VIDFORCE: PREDICTING FORCE-AWARE ARTICULATION BY OBSERVING HUMANS

Anonymous authors
Paper under double-blind review

ABSTRACT

People can anticipate how an object moves and the effort an interaction may require by watching others. How can embodied agents acquire this understanding from video? Force measurements provide direct sensory supervision, but require specialized equipment and remain limited in quantity. We introduce VidForce, which estimates an interaction trajectory together with a force profile from a third-person RGB video; 2D contact clicks and textual object–action prompts are optional. Our approach makes limited instrumented measurements useful for interpreting human interactions through learned physical prototypes as a prior. The prototypes share learned force levels and temporal profiles across interactions of comparable effort, with visual and semantic cues determining their contribution to each observation. Geometric and vision–language features adapt this prior to the observed interaction, while a joint conditional flow refines an initial tracked trajectory together with the force profile. Paired human and instrumented-gripper recordings provide measured force references for supervising human video, and a separately trained torque model demonstrates that the formulation extends to other sensory quantities. We evaluate trajectory recovery and force estimation through baseline comparisons and controlled ablations on held-out interactions from the Hoi! dataset. On this evaluation subset, VidForce achieves 6.6 cm ATE, peak-force MAE and RMSE of 4.81 N and 5.92 N, respectively, and force-profile RMSE of 4.95 N. On real-world videos, VidForce infers motion and force reliably and consistently despite the variability in human execution. We further deploy VidForce on a Spot robot, where trajectories and force profiles estimated from human demonstrations drive real-world articulated-object manipulation, and report success rates across repeated trials. **Project Page:** `https://vid-force.github.io/`

1 INTRODUCTION

Watching another person interact with an object reveals more than its motion. We can anticipate how it articulates and form an expectation of the effort involved, using observed actions to guide our own (Reichelt et al., 2013). Giving robots a comparable understanding would make human video a source of both motion references and sensory expectations for planning and interaction. We study this capability through third-person video: estimating the trajectory of an articulated interaction together with its force or torque profile.

Recent methods extract rich geometric and motion representations from interaction videos. CoTracker3 (Karaev et al., 2025) follows image points, D4RT (Zhang et al., 2026) reconstructs and tracks dynamic scenes, and EgoInfinity (Wang et al., 2026a) recovers hand–object motion. VidBot (Chen et al., 2025) and AFUN (Wang et al., 2026b) predict interaction trajectories from an RGB observation and a language prompt. While these capabilities capture how an interaction unfolds, they do not estimate the force or torque required to execute it. Supervising such estimates, however, comes with two critical challenges: (i) force and torque data require specialized measurement equipment, and such instrumented interaction data are therefore far scarcer than ordinary videos. (ii) Even with the limited available data, how can the extracted knowledge be transferred so that models can predict from uninstrumented human demonstrations?

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.

We address this gap by transferring sensory information across interactions that require comparable effort. VidForce captures this structure with a learned set of force prototypes, each encoding a characteristic force level and temporal profile. Vision–language features route each observation to relevant prototypes, while observed geometry and appearance adapt the resulting prior to the observed interaction. By organizing these prototypes around measured effort rather than predefined object categories, the model connects limited instrumented supervision to uninstrumented human video without supervising or fixing any object taxonomy. To adapt the prototype to the observed instance, a joint conditional flow couples trajectory and force prediction. Given an initial trajectory estimate from a motion tracker, the flow refines this proposal and the adapted force prior. During training, force measurements from instrumented-gripper recordings provide reference targets for paired human videos. At inference, the contact point and prompt may be supplied or inferred from the video. The same formulation can be extended to other sensory quantities; we demonstrate this with a separately trained torque model.

- We formulate sensory-aware articulation understanding from third-person video: jointly recovering interaction trajectory and a force profile, with optional contact and language guidance.
- We introduce VidForce, a model that predicts the effort of an observed interaction in the form of force profiles. The model organizes observations by measured effort into learned force prototypes and jointly refines the resulting prior and a tracked trajectory. This design makes effective use of the limited available data without relying on predefined object categories.
- We evaluate trajectory and sensory prediction against matched baselines and isolate the contributions of key model components and input sources through controlled ablations. We further assess video-only inference on real-world recordings outside the training dataset and conduct success-rate trials on a Spot robot using the predicted trajectories and force profiles for manipulation.

2 Related Work

Interaction and articulation from human video. Human video can supply motion cues for robot learning without robot action labels: point tracks condition policies (Wen et al., 2024; Jain et al., 2024), reconstructed object motion supports imitation (Kerr et al., 2024), and object-centric 3D motion fields provide metric geometry (Yin et al., 2025). CoTracker3 (Karaev et al., 2025) tracks points through occlusion, D4RT (Zhang et al., 2026) combines tracking with dynamic 3D reconstruction, EgoInfinity (Wang et al., 2026a) recovers metric hand–object motion, and Articulate-Anything (Le et al., 2025) constructs articulated simulation assets. Affordance models such as VidBot (Chen et al., 2025) and AFUN (Wang et al., 2026b) use visual observations and language prompts to predict interaction trajectories. These methods capture where and how an interaction unfolds, but not the required force or torque; VidForce builds on D4RT by initializing from its trajectory estimate and conditioning joint trajectory–force prediction on the full motion field.

Force inference from video. Ehsani et al. (2020) infer force by matching simulated object motion to video, while ForceSight (Collins et al., 2024) predicts a single robot-centric force goal from RGB-D and language. VidForce instead jointly predicts a temporal force profile and interaction trajectory from third-person human video, using paired human and gripper recordings to transfer measured force supervision. Recent methods use measured force to transfer human demonstrations to robots: FeelTheForce (Adeniji et al., 2025) learns contact-driven policies from demonstrations with hand-force measurements, and ReForce (Wu et al., 2026) retargets hand motion while accounting for contact forces. VidForce addresses the complementary problem of estimating this missing force signal from video, where robots can use the predicted force profile for force-aware control (Sec. 4.3).

Sensory supervision. Measured sensory signals can ground visual representations in physical interaction. Touch and Go (Yang et al., 2022) learns visual–tactile correspondence, TVL (Fu et al., 2024) aligns touch with vision–language representations, FEEL (Dessalene et al., 2026) pairs egocentric video with measured hand forces for pretraining, and TouchAnything (Zhou et al., 2026) predicts bimanual pressure maps. Hoi! (Engelbracht et al., 2026) provides paired bare-hand and instrumented-gripper recordings across viewpoints; VidForce uses the gripper measurements to supervise gripper videos and provide force references for paired human interactions, then shares this supervision through learned prototypes organized by effort.

Learned prototype and mixture priors. Mixture-of-experts models (Jacobs et al., 1991; Shazeer et al., 2017) route inputs to jointly learned specialists, discrete codebooks (van den Oord et al.,

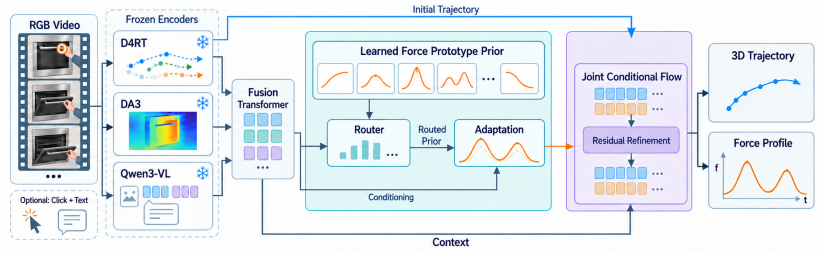


Figure 1: **VidForce overview.** The model takes an RGB video, with an optional contact click and text describing the object and its motion. Frozen D4RT, DA3, and Qwen3-VL backbones extract motion, geometry, visual, and language features. A fusion transformer combines these features into a shared representation that routes and adapts learned force prototypes to form a video-specific force prior. A joint conditional flow refines this prior together with the initial tracked trajectory, producing the final 3D trajectory and force profile.

2017) learn finite latent sets and serve as human-motion priors (Lucas et al., 2022), and prototype networks (Snell et al., 2017; Chen et al., 2019) represent classes through learned exemplars. VidForce combines these ideas but organizes its prototypes by measured effort: a routing loss associates each training interaction with nearby force levels, while the levels, temporal shapes, and semantic keys are learned jointly with the router. Their log-force levels and temporal shapes are mixed softly, yielding a force prior without class labels or a reconstruction objective.

3 METHOD

We present an overview of our method in Figure 1. Given a video clip V, a sequence of 32 RGB frames showing one interaction from a static third-person camera and cropped to an object-centric window around the manipulated object, VidForce predicts a camera-space contact trajectory $\mathbf{P} = (\mathbf{p}_i)_{i=1}^{N}$ at N=16 waypoints, a force-magnitude profile $\mathbf{f} \in \mathbb{R}^{L}_{\geq 0}$ over L=32 samples of normalized interaction time, and a separate peak estimate F. A contact point and a text prompt q of the object and its motion are optional inputs. Visual and language features route the observation to learned force prototypes, which initialize the force prediction. A joint conditional flow then refines this prediction together with the tracked trajectory. The same formulation applies to other sensory channels; a separately trained model predicts sensor-frame torque.

3.1 REPRESENTING THE OBSERVED INTERACTION

Three frozen models extract complementary information (motion, depth, semantics) from the clip and atrainable transformer fuses these representations, while the tracks provide an initial trajectory proposal. D4RT tracks points throughout the clip to form a motion field. The median displacement of valid tracks near the contact defines the initial trajectory proposal $\mathbf{B}$. DA3 (Lin et al., 2026) estimates depth and camera intrinsics $\mathbf{K}$. Qwen3-VL (Bai et al., 2025) provides visual features and the prompt embedding $\mathbf{e}(q)$. Rather than a generic text embedding, $\mathbf{e}(q)$ contains the square-rooted probabilities assigned to 256 answer tokens and the two actions when Qwen3-VL is asked which part is manipulated and whether it opens or closes (Appendix A.1). The fusion transformer reads these features together with track reliability, motion statistics, and an embodiment flag (human or gripper), and produces context tokens $\mathbf{M}$ and pooled features $\mathbf{g}$.

Coordinates. Each waypoint $\mathbf{p}_i$ is a 3D point in the camera frame. It is encoded as $\mathbf{x}_i$ by its normalized image offset from the contact and its log depth relative to the contact depth z_c (Eq. 6, Appendix A.1). This places the encoded trajectory $\mathbf{X}=(\mathbf{x}_i)_{i=1}^{N} \in \mathbb{R}^{N\times 3}$, tracked proposal $\mathbf{B}$, and predicted residuals in the same scale-free coordinate frame. The encoding is invertible given $\mathbf{K}$, so metric motion can be recovered exactly, at the scale estimated by DA3.

3.2 A LEARNED PHYSICAL PROTOTYPE PRIOR

The sensory prior consists of learned force prototypes that capture force levels and temporal profiles shared across interactions of comparable effort. A router weights these prototypes using visual and

language features, and an adaptation head adjusts the resulting prior to the observed interaction. The prior can be of arbitrary size, and we choose $K{=}10$ prototypes for all our experiments as the largest K for which every prototype remains in use (Appendix C.7). Each prototype comprises a log-force level a_k, a shape $\mathbf{s}_k \in \mathbb{R}^L$, and a learned semantic key $\mathbf{e}_k$. The level describes how much force is expected, the shape describes how it varies through the interaction, and the key is matched against language features during routing. We normalize the force profile by the separately predicted peak, $\boldsymbol{\sigma} = \mathbf{f}/F$, separating magnitude and temporal shape.

Routing. For prototype k, the router combines a learned score from the pooled video features with the similarity of prompt embedding $\mathbf{e}(q)$ and semantic key $\mathbf{e}_k$: $\ell_k = r_k(\mathbf{g}) + \beta \cos(\mathbf{e}(q), \mathbf{e}_k)$, where β is learned. A softmax over the scores mixes the prototypes into the prior peak and shape:

$$\boldsymbol{\pi} = \mathrm{softmax}(\boldsymbol{\ell}), \quad \log \bar{F} = \sum_k \pi_k a_k, \quad \bar{\boldsymbol{\sigma}} = \left[\textstyle\sum_k \pi_k \mathbf{s}_k\right]_+. \tag{1}$$

Mixing peaks in the log domain makes partial weights interpolate force levels geometrically rather than additively. The operator $[\cdot]_+$ keeps the mixed shape nonnegative. We initialize the levels from quantiles of the training peaks, the shapes from the mean training profile, and the keys randomly. All prototype parameters are then learned jointly with the router as a mixture of experts (Jacobs et al., 1991). The *global prior* is the median measured peak and mean normalized profile over the training sequences. It randomly replaces the routed prior for 30% of training sequences as prior dropout and serves as the no-prototype control in Sec. 4.

Routing loss. During training, prototypes whose levels are closer to the target peak receive larger target weights. Let $\delta_k = (\log \max(F, \epsilon) - \mathrm{sg}(a_k))/\tau_r$ be the scaled distance to prototype k. These distances define target weights $\rho_k(F)$ and a cross-entropy loss that trains the predicted weights π_k:

$$\boldsymbol{\rho}(F) = \mathrm{softmax}_k(-\delta_k^2/2), \qquad \mathcal{L}_{\mathrm{route}} = -\sum_k \rho_k(F) \log \pi_k. \tag{2}$$

Through stop-gradient, we hold prototype levels fixed when forming these targets. The width τ_r controls how broadly credit is shared among nearby levels. Prototype values learn through sensory prediction, while this loss teaches the router which prototypes to favor. Together, these losses let the prototypes self-organize around measured effort rather than object categories. At inference, $\boldsymbol{\pi}$ is computed from the video features and, when supplied, the prompt; no measurement enters.

Instance adaptation. An adaptation head adjusts the shared prior to the observed interaction by predicting a scalar log-peak offset $d(\mathbf{g})$ and a shape offset $\boldsymbol{\eta}(\mathbf{g})$. The adapted estimates are $\bar{F}' = \bar{F} \exp(c \tanh(d(\mathbf{g})/c))$ and $\bar{\boldsymbol{\sigma}}' = [\bar{\boldsymbol{\sigma}} + \boldsymbol{\eta}(\mathbf{g})]_+$, where $c = \ln 2$ bounds the peak multiplier to a factor of two. A log-peak Huber loss and a shape squared-error loss jointly train the prototypes and adaptation head.

3.3 Joint conditional flow for motion and effort

A joint conditional flow predicts residual corrections to the trajectory proposal and adapted force prior, allowing motion and effort to exchange information during refinement. The resulting residual is decoded into the final trajectory, peak force, and force profile. The correction has three parts—waypoint offsets from the tracked trajectory, a log-scale change in peak force, and changes to the temporal shape—concatenated into one residual state:

$$\mathbf{r} = \left[\mathrm{vec}(\mathbf{X} - \mathbf{B}),\ \log \frac{\max(F, \epsilon)}{\bar{F}'},\ \boldsymbol{\sigma} - \bar{\boldsymbol{\sigma}}'\right] \in \mathbb{R}^{3N+1+L}. \tag{3}$$

The operator vec stacks the waypoint offsets into a vector. The adapted prior enters the target with gradients detached, so the flow learns residuals around a fixed reference and the prior is shaped only by its own loss. A DiT-style transformer velocity field (Peebles & Xie, 2023) processes the residual tokens jointly. It attends to $\mathbf{M}$ and is conditioned on $\mathbf{g}$, the tracked proposal, and motion-derived features. Waypoint tokens receive tracked motion around the contact, while force tokens receive speed, acceleration, and displacement statistics. This joint conditioning lets motion and effort inform one another without imposing a force–acceleration relation.

We train it with flow matching (Lipman et al., 2023) along the straight paths of rectified flow (Liu et al., 2023): with $\tilde{\mathbf{r}}$ the target standardized per coordinate, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and flow time $t \sim \mathcal{U}[0, 1]$

(distinct from interaction time),

$$\mathbf{y}_t = (1-t)\boldsymbol{\epsilon} + t\widetilde{\mathbf{r}}, \qquad \mathcal{L}_{\text{flow}} = \mathbb{E}\Big[\|v_\theta(\mathbf{y}_t, t; \mathcal{C}) - (\widetilde{\mathbf{r}} - \boldsymbol{\epsilon})\|_\Omega^2\Big], \tag{4}$$

where $\mathcal{C}$ is the video and motion context and Ω masks missing targets and balances the motion and sensory terms (Appendix A.3).

At inference, 20 Euler steps from $\mathbf{y}_0 = \mathbf{0}$ produce the residual. We add this residual to the trajectory proposal and adapted prior, clipping the log-peak change to $\pm b$, where $b = 0.5$. Inverse projection then recovers metric motion. Integration from the origin yields a single deterministic estimate; starting from sampled noise instead leaves the force errors unchanged within seed variation and worsens ATE by 0.2 cm, and averaging five noise draws matches the zero start (Appendix C.8).

3.4 Training and inference

Training transfers the few measured forces to human video through paired recordings, and inference needs only the clip, with the contact and the prompt supplied by a human or a VLM.

Training. Hoi! (Engelbracht et al., 2026) supplies tracked trajectories for human and gripper recordings (Sec. 4.1) and force measurements for the gripper. A human recording inherits the peak and normalized profile of the paired gripper recording of the same episode, possibly from another camera, as a down-weighted reference for the human's effort. The training loss is

$$\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda_{\text{peak}}\mathcal{L}_{\text{peak}} + \lambda_{\text{prior}}\mathcal{L}_{\text{prior}} + \lambda_{\text{route}}\mathcal{L}_{\text{route}} + \lambda_{\text{cons}}\mathcal{L}_{\text{cons}} + \mathcal{L}_{\text{aux}}, \tag{5}$$

Here, $\mathcal{L}_{\text{flow}}$ is defined in Eq. 4, $\mathcal{L}_{\text{peak}}$ is a Huber loss in newtons on the peak decoded in one step from the current flow state, and $\mathcal{L}_{\text{prior}}$ combines a log-peak Huber loss with shape squared error on the adapted prior. The routing loss $\mathcal{L}_{\text{route}}$ is defined in Eq. 2, $\mathcal{L}_{\text{cons}}$ is a cross-entropy that aligns the language-key scores with the detached video-based routing, and $\mathcal{L}_{\text{aux}}$ contains auxiliary scale, peak-count, and contact-localization terms. We set $\lambda_{\text{prior}} = 1$ and the other weights to 0.5 (Appendix A.4). Missing targets are masked. The few force targets are fit long before the trajectory converges. Once the force error has leveled off, a frozen exponential-moving-average copy of the model becomes the source of all force targets. This holds the force function fixed while trajectory training continues, allowing one checkpoint to serve both outputs (Appendix A.4).

Inference. Inference requires no force or torque measurements. The model takes a clip spanning one interaction; untrimmed recordings are delimited to the span in which the object moves, following the data pipeline of Sec. 4.1 (Appendix D). A manual click centers the window on the contact. Without a click, the vision–language model proposes the contact, optionally refined by a localization head. The motion field is then re-anchored on this proposal in a second pass (*self-click*). The object–action prompt can likewise be supplied or written by the same model from the frames (*self-prompt*); the questions are given in Appendix A.5. The torque model follows the same procedure with sensor-frame torque targets and its own prototypes.

4 Experiments

We evaluate trajectory recovery, force and torque estimation, and the contribution of the learned prototype prior and the joint flow (Sec. 4.2). The prior study also tests limited sensory supervision (Table 3); separate analyses examine paired transfer to human video and input guidance (Sec. 4.2). Real-world videos and a robot deployment examine predictions beyond the recorded dataset (Sec. 4.3); the data, protocol, and baselines are described first (Sec. 4.1).

4.1 Experiment Setup

Data. We train and evaluate on Hoi! (Engelbracht et al., 2026), which contains paired human and instrumented-gripper recordings of everyday articulated-object interactions viewed from two static cameras. Gripper recordings provide measured force and torque, while paired human recordings support transfer to bare-hand video (Sec. 3.4). Each sequence represents one interaction from one camera and one embodiment, sampled to 32 frames, cropped to the model's object window (Sec. 3), and annotated with the first-frame contact and the dataset prompt. Because occlusions can hide the contact or articulated motion, we verify contact visibility by a depth-ray test, and check whether at

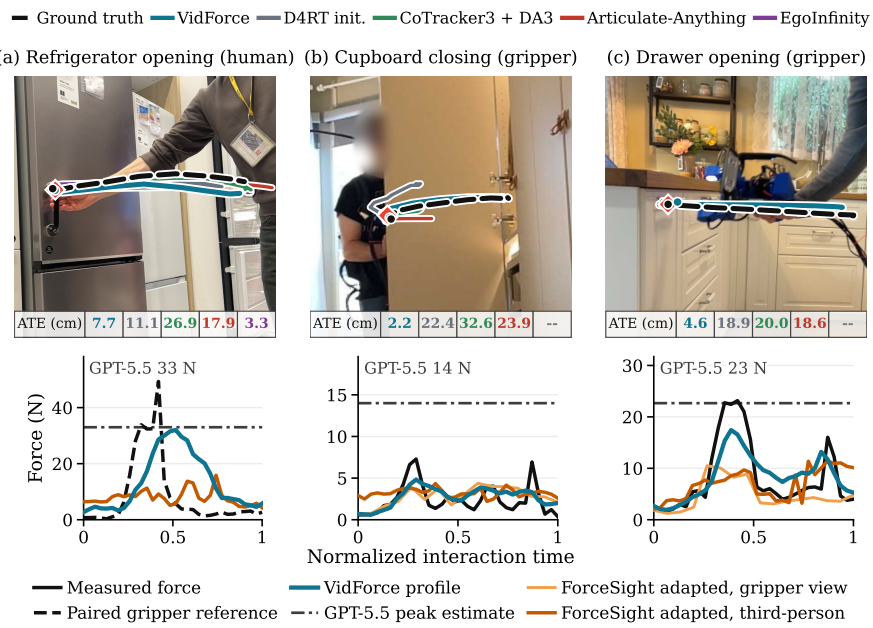


Figure 2: **Trajectory and force predictions.** Top: ground-truth and predicted contact trajectories in the object-centric window. All trajectory methods receive the manual click ◆, and the lower strip reports each method's ATE in cm using its line color. Bottom: predicted force profiles and peak estimates against measured force; the human sequence uses its paired gripper reference.

least half of the motion is in view, and the trajectory plausible.Using these checks, we retain 630 sequences (343 gripper and 287 human). Appendix B details the pipeline and rejected examples.

Training. We train on a train split of 531 sequences (295 gripper, 236 human), increasing data diversity through shifted crops, rescaling around the object, and flipping (Appendix B.4). Sequences with measured force targets have weight 1 and paired references weight 0.5 in the peak and shape terms. On 30 % of the training recordings, the contact is perturbed and the click-seeded tracks are dropped (click-free rows), so that the model also works from a proposed contact. More details can be found in Appendix B.4 and A.2.

Evaluation. The evaluation set (**Eval**) contains 99 sequences (48 gripper, 51 human) with a *manual click* and a *manual prompt*. It combines 47 sequences (23 gripper) from an *unseen room type*—the only living room among the 20 Hoi! scenes—and 52 sequences (25 gripper) from interactions with *held-out objects* in ten training scenes. *Self-click* and *self-prompt* (Sec. 3.4) replace the *manual click* and *manual prompt* with Qwen3-VL proposals using the same checkpoints. Cached self-click runs retain the annotated contact depth, so they evaluate the proposed image coordinate rather than fully annotation-free inference. Tracker baselines receive the same proposed point (Appendix C.1). Unless stated otherwise, every evaluation uses three training seeds and reports mean ± sample SD at the final checkpoint; deterministic trajectory baselines are run once.

Metrics. ATE (cm) is the mean waypoint error per sequence after translation-only alignment of the predicted and measured starts, reported as the median across sequences. Peak MAE and RMSE compare one predicted peak with one measured peak per instrumented sequence. Profile RMSE is the RMS error over the 32 normalized-time samples, aggregated by RMS across sequences. Force errors are reported in N and torque errors in N m.

Baselines. No prior method jointly predicts trajectory and force, so we use separate baselines.
All trajectory baselines receive the same clip and contact and are evaluated at the same 16 waypoints. D4RT provides the tracked initialization without learned refinement. CoTracker3 tracks the contact pixel and lifts it to 3D using DA3 depth. Articulate-Anything (Le et al., 2025) reconstructs an articulated asset from the video; we register it to the contact and CoTracker3 track, then animate its predicted joint range at constant speed. EgoInfinity (Wang et al., 2026a) reconstructs the hand and uses the joint nearest the contact; we evaluate it only on human interactions.
For force, GPT-5.5 (OpenAI, 2026) estimates the peak from eight frames, with or without object–

Table 1: **Trajectory estimation.** ATE for manual and Qwen3-VL contacts. VidForce is best for both embodiments.

ATE (cm) ↓	All (99)	Human (51)	Gripper (48)
Manual click			
Stay at contact (ref.)	17.5	18.7	13.2
Category mean (ref.)	16.5	19.1	13.0
D4RT init.	11.8	11.1	12.1
CoTracker3 + DA3	13.3	14.2	13.1
Articulate-Anything (adapted)	17.5	17.3	18.7
EgoInfinity (hands)	–	15.1	–
VidForce	6.6±0.4	7.9±0.8	5.6±0.2
Self-click (Qwen3-VL-proposed contact)			
D4RT initialization	12.0	13.3	11.6
CoTracker3 + DA3 (per-frame depth)	12.4	16.5	11.4
VidForce	7.0±0.3	8.8±0.4	5.8±0.2

Table 2: **Force and torque estimation.** GPT-5.5 uses eight frames. †Gripper-view ForceSight is scored once per episode.

Error ↓	Peak MAE	Peak RMSE	Profile RMSE
Force (N)			
GPT-5.5	6.85	11.88	–
GPT-5.5 + text	6.98	11.43	–
ForceSight: gripper view†			
Zero-shot	12.33	19.53	–
Adapted	7.56±0.28	10.84±0.35	4.71±0.16
ForceSight: third-person view			
Zero-shot	13.75	20.96	–
Adapted	10.30±0.75	16.60±0.92	7.55±0.27
VidForce	4.81±0.38	5.92±0.42	4.95±0.20
Sensor-frame torque (N m)			
GPT-5.5	0.693	0.842	–
GPT-5.5 + text	0.550	0.725	–
VidForce	0.301±0.021	0.384±0.023	0.315±0.003

Table 3: **Limited force supervision.** Peak MAE (N) with force labels on a share of the training episodes; at 50 %, the prototypes beat the global prior at 100 % (bold). Full metrics: Table 21.

Force labels	12 %	25 %	50 %	100 %
Global prior	8.48±0.25	7.20±0.75	6.44±0.25	**5.61±0.28**
Prototypes (VidForce)	7.53±0.37	6.27±0.46	**5.23±0.28**	4.81±0.38

action text. ForceSight (Collins et al., 2024) predicts the force a robot should apply from one RGB-D frame and a prompt; its released model was trained on a robot wrist camera. We evaluate it in the Hoi! gripper-mounted view—closest to its training setting—and in our third-person view. We test the released checkpoint zero-shot, a fine-tuned version on our data, and an adapted variant that replaces the force-goal output with a per-frame force prediction. Appendix C.2 details all baselines.

4.2 Results and Discussion

Trajectory estimation. VidForce achieves the lowest ATE among all baselines, reducing the error from 11.8 cm for its D4RT initialization to 6.6 ± 0.4 cm (Table 1). Both video-free references and Articulate-Anything remain substantially worse, showing that neither category-level motion nor a registered articulation alone recovers the observed instance trajectory.

The largest gains occur when contact-seeded tracking fails to follow the object, particularly under gripper occlusion. The same ordering holds for human and gripper recordings and for Qwen3-VL-proposed contacts (Table 1). The examples in Fig. 2 illustrate this pattern. EgoInfinity follows the visible hand during the refrigerator opening (a), but the trackers and registered asset miss the cupboard motion (b), and gripper occlusion leaves the drawer baselines near their starting point (c). VidForce recovers the object motion in both cases.

Force and torque estimation. VidForce achieves the lowest peak and profile errors among the third-person force methods (Table 2), reducing peak MAE from 6.85 N for GPT-5.5 to 4.81 ± 0.38 N. Errors remain concentrated in the heaviest interactions, which the model tends to underestimate. ForceSight is strongly affected by camera viewpoint. Its released model predicts nearly constant low force goals. The adapted variant calibrates the force scale and adds a profile, but remains less accurate than VidForce in the third-person view (Table 2; Fig. 2; Appendix C.2). Table 2 also evaluates sensor-frame torque, validating the applicability of our general approach to other modalities.

How data-efficient is VidForce? Force data is expensive to acquire. We test data-efficiency by retraining the learned prototypes and the global prior by partially withholding measurements from training episodes, stratified by object and action; withheld episodes keep their videos and motion labels but contribute no force targets, paired references, or prior statistics (Table 3). We observe that even with a quarter of the measurement data, VidForce still outperforms all baselines of Table 2.

Are prototype priors the best architectural alternative? With instance adaptation and joint flow fixed, learned prototypes with routing supervision outperform both a parameter-matched MLP prior,

Table 4: **Ablations.** Each block varies one design choice with the rest fixed; the full model (last row) is the reference for every block except the motion cues, which have their own control, and the human-video block, whose force errors are scored against the paired gripper force references. Peak RMSE and full settings in Appendix C.

Variant	ATE (cm) ↓	Peak MAE (N) ↓	Profile RMSE (N) ↓
Prior type			
Direct MLP prior (matched size)	6.6±0.4	5.79±0.17	5.72±0.02
Global prior + adaptation	6.7±0.1	5.61±0.28	5.73±0.05
Prototypes, routing loss off	6.9±0.4	5.64±0.21	5.53±0.15
Word list (categorical), LLM-assigned labels	7.1±0.5	4.86±0.25	5.00±0.04
Word list (categorical), annotated labels	7.1±0.4	4.94±0.24	5.15±0.07
Flat clusters (k-means, K=10)	6.9±0.3	5.32±0.34	5.25±0.05
Human video			
With paired gripper force references (control)	7.9±0.8	10.64±0.50	7.88±0.17
Without paired gripper force references	7.5±0.5	18.79±4.14	9.56±1.45
VidForce (full model)	6.6±0.4	4.81±0.38	4.95±0.20

Variant	ATE (cm) ↓	Peak MAE (N) ↓	Profile RMSE (N) ↓
Prediction head			
One-shot regression	10.3±1.3	6.73±0.38	6.47±0.19
Split trajectory / force flows	6.9±0.5	5.33±0.46	5.23±0.13
Motion cues			
All motion cues (control)	7.6±0.5	5.38±0.23	5.38±0.20
No dense motion field	10.1±0.6	6.98±0.34	6.37±0.14
Guidance (full model uses manual click + manual prompt)			
Self-click + manual prompt	7.0±0.3	4.53±0.42	–
Manual click + self-prompt	6.8±0.5	5.85±0.10	–
Self-click + self-prompt	7.7±0.7	5.69±0.39	–
VidForce (full model)	6.6±0.4	4.81±0.38	4.95±0.20

which maps the video and prompt features directly to a peak and profile, and the shared global prior of Sec. 3.2 in peak and profile error (Table 4, *Prior type*). Removing routing supervision raises peak MAE from 4.81 to 5.64 N, indicating that prototype capacity alone does not account for the gain. Appendix C.4 confirms these findings in an extended ablation of the learned routing.

Does the joint flow improve the correction, and what do the motion cues contribute? Flow matching improves prediction over one-shot regression, which predicts the same residual in a single step with the same network (Table 4, *Prediction head*): The joint flow also improves on separate trajectory and force streams that share the conditioner but do not attend to each other (4.81 vs. 5.33 N peak MAE). The dense motion field contributes to both outputs: removing it significantly increases force prediction errors. Appendix C.6 reports the complete comparison.

Does paired supervision improve force transfer to human video? Paired supervision carries measured gripper force to human video (Table 4, *Human video*; Appendix C.9 for more): without it, the human-video peak disagreement with the paired gripper references rises from 10.64 to 18.79 N and the profile error from 7.88 to 9.56 N, while the trajectory error stays similar (7.9 vs. 7.5 cm).

Can the click and prompt be automated? Both generated inputs keep force prediction close to the manual setting (Table 4, *Guidance*). The automated self-click has errors within std of the full model, while the automated prompt slightly reduces force prediction accuracy.

How do learned prototypes compare with other prior types? Learned prototypes and the word list (categorical), which groups the training measurements by four object names and two actions that a language model reads from the prompt, give comparable force accuracy (Table 4, *Prior type*). Therefore, there is no added benefit to explicitly categorizing interactions. Flat clusters, which assign each prompt to its nearest of ten k-means clusters, are less accurate than the other options.

4.3 Real-World Interactions and Robot Deployment

We test VidForce beyond Hoi! through separate video and robot studies (protocols in Appendix D).

Real-world videos. We record seven phone videos of a person repeatedly opening and closing six kitchen objects—a refrigerator door, freezer, cabinet door, drawer, oven door, and dishwasher door—at varying speeds and extents. We divide them into 46 episodes of 32 frames, each spanning contact to the end of motion. We evaluate action and motion direction, consistency across repetitions of different durations, and agreement between predicted peaks and the scale readings.

With self-click and self-prompt, VidForce produces distinct force estimates across actions and objects (Fig. 3b). Every object receives a higher predicted peak for opening than for closing, with opening peaks ranging from 21 N for the cabinet door to 43 N for the oven door. The routed prior assigns 31 N to 20 of these 21 openings, so instance adaptation and joint refinement produce most of the variation between objects. Prompt generation reads the action correctly in 36 of the 46 episodes, with most misreads on refrigerator closings.

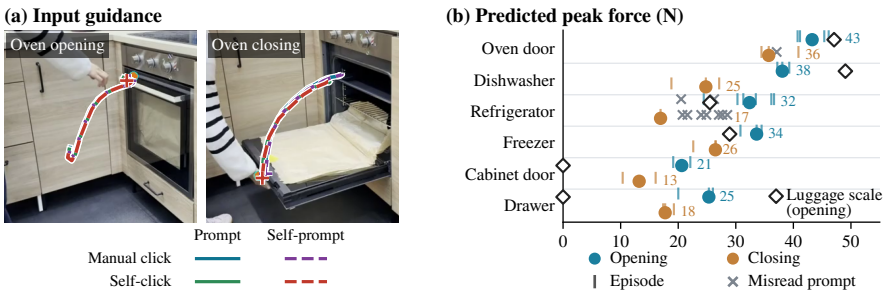


Figure 3: **Real-world videos.** (a) Predicted trajectories under the four input combinations; plus signs mark the contact used. (b) Predicted peak forces with self-click and self-prompt; medians exclude misread prompts. Diamonds mark the opening force measured with a luggage scale (openings only).

Table 5: **Spot trials.** Left: Spot opening the oven door and drawer. Right: successes out of five per viewpoint, without and with the predicted force; failures: F force, T trajectory, P partial opening.



	Drawer (prismatic)			Oven door (revolute)	
Viewpoint	Load	Without force	With force	Without force	With force
1	empty	5/5	5/5	0/5 (4F, 1T)	3/5 (1P, 1T)
2	∼5 kg	4/5 (1T)	4/5 (1T)	0/5 (4F, 1T)	4/5 (1T)

Figure 3a qualitatively compares the predicted trajectories and force profiles for oven opening and closing. The four input combinations produce nearly identical trajectories and similar peak estimates. For closing, the predicted peak varies slightly more between 37 to 41 N.

Robot deployment. Spot records human demonstrations with its arm camera, and VidForce turns them into manipulation plans with self-click and self-prompt. We evaluate both joint types, a prismatic drawer and a revolute oven door, from two viewpoints, executing each plan with and without the predicted force profile as an additional controller signal (Table 5; Appendix D.3). The predicted trajectories carry over across joint types and viewpoints, with similar success from both viewpoints. The force profile matters most for demanding interactions: it enables Spot to open the high-effort oven door, whereas the low-effort drawer opens reliably from the trajectory alone, even when loaded.

4.4 Limitations and Future Directions

VidForce estimates typical interaction effort. Direct human-force measurements would complement paired gripper supervision and support person-specific predictions, while broader force data would help distinguish visually similar objects with different resistance and refine estimates for high-effort interactions. Our evaluation focuses on visible interactions, with metric scale estimated from monocular depth. *Future work* will expand object-centric data to interactions with distinct force signatures, such as stapler use and USB insertion at varying speeds. We plan to extend tracking to moving cameras and multiple viewpoints, and explore joint trajectory and force prediction from a single image.

5 Conclusion

VidForce brings sensory estimation to articulated-object understanding by jointly predicting a contact trajectory and force profile from third-person video. Learned physical prototypes share limited instrumented measurements across interactions of comparable effort; geometric and vision–language features adapt this prior to the observation, and a conditional flow refines it together with a tracked trajectory. VidForce improves trajectory recovery over every baseline and achieves the lowest force errors among third-person methods on held-out Hoi! interactions. A separately trained torque model further shows that the formulation can extend to other sensory quantities. The model also produces consistent motion and force predictions on real-world videos, and its outputs can be exported as manipulation plans for Spot. These results show how limited instrumented data can enrich ordinary interaction videos with estimates of both how objects move and how much effort their manipulation may require.

AI USE STATEMENT

LLM-based assistants were used while preparing this paper. They helped edit the text for clarity and consistency, draft and revise parts of sections under the authors' direction, write scripts that produce figures and tables from the logged results, and format references. Coding agents were used in implementing and debugging training, evaluation, and baseline code and in running experiments; the authors reviewed this code and checked the reported results against the experiment logs. Separately, Qwen3-VL and GPT-5.5 are components of VidForce and baselines in our experiments, as described in Sections 3 and 4. The authors take full responsibility for the content of this paper.

REFERENCES

Ademi Adeniji, Zhuoran Chen, Vincent Liu, Venkatesh Pattabiraman, Raunaq Bhirangi, Siddhant Haldar, Pieter Abbeel, and Lerrel Pinto. Feel the force: Contact-driven learning from humans. *arXiv preprint arXiv:2506.01944*, 2025.

Shuai Bai et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K. Su. This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Hanzhi Chen, Boyang Sun, Anran Zhang, Marc Pollefeys, and Stefan Leutenegger. VidBot: Learning generalizable 3d actions from in-the-wild 2d human videos for zero-shot robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27661–27672, 2025. doi: 10.1109/CVPR52734.2025.02576.

Jeremy A. Collins, Cody Houff, You Liang Tan, and Charles C. Kemp. ForceSight: Text-guided mobile manipulation with visual-force goals. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10874–10880, 2024.

Eadom Dessalene, Botao He, Michael Maynord, Yonatan Tussa, Pavan Mantripragada, Yianni Karabatis, Nirupam Roy, and Yiannis Aloimonos. FEEL (force-enhanced egocentric learning): A dataset for physical action understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science. Springer, 2026. doi: 10.1007/978-3-032-37602-2_31.

Kiana Ehsani, Shubham Tulsiani, Saurabh Gupta, Ali Farhadi, and Abhinav Gupta. Use the force, luke! learning to predict physical forces by simulating effects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 221–230, 2020.

Tim Engelbracht, René Zurbrügg, Matteo Wohlrapp, Martin Büchner, Abhinav Valada, Marc Pollefeys, Hermann Blum, and Zuria Bauer. Hoi! – a multimodal dataset for force-grounded, cross-view articulated manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8880–8890, June 2026.

Letian Fu, Gaurav Datta, Huang Huang, William Chung-Ho Panitch, Jaimyn Drake, Joseph Ortiz, Mustafa Mukadam, Mike Lambeta, Roberto Calandra, and Ken Goldberg. A touch, vision, and language dataset for multimodal alignment. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235, pp. 14080–14101. PMLR, 2024. URL `https://proceedings.mlr.press/v235/fu24b.html`.

Gemini Team, Google. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79. URL `https://www.cs.toronto.edu/~fritz/absps/jjnh91.pdf`.

Vidhi Jain, Maria Attarian, Nikhil J. Joshi, Ayzaan Wahid, Danny Driess, Quan Vuong, Pannag R. Sanketi, Pierre Sermanet, Stefan Welker, Christine Chan, Igor Gilitschenski, Yonatan Bisk, and Debidatta Dwibedi. Vid2robot: End-to-end video-conditioned policy learning with cross-attention transformers. In *Robotics: Science and Systems (RSS)*, 2024.

Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–10, 2025. URL `https://arxiv.org/abs/2410.11831`.

Justin Kerr, Chung Min Kim, Mingxuan Wu, Brent Yi, Qianqian Wang, Ken Goldberg, and Angjoo Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. In *Conference on Robot Learning (CoRL)*, 2024.

Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. Articulate-Anything: Automatic modeling of articulated objects via a vision-language foundation model. In *International Conference on Learning Representations (ICLR)*, pp. 17578–17602, 2025. URL https://arxiv.org/abs/2410.13882.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations (ICLR)*, 2026.

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 (ICLR)*, 2023. URL https://arxiv.org/abs/2210.02747.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Thomas Lucas, Fabien Baradel, Philippe Weinzaepfel, and Grégory Rogez. PoseGPT: Quantization-based 3D human motion generation and forecasting. In *European Conference on Computer Vision (ECCV)*, 2022.

OpenAI. GPT-5.5 (model snapshot gpt-5.5-2026-04-23). Chat completions API, 2026. Accessed September 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182, 2023.

Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. WiLoR: End-to-end 3D hand localization and reconstruction in-the-wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Andreas F. Reichelt, Alyssa M. Ash, Lee A. Baugh, Roland S. Johansson, and J. Randall Flanagan. Adaptation of lift forces in object manipulation through action observation. *Experimental Brain Research*, 228(2):221–234, 2013. doi: 10.1007/s00221-013-3554-9.

RHOS Team. Open-d4rt: An open reimplementation of D4RT. https://github.com/Lijiaxin0111/Open-d4rt, 2026. Apache-2.0; checkpoint OpenD4RT_32CLIP_9Dataset_NoAUG, accessed 2026-09-03.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6), 2017.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Gaotian Wang, Kejia Ren, Andrew Morgan, Yiting Chen, Howard H. Qian, Podshara Chanrungmaneekul, and Kaiyu Hang. EgoInfinity: A web-scale 4d hand-object interaction data engine for any-view robot retargeting and video-to-action robot learning. *arXiv preprint arXiv:2606.17385*, 2026a.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate monocular geometry with metric scale and sharp details. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

Zhaoning Wang, Yi Zhong, Jiawei Fu, Henrik I. Christensen, and Jun Gao. AFUN: Towards an affordance foundation model for functionality understanding. *arXiv preprint arXiv:2606.02551*, 2026b.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. In *Robotics: Science and Systems (RSS)*, 2024.

Yuhang Wu, Lingqi Zeng, Changwei Jing, Jianglong Ye, and Xiaolong Wang. ReForce: Learning force-aware retargeting for dexterous manipulation. *arXiv preprint arXiv:2608.15560*, 2026.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A SimulAted part-based interactive ENvironment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Fengyu Yang, Chenyang Ma, Jiacheng Zhang, Jing Zhu, Wenzhen Yuan, and Andrew Owens. Touch and go: Learning from human-collected vision and touch. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2211.12498.

Zhao-Heng Yin, Sherry Yang, and Pieter Abbeel. Object-centric 3D motion field for robot learning from human videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K. Barral, Raia Hadsell, Zoubin Ghahramani, Andrew Zisserman, Junlin Zhang, and Mehdi S. M. Sajjadi. Efficiently reconstructing dynamic scenes one D4RT at a time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7382–7392, 2026.

Jianyi Zhou, Ziteng Gao, Feiyang Hong, Zirui Liu, Guannan Zhang, Weisheng Dai, Ruichen Zhen, Chuqiao Lyu, Haotian Wu, Yinian Mao, Xushi Wang, Yuxiang Jiang, Wenbo Ding, and Shuo Yang. TouchAnything: A dataset and framework for bimanual tactile estimation from egocentric video. *arXiv preprint arXiv:2605.13083*, 2026. URL https://arxiv.org/abs/2605.13083.

APPENDIX CONTENTS

A Implementation Details **14**
A.1 Inputs and representation . . . 14
A.2 Prototype prior . . . 15
A.3 Joint conditional flow . . . 15
A.4 Training . . . 17
A.5 Inference with generated inputs . . . 18
B Data and Processing **18**
B.1 Source recordings . . . 18
B.2 Quality filter . . . 19
B.3 Sequences, splits, and pairs . . . 19
B.4 Training windows and augmentation . . . 21
C Supplementary Evaluation **21**
C.1 Evaluation protocol . . . 21
C.2 Baseline implementation . . . 22
C.3 Dataset partitions and re-framing . . . 26
C.4 Contribution of the learned prior . . . 28
C.5 Prior contribution with limited force supervision . . . 28
C.6 Joint prediction and motion cues . . . 29
C.7 Additional ablations and prototype count . . . 30
C.8 Inference start point . . . 30
C.9 Transfer to human videos . . . 30
C.10 Prior types . . . 31
C.11 Additional torque results . . . 32
C.12 Supplied and generated guidance . . . 32
C.13 Additional qualitative results . . . 34
D Real-World Interactions and Robot Deployment **34**
D.1 Real-world videos . . . 35
D.2 Manual and generated inputs . . . 35
D.3 Robot deployment . . . 36

A IMPLEMENTATION DETAILS

A.1 INPUTS AND REPRESENTATION

Object-centric video clip. The input is a square window of the recording around the manipulated object, sampled at $T = 32$ frames over the interaction; the frozen models run on this window, and $\mathbf{K}$ denotes its intrinsics. For recorded evaluations with a manual click the window is sized from the annotated articulation ($1.3\times$ its largest image-space reach from the contact, at least $0.4\times$ the frame's longer side); in the automatic setting it spans $0.55\times$ the longer side around the proposed contact. In training the window is shifted and rescaled so that the contact does not sit at a fixed position (Appendix B.4).

Frozen models and memory tokens. Three frozen models turn the clip into the tokens of Table 6. ***D4RT***, through its open reimplementation (RHOS Team, 2026), tracks a 32×32 lattice of points and a block of points seeded at the contact through the 32 frames. The lattice tracks become a motion field of 18 channels per cell and frame (displacements, contact offsets, visibility, confidence, normals, and depth consistency); a temporal convolution reduces it to 8 steps, and a 4×4 patchification turns it into 64 field tokens, with 8 timeline tokens summarizing the motion-weighted sequence. The contact-seeded tracks yield two contact-track tokens and the tracked initialization $\mathbf{B}$, and D4RT's encoder features, pooled to $8 \times 4 \times 4$, enter as 128 tracker tokens. ***DA3*** (Lin et al., 2026) provides the depth map, the intrinsics $\mathbf{K}$, and the contact depth z_c; a scale token carries the metric-scale fit and per-frame scale statistics. ***Qwen3-VL-8B*** (Bai et al., 2025) provides 16 pooled vision tokens on each of 8 keyframes, a four-token embedding of the prompt, and the contact proposal that serves as the contact cue. It also produces the 258-dimensional prompt embedding $\mathbf{e}(q)$ read by the router: the square roots of the probabilities it assigns to a fixed vocabulary of 256 answer tokens and to the two actions when asked, given the prompt, which object part is manipulated and whether it is opened or closed.

A fusion transformer reads all of these tokens and returns the context $\mathbf{M}$; the pooled features $\mathbf{g}$ are the mean over its output tokens. It has three pre-norm layers of width 512 with eight attention heads, a feed-forward width of 1024, and dropout 0.1.

Table 6: **Memory tokens of the fusion transformer.** One row per token type of the object-centric clip: its count, the frozen model or signal that produces it, and what it encodes. The D4RT tracks yield the field, timeline, contact-track, and tracker tokens, DA3 the scale token, and Qwen3-VL the vision, prompt, and contact-cue tokens.

Token	Count	Source	Encodes
Field	64	D4RT lattice tracks	tracked motion field, 18 channels per cell and frame
Timeline	8	D4RT lattice tracks	motion-weighted temporal sequence
Contact tracks	2	contact-seeded D4RT tracks	mean track curve and its dispersion around the contact
Scale	1	DA3 depth and tracks	metric-scale fit and per-frame scale statistics
Tracker features	128	D4RT encoder, pooled to $8 \times 4 \times 4$	learned geometry and motion representation
Vision	128	Qwen3-VL, 16 pooled tokens on 8 keyframes	appearance and semantics
Prompt	4	Qwen3-VL text embedding	the object–action prompt
Contact cue	1	Qwen3-VL proposal	proposed hand or gripper point; dropped with probability 0.5 in training
Embodiment and mode	1	metadata	human or gripper; click-free and prior-free flags

Coordinates and tracked initialization. A camera-space waypoint $\mathbf{p}_i = (X_i, Y_i, Z_i)^\top$ projecting to pixel (u_i, v_i) of the window is encoded relative to the contact as

$$\mathbf{x}_i = \big(u_i/(w-1) - c_u,\ v_i/(h-1) - c_v,\ \log(Z_i/z_c)\big)^\top, \tag{6}$$

with (w, h) the window size in pixels, (c_u, c_v) the normalized contact coordinates, and z_c the DA3 depth at the contact; $\mathbf{K}$ and z_c invert the encoding to metric coordinates. The tracked initialization takes, for each waypoint, the componentwise median over the contact-seeded tracks $\mathcal{S}_i$ visible at both the contact frame t_c and the waypoint frame t_i (at least eight):

$$\mathbf{b}_i = \operatorname*{med}_{s \in \mathcal{S}_i} \left[\mathbf{u}_s(t_i) - \mathbf{u}_s(t_c),\ \log \frac{z_s(t_i)}{z_s(t_c)} \right]. \tag{7}$$

For human recordings the lattice cells within 0.06 of the contact replace the seeded tracks, which the hand occludes more often; waypoints without surviving tracks copy the nearest tracked waypoint or stay at the contact.

A.2 Prototype prior

The prior of Sec. 3.2 is trained with the settings in Table 7. All prototypes are initialized from the *global prior*, the median measured peak and the mean normalized profile of the training sequences, which also stands in for the routed prior on prior-dropout rows and serves as the no-prototype control in Sec. 4; the levels are then spread to the quantiles of the measured training peaks, so the prototypes start as a force ladder. The number of prototypes, $K = 10$, is the largest tested value in the sweep of Table 24 at which every prototype stays in use.

A.3 Joint conditional flow

The flow of Sec. 3.3 operates on a residual state of 49 tokens: 16 waypoint tokens (three coordinates each, relative to the tracked proposal), one peak token (the log ratio to the adapted prior peak), and 32 shape tokens (one per profile sample, relative to the adapted shape). Table 8 lists its settings; the target uses the adapted prior with gradients detached, so the flow learns residuals around a fixed reference and the prior is shaped only by its own loss.

Table 7: **Settings of the prototype prior.** Initialization and size of the prototypes, the router and its two losses, the instance-adaptation head with its bound and loss, and prior dropout, which substitutes the global prior on a share of the training sequences.

Setting	Value
Prototypes	$K = 10$ (sweep in Table 24); levels start at evenly spaced quantiles of the measured training peaks (5.0 to 49.3 N), shapes at the mean normalized training profile, keys (258-dimensional) at random
Router	two-layer network of width 512 on $\mathbf{g}$ giving one score per prototype, plus the key cosine with a learned scale β (initialized at 10); soft assignment in training and at inference
Routing loss	$\tau_r = 0.3$; all recordings with a force target; weight 0.5
Consistency loss	cross-entropy of the key scores toward the prototype selected by the video score (target detached); recordings with a prompt; weight 0.5
Adaptation head	two-layer network of width 512 on $\mathbf{g}$ giving the log-peak offset and the L shape offsets; zero-initialized output; bound $c = \ln 2$, i.e. the peak changes by at most a factor of two
Adaptation loss	Huber on the log peak (threshold 0.5) plus squared error on the shape; weight 1
Prior dropout	30 % of training sequences use the global prior instead of the routed prior, with the adaptation still applied

Table 8: **Settings of the joint conditional flow.** The residual state and its standardization, the DiT-style velocity network, what the waypoint and force tokens read, the loss weights Ω and the one-step peak term, and the sampling and peak-bound settings used at inference.

Setting	Value
State	81 coordinates in 49 tokens, standardized per coordinate over the training sequences; unsupervised tokens are masked
Network	4 transformer blocks, width 512, 8 heads: self-attention over the tokens, cross-attention to $\mathbf{M}$, modulation by $\mathbf{g}$ and the flow time; zero-initialized outputs
Token inputs	waypoint tokens read the tracked proposal and 13 channels of tracked motion around the contact; force tokens read speed, acceleration, and displacement statistics of the tracked proposal
Loss weights Ω	span mask on the waypoints; force weight 1 (measured), 0.5 (paired reference), 0 (none), scaled by $\sqrt{\text{peak}/\text{global peak}}$ within $[0.25, 4]$; force term counted twice
One-step peak term	Huber loss (5 N) on the peak decoded from the one-step estimate, weighted by t; weight 0.5
Sampling	20 Euler steps from zero; one deterministic sample
Peak bound b	0.5 in log space

Decoding. After the standardization is reversed, the waypoint residual is added to the tracked proposal and mapped to metric coordinates with Eq. 6, and the force outputs are

$$\widehat{F} = \bar{F}' \exp(\mathrm{clip}(\widehat{\rho}, -b, b)), \qquad \widehat{\mathbf{f}} = \widehat{F}\,[\bar{\boldsymbol{\sigma}}' + \widehat{\boldsymbol{\delta}}]_+, \tag{8}$$

with $\widehat{\rho}$ the predicted log-peak residual and $\widehat{\boldsymbol{\delta}}$ the predicted shape residuals. The shape is not renormalized after decoding, so $\widehat{F}$ can differ from the maximum of $\widehat{\mathbf{f}}$; peak and profile errors are therefore reported separately.

A.4 Training

Every training sequence supervises the trajectory. Force targets come from the instrumented gripper recordings and, for human recordings that have a gripper recording of the same episode, from that paired recording at half weight; human recordings without a partner supervise the trajectory only. The model has 42.9 M trainable parameters on top of the frozen backbones. All training and evaluation were performed on four NVIDIA A100 GPUs. Tables 9 and 10 give the schedule and the loss terms.

Table 9: **Training schedule.** Optimizer and learning-rate schedule, the number of updates and batch size, weight averaging, the point at which the frozen teacher takes over the force targets, and the dropout and augmentation applied to the training sequences.

Setting	Value
Optimizer	AdamW, learning rate 5×10^{-4}, weight decay 0.01 (prototypes exempt); linear warm-up over the first 5 % of training, then cosine decay to zero
Length and batch	12,000 updates with batches of 128 (32 per GPU); three seeds; the final checkpoint is used
Weight averaging	exponential moving average of the weights with decay 0.999; evaluation uses the averaged weights
Force teacher	at update 2000 the averaged weights are frozen as a teacher whose predictions become the force target of every training sequence, all with the same weight, for the rest of training
Dropout	encoder 0.1; prompt 0.2; prior 0.3 (the global prior is substituted); contact cue 0.5; click-free path on 30 % of recordings
Augmentation	shifted and rescaled windows with mirror images, contact and depth jitter, horizontal flips (Appendix B.4)

Table 10: **Loss terms of Eq. 5.** Each term with its definition and weight: the flow-matching loss with the one-step peak term, the prior losses (adapted prior, routing, consistency), and the auxiliary scale, peak-count, and contact-localization terms. Targets that a sequence lacks are masked.

Term	Definition	Weight
Flow	velocity error of Eq. 4 with the weights of Table 8	1
One-step peak	Huber loss in newtons (threshold 5 N) on the peak decoded from the one-step estimate	0.5
Adapted prior	Huber loss on the log peak (threshold 0.5) plus squared error on the shape	1
Routing	cross-entropy of Eq. 2	0.5
Consistency	cross-entropy of the key scores toward the video-selected prototype	0.5
Metric scale	smooth-ℓ_1 error on the log scale of the tracked depth	1
Peak count	cross-entropy over the number of force peaks	0.2
Contact localization	cross-entropy over the 64 field tokens toward a Gaussian heatmap at the contact	0.5

Why a force teacher. Measured during development: without a teacher, the validation force error reached its minimum early in training and had risen by about 15 % by the time the trajectory error converged, so no single checkpoint served both outputs, and freezing the force weights alone did not help because the shared features keep moving. The teacher step is therefore fixed early, where the

force error had leveled off in those runs. With the teacher, the force error keeps decreasing slowly after the switch while the trajectory error still falls by roughly half, and the final checkpoint is used for both outputs.

A.5 Inference with generated inputs

At inference the model needs the clip and, optionally, a contact click and a prompt; either can be generated by the frozen Qwen3-VL-8B. The evaluation protocol for generated inputs is given in Appendix C.12.

Self-click. The contact is the point Qwen3-VL proposes for where the hand or gripper touches the object. The object-centric window is re-centered on it (side $0.55\times$ the longer image side) and the tracks and trajectory proposal are recomputed there. A localization head can refine the proposal within 1.5 grid cells, after which the motion field is re-anchored in a second pass; the head is trained on the click-free training rows, where the contact is perturbed by 1.5 grid cells and the click-seeded track tokens are dropped.

Self-prompt. The prompt is written by the same model. For the recorded evaluations it is composed from the model's answers about the manipulated part and its opening degree at the start and end of the interaction; for the real-world and robot videos it is the model's answer to one free-form question over eight frames. A generated prompt is encoded exactly like a supplied one.

Questions. The questions below are used verbatim; nothing is fine-tuned.

Contact proposal (self-click). Asked on the first frame of the interaction; ⟨action⟩ holds the supplied prompt or, in the self-prompt setting, the model's own prompt. The answer is a point in normalized image coordinates.

> The image shows a person's hand or a robot gripper about to ⟨action⟩. Point to the exact pixel where the hand or gripper touches (or is about to touch) the object it manipulates—the contact point on the handle, knob, edge or surface. Answer with JSON only: {"point_2d": [x, y]}

Self-written prompt, recorded evaluations (self-prompt). Two questions on windows around the proposed contact at the start and the end of the interaction. The action is the sign of the change in opening degree (at least 15 points; smaller changes fall back to a two-image comparison), and the prompt is composed as "⟨action⟩ ⟨attributes⟩ ⟨part⟩", for example "open white toilet lid".

> (1) These two images are the first and the last frame of one short manipulation of an articulated object by a person's hand or by a robot gripper. Name the object part that is manipulated (the part that moves: a door, drawer, lid, . . .). Answer with JSON only [fields *object* and *attributes*].
>
> (2) Look at the ⟨part⟩ in this image. How open is it right now? Judge only the current geometry of the part: 0 means fully closed (flush with its frame or fully pushed in), 100 means fully open (swung or pulled out as far as it goes). Answer with JSON only: {"open_percent": ⟨integer from 0 to 100⟩}

Self-written prompt, real-world videos and robot deployment. One free-form question over eight frames spanning the episode; the answer is used as the prompt without editing.

> The frames are in temporal order and show a person acting on an object in a home. In one short imperative sentence, say what the person does and to which part of which object, as an prompt someone could follow. Answer with the prompt only.

B Data and Processing

B.1 Source recordings

Hoi! (Engelbracht et al., 2026) captures people and a hand-held, instrumented gripper operating the same articulated furniture and appliances, filmed by two static third-person cameras. We use the exocentric (third-person) recordings of the hand and Hoi!-gripper embodiments that were available on the Hoi! servers at the time of this work: 25 sessions in 20 indoor scenes, 18 of which have

gripper recordings and therefore force measurements. A scene, session, and interaction span define an *episode*; our unit, a *sequence*, is one camera view of an episode in one embodiment (Hoi! counts the two views as one sequence). Nearly every episode was performed by both embodiments, so it has up to four third-person recordings, 1,434 recordings of 395 episodes in total. Gripper recordings carry a tracked tool trajectory and force–torque measurements, human recordings a tracked hand trajectory and no force (below). For 191 recordings the release lacks an input the pipeline needs (gripper calibration or poses, articulation annotation, or hand landmarks); they are set aside before any quality assessment, a matter of availability rather than quality.

Ground-truth trajectories. All Hoi! streams share one world frame (QR-tag time synchronization, visual localization against a laser scan of the scene). Gripper: the SLAM pose of the Aria device mounted on the gripper, mapped by the calibrated device-to-tool transform to the tool point; the force–torque sensor supplies the wrench in its own frame. Human: the thumb fingertip of the manipulating hand from the Aria glasses' hand tracking (landmark 0), mapped from the device to the world frame by the glasses' SLAM pose. Both are expressed in the iPhone camera frame through its aligned pose at the observation frame and resampled to 16 waypoints over the interaction span.

Force and torque targets. The gripper carries a six-axis Bota force–torque sensor; the release provides its compensated and filtered external wrench, and the gripper's contact trigger marks when the tool touches the object. The force target is the magnitude of the measured force, and the torque target is the magnitude of the torque in the sensor frame, as logged and without a shift to the contact point. Both are restricted to the contact span, from the first to the last contact sample. The peak is the maximum magnitude over the raw contact samples, and the 32-sample profile is the Savitzky–Golay-smoothed magnitude on a uniform grid over the span, rescaled so that its maximum equals the peak.

B.2 Quality filter

The filter keeps the recordings in which the interaction can actually be seen; in the third-person views the operator often walks the gripper rig between the camera and the object, and hands and contacts leave the frame. Every recording is tested at its annotated contact with a *depth-ray gap*, the distance between the three-dimensional contact and the depth image at its projection: up to 0.20 m the contact is visible, up to 0.45 m borderline, beyond that occluded. Recordings occluded by the person, with the contact outside the image, with less than half of the articulation path in view, or with an implausibly long path (corrupted tracking) are rejected; borderline gaps are ambiguous; the rest pass. Human recordings are trimmed to the span in which the object moves, from grip onset to the first confident stop, release, or a per-scene cap; the median trimmed span covers 7 of the 16 waypoint frames, whereas gripper spans are used as delimited by the release. Two thirds of the buildable recordings pass, 6 % are ambiguous and unused, and 28 % are rejected; Fig. 4 shows two recordings per tag.

B.3 Sequences, splits, and pairs

A recording becomes a sequence when it passes the filter, can be calibrated, and shows the contact and the object in its first frame (Table 11); intrinsics come from the gripper recording of the same episode and camera, so human recordings without a buildable gripper counterpart remain unused. The sequences come from 280 episodes in 17 scenes. Splits were fixed per episode before filtering (Table 12): the living room is held out as a whole, and a stratified set of episodes from ten training scenes forms the held-out interactions. Every human recording is paired with the gripper recording of its episode, from the same camera where one exists; the partner's peak and normalized profile serve as its reference targets and describe gripper force rather than the person's effort. 201 of the 236 human training sequences have a partner (169 same camera, 32 other camera), and 40 evaluation human sequences have a measured evaluation partner (Appendix C.9). The release names no objects; for per-object reporting we name each episode by the part its gripper prompt mentions, with a visual check of the few evaluation episodes without a descriptive prompt. Measured peaks differ between objects and between opening and closing (Table 13); the spread within an object and action is what the instance adaptation and the flow have to explain.

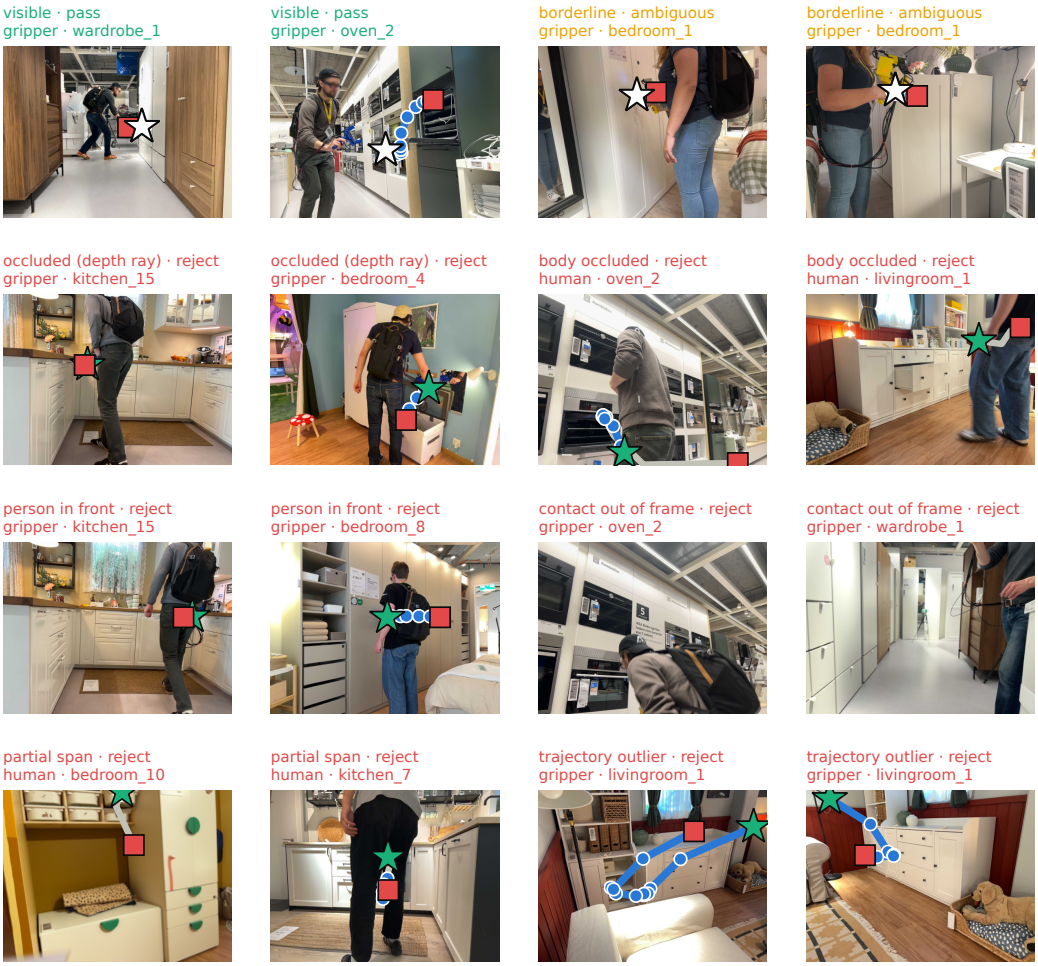


Figure 4: **Filter examples.** Two random recordings per occlusion tag at their first frame, with the annotated path —, its supervised span •••, the first ★ and last ■ annotated point, and the annotated contact ☆.

Table 11: **Data funnel.** Every third-person recording of the release, per embodiment and camera, counted at the first gate it fails: not buildable from the release, rejected or ambiguous in the quality filter, no gripper counterpart, contact or object not visible in the first frame, or corrupted motion capture; the last row gives the sequences that remain.

	Gripper		Human		
	Camera 1	Camera 2	Camera 1	Camera 2	All
Third-person recordings	350	372	362	350	1434
Not buildable from the release	86	86	7	12	191
Rejected by the quality filter	64	107	96	87	354
Ambiguous filter verdict	4	6	28	33	71
No gripper counterpart	0	0	80	46	126
Contact or object not visible in the first frame	9	14	13	17	53
Corrupted motion capture	2	1	3	3	9
Sequences	185	158	135	152	630

Table 12: **Sequences by split and embodiment.** Training, unseen-scene, and held-out-interaction sequences of the gripper and the person. Measured force exists for every gripper sequence and for none of the human sequences, whose force targets are the paired gripper references.

Split	Gripper	Human	Total
Training	295	236	531
Unseen scene	23	24	47
Held-out interactions	25	27	52
All	343	287	630

Table 13: **Measured peak force by object and action.** Count, median, and 10th and 90th percentiles of the measured peak over the 343 gripper sequences, for opening and closing each object; the spread within an object and action is what instance adaptation and the flow have to explain.

Object	Action	n	Median (N)	P10 (N)	P90 (N)
Drawer	open	58	14.6	5.9	26.9
Drawer	close	67	12.5	7.4	18.9
Cupboard door	open	64	13.2	7.5	21.9
Cupboard door	close	69	7.5	4.8	13.8
Oven door	open	33	48.6	31.0	68.8
Oven door	close	27	29.3	18.9	46.9
Refrigerator door	open	8	41.8	39.5	54.5
Refrigerator door	close	7	24.6	14.6	25.0
Other	open	7	11.2	8.3	27.9
Other	close	3	15.3	9.1	58.0

B.4 Training windows and augmentation

Training windows are shifted and rescaled so that the model cannot read the contact off a fixed image position. Each training sequence contributes three windows with side 1.2–$2.2\times$ the canonical window and the contact placed anywhere within the central $70\,\%$ of the window, each with its mirror image; trajectories, force profiles, intrinsics, and prompts are transformed consistently, and the canonical window is sampled with probability 0.3. Further perturbations are one-cell contact jitter, log-depth jitter with scale 0.05, and horizontal flips with probability 0.5; the encoder, prompt, prior, and contact-cue dropout rates are listed with the training schedule in Appendix A.4. The same construction defines the re-framed evaluation set Eval$'$ (Appendix C.1).

C Supplementary Evaluation

C.1 Evaluation protocol

Sets. Eval, the evaluation set of the main text, is the evaluation split of Appendix B.3: the 99 sequences of the unseen living room (47) and of the held-out interactions in ten training scenes (52), 48 of them instrumented. Eval$'$ applies the training-window construction of Appendix B.4 to every Eval sequence: three windows with side 1.2–$2.2\times$ the canonical window that contain the whole articulation with the contact off-center, and their mirror images, six views per sequence (594 views, 288 instrumented). Views of one sequence are correlated observations and count as views, not as additional interactions.

Scoring. Every method receives the evaluation window and the same contact and is scored on the same 16 waypoints over the retained articulation span; on Eval$'$ every window is a row, mirrored windows against the mirrored reference. ATE aligns the starting point by translation only. Force metrics use the 48 measured gripper sequences (288 views on Eval$'$); human references are never scored as measurements. The gripper-view ForceSight rows see the gripper-mounted camera and are scored once per episode (39 episodes of the same interactions). A dash marks a setting that was not evaluated. Nothing was tuned on Eval; the frozen export reproduces the model evaluator's

D4RT-initialization ATE to within 0.01 cm, and the baseline metric code was checked against it. Generated inputs (self-click, self-prompt) follow the protocol of Appendix C.12.

C.2 Baseline implementation

Table 14 summarizes what each baseline does and which of the evaluated outputs it provides. Table 15 lists ATE and ADE of every trajectory baseline and of our model on Eval and Eval′, Fig. 5 shows them on one episode in both embodiments, and Table 16 the GPT-5.5 and ForceSight variants; all are scored as in Appendix C.1.

Table 14: **Baselines at a glance.** For each method: approach, input and camera view (T: third-person, G: gripper-mounted), whether it was trained on Hoi!, and which outputs it provides (trajectory, peak force, force profile). Only VidForce provides all three, and its contact point and prompt are optional.

Method	Approach	Input (view)	Trained	Traj.	Peak	Profile
D4RT initialization	contact-seeded tracks	clip, contact (T)	–	✓	–	–
CoTracker3 + DA3	tracked contact pixel	clip, contact (T)	–	✓	–	–
Articulate-Anything	Gemini-built asset	5 frames, contact (T)	–	✓	–	–
EgoInfinity	hand reconstruction	clip, contact (T, human)	–	✓	–	–
GPT-5.5	prompted peak estimate	8 frames, text (T)	–	–	✓	–
ForceSight, zero-shot	force goal	RGB-D frame, text (G, T)	–	–	✓	–
ForceSight, fine-tuned	goal fit to our force	RGB-D frame, text (G, T)	fine-tuned	–	✓	–
ForceSight, adapted	per-frame force	RGB-D frame, text (G, T)	trained	–	✓	✓
VidForce	prior + joint flow	clip, contact, text (T)	trained	✓	✓	✓

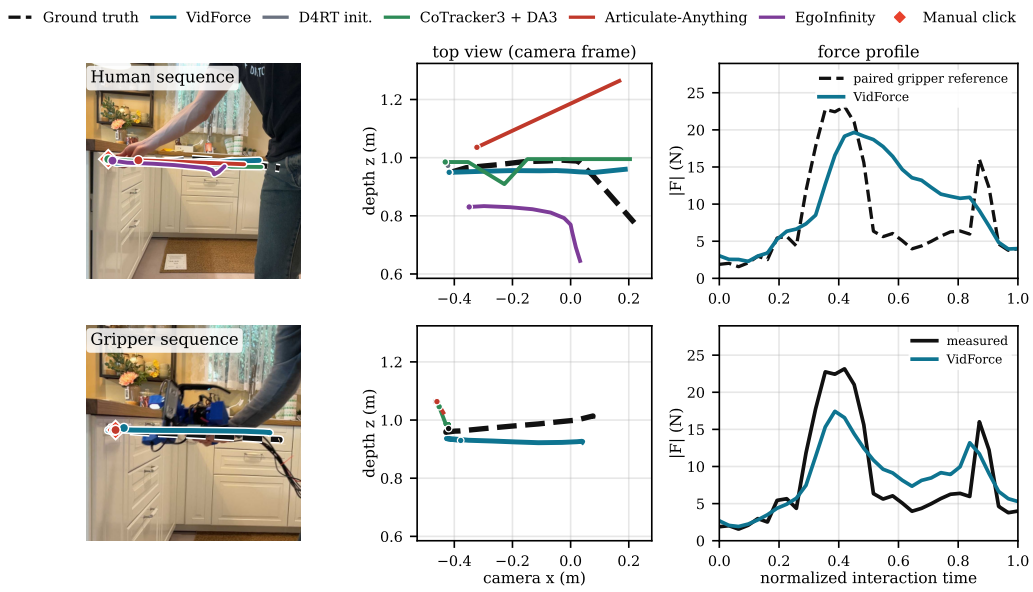
D4RT initialization. The proposal $\mathbf{B}$ of Eq. 7 decoded to metric coordinates, i.e. the input our flow refines: contact-seeded D4RT tracks through the 32 frames, per-waypoint median image displacement and log-depth ratio over the tracks visible at both the contact and the waypoint frame, mapped to camera coordinates with the DA3 contact depth and $\mathbf{K}$. Waypoints without surviving tracks copy the nearest tracked waypoint.

CoTracker3 with monocular depth. The contact pixel is tracked through the 32 frames with the offline CoTracker3 model, backward from the contact frame, inside the evaluation window; each tracked pixel is lifted with the DA3 depth at that pixel and the frame intrinsics, occluded or invalid depth holds the last valid value, and the sequence is linearly resampled to the 16 waypoint times. The main variant smooths the lifted depth with a five-frame temporal median; the per-frame-depth variant is listed as well.

EgoInfinity. EgoInfinity reconstructs MANO hands from a static-camera video and has no contact interface, so it runs on the 51 human sequences only. Each sequence is encoded as a video of its 32 exported frames and processed with MoGe-2 (Wang et al., 2025) depth and focal length, WiLoR (Potamias et al., 2025) hands with MANO (Romero et al., 2017), and the pipeline's motion stabilization; object reconstruction and tracking are disabled. The hand whose 2D joint is nearest the contact at the contact frame is followed and that joint is the contact proxy; sequences without a hand track fall back to the stationary prediction. The main row reads the trajectory in EgoInfinity's own reconstructed camera; a second row re-projects its joints with our camera intrinsics and rescales the trajectory so that its start depth equals the DA3 depth at the contact.

Articulate-Anything. Gemini 2.5 Flash (Gemini Team, Google, 2025) with five frames per call, PartNet-Mobility (Xiang et al., 2020) retrieval, and one actor–critic iteration (the original method used a different vision–language model). The inferred asset is placed at the contact with yaw and scale fitted to the CoTracker3 track, and its trajectory follows the predicted joint range at constant speed. Uncompilable or unrenderable outputs (5 of 99 Eval sequences, 40 of 594 re-framed windows) fall back to the stationary prediction and stay in the denominator, so the comparison includes reconstruction and registration errors.

Global prior with adaptation. This is our own model with the prototypes removed. Instead of K routed prototypes, every sequence starts from one shared prior: the median measured training



ATE (cm)	VidForce	D4RT init.	CoTracker3 + DA3	Articulate-Anything	EgoInfinity
Human	**5.0**	31.6	5.9	14.0	10.3
Gripper	**4.6**	18.9	20.0	18.6	—

Figure 5: **Trajectory baselines on one episode.** A kitchen drawer opened by the person and by the gripper, seen from the same camera; all methods use the manual click ◆, and dots mark each start. VidForce is the seed-0 checkpoint.

Table 15: **Trajectory error of every baseline.** Median over sequences on Eval and Eval′ (cm); ATE aligns the start by translation, ADE averages the raw waypoint error and so includes the contact-depth error. All: 99 sequences (594 views on Eval′); Human: the 51 human sequences (306 views). CoTracker3 is lifted with per-frame DA3 depth, raw or median-smoothed over five frames; EgoInfinity is read in its own camera or re-projected with our intrinsics at the DA3 contact depth, human sequences only; Articulate-Anything is our adaptation with Gemini 2.5 Flash. Baselines are single runs.

	Eval				Eval′			
	ATE (cm) ↓		ADE (cm) ↓		ATE (cm) ↓		ADE (cm) ↓	
Method	All	Human	All	Human	All	Human	All	Human
Tracking								
D4RT initialization	11.8	11.1	23.7	23.0	12.8	13.9	25.4	25.4
CoTracker3 + DA3 depth, per-frame depth	13.7	14.2	20.6	21.9	13.5	14.2	21.0	21.2
CoTracker3 + DA3 depth, depth smoothed over 5 frames	13.3	14.2	20.3	21.1	12.6	13.4	21.2	21.4
Reconstruction								
Articulate-Anything	17.5	17.3	30.4	28.9	18.9	18.9	31.6	35.1
EgoInfinity, its own camera model	–	15.1	–	24.8	–	15.8	–	49.1
EgoInfinity, our camera intrinsics + DA3 depth	–	14.2	–	28.1	–	15.9	–	30.3
VidForce	6.6 ±0.4	7.9 ±0.8	11.4 ±1.1	14.8 ±2.0	7.2 ±0.3	8.2 ±0.5	13.2 ±0.2	16.2 ±1.1

peak and the mean normalized training profile, the same statistics that initialize the prototypes, kept fixed during training. Everything else is unchanged and learned as in the main model: the instance adaptation corrects the shared prior from the video features, and the joint flow refines trajectory and force together with the same conditioner, losses, and schedule. The arm therefore isolates what the prototypes and their routing add over a shared starting point that the model still adapts to each video; read before adaptation, its error is that of predicting the training median for every sequence (Table 20). It is trained for force and for torque like the main model.

GPT-5.5. Chat completions with the model snapshot `gpt-5.5-2026-04-23`, a strict JSON output schema, and low reasoning effort; eight frames evenly spaced over the articulation span, either the full frame or the object-centric window, with or without the recorded prompt and the object and action words, or the words and the room type alone (text only); the main text quotes the full-frame rows, Table 16 all of them. Three calls per input are averaged (sampling is the model's default, which cannot be fixed); identical full-frame inputs share calls across re-framed windows. The torque variant asks for the peak torque magnitude about the sensor frame in N m. These are adaptations for peak estimation, not models trained on the Hoi! targets.

GPT-5.5 questions. The system message and the user templates are used verbatim; $\langle\cdot\rangle$ marks the fields filled per sequence.

> *System.* You are a careful physical-interaction analyst. You estimate the peak contact force (in newtons) needed at the hand or gripper contact point to perform an articulated-object interaction such as opening or closing a cupboard door, a drawer, an oven door or a fridge door. Think about the object type, its size and mass, hinge or slider friction, seals and dampers, and the motion visible in the frames. Answer with JSON only.
>
> *Frames + text.* You see 8 frames, evenly spaced over the interaction, from one fixed third-person camera (full frame). The interaction is: "⟨prompt⟩". The manipulated object is a ⟨object⟩ being ⟨opened/closed⟩. The contact point of the ⟨hand/robot gripper⟩ is marked by nothing; infer it from the frames. Estimate the PEAK force in newtons applied at the contact point over the whole interaction (the maximum of the force magnitude). Typical values for household articulated objects range from a few newtons to a few tens of newtons. Reply with JSON of the form {"object": "...", "interaction": "...", "peak_force_newtons": ⟨number⟩, "confidence": "low|medium|high", "reasoning": "one or two sentences"}.
>
> *Frames only.* You see 8 frames, evenly spaced over one manipulation, from one fixed third-person camera (full frame). First identify what articulated object is manipulated and what the interaction is (open or close, which part). Then estimate the PEAK force in newtons applied at the contact point over the whole interaction (the maximum of the force magnitude). Typical values . . . [as above].
>
> *Text only.* A person or a robot gripper performs this interaction with an articulated household object: "⟨prompt⟩". The object is a ⟨object⟩ being ⟨opened/closed⟩, in a ⟨room type⟩. There is no image. Estimate the typical PEAK force in newtons applied at the contact point over the whole interaction (the maximum of the force magnitude). Reply with JSON . . . [as above].

ForceSight. ForceSight (Collins et al., 2024) predicts, from one RGB-D frame and an prompt, the force goal of a robot's next action; the released model is an RGB-D ViT-L with a frozen T5 text encoder, trained on a robot's wrist camera. Every variant sees frames squashed to 224^2 with DA3 depth: in the third-person view the 32 frames of the object window our model sees, and in the gripper-mounted view, the view of the Hoi! zero-shot protocol (Engelbracht et al., 2026), the whole image at 16 contact-anchored times per episode (39 Eval episodes); the peak is the maximum over frames. *Zero-shot* rows run the released checkpoint with the authors' code and score the magnitude of its applied-force goal $|F_A|$; Fig. 6 draws its fingertip heat map and force goals. *Fine-tuned* rows train all weights of the released checkpoint so that $|F_A|$ matches the measured force of each frame; the model keeps its force-goal output and is compared on the peak. Fine-tuning places no loss on the fingertip head, which then finds a fingertip in a quarter of the frames instead of most, so the heat maps in Fig. 6 are those of the zero-shot model. *Adapted* ForceSight re-implements the architecture with the same ViT-L backbone and frozen T5 text path, replaces the goal outputs, most of which have no labels for human video, with one force output per frame, and trains it from ImageNet-21k initialization; its per-frame predictions form the profile. Fine-tuned and adapted models share data and schedule: in the third-person view, the 496 training sequences with a force target on the object window (295 measured gripper sequences and 201 paired human references at weight 0.5, as for our model); in the gripper-mounted view, the 199 measured training episodes of that view; 8,000

updates, batch 32, L2 loss, AdamW at 5×10^{-5}. In both views, adapted ForceSight matches the fine-tuned release within one seed standard deviation in peak error (Table 16), so the adaptation does not weaken the baseline; trained on the object window with paired references it is the matched baseline of Table 2, and trained on the measured rows alone it performs similarly. None of the variants predicts a trajectory.

Table 16: **Force baselines.** Errors on the instrumented evaluation sequences with VidForce for comparison. Per view, ForceSight is listed zero-shot, fine-tuned (all weights), and adapted; * marks the matched baseline of Table 2, and zero-shot and fine-tuned rows are scored on the peak only. Gripper-mounted rows cover the 39 episodes once each with 16-frame profiles; "paired refs." adds the gripper measurements copied to the paired human sequences.

			Force error (N) ↓		
Model	View	Training data	Peak MAE	Peak RMSE	Profile RMSE
Training median (reference)	no video	measured	8.98	15.88	–
Object–action mean (reference)	no video	measured	6.09	8.57	–
GPT-5.5	8 full frames	none	6.85	11.88	–
GPT-5.5	8 full frames + text	none	6.98	11.43	–
GPT-5.5	8 object-crop frames	none	7.75	12.35	–
GPT-5.5	8 object-crop frames + text	none	7.99	11.98	–
GPT-5.5	text only	none	13.03	15.17	–
ForceSight zero-shot	gripper-mounted	none	12.33	19.53	–
ForceSight fine-tuned	gripper-mounted	measured	7.26 ± 0.49	10.29 ± 0.87	–
ForceSight adapted	gripper-mounted	measured	7.56 ± 0.28	10.84 ± 0.35	4.71 ± 0.16
ForceSight zero-shot	third-person	none	13.75	20.96	–
ForceSight fine-tuned	third-person	measured + paired refs.	10.16 ± 0.26	16.79 ± 0.49	–
ForceSight adapted*	third-person	measured + paired refs.	10.30 ± 0.75	16.60 ± 0.92	7.55 ± 0.27
ForceSight adapted	third-person	measured	10.02 ± 0.27	15.93 ± 0.35	7.39 ± 0.18
VidForce	third-person	measured + paired refs.	4.81 ± 0.38	5.92 ± 0.42	4.95 ± 0.20

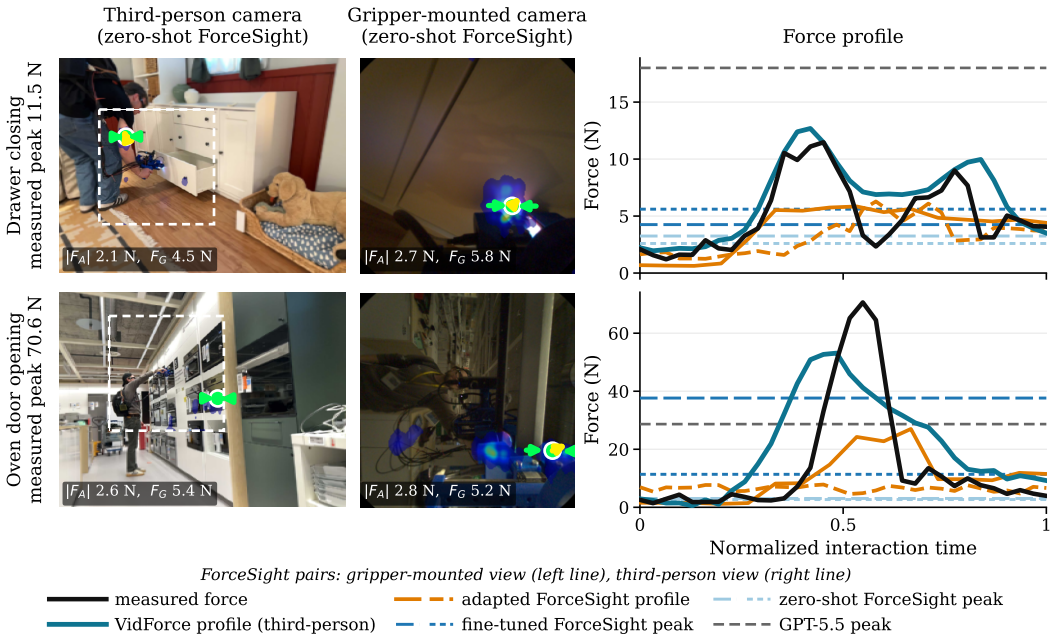


Figure 6: **ForceSight on two episodes.** Zero-shot ForceSight at the frame of the measured peak: fingertip heat map, decoded fingertip ●, applied-force goal ➔ and grip-force goals ➤➤, with their magnitudes $|F_A|$ and F_G; in the third-person view it sees only the object window ⬚.

C.3 DATASET PARTITIONS AND RE-FRAMING

Performance is similar across the two Eval partitions. On the unseen living room and the held-out interactions in seen scenes, VidForce obtains ATEs of 6.7 ± 0.3 and 6.6 ± 0.4 cm and peak MAEs of 4.77 ± 0.54 and 4.84 ± 0.24 N, respectively (Table 18). Table 17 provides the corresponding object-level breakdown.

Table 17: **Breakdown by partition and object.** We report ATE and force errors with openings and closings combined; the category mean predicts the mean training peak of the object–action category. n denotes all sequences and n_F the instrumented subset used for the measured peak and force errors.

		Measured	ATE (cm) ↓		Peak MAE (N) ↓		VidForce (N) ↓	
Subset	n (n_F)	peak (N)	D4RT init.	VidForce	Cat. mean	VidForce	Peak RMSE	Profile RMSE
Partition								
Unseen scene	47 (23)	11.0	11.1	6.7 ± 0.3	4.14	4.77 ± 0.54	5.67 ± 0.62	3.10 ± 0.22
Held-out interactions	52 (25)	22.2	11.9	6.6 ± 0.4	7.88	4.84 ± 0.24	6.13 ± 0.25	6.18 ± 0.24
Object								
Drawer	40 (20)	14.7	7.8	5.3 ± 0.6	3.30	4.54 ± 0.93	5.41 ± 0.85	3.50 ± 0.38
Cupboard door	33 (18)	8.1	13.7	6.8 ± 0.2	4.85	4.26 ± 0.23	5.08 ± 0.34	2.50 ± 0.10
Oven door	9 (3)	69.4	11.8	7.6 ± 1.6	25.28	11.30 ± 2.32	11.84 ± 2.00	14.03 ± 0.42
Refrigerator door	9 (4)	32.6	23.0	9.6 ± 0.6	6.69	6.05 ± 1.83	6.59 ± 1.97	6.61 ± 0.59
Other	8 (3)	10.2	13.9	9.5 ± 0.8	12.07	1.77 ± 0.33	2.09 ± 0.17	4.09 ± 0.13

Table 18: **Results by evaluation partition.** Trajectory and force errors for the unseen scene and held-out interactions in seen scenes, together with the combined Eval result.

Partition	Counts	ATE (cm) ↓	Peak MAE (N) ↓	Peak RMSE (N) ↓
Unseen scene	47 / 23	6.7 ± 0.3	4.77 ± 0.54	5.67 ± 0.62
Held-out interactions, seen scenes	52 / 25	6.6 ± 0.4	4.84 ± 0.24	6.13 ± 0.25
Combined Eval	99 / 48	6.6 ± 0.4	4.81 ± 0.38	5.92 ± 0.42

Eval′ tests sensitivity to framing rather than new interactions: it applies six shifted, rescaled, or mirrored views to each Eval sequence. Under re-framing, VidForce stays close to its canonical accuracy, with 7.2 ± 0.3 against 6.6 ± 0.4 cm ATE and 5.02 ± 0.32 against 4.81 ± 0.38 N peak MAE, and its trajectory advantage remains clear: on Eval′, D4RT and CoTracker3 score 12.8 and 12.6 cm ATE, respectively (Table 15).

Fig. 7 illustrates this behavior across three interactions. Applied to the real-world videos of Sec. 4.3, the same re-framings leave most peak-force estimates close to their canonical values.

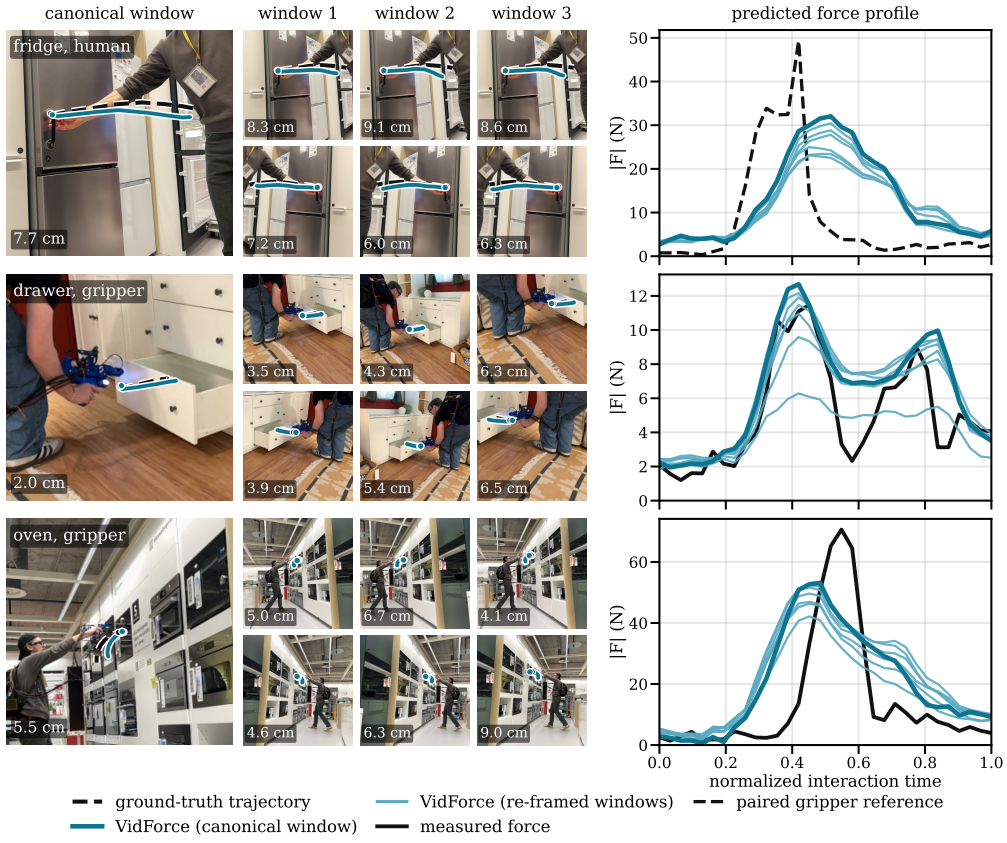


Figure 7: **Predictions under re-framing.** Three interactions in the canonical view (left), three crops and their mirrors (center), and the corresponding force predictions (right). Trajectory overlays report ATE; force references are measured or, for the human sequence, paired.

C.4 CONTRIBUTION OF THE LEARNED PRIOR

Controlled comparison. Table 19 compares the learned prototypes with a parameter-matched MLP prior, a fixed global prior (Appendix C.2), and the same ten prototypes without $\mathcal{L}_{\text{route}}$. The MLP prior replaces the prototype path with one MLP that maps the router's pooled video features and the prompt embedding to a log-peak offset from the global prior, bounded to a factor of eight, and a 32-sample shape residual; it is zero-initialized, so it starts from the global prior, and its hidden width gives it the same number of parameters as the prototype path. All variants were retrained with the same conditioner, instance adaptation, joint flow, data, and budget. Eval uses manual click and manual prompt; force errors cover 48 instrumented sequences. The full model has the lowest mean peak and profile errors, while ATE varies little across priors (Table 20). The peak-MAE ordering also holds on Eval′ (Table 23).

Table 19: **Prior parameterization and routing.** With instance adaptation and joint flow fixed, learned prototypes with routing supervision achieve the lowest force errors.

Prior	Peak MAE (N) ↓	Peak RMSE (N) ↓	Profile RMSE (N) ↓
Direct MLP prior (matched size)	5.79 ± 0.17	8.68 ± 0.03	5.72 ± 0.02
Global prior + adaptation	5.61 ± 0.28	7.64 ± 0.21	5.73 ± 0.05
Prototypes, routing loss off	5.64 ± 0.21	7.26 ± 0.34	5.53 ± 0.15
Prototypes + routing loss (VidForce)	4.81 ± 0.38	5.92 ± 0.42	4.95 ± 0.20

Table 20: **Prior-stage readouts.** Peak MAE read from the same checkpoint before adaptation (prior), after adaptation (adapted), and after the joint flow (final); ATE is the final trajectory error. Final RMSEs are in Table 19.

	ATE (cm) ↓	Peak MAE (N) ↓		
Prior	Final	Prior	Adapted	Final
Direct MLP	6.6 ± 0.4	8.35 ± 0.04	5.78 ± 0.17	5.79 ± 0.17
Global prior	6.7 ± 0.1	8.99 ± 0.00	6.95 ± 0.29	5.61 ± 0.28
Prototypes, no routing loss	6.9 ± 0.4	7.91 ± 0.13	5.68 ± 0.19	5.64 ± 0.21
Prototypes (VidForce)	6.6 ± 0.4	6.88 ± 0.29	4.84 ± 0.31	4.81 ± 0.38

Prior and instance adaptation. The full model's prior has the lowest mean peak MAE at every stage (Table 20). Adaptation accounts for most of its peak-error reduction: peak RMSE falls from 11.28 to 5.94 N, close to the final 5.92 N. The global-prior variant relies more on the flow, which lowers peak MAE from 6.95 to 5.61 N. These are intermediate readouts of trained models, not ablations retrained without adaptation or flow; Table 22 separately compares prediction heads.

Routing and prototype organization. Without the routing loss, only 4.0 ± 1.0 of ten prototypes receive the largest weight for any Eval sequence, compared with all ten in the full model. The mean effective number per sequence, $\exp(-\sum_k \pi_k \log \pi_k)$, falls from 7.0 ± 0.1 to 2.9 ± 0.6, indicating more concentrated assignments. The semantic diagnostic finds stronger alignment with measured-force terciles than with object–action categories on labelled training sequences (adjusted mutual information 0.61 vs. 0.16). On Eval, prototype levels correlate with their members' median measured peaks (Spearman 0.67), with force and category alignment at 0.16 and 0.21. These diagnostics support an effort-based organization, most clearly on the training sequences.

C.5 PRIOR CONTRIBUTION WITH LIMITED FORCE SUPERVISION

Measured subsets. The study reduces force supervision while retaining the video pool of 531 sequences and its motion labels. Nested subsets of 200 measured training episodes are stratified by object and action using one seeded permutation. An episode is retained or withheld as a whole, including both camera views and its paired human references. Withheld measurements enter neither the losses nor the paired targets, prior initialization, or normalization; their videos remain available without force labels. The three seeds measure training variation on these fixed subsets, not variation across independent subset draws.

Training controls. Within each prior family, the architecture, prototype count where applicable, and schedule are fixed across fractions. The global batch size is 128 throughout. At 100 %, Tables 3 and 21 report the main VidForce model and the global prior of Table 4, both trained with all 295 measured sequences.

Table 21: **Limited force supervision, all metrics.** Measured episodes (sequences) per fraction: 12 % = 24 (39), 25 % = 50 (74), 50 % = 100 (142), 100 % = 200 (295).

		Eval, force in N ↓				Eval′, force in N ↓	
Prior	Fraction	ATE (cm)	Peak MAE	Peak RMSE	Profile RMSE	Peak MAE	Peak RMSE
Global	12 %	6.6 ± 0.2	8.48 ± 0.25	14.77 ± 0.17	7.52 ± 0.07	8.40 ± 0.10	14.78 ± 0.01
	25 %	6.6 ± 0.7	7.20 ± 0.75	10.31 ± 0.42	6.53 ± 0.22	7.23 ± 0.53	10.33 ± 0.46
	50 %	6.6 ± 0.2	6.44 ± 0.25	9.32 ± 0.33	5.97 ± 0.07	6.70 ± 0.45	9.77 ± 0.62
	100 %	6.7 ± 0.1	5.61 ± 0.28	7.64 ± 0.21	5.73 ± 0.05	5.73 ± 0.22	8.02 ± 0.17
Prototypes	12 %	6.7 ± 0.4	7.53 ± 0.37	12.99 ± 0.38	7.15 ± 0.05	7.57 ± 0.31	12.96 ± 0.48
	25 %	6.8 ± 0.1	6.27 ± 0.46	8.23 ± 1.37	6.19 ± 0.14	6.28 ± 0.49	8.07 ± 1.34
	50 %	6.6 ± 0.6	5.23 ± 0.28	6.89 ± 0.70	5.80 ± 0.23	5.25 ± 0.48	6.83 ± 1.01
	100 %	6.6 ± 0.4	4.81 ± 0.38	5.92 ± 0.42	4.95 ± 0.20	5.02 ± 0.32	6.27 ± 0.49

The learned prototypes have lower mean peak MAE and profile RMSE than the global prior at every measured fraction (Table 21). With half the measured episodes, their peak MAE is below that of the full-data global prior (5.23 vs. 5.61 N), and their profile RMSE stays within 0.1 N of it (5.80 vs. 5.73 N), so half the measurements suffice for peak prediction.

C.6 Joint prediction and motion cues

Prediction heads. The study retains the learned prototype prior and instance adaptation and changes the prediction head or its motion cues (Table 22). The one-shot head directly regresses the standardized residual target with the same network architecture. The split-flow variant processes trajectory and force tokens in separate streams, retaining the shared conditioner but removing direct exchange between the two outputs; a block-diagonal self-attention mask restricts waypoint and force tokens to their own stream without changing the parameters. All learned variants use the same data and training budget and are evaluated with manual click and manual prompt over three seeds.

Table 22: **Prediction heads and motion cues, full comparison.** The tracked initialization is decoded without refinement; the middle block replaces the joint flow with other prediction heads, and the lower block disables click-free training throughout.

		Force error (N) ↓		
Configuration	ATE (cm) ↓	Peak MAE	Peak RMSE	Profile RMSE
Tracked initialization	11.8	–	–	–
Joint flow (VidForce)	6.6 ± 0.4	4.81 ± 0.38	5.92 ± 0.42	4.95 ± 0.20
One-shot regression	10.3 ± 1.3	6.73 ± 0.38	10.20 ± 0.55	6.47 ± 0.19
Split trajectory / force flows	6.9 ± 0.5	5.33 ± 0.46	6.57 ± 0.70	5.23 ± 0.13
All motion cues (control)	7.6 ± 0.5	5.38 ± 0.23	6.77 ± 0.42	5.38 ± 0.20
No lattice field; tracker tokens kept	7.8 ± 0.8	5.00 ± 0.91	6.61 ± 1.33	5.57 ± 0.09
No dense motion field	10.1 ± 0.6	6.98 ± 0.34	9.00 ± 1.13	6.37 ± 0.14

Prediction-head comparison. The joint flow of VidForce reaches 6.6 ± 0.4 cm ATE and 4.81 ± 0.38 N peak MAE, improving on one-shot regression (10.3 cm, 6.73 N) and on split trajectory and force flows (6.9 cm, 5.33 N), with lower profile RMSE than both (4.95 against 6.47 and 5.23 N). These results support flow-based refinement that predicts both outputs together.

Motion cues. The dense-field ablation removes both lattice-field and tracker tokens. Because it cannot train the click-free path, its control also disables that path while retaining all motion cues. Removing the field increases both trajectory and force errors relative to this control. Retaining tracker tokens while removing only the lattice largely recovers trajectory accuracy (7.8 vs. 7.6 cm),

so the tracker tokens and the lattice carry overlapping motion cues. The study therefore supports the motion cues as a whole.

Re-framed evaluation. Table 23 reports the same variants on Eval′. Removing the dense motion field increases ATE from 8.4 to 12.7 cm relative to its control, consistent with the Eval result.

C.7 Additional ablations and prototype count

Table 23 extends the prior and flow comparisons of Tables 19 and 22 to Eval′. Table 24 varies the number of prototypes at fixed recipe. Across $K \in \{5, 10, 15, 20, 30\}$, mean peak MAE stays between 4.81 and 5.32 N, so accuracy is stable across prototype counts. Every prototype is used at K=5 and K=10, while larger settings leave some unused. We retain K=10, the largest tested setting with full usage.

Table 23: **Mechanism and flow arms on Eval′.** 594 views, 288 of them instrumented.

		Force error (N) ↓		
Configuration	ATE (cm) ↓	Peak MAE	Peak RMSE	Profile RMSE
Direct MLP prior	7.4 ± 0.1	6.17 ± 0.30	9.18 ± 0.27	5.84 ± 0.11
Global prior + adaptation	7.7 ± 0.5	5.73 ± 0.22	8.02 ± 0.17	5.85 ± 0.15
Prototypes, routing loss off	7.4 ± 0.3	5.86 ± 0.40	7.68 ± 0.60	5.69 ± 0.09
One-shot regression head	11.9 ± 0.8	6.77 ± 0.32	10.44 ± 0.41	6.51 ± 0.18
Split trajectory / force streams	7.5 ± 0.1	5.60 ± 0.41	7.00 ± 0.72	5.39 ± 0.13
Control: no click-free path	8.4 ± 0.1	5.61 ± 0.10	7.26 ± 0.41	5.72 ± 0.13
No lattice field, tracker tokens kept	8.7 ± 0.5	5.28 ± 0.59	7.53 ± 1.20	5.85 ± 0.25
No dense motion field	12.7 ± 0.6	6.97 ± 0.31	9.21 ± 1.08	6.34 ± 0.31
VidForce	7.2 ± 0.3	5.02 ± 0.32	6.27 ± 0.49	5.11 ± 0.12

Table 24: **Number of prototypes.** Fixed recipe on the evaluation set. *Used*: prototypes that carry the largest weight for at least one sequence; *effective*: exponential of the entropy of that usage over the 48 instrumented sequences.

				Force error (N) ↓	
K	Used (of K)	Effective	ATE (cm) ↓	Peak MAE	Peak RMSE
5	5.0 ± 0.0	4.3 ± 0.3	6.7 ± 0.7	4.83 ± 0.47	6.12 ± 0.72
10 (ref.)	10.0 ± 0.0	6.5 ± 0.4	6.6 ± 0.4	4.81 ± 0.38	5.92 ± 0.42
15	14.0 ± 0.0	7.7 ± 0.2	6.7 ± 0.9	5.32 ± 0.46	6.66 ± 0.41
20	16.3 ± 1.5	9.1 ± 0.9	6.5 ± 0.4	4.90 ± 0.09	6.13 ± 0.26
30	25.0 ± 2.0	12.3 ± 0.2	7.0 ± 0.3	4.87 ± 0.57	6.19 ± 0.93

C.8 Inference start point

The flow is integrated from $\mathbf{y}_0 = \mathbf{0}$ (Sec. 3.3), whereas training draws $\mathbf{y}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Table 25 repeats the Eval protocol with the same checkpoints and 20 Euler steps but starts from sampled noise in the standardized state space, with five fixed draws per sequence: *single draw* reports each seed's mean over the five draws, and *five-draw average* decodes the mean of the five residual states (clipping each draw before averaging changes no number by more than 0.002). Force errors are equal within noise: the differences are below the draw-to-draw SD of 0.07 N and the seed SD of 0.38 N. A single draw is 0.16 cm worse in ATE on every seed (draw-to-draw SD 0.18 cm), and the five-draw average matches the zero start, so the origin acts as the ensemble mean at one fifth of the cost. The sampled starts have norm 8.1 ± 1.1 over the dimensions the network reads, against 0 for the zero start.

C.9 Transfer to human videos

Reference agreement. The 40 matched Eval human recordings each have a same-episode gripper partner with measured force in Eval. Table 26 compares the human-video prediction with that

Table 25: **Start point of the flow at inference.** Zero is the paper's setting; the noise rows start from $\mathbf{y}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with five fixed draws per sequence.

		Force error (N) ↓		
Start	ATE (cm) ↓	Peak MAE	Peak RMSE	Profile RMSE
Zero (ref.)	6.6 ± 0.4	4.81 ± 0.38	5.92 ± 0.42	4.95 ± 0.20
Noise, single draw	6.8 ± 0.5	4.78 ± 0.36	5.90 ± 0.41	4.94 ± 0.20
Noise, five-draw average	6.6 ± 0.5	4.77 ± 0.36	5.89 ± 0.41	4.92 ± 0.20

partner's peak and 32-sample profile, aligned by normalized interaction time. The comparison thus measures how well the measured gripper force transfers to the human video of the same interaction. GPT-5.5 uses the questions of Appendix C.2; the frames-and-text variant names the hand as the contact.

Table 26: **Human video against the paired gripper reference.** 40 matched evaluation recordings; peak errors and bias (mean signed difference) compare the predicted peak with the partner's measured peak, profile RMSE compares the 32-sample profiles by normalized time, and GPT-5.5 estimates the peak only. Gripper MAE is each model's peak MAE on the 48 measured gripper sequences.

		Human video vs. paired reference (N) ↓			
Model	Gripper MAE (N) ↓	Peak MAE	Peak RMSE	Profile RMSE	Bias
VidForce (paired references)	4.81 ± 0.38	10.64 ± 0.50	14.09 ± 0.72	7.88 ± 0.17	2.1 ± 0.2
Same recipe, no paired references	4.89 ± 0.37	18.79 ± 4.14	21.21 ± 3.63	9.56 ± 1.45	12.9 ± 6.2
ForceSight adapted (third-person), paired references	10.30 ± 0.75	11.00 ± 0.46	17.64 ± 0.83	8.29 ± 0.05	-10.4 ± 0.7
ForceSight adapted (third-person), measured rows only	10.02 ± 0.27	12.08 ± 0.29	19.18 ± 0.46	8.61 ± 0.03	-11.6 ± 0.4
ForceSight released checkpoint (one run)	13.75	16.11	23.61	9.33	-16.1
GPT-5.5 (8 frames + text)	6.98	8.32	13.09	–	-0.7
GPT-5.5 (8 frames)	6.85	7.95	13.72	–	-1.8

Ablation setup. The ablated model is trained on the same 531 sequences as VidForce, with the same videos, trajectory labels, schedule, and settings; only the paired force targets of the 201 human training sequences are removed. Without them, these sequences still supervise the trajectory and motion but receive no measured force. The self-distillation teacher of the training schedule (Appendix A.4) is kept: from update 2,000 its predictions become the force target of every training sequence, the human ones included, but in this run the teacher is the ablated model's own average and has never seen a paired reference. The comparison therefore isolates what the paired gripper force adds rather than the effect of less training data.

Effect of paired supervision. Removing paired targets changes measured-gripper peak MAE little (4.89 vs. 4.81 N), but raises human-video peak disagreement from 10.64 to 18.79 N and mean signed bias from 2.1 to 12.9 N. Profile disagreement also increases (7.88 to 9.56 N). On measured gripper recordings, profile RMSE rises from 4.95 ± 0.20 to 5.32 ± 0.06 N; on Eval′, peak MAE remains similar (4.96 ± 0.53 N without pairing, 5.02 ± 0.32 N with it).

Comparison with other predictors. VidForce improves on adapted ForceSight with paired supervision in every reference metric, with a peak RMSE of 14.09 against 17.64 N and a mean bias of 2.1 against -10.4 N. GPT-5.5 estimates the peak only: it reaches 8.32 N with text and 7.95 N from frames alone on the reference peaks, with 6.98 and 6.85 N on measured gripper recordings against 4.81 N for VidForce. Among profile predictors, VidForce has the lowest mean reference RMSE (7.88 ± 0.17 N).

C.10 Prior types

Four prior families. We compare a *global prior*, which supplies one shared peak and normalized profile; a *word list* with a fixed vocabulary, which groups measurements by four object names and two actions read from the prompt by a language model; *flat clusters*, which assign each prompt to its nearest k-means cluster (K=10); and *learned prototypes*, whose force levels, profiles, and semantic

keys are optimized with soft routing from video and text (K=10; Sec. 3.2). The reference designs share the motion representation, instance adaptation, and joint flow; Table 27 compares them.

Construction and controls. Force statistics use training sequences only. The clustering reference uses all prompts, including those of evaluation sequences, to construct its groups. Category and cluster tokens are dropped for part of training so that these references also run from the prompt alone.

Table 27: **Prior types.** Manual click and manual prompt; peak MAE and profile RMSE of these rows also appear in Table 4.

		Force error (N) ↓		
Prior type	ATE (cm) ↓	Peak MAE	Peak RMSE	Profile RMSE
Global prior + adaptation	6.7 ± 0.1	5.61 ± 0.28	7.64 ± 0.21	5.73 ± 0.05
Word list (categorical), LLM-assigned labels	7.1 ± 0.5	4.86 ± 0.25	5.85 ± 0.53	5.00 ± 0.04
Word list (categorical), annotated labels	7.1 ± 0.4	4.94 ± 0.24	5.91 ± 0.27	5.15 ± 0.07
Flat clusters (k-means, K=10)	6.9 ± 0.3	5.32 ± 0.34	6.56 ± 0.21	5.25 ± 0.05
Learned prototypes (VidForce)	6.6 ± 0.4	4.81 ± 0.38	5.92 ± 0.42	4.95 ± 0.20

C.11 Additional torque results

Torque models differ from their force counterparts only in the supervised channel: sensor-frame torque magnitude from the same wrench sensor. They use the same sequences and training recipe, with prototypes learned separately for torque. Targets are scaled so that the Huber threshold has the same relative position as for force and are reported in N m. Table 28 compares the retained prior families.

Table 28: **Sensor-frame torque.** Comparison of GPT-5.5 with models trained using alternative prior types.

	Torque error (N m) ↓		
Model	Peak MAE	Peak RMSE	Profile RMSE
GPT-5.5 (8 frames)	0.693	0.842	–
GPT-5.5 (frames + text)	0.550	0.725	–
Global prior	0.283 ± 0.008	0.353 ± 0.011	0.332 ± 0.007
Word list, annotated labels	0.256 ± 0.010	0.331 ± 0.013	0.305 ± 0.006
Word list, LLM-assigned labels	0.281 ± 0.016	0.375 ± 0.015	0.327 ± 0.005
Flat clusters (k-means)	0.308 ± 0.012	0.398 ± 0.030	0.334 ± 0.008
VidForce, learned prototypes			
K=10	0.301 ± 0.021	0.384 ± 0.023	0.315 ± 0.003
K=20	0.285 ± 0.026	0.376 ± 0.041	0.314 ± 0.006

The learned formulation transfers directly from force to torque: both prototype configurations improve peak estimation over GPT-5.5 and achieve the lowest profile errors among priors that do not use annotated labels. Increasing the number of prototypes further reduces peak error, suggesting that torque benefits from a larger prototype set. Under re-framing, the prototype model stays close to its canonical result (0.298 ± 0.021 against 0.301 ± 0.021 N m). Text improves the GPT-5.5 estimate but does not close the gap to the trained models.

C.12 Supplied and generated guidance

Evaluation protocol. Table 29 compares, for the same checkpoints, the annotated contact with the vision–language model's proposal used directly (crop re-centered; the self-click rows of the main text) and with the proposal refined by the localization head, and the recorded prompt with the self-written one; the tracker baselines receive the same proposal in Table 1. The generation of the

proposal and of the self-written prompt is described in Appendix A.5, the cached-depth convention in Sec. 4.1: all rows keep the DA3 depth at the annotated contact, so the comparison tests the image contact and the textual guidance while holding the depth fixed. Table 29 adds peak RMSE and the localization-head refinement to the four combinations of Table 4 (*Guidance*).

Table 29: **Guidance variants.** Self-click uses Qwen3-VL's contact proposal directly (crop re-centred) and self-prompt its self-written prompt; contact error is the median pixel distance from the annotated contact, omitted when the contact is supplied; refinement applies the localization head to the proposal. All rows keep the annotated contact depth.

				Force error (N) ↓	
Contact point	Prompt	Contact error (px) ↓	ATE (cm) ↓	Peak MAE	Peak RMSE
manual click	manual prompt	–	6.6 ± 0.4	4.81 ± 0.38	5.92 ± 0.42
self-click	manual prompt	25	7.0 ± 0.3	4.53 ± 0.42	5.63 ± 0.58
manual click	self-prompt	–	6.8 ± 0.5	5.85 ± 0.10	7.58 ± 0.26
self-click	self-prompt	27	7.7 ± 0.7	5.69 ± 0.39	7.85 ± 0.97
self-click, refined	manual prompt	18	7.0 ± 0.6	4.54 ± 0.49	5.57 ± 0.64

Contact proposals. With the recorded prompt, the self-click proposal lies 25 px from the annotation on median. Localization-head refinement reduces this distance to 18 px while ATE stays at 7.0 cm and peak MAE at 4.54 N (4.53 N without refinement), so the predictions are insensitive to this localization difference. Table 1 compares trajectory baselines given the same proposed contact.

Prompt sensitivity. With self-prompt, peak MAE stays within 1.04 N (manual click) and 1.16 N (self-click) of the manual prompt, and peak RMSE within 1.66 and 2.22 N. The generated prompt identifies opening versus closing correctly in 92 of the 99 sequences, including 46 of the 48 instrumented ones. With manual click, substituting predictions obtained with the recorded prompt for the two misread instrumented sequences changes peak MAE and RMSE by at most 0.12 N across seeds. This diagnostic points to the prompt's wording beyond its action label. Appendix D.2 separately examines prompt changes and misread actions on real-world videos, where force is unmeasured.

C.13 Additional qualitative results

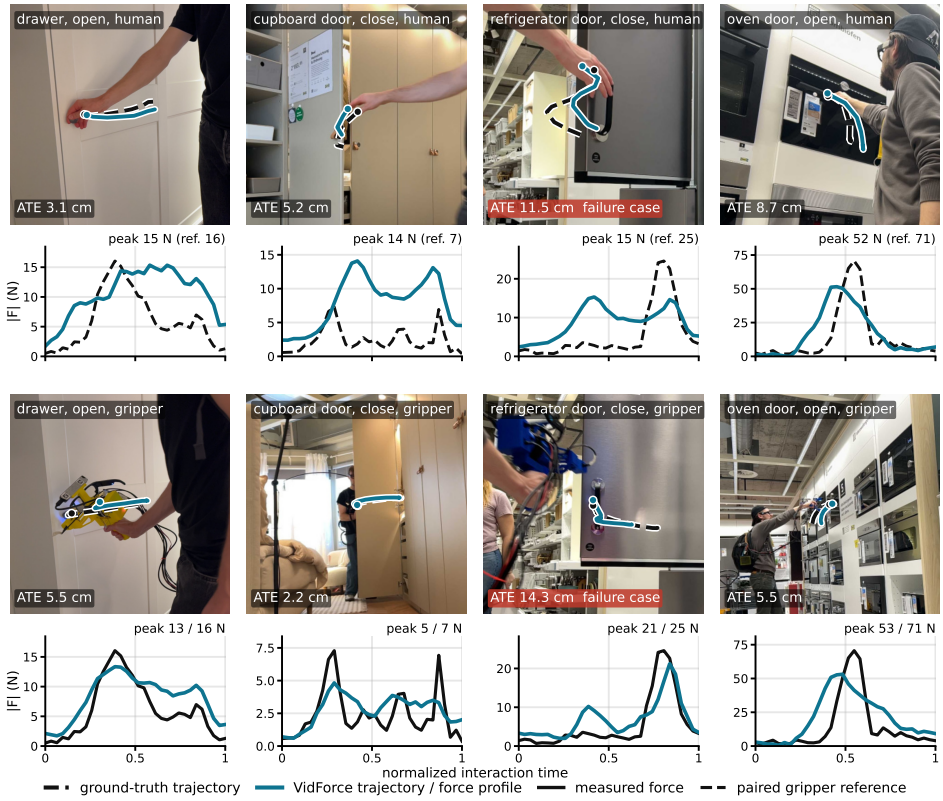


Figure 8: **Qualitative evaluation examples.** Paired human (top) and gripper (bottom) episodes for four interactions. Trajectory overlays report ATE; force plots compare predictions with measured force or paired gripper references. Peak labels show predicted/reference values.

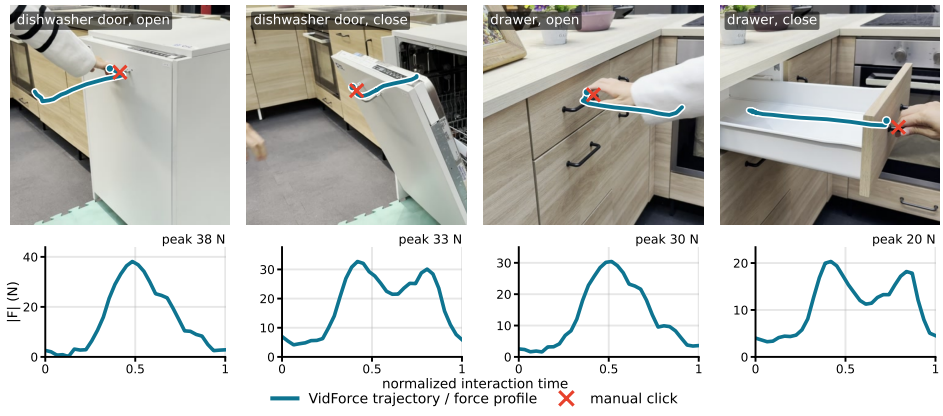


Figure 9: **Real-world videos.** Predicted trajectories and force profiles.

D Real-World Interactions and Robot Deployment

This appendix details the real-world video protocol, supplementary analyses, and robot deployment of Sec. 4.3.

D.1 Real-world videos

Seven tripod-mounted phone videos show one person opening and closing six kitchen objects: a refrigerator door, freezer, cabinet door, top drawer, oven door, and dishwasher door. We split them into 46 episodes, from the hand reaching the handle until motion stops, following the motion-span criterion of Appendix B, and sample 32 frames per episode. The manual inputs are the operator's prompt (e.g., "open the oven door") and a manually annotated contact pixel on the 640-pixel frame. Self-click uses the vision–language model's contact proposal; self-prompt asks the model over eight frames what the person does and to which part of which object, without naming the action or appliance. We evaluate all four input combinations on the same episodes, assess repeatability with the manual inputs, and identify action misreads by comparing generated and operator-provided actions.

D.2 Manual and generated inputs

Fig. 10 shows that predicted peaks vary with both the object and the action, with opening generally requiring more force than closing. Although prototype routing often selects a common force level, 31 N for 20 of the 21 correctly read openings with generated inputs, instance adaptation and the joint flow separate the interactions using the observed video.

Self-click produces predictions close to those obtained with the manual contact (median peak change 0.3 N, at most 2.0 N), while the generated prompt steers the estimated effort. It reads the action correctly in 36 of the 46 episodes, 14 with the operator's wording and 22 reworded; seven of the ten misreads are refrigerator closings read as openings, for which the routed prior rises from 17–18.5 to 31 N. In the oven-closing example of Fig. 3, the reworded prompt adds a purpose clause and the predicted peak moves from 37 to 41 N. Trajectory direction matches the demonstrated action in all 46 episodes under every input combination, even when the prompt is wrong; the tracked initialization alone moves less than 1 cm or the wrong way in 13 to 16 of them. Repeated recordings also yield similar peak estimates, with oven opening showing the greatest variation.

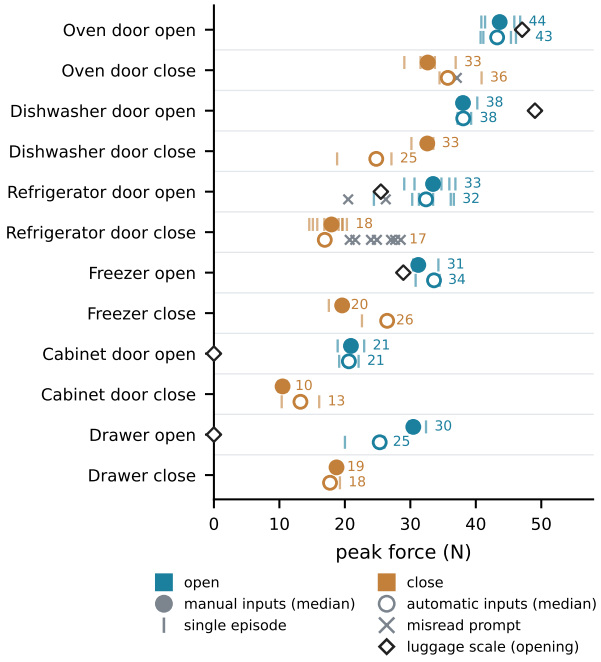


Figure 10: **Real-world peaks by input source.** Filled and open circles show median predictions with manual inputs (operator click and prompt) and automatic inputs (self-click and self-prompt), respectively. Ticks mark individual episodes and crosses misread prompts, which are excluded from the medians; blue denotes opening and orange closing. Diamonds mark the opening force measured with a luggage scale, as in Fig. 3b.

D.3 ROBOT DEPLOYMENT

Spot records a person operating a kitchen unit with its arm RGB-D camera from two viewpoints (Fig. 11). The model proposes the contact and generates the prompt, so all robot trials use self-click and self-prompt. VidForce predicts a contact trajectory and force profile from the demonstration, which are exported as a manipulation plan: the grasp, the trajectory, and the predicted force profile as an additional controller signal. Execution trials cover a prismatic and a revolute joint, the drawer and the oven door. From each viewpoint, Spot performs five openings of each object with and without the predicted force profile, 40 openings in total. The drawer is empty for the first viewpoint and holds an object of about 5 kg for the second. A trial succeeds when the robot opens the panel to the demonstrated state. Failures fall into force errors, where the arm stalled or reached its force limit; trajectory errors, where the gripper left the demonstrated path; and partial openings, where the panel opened partway before the gripper slipped off, the arm stowed, or it could not follow the rest of the trajectory. A partial opening counts as a failure (Table 30).



Figure 11: **Spot demonstrations.** A person opening the drawer (left) and the oven door (right), recorded with Spot's arm camera from the two viewpoints per object used in the robot trials.

On the real robot, both conditions run the same Cartesian impedance controller on Spot's arm, streamed through the SDK's arm impedance command in a gravity-aligned world frame: the commanded wrench at the tool is $\mathbf{F} = \mathbf{K}_{\mathrm{p}}(\mathbf{x}_{\mathrm{ref}} - \mathbf{x}) + \mathbf{K}_{\mathrm{d}}(\dot{\mathbf{x}}_{\mathrm{ref}} - \dot{\mathbf{x}}) + \mathbf{F}_{\mathrm{ff}}$, with fixed diagonal stiffness and damping and a cap on the total commanded force. The reference pose $\mathbf{x}_{\mathrm{ref}}(t)$ is the predicted contact-point trajectory placed in the robot frame with the recording's camera pose, time-scaled for the robot, with the gripper orientation set at the grasp and either held fixed or turned with the panel along the path. The only difference between the two conditions is the **feed-forward force term**. Without force, $\mathbf{F}_{\mathrm{ff}} = \mathbf{0}$, so any interaction force arises only from the hand lagging behind the reference, i.e., from position tracking error. With force, $\mathbf{F}_{\mathrm{ff}}(t)$ is the predicted force profile, scaled to the predicted peak and applied along the local path tangent, while the stiffness along the path is reduced so that the predicted force, rather than the tracking error, drives the interaction; lateral and vertical stiffness stay high to keep the hand on the path. Both conditions share identical gains, gripper handling and stop rules, and the stop rules act only on the measured end-effector force and on progress, never on the command, so the comparison isolates the contribution of the predicted force profile.

Table 30: **Spot trials in full.** Outcomes of the five openings per object, viewpoint, and force condition; a partial opening counts as a failure. The successes are those of Table 5.

Object (joint)	Viewpoint	Force profile	Success	Force error	Trajectory error	Partial opening
Drawer (prismatic)	1 (empty)	Without	5	0	0	0
		With	5	0	0	0
	2 (∼5 kg)	Without	4	0	1	0
		With	4	0	1	0
Oven door (revolute)	1	Without	0	4	1	0
		With	3	0	1	1
	2	Without	0	4	1	0
		With	4	0	1	0