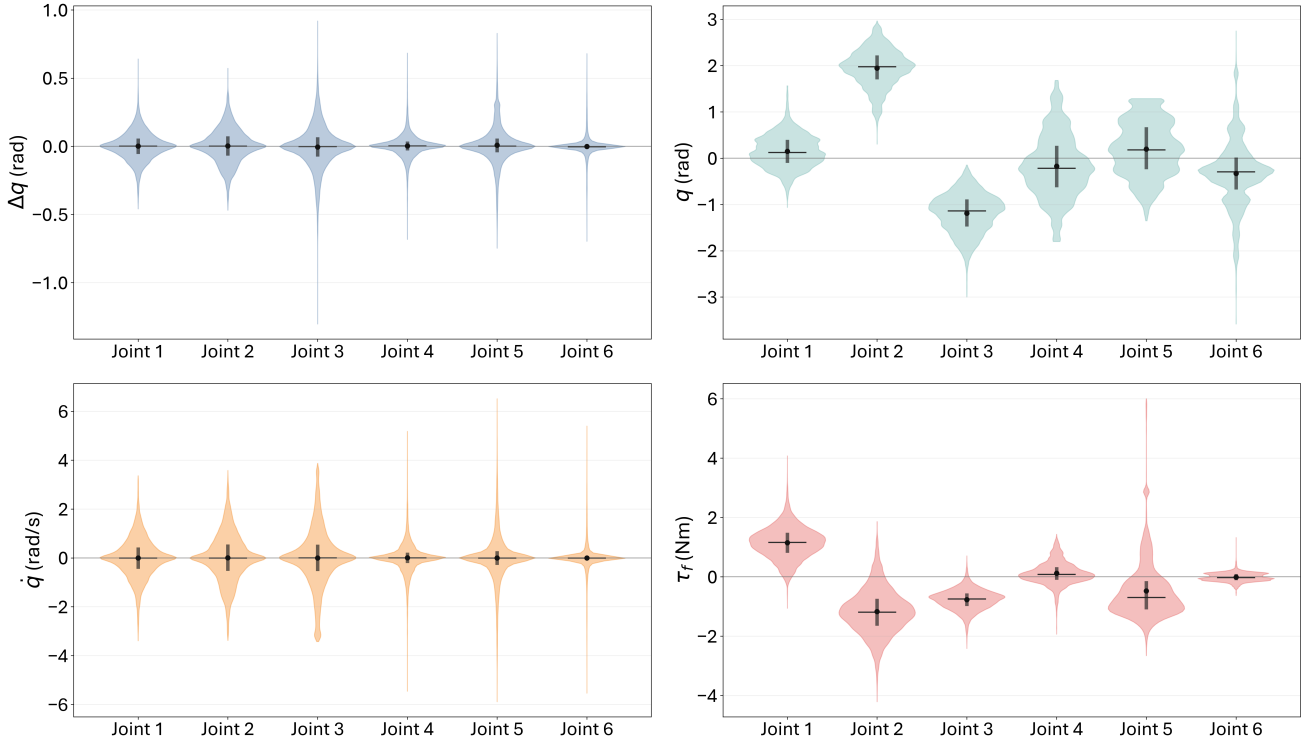


Fig. 8. Distribution of the 10-minute free-motion dataset used to train NEXT on the Piper arm. The plots show coverage of joint-position error Δq , joint position q , joint velocity $\dot{q}$, and free-space torque τ_f across all six joints.

TABLE VI

AVERAGE L_1 JOINT TORQUE ERROR ON THE AGILEX PIPER IN FREE-SPACE SETTINGS. ERRORS ARE MEASURED AGAINST ZERO EXTERNAL TORQUE. NEXT ACHIEVES THE LOWEST ERROR.

External Torque Method	Error (Nm) ↓
Free Motion (GT)	0.000 ± 0.000
FILIC	0.435 ± 0.225
Disturbance Observer	0.248 ± 0.132
NEXT (Ours)	0.018 ± 0.012

K. Related Work on Policy Learning with Dataset Reweighting

In reinforcement learning, AWR [33], CRR [38], and AWAC [30] emphasize high-advantage actions. Robotics imitation-learning methods also reweight or select data: Re-Mix adapts DoReMi-style group optimization [40, 15]; DataMIL uses data-model estimators [18, 10]; DemInf scores state-action mutual information [16]; and SCIZOR combines progress prediction with clustering [45]. Other methods use learned values or rewards [28, 8], rollout success classifiers [7], or influence functions [1]. FIRST requires neither reward models nor policy rollouts: it derives a task-relevant sampling signal directly from external torque.

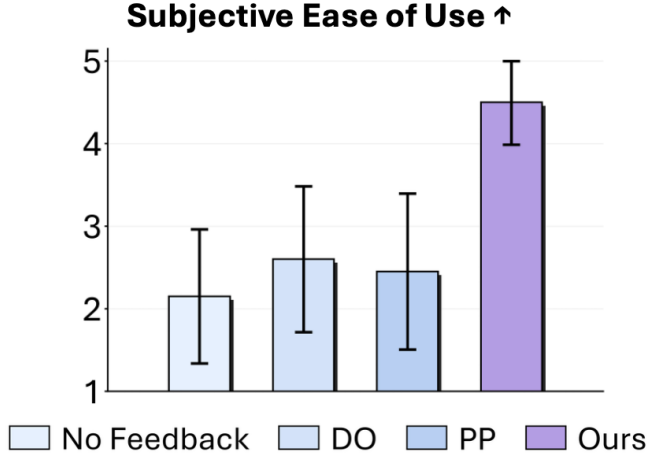


Fig. 9. User-study results comparing external torque feedback methods on the AgileX Piper.

L. Full FIRST Formulation

At timestep t , the policy conditions on $\mathbf{o}_t = (\mathbf{I}_t^{1:M}, \mathbf{q}_t, \hat{\tau}_{\text{ext},t})$ and predicts an action chunk $\hat{\mathbf{a}}_{t:t+k}$. For the flow-matching policy, let $\mathbf{x}_1 = \mathbf{a}_{t:t+k}$, $\mathbf{x}_0 \sim \mathcal{N}(0, I)$, and $\mathbf{x}_\alpha = (1 - \alpha)\mathbf{x}_0 + \alpha\mathbf{x}_1$ for $\alpha \sim \mathcal{U}(0, 1)$. We optimize [27]

$$\mathcal{L}_{\text{FM}} = \mathbb{E} \left[\|v_\theta(\mathbf{x}_\alpha, \alpha, \mathbf{o}_t) - (\mathbf{x}_1 - \mathbf{x}_0)\|_2^2 \right]. \quad (12)$$

The contact score is $f_t = \|\hat{\tau}_{\text{ext},t}\|_1$. Hysteresis thresholding defines $c_t \in \{0, 1\}$, and contact onsets are

$$\mathcal{T}_{\text{onset}} = \{t \mid c_{t-1} = 0, c_t = 1\}. \quad (13)$$

For control frequency F , phase labels are

$$s_t = \begin{cases} \text{Contact}, & c_t = 1, \\ \text{Pre-Contact}, & \exists t_o \in \mathcal{T}_{\text{onset}} : t_o - F \leq t < t_o, \\ \text{Free Space}, & \text{otherwise.} \end{cases} \quad (14)$$

FIRST then samples datapoints according to Eq. 2; it changes neither the demonstrations nor the regression objective.

M. Evaluation Protocol

For Franka contact evaluation, we collect five minutes of held-out motion while a human applies forces and compare estimates against the factory external-torque signal. For free-space evaluation, we collect five minutes without contact and use zero torque as ground truth, treating the Franka signal as an additional baseline. Piper contact performance is assessed through teleoperation because the platform lacks a dedicated torque reference.

For policy learning, each task contains 250 demonstrations collected with two wrist and two overhead RGB cameras on a bimanual Piper system. Each policy is evaluated over 20 rollouts. We use a flow-matching transformer with action chunk size 30 as the primary architecture and repeat the study with ACT. The visual task overview is shown in main-paper Fig. 6.

N. Behavior Cloning Policy Architecture and Training Hyperparameters

Our behavior cloning policy takes RGB images, proprioceptive inputs, and force estimates as input. Each modality is tokenized before being passed to either an ACT or Flow Matching DiT policy architecture. Images are encoded with a DINOv3-Base encoder [37], while proprioceptive and force inputs are projected into tokens using MLP layers.

Proprioceptive inputs include joint positions and gripper widths, and force inputs correspond to estimated external joint torques. We use only the most recent observation at the current timestep. The action space consists of absolute joint angles and gripper width, with an action prediction horizon of 30. Training details for both ACT and Flow Matching policies are provided in Table VII.

O. Task Description

We evaluate FIRST along with baselines on five long-horizon, contact-rich manipulation tasks:

- *LEGO assembly (LEAP Hand V2 [35])*: bimanual hand assembly of two small LEGO blocks onto a bigger LEGO block.
- *NIST belt assembly (gripper)*: bimanual gripper manipulation of two belts around two pulley sets.
- *NIST insertion (gripper)*: insertion of two pulley sets onto pegs on a NIST board.
- *Tool Clean Up (gripper)*: screwdriver insertion followed by case assembly.
- *Cap screwing (gripper)*: cap pickup, alignment, and screwing onto a container until tightened.

P. Task Progress Rate Calculation

We report task progress rate in addition to binary success rate to evaluate partial completion in multi-stage manipulation tasks. Each task is divided into several stages, and each stage is marked as either completed (s) or failed (f) for every rollout. The task progress rate is calculated as the percentage of completed stages across all rollouts.

The task stages are listed in Table VIII. For example, the LEGO task consists of grasping, aligning, and inserting both the red and green cubes. The NIST Belt task includes picking up and aligning both belt components before pushing them into place. The NIST Insertion task measures progress through picking, aligning, and inserting the black part, followed by picking up the aluminum part. The Tool Clean Up task evaluates whether the robot can pick up, align, and insert the driver, then assemble the box.

For the cap screwing task, we use a modified progress score because a policy can fail by continuing to rotate the cap after it is already tightened. Therefore, we subtract a penalty based on the number of over-turns. Each over-turn reduces the progress score by 0.02. This penalizes policies that complete the screwing motion but do not stop properly.

TABLE VII
POLICY ARCHITECTURE AND TRAINING HYPERPARAMETERS.

Hyperparameter	Value
Behavior Policy Training	
Optimizer	AdamW
Base learning rate	3×10^{-4}
Weight decay	0.05
Optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
Batch size	128
Learning-rate schedule	Cosine decay
Total epochs	15–20
Warmup steps	500
Augmentation	RandomResizeCrop
GPU	NVIDIA A6000, 48 GB
Wall-clock time	12–24 hours
Action Chunking Transformer Details	
Encoder layers	4
Decoder layers	6
MHSA heads	8
Hidden dimension	768
Feed-forward dimension	3072
Dropout	0.1
Positional encoding	Sinusoidal
Action chunk	30
Flow-Matching Details	
Decoder layers	6
MHSA heads	8
Hidden dimension	768
Feed-forward dimension	3072
Dropout	0.1
Positional encoding	Sinusoidal
Action chunk	30
Evaluation steps	10
Time sampling	$\tau \sim \text{Beta}(1.0, 1.5)$
ODE solver	Euler
Noise distribution	$\mathcal{N}(0, I)$

TABLE VIII
STAGE DECOMPOSITION USED FOR TASK-PROGRESS CALCULATION.

Task	Stages
LEGO	Red grasp, red align, red insert, green grasp, green align, green insert
NIST Belt	Pick orange, align big, align small, push, pick gray, align gray, push
NIST Insertion	Pick black, align black, insert black, pick aluminum
Tool Clean Up	Pick screwdriver, align screwdriver, insert screwdriver, assemble box
Cap Screwing	Pick cap, align cap, screw on, end properly

Q. Imitation Learning with Force Information Baseline Implementations

Base Policy: We use the base ACT or flow-matching policy described in Sec. IV-B. The policy takes multi-view RGB images and robot proprioception, including joint positions and gripper states, as input and predicts a chunk of future joint and gripper commands.

Base Policy + Torque: This baseline augments the base policy with external joint torque predictions from NEXT. The

predicted torque signal is embedded with an MLP and added to the policy context as an additional observation token.

FACTR [25]: FACTR keeps the policy architecture unchanged and applies a visual curriculum to all image observations. Following the original FACTR implementation, we use a Gaussian blur operator with a linear curriculum schedule. Specifically, at training step n , the blur strength is set as

$$\sigma_n = \sigma_0 \left(1 - \frac{n}{N}\right),$$

where σ_0 is the initial blur strength and N is the total number of curriculum steps. Thus, the input images are strongly blurred at the beginning of training, and the blur is gradually removed as training progresses.

TA-VLA [46]: TA-VLA uses the predicted torque/contact signal as an auxiliary learning target in addition to the action. During inference, only the robot-action dimensions are executed, while the auxiliary torque/contact prediction is used only to shape the learned representation during training.

TABLE IX
BEST RE-SAMPLING RATIOS USED BY FIRST FOR EACH TASK. RATIOS ARE REPORTED AS $w_F : w_{PC} : w_C$, CORRESPONDING TO FREE-MOTION, PRE-CONTACT, AND CONTACT SAMPLING WEIGHTS.

Task	Best Ratio
LEGO Assembly	1 : 5 : 1
NIST Belt Assembly	1 : 5 : 1
NIST Insertion	1 : 5 : 1
Tool Clean Up	1 : 5 : 1
Cap Screwing	1 : 3 : 3

R. Segmenting Trajectories into Contact Phases Visualization

Figure 10 shows an example of phase labels produced by our hysteresis-threshold-based labeling procedure. The estimated external torque from NEXT remains near zero during free motion and rises clearly during physical interaction, allowing the trajectory to be separated into free-motion, pre-contact, and contact phases. The pre-contact label captures the short interval immediately before contact onset, where precise alignment is required, while the contact label captures periods of sustained interaction. This sensor-level signal enables automatic segmentation of demonstrations even on robot arms without dedicated force sensors, providing the phase annotations used by FIRST for force-informed re-sampling.

S. Task Progress Evaluation on ACT

We also apply FIRST to ACT, in addition to the flow-matching policy presented in Section IV-B. As shown in Figure 11, we observe a similar trend: up-sampling pre-contact and contact phases improves task progress over the baselines.

T. Selected Re-Sampling Ratios for Each Task

We report the task-specific re-sampling ratios used by FIRST in Table IX. All ratios are written as $w_F : w_{PC} : w_C$, where w_F , w_{PC} , and w_C denote the sampling weights for free-motion, pre-contact, and contact samples, respectively. For most tasks, FIRST uses 1 : 5 : 1, which up-samples pre-contact samples by $5\times$ relative to free-motion and contact samples. For Cap Screwing, FIRST uses 1 : 3 : 3, up-sampling both pre-contact and contact samples, since sustained contact is central to successful execution. These ratios are selected based on task progress for each task.

U. Up-Sampling Magnitude Ablation

Figure 12 ablates the effect of up-sampling magnitude on task progress. We define the magnitude m as the sampling weight assigned to the up-sampled phase relative to free motion. For pre-contact up-sampling, the phase weights are

$$w_F : w_{PC} : w_C = 1 : m : 1,$$

whereas for pre-contact + contact up-sampling, the phase weights are

$$w_F : w_{PC} : w_C = 1 : m : m.$$

We run this ablation on NIST Belt Assembly, NIST Insertion, and Tool Clean Up using ACT. Due to limited time

and compute, we additionally run the flow-matching ablation only on Tool Clean Up. Across tasks, moderate up-sampling improves task progress, with performance generally peaking around $m = 5$. Larger values lead to saturation or degradation, suggesting that overly aggressive up-sampling can over-bias the training distribution toward contact-relevant segments and reduce coverage of the full task trajectory.

V. Up-Sampling Policies Without Force Input

While Section IV-B evaluates FIRST on policies that condition on external joint torque, FIRST can also be applied without using force as a policy input. In this setting, NEXT is used only offline to segment demonstrations into free-motion, pre-contact, and contact phases. These phase labels are then used by FIRST to re-sample the training data, increasing the frequency of pre-contact and contact examples in each minibatch. Importantly, the policy itself receives only the standard visual and proprioceptive observations, without the predicted external joint torque.

Table X highlights two trends. First, when comparing methods within the columns without external joint torque, phase-based up-sampling improves over standard behavior cloning. For both ACT and Flow Matching, using PC or PC+C up-sampling generally increases task success compared to no up-sampling, showing that NEXT-derived phase labels are useful even when force estimates are used only at training time for data re-sampling. This indicates that FIRST can improve standard behavior cloning by reshaping the training distribution toward task-critical segments where alignment and contact-sensitive corrections are most important.

Second, when comparing horizontally between policies trained with and without external joint torque, the force-conditioned policies consistently achieve higher success rates. This shows that while phase-based up-sampling alone provides a clear benefit, directly feeding the predicted external joint torque into the policy provides additional state information that further improves contact-rich manipulation. Overall, the best performance is achieved when both components are used together: NEXT provides external joint torque as a policy input, while FIRST uses NEXT-derived phase labels to up-sample the most relevant pre-contact and contact phases during training.

W. Inputting Joint Current vs. External Joint Torque on Policy Performance

We further examine whether raw joint current is sufficient for contact-aware policy learning, or whether the learned external joint torque estimate provides a more useful representation. Prior work such as TA-VLA [46] uses readily available joint current measurements from low-cost robot arms as a force-related input for imitation learning. However, joint current is only an indirect measure of physical interaction: it contains actuator effort required for both free-space motion and external contact, as well as hardware- and controller-dependent effects. As a result, the policy must implicitly disentangle contact-induced torque from free-space dynamics.

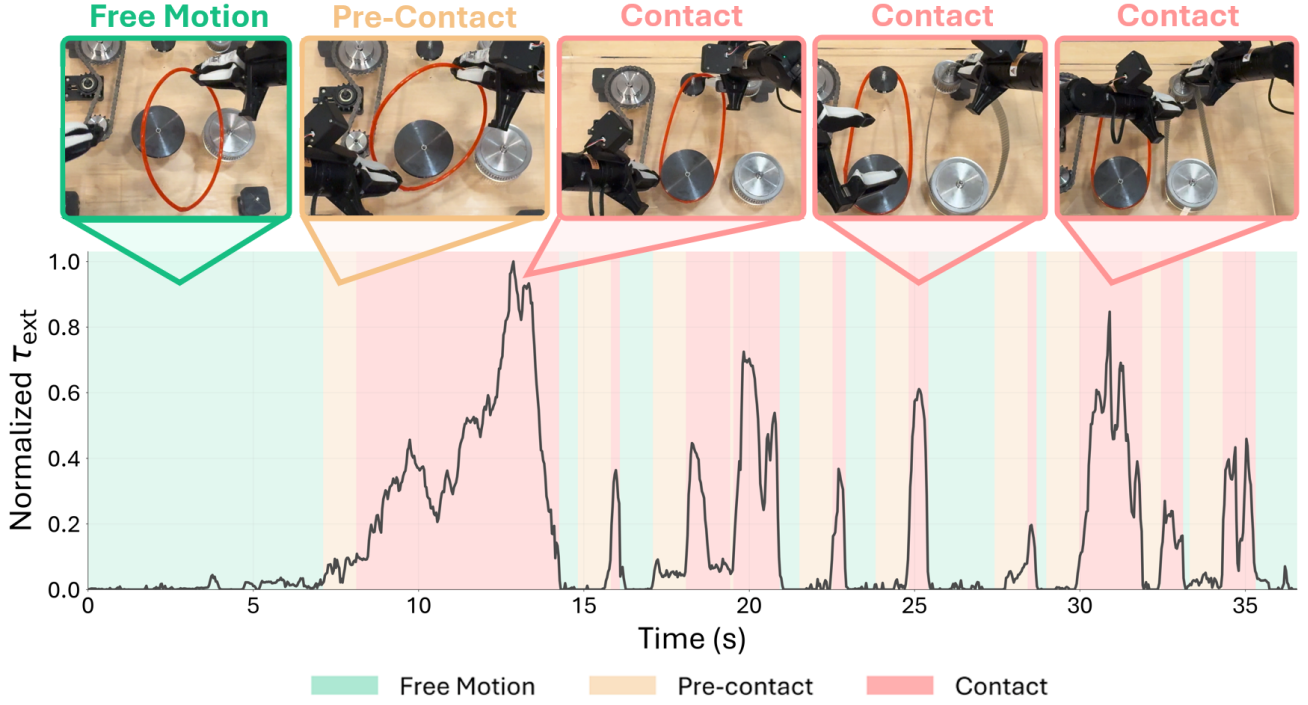


Fig. 10. **Contact-phase segmentation from learned external torque.** The normalized τ_{ext} from NEXT is used to segment demonstrations into free-motion, pre-contact, and contact phases. The image overlays show that the predicted labels align with the robot’s interaction state during the task.

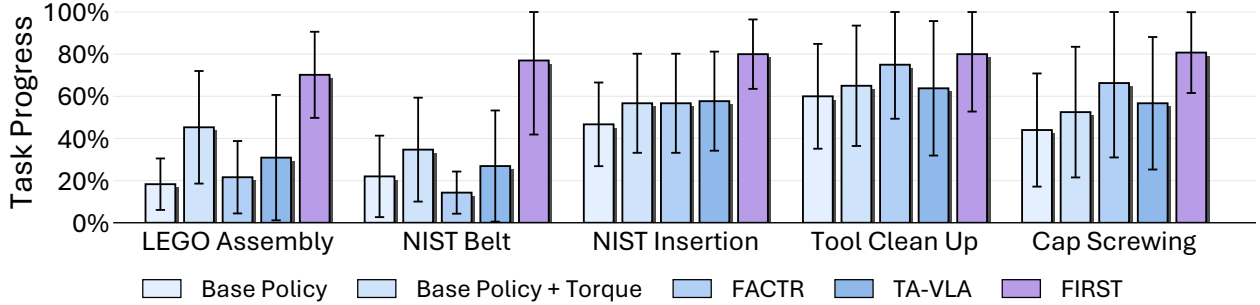


Fig. 11. Task progress across five contact-rich manipulation tasks using the ACT policy.

In contrast, NEXT explicitly estimates external joint torque by subtracting a learned free-space torque prediction from the measured motor torque. This distinction is important because estimating a clean external torque signal from current alone generally requires temporal context. As shown in our NEXT free-space inverse dynamics ablations in Appendix E, longer proprioceptive histories improve torque estimation, suggesting that instantaneous current and state observations are insufficient to reliably separate free-space actuation from contact. A standard behavior cloning policy, however, typically receives only the current observation rather than a long proprioceptive history such as the 50-step history used by NEXT. While one could provide such a history directly to the policy, this increases the burden on the policy to simultaneously infer dynamics, localize contact, and solve the visuomotor task. In practice, long proprioceptive histories may also encourage overfitting to trajectory-specific motion patterns rather than

using vision appropriately to react to scene variation. NEXT avoids this issue by using history in a specialized force-estimation module, while providing the policy with a compact, contact-relevant external torque signal.

We compare policies conditioned on raw joint current and learned external joint torque on two contact-sensitive tasks: NIST Insertion and Cap Screwing. Both tasks require force cues for successful execution: NIST Insertion depends on detecting alignment and contact during insertion, while Cap Screwing requires identifying when the cap has been fully tightened. As shown in Table III, conditioning on joint current improves performance on NIST Insertion compared to the base ACT policy, but does not improve Cap Screwing. In contrast, conditioning on learned external joint torque achieves the best performance on both tasks. Qualitatively, we observe that policies conditioned on raw joint current often continue rotating the cap after it is already tightened, leading to over-

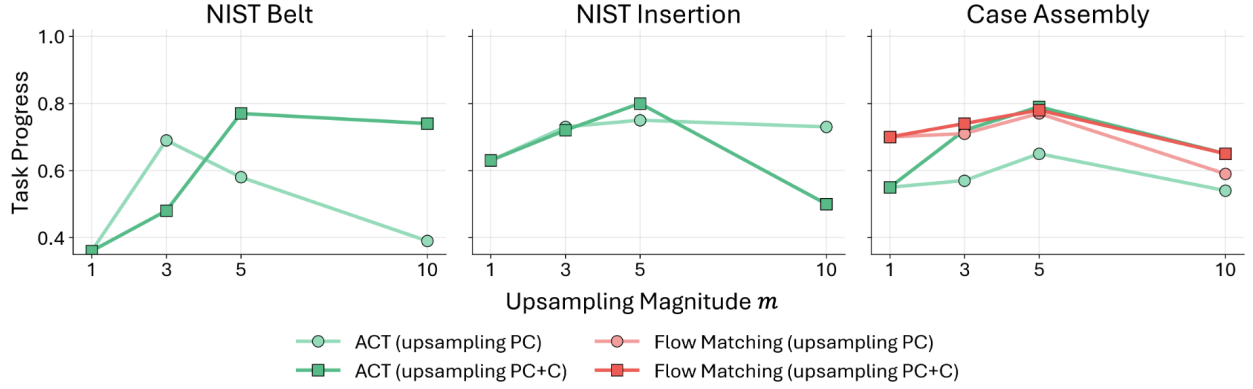


Fig. 12. Moderate up-sampling generally improves performance, with the best results around $m = 5$, while larger values lead to saturation or degradation.

TABLE X
EFFECT OF PHASE-BASED UP-SAMPLING ON POLICIES TRAINED WITH AND WITHOUT EXTERNAL JOINT TORQUE. RESULTS ARE REPORTED AS TASK SUCCESS RATES.

Policy	Up-sampling Strategy	Without τ_{ext}		With τ_{ext}	
		Belt Assembly	Tool Clean Up	Belt Assembly	Tool Clean Up
ACT	None	0.220	0.600	0.347	0.650
	PC	0.360	0.600	0.687	0.650
	PC+C	0.283	0.763	0.770	0.788
Flow Matching	None	0.150	0.463	0.431	0.525
	PC	0.213	0.575	0.767	0.805
	PC+C	0.241	0.525	0.649	0.775

screwing failures. Policies conditioned on learned external torque are better able to detect the tightened contact state and terminate the screwing motion appropriately.

The attention analysis in Figure 13 further supports this observation. To visualize how the policy uses force-related information during rollout, we compute the average cross-attention from the action tokens to the observation tokens in the first decoder layer. The decoder attends over both visual tokens and force-related tokens, but in Figure 13 we plot only the attention mass assigned to the force-related tokens: raw joint-current tokens for the current-conditioned policy, and learned external-torque tokens for our policy. This visualization therefore measures how strongly the policy attends to the force input when predicting actions.

During cap-screwing manipulation, the policy conditioned on learned external torque shows a clear increase in attention around key contact events, especially when the cap becomes tightened and the screwing motion should stop. In contrast, the policy conditioned on raw joint current exhibits weaker and less event-aligned attention to its current tokens. While attention is only a diagnostic and does not by itself prove causality, this pattern suggests that learned external joint torque provides a cleaner and more task-relevant contact representation than raw joint current for force-sensitive manipulation.

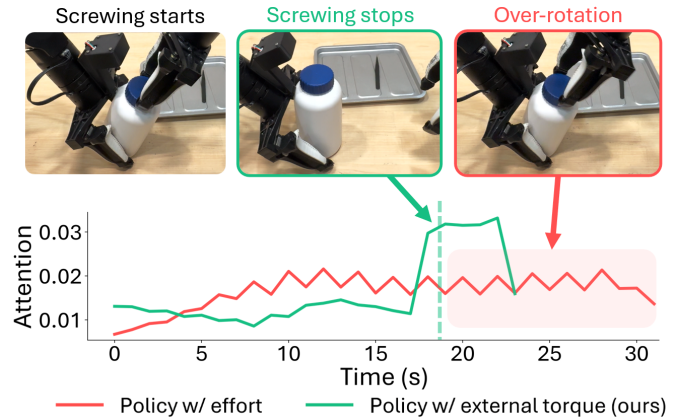


Fig. 13. Attention values during cap-screwing manipulation. Learned external torque produces clearer attention around key contact events than raw joint current.

X. Per-Batch Up-Sampling vs. Weighted Regression

We further compare FIRST against a weighted-regression baseline, where the loss is reweighted according to whether each sample belongs to the free-motion, pre-contact, or contact phase. Both approaches aim to emphasize contact-relevant data, but they do so in different ways: weighted regression modifies the optimization objective, whereas FIRST changes the training distribution by increasing the frequency of infor-

TABLE XI
COMPARISON OF PHASE-EMPHASIS MECHANISMS FOR LEGO ASSEMBLY
USING ACT.

Phase-Emphasis Mechanism	Phase	Progress
None	–	0.453
Weighted Regression	PC	0.345
	PC+C	0.478
FIRST (Ours)	PC	0.573
	PC+C	0.501

mative samples within each batch.

As shown in Table XI, weighted regression either degrades performance or provides only limited improvement over the no up-sampling baseline. In contrast, weighted sampling consistently improves LEGO assembly performance. We observe similar qualitative trends in other contact-rich tasks. This suggests that directly increasing the loss contribution of sparse contact-related samples can make optimization less stable, potentially causing the policy to overfit to high-weighted segments or ignore the broader trajectory context. By contrast, per-batch up-sampling preserves the original regression objective while exposing the policy more frequently to critical pre-contact and contact transitions, leading to more stable policy learning.