Appendix

This appendix defines the evaluation subgoals from Sec. V and provides additional implementation details for Sec. IV. Appendix A documents the trajectory latent space of Sec. IV-C: the complete 113-dimensional style fingerprint (Appendix A.2), the encoder architecture (Appendix A.3), and training details (Appendix A.4). Appendix C documents the style alignment objective of Sec. IV-D, including the full gradient chain rule before the collapse to Eq. (16). Appendix D documents the timing alignment objective from the same section, and Appendix B the joint-space alignment objective. Appendix E documents the hyperparameters used during planning and VLA fine-tuning. Appendix F gives the task-specific scoring criteria.

A. Trajectory Latent Space

1) Trajectory inputs: A timed segment $\xi = ((q_i, t_i))_{i=1}^{n}$, with $t_1 = 0$, is resampled at $F = 15\,\text{Hz}$ to configurations $\tilde{q} \in \mathbb{R}^{\tilde{n}\times 7}$. With D the central-difference operator of step $1/F$, the encoder input stacks positions, velocities, accelerations, and jerks:

$$Y = [\, Y_q \mid Y_v \mid Y_a \mid Y_j \,] = [\, \tilde{q} \mid D\tilde{q} \mid D^2\tilde{q} \mid D^3\tilde{q} \,] \in \mathbb{R}^{\tilde{n}\times 28}. \tag{9}$$

Each channel is then standardized,

$$\bar{Y} = (Y - \mu_c) \oslash \sigma_c, \tag{10}$$

where $\mu_c, \sigma_c \in \mathbb{R}^{28}$ are fixed per-channel means and standard deviations over the encoder's training dataset.

2) Motion descriptors: The auxiliary head g_ψ predicts the 113-dimensional descriptor vector $f(\xi)$ from a sampled latent vector. Table III lists the 14 descriptors computed independently for each of the 7 joints (98 entries). Table IV lists the 15 whole-trajectory descriptors. We denote the speed as:

$$v(t) = \|\dot{q}(t)\|_2. \tag{11}$$

We write $\mathcal{F}\{\cdot\}$ for the discrete Fourier transform, $S(\nu)$ for the normalized power spectrum of $v(t)$ at frequency ν, and N_{freq} for the number of frequency bins used in the spectral sums. In the smoothness descriptor, $v_{\max}$ is the maximum of $v(t)$. Spectral bands are low $[0, 0.5)$ Hz, mid $[0.5, 2)$ Hz, and high $[2, 7.5)$ Hz, with the last bounded by the Nyquist frequency at $F = 15\,\text{Hz}$.

Descriptor	Description	Computation
Velocity		
`vel_mean`	mean joint speed	$\text{mean}_t\lvert\dot{q}^{(j)}\rvert$
`vel_std`	variability of joint speed	$\text{std}_t\lvert\dot{q}^{(j)}\rvert$
`vel_max`	peak joint speed	$\max_t \lvert\dot{q}^{(j)}\rvert$
`vel_p95`	near-peak speed, outlier-robust	95th pct. of $\lvert\dot{q}^{(j)}\rvert$
Acceleration and jerk		
`acc_mean`	mean acceleration magnitude	$\text{mean}_t\lvert\ddot{q}^{(j)}\rvert$
`acc_std`	variability of acceleration	$\text{std}_t\lvert\ddot{q}^{(j)}\rvert$
`acc_max`	peak acceleration	$\max_t \lvert\ddot{q}^{(j)}\rvert$
`jerk_mean`	mean jerk magnitude	$\text{mean}_t\lvert\dddot{q}^{(j)}\rvert$
`jerk_max`	peak jerk	$\max_t \lvert\dddot{q}^{(j)}\rvert$
Extent and reversals		
`rom`	range of motion	$\max_t q^{(j)} - \min_t q^{(j)}$
`vsign`	direction-reversal rate	sign changes of $\dot{q}^{(j)}$, per t_n
Spectral (share of velocity power per band)		
`bandfrac_low`	share that is slow, gross motion	frac. of $\lvert\mathcal{F}\{\dot{q}^{(j)}\}\rvert^2$ in $[0, 0.5)$ Hz
`bandfrac_mid`	share from corrections, wobble	frac. of $\lvert\mathcal{F}\{\dot{q}^{(j)}\}\rvert^2$ in $[0.5, 2)$ Hz
`bandfrac_high`	share from fast micro-motion	frac. of $\lvert\mathcal{F}\{\dot{q}^{(j)}\}\rvert^2$ in $[2, 7.5)$ Hz

TABLE III: Per-joint motion descriptors. Fourteen descriptors for each of seven joints yield 98 entries.

3) Encoder architecture: Figure 5 shows the full trajectory encoder architecture. It consists of two branches:

a) Spectral branch: This branch uses a variety of kernel-dilation pairs to capture motion behaviors at different temporal scales. Their outputs are masked-pooled over time by the mean and standard deviation of the absolute activations, giving a 384-dimensional summary.

b) Temporal branch: This branch captures the ordering of motion, such as acceleration during free-space transit followed by deceleration on approach to an object.

4) Training details: The trajectory encoder is trained in a self-supervised manner. For each trajectory, the objective combines descriptor prediction with Gaussian KL regularization:

$$\begin{aligned}\mathcal{L}(\phi, \psi; \xi) = \mathbb{E}_\epsilon\Big[\big\|w \odot \big(g_\psi(z) - f(\xi)\big)\big\|_2^2\Big] \\ + \beta\, D_{\text{KL}}\big(\mathcal{N}\big(E_\phi(\xi), \text{diag}(\sigma_\phi^2(\xi))\big) \,\big\|\, \mathcal{N}(0, I)\big)\,,\end{aligned} \tag{12}$$

Descriptor	Description	Computation
Path geometry		
`path_len_per_s`	joint-space distance per second	$\big(\sum_{i=1}^{\tilde{n}-1} \|\tilde{q}_{i+1} - \tilde{q}_i\|_2\big)/t_n$
`straightness`	how direct the path is	$\|\tilde{q}_{\tilde{n}} - \tilde{q}_1\|_2$ / path length
Speed		
`speed_mean`	mean overall speed	$\text{mean}_t\, v(t)$
`speed_std`	speed variability	$\text{std}_t\, v(t)$
`speed_max`	peak overall speed	$\max_t v(t)$
`frac_moving`	fraction of time in motion	frac. of frames with $v(t) > 0.02\,\text{rad/s}$
Smoothness		
`sparc`	spectral arc length $\to 0$ is smoother [28]	arc length of the normalized speed spectrum up to the 10 Hz cutoff (limited by $F/2$), truncated below amplitude 0.05
`ldlj`	log dimensionless jerk; higher is smoother [29]	$-\log\big((t_n^3/v_{\max}^2)\int \ddot{v}^2\, dt\big)$
`submov_per_sec`	submovement rate	peaks in $v(t)$ with prominence $\geq 0.05 \max_t v(t)$, per second
Spectral (shape of the speed spectrum)		
`spec_centroid`	where the spectrum is centered	$\sum_\nu \nu S(\nu)$
`spec_bandwidth`	how spread out the spectrum is	$\big(\sum_\nu (\nu - \text{centroid})^2 S(\nu)\big)^{1/2}$
`spec_entropy`	how broadband the motion is	$-\sum_\nu S(\nu)\log S(\nu)$ / $\log N_{\text{freq}}$
`bandfrac_low_speed`	slow share of the speed profile	frac. of $\|\mathcal{F}\{v\}\|^2$ in $[0, 0.5)$ Hz
`bandfrac_mid_speed`	mid share of the speed profile	frac. of $\|\mathcal{F}\{v\}\|^2$ in $[0.5, 2)$ Hz
`bandfrac_high_speed`	fast share of the speed profile	frac. of $\|\mathcal{F}\{v\}\|^2$ in $[2, 7.5)$ Hz

TABLE IV: Whole-trajectory motion descriptors (15 entries). $S(\nu)$ is the normalized power spectrum of the speed signal $v(t)$ in Eq. (11).

where the encoder outputs a mean $E_\phi(\xi) \in \mathbb{R}^d$, with $d = 16$, and standard deviations $\sigma_\phi(\xi)$, and the latent is sampled as

$$z = E_\phi(\xi) + \sigma_\phi(\xi) \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{13}$$

Planning uses only the mean $E_\phi(\xi)$. The weights w balance the different natural scales of the 113 descriptors. We use $\beta = 0.005$ and apply free bits to the KL term to discourage latent collapse while retaining information useful for descriptor prediction.

The encoder is trained on reference trajectories together with random crops. Cropping increases the number of training examples and exposes the encoder to partial motions of different lengths, matching the segments queried during planning. Appendix D explains why these crops also motivate a boundary-speed penalty during retiming. The channel statistics (μ_c, σ_c) are computed from the reference inputs and kept fixed during training and planning. After training, we freeze ϕ and fit the latent reference mean μ and covariance Σ to embeddings of reference trajectories and their crops. The inverse covariance Σ^{-1} is used in c_{traj} (Eq. (4)). Table V lists the training hyperparameters.

Hyperparameter	Value
Data	
Random crops per episode	6
Crop length (fraction of episode)	$\mathcal{U}[0.4,\ 1.0]$
Encoder rate F	15 Hz
Model	
Latent dimension d	16
Descriptor dimension	113
Dropout	0.3
Objective	
KL weight β	0.005, linear warmup over 10 epochs
Free-bits threshold	0.10 nats per latent dimension
Optimization	
Optimizer	Adam, weight decay 10^{-4}
Learning rate	7×10^{-4}, cosine annealing
Batch size	128
Epochs $\times$ steps per epoch	60×800
Gradient-norm clip	5.0

TABLE V: Trajectory encoder training hyperparameters.

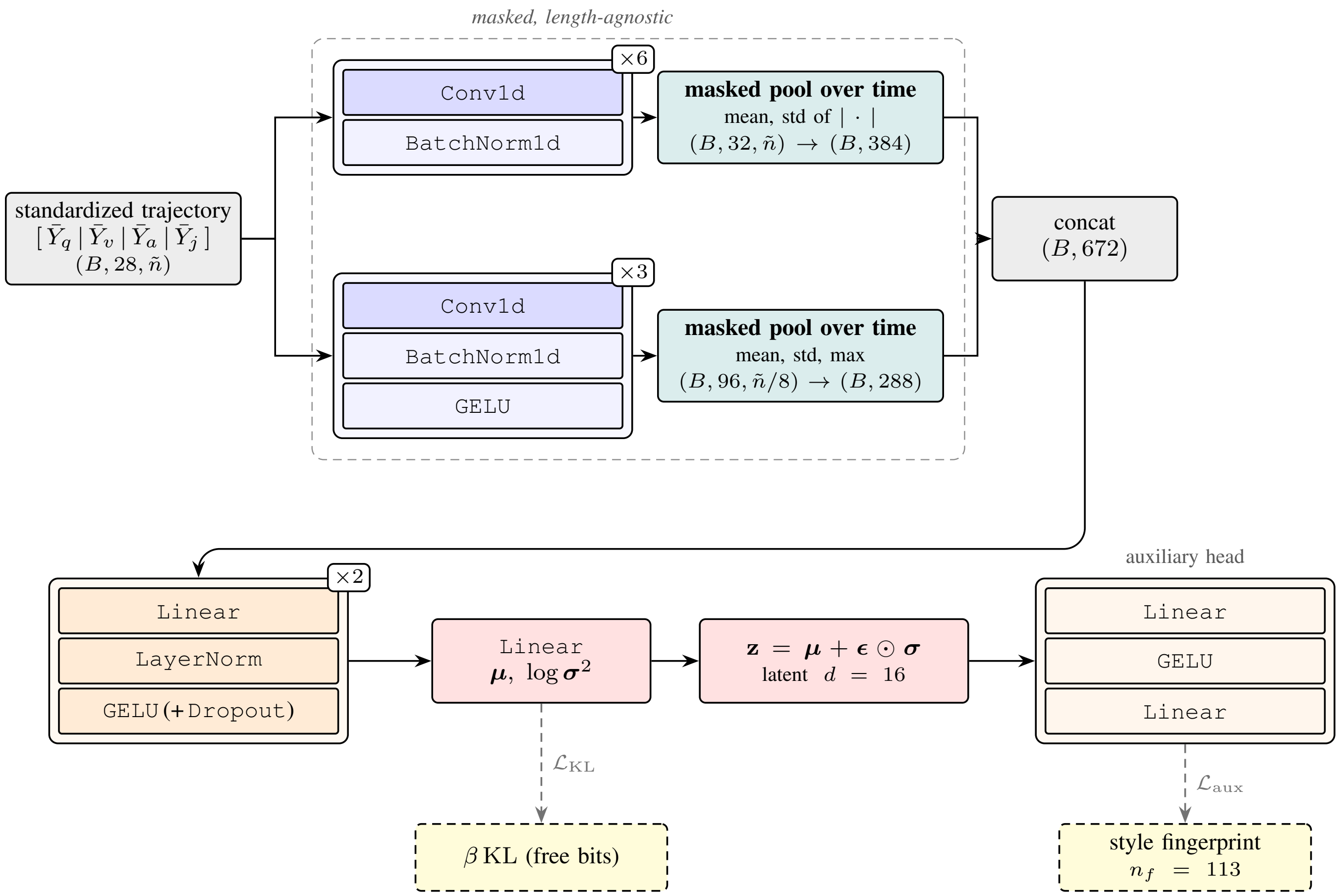

Fig. 5: Trajectory encoder architecture.

B. Endpoint Selection

1) Fitting the configuration density: The configuration density p_{conf} is a 16-component full-covariance Gaussian mixture over the 7 joint angles, fitted to 987,631 frames from 4,000 reference episodes. The configuration cost is $c_{\mathrm{conf}}(q) = -\log p_{\mathrm{conf}}(q)$.

2) IK candidate generation and grasp symmetry: For a candidate end-effector pose x, we obtain a finite set of IK solutions $\mathcal{Q}(x)$ by searching from multiple initial guesses. For grasps, we also consider the symmetry of the parallel-jaw gripper. Let R_π denote a rotation of π about the gripper's local approach axis. When the grasp model admits both x and xR_π, we solve IK for both representatives and select

$$q^\star = \operatorname*{arg\,min}_{q \in \mathcal{Q}(x) \cup \mathcal{Q}(xR_\pi)} c_{\mathrm{conf}}(q). \tag{14}$$

The stored grasp is updated to match the selected representative. For poses without this symmetry, we rank only $\mathcal{Q}(x)$. Candidates that fail a forward-kinematics check against their requested pose are discarded. If none remain, we fall back to the planner's baseline IK procedure. The associated grasps, placements, and connecting motions remain subject to the planner's feasibility checks.

3) Effect of including the rotated grasp: IK branch selection for a fixed pose considers alternative arm configurations but keeps the end-effector orientation fixed. Including the rotated grasp also allows the planner to choose between the two equivalent grasp orientations. Table VI quantifies the effect on 512 endpoints from two scenes. We report the fraction of selected configurations whose c_{conf} exceeds the 99th percentile of this cost on held-out reference data. Including the rotated grasp reduces this fraction in both scenes.

Candidate pool	Semantic Reasoning	Geometric Constraints
Best branch of the given pose only	37–42%	37–42%
Best branch over both grasp orientations	5%	7%

TABLE VI: Share of 512 selected endpoint configurations outside the 99th percentile of the reference configuration density.

C. Path Optimization

1) Evaluation pipeline: To evaluate $c_{\text{traj}}(\xi)$, we resample the segment from the trajectory optimizer's rate $1/\delta t$ to the encoder rate $F = 15\,\text{Hz}$. We then form Y, standardize it, evaluate the mean embedding $E_\phi(\xi)$, and compute the squared Mahalanobis cost in Eq. (4). The planner weights this cost by λ_{path} in Eq. (6).

2) Gradient computation and caching: Gradients propagate through the frozen encoder, channel standardization, finite differences, and resampling. For the Jacobians below, waypoint arrays and feature matrices are flattened in a fixed order. Expanding the chain rule gives

$$\begin{aligned}\frac{\partial c_{\text{traj}}(\xi)}{\partial q_{1:n}} &= 2\big(E_\phi(\xi)-\mu\big)^\top \Sigma^{-1}\frac{\partial E_\phi(\xi)}{\partial \bar{Y}} \\ &\quad\cdot \left[\sum_{b\in\{q,v,a,j\}} \frac{\partial \bar{Y}}{\partial Y_b}\frac{\partial Y_b}{\partial \tilde{q}_{1:\tilde{n}}}\right]\frac{\partial \tilde{q}_{1:\tilde{n}}}{\partial q_{1:n}}.\end{aligned} \tag{15}$$

The derivative with respect to $\bar{Y}$ is through the neural network following preprocessing. At fixed sample times, standardization, finite differencing, and resampling form an affine map, so their Jacobian is constant. We precompute $J = \partial\bar{Y}/\partial q_{1:n}$ and reuse it across solver iterations, obtaining

$$\frac{\partial c_{\text{traj}}(\xi)}{\partial q_{1:n}} = 2\big(E_\phi(\xi)-\mu\big)^\top \Sigma^{-1}\frac{\partial E_\phi(\xi)}{\partial \bar{Y}}\, J. \tag{16}$$

Only the embedding's offset from the reference mean and the encoder Jacobian vary with the current trajectory. A change in the sample times or number of waypoints requires updating the preprocessing operator. During path optimization, only the interior waypoint configurations are updated; the endpoints remain fixed.

D. Retiming

1) Time-scaling implementation: We use $M = 64$ equal arc-length intervals in the time parameterization of Eq. (7), with time fractions

$$\alpha_m(\theta) = \frac{\exp(\tanh\theta_m)}{\sum_{j=1}^{M}\exp(\tanh\theta_j)}. \tag{17}$$

This applies an elementwise $\tanh$ to θ before softmax normalization. Softmax alone can assign an arbitrarily small fraction of the total duration to an interval, leading to near-zero traversal times and large velocities or accelerations. Bounding the logits with $\tanh$ limits the ratio between any two interval durations to at most e^2, preventing individual time fractions from approaching zero. This helps avoid allocations that violate the robot's motion limits, but does not itself enforce velocity or acceleration limits, which are checked separately.

The time scaling changes progress along the fixed path and preserves the endpoint configurations, including those at gripper events. The speed $\dot{s}_{\theta,\tau}(t)$ is expressed in normalized path coordinates. The corresponding joint-space speed used in the boundary penalty is

$$v(t) = \big\|\gamma'\big(s_{\theta,\tau}(t)\big)\big\|_2\,\dot{s}_{\theta,\tau}(t). \tag{18}$$

2) Boundary-speed penalty: Let $v_{\text{start}}(\xi)$ and $v_{\text{end}}(\xi)$ denote the joint-space speeds near a segment's boundaries, and let $\bar{v}_{\text{grip}}$ be the mean speed measured immediately before and after gripper events in the reference demonstrations. We define

$$\begin{aligned}c_{\text{boundary}}(\xi) &= \big(v_{\text{start}}(\xi)-\bar{v}_{\text{grip}}\big)^2 \\ &\quad + \big(v_{\text{end}}(\xi)-\bar{v}_{\text{grip}}\big)^2.\end{aligned} \tag{19}$$

The full retiming objective combines the trajectory cost with this penalty, weighted by $\lambda_{\text{boundary}}$:

$$\min_{\theta,\,\tau>0} \quad c_{\text{traj}}(\xi_{\theta,\tau}) + \lambda_{\text{boundary}} c_{\text{boundary}}(\xi_{\theta,\tau}). \tag{20}$$

The reference density is fitted to embeddings of trajectories and random crops. In these windows, the median joint speed at the first and last frames is 0.296 rad/s, compared with 0.335 rad/s in their interiors. Thus, the latent cost alone does not reliably encourage slow motion at segment boundaries. Without the boundary term, the retiming optimizer produces boundary speeds of 0.30–0.37 rad/s. The reference target $\bar{v}_{\mathrm{grip}} = 0.07$ rad/s is measured by averaging joint speeds at frames immediately before and after gripper events in reference episodes.

3) Optimization and candidate selection: We optimize Eq. (20) using Adam [30] from 16 initial durations in parallel, using gradients through the path interpolation and frozen encoder. Each candidate is resampled at the robot's control timestep and rejected if it exceeds velocity or acceleration limits. Each survivor is then resampled at exactly $F = 15$ Hz and evaluated with c_{traj}; the candidate with the lowest cost is returned. The boundary penalty therefore contributes to optimization, while the final ranking uses the latent trajectory cost.

E. Planner and Fine-Tuning Hyperparameters

Table VII lists the planner hyperparameters used for data generation, and Table VIII lists the VLA fine-tuning hyperparameters. All other planner settings use the defaults of the base TAMP system.

Hyperparameter	Value
Endpoint selection	
IK solutions per candidate pose ($\lvert\mathcal{Q}(x)\rvert$)	12
Path optimization	
Trajectory-cost weight (λ_{path})	25,000
Retiming	
Arc-length intervals (M)	64
Parallel initial durations	16
Adam iterations	500
Learning rate, time-allocation, (θ)	0.08
Learning rate, total duration (τ)	0.0375
Boundary-penalty weight ($\lambda_{\mathrm{boundary}}$)	50
Gripper-event speed target ($\bar{v}_{\mathrm{grip}}$)	0.07 rad/s

TABLE VII: Planner hyperparameters used for data generation.

Hyperparameter	Value
Base policy	$\pi_{0.5}$-DROID
Action horizon	16
Optimizer	AdamW (β_1=0.9, β_2=0.95)
Peak learning rate	2.5×10^{-5}
Learning-rate schedule	1,000 warmup steps, cosine decay to 2.5×10^{-6}
Gradient-norm clip	1.0
EMA decay	0.99
Batch size	256
Training steps	20,000

TABLE VIII: VLA fine-tuning hyperparameters, shared by all data sources.

F. Task Evaluation Subgoals

We use the binary subgoals in Table IX to score evaluation rollouts. A rollout succeeds when it meets every subgoal for its task, and task progress is the fraction of subgoals met at termination. We report the mean progress and the success rate over rollouts.

G. Failure Breakdown

Table X breaks down the failed rollouts of each policy by failure mode. *Grasping* failures miss or drop an object during pickup. *Manipulation/placement* failures grasp the object but release it in the wrong place or disturb other objects. *Joint-space misalignment* failures move the arm into unusual postures from which it cannot recover or complete the task. *Semantic* failures act on the wrong object or goal, or ignore the instruction. The two *timeout* modes separate policies that stop making

Subgoal	Criterion
Geometric Constraints (7 subgoals)	
`blue_toy_on_plate`	Blue toy ends up on the plate
`brown_toy_on_plate`	Brown toy ends up on the plate
`pink_toy_on_plate`	Pink toy ends up on the plate
`blue_toy_lifted`	Blue toy is lifted off the table at some point
`brown_toy_lifted`	Brown toy is lifted off the table at some point
`pink_toy_lifted`	Pink toy is lifted off the table at some point
`items_spaced`	At least two toys are on the plate, and no placed toys touch
Semantic Reasoning (7 subgoals)	
`blue_toy_in_bowl`	Blue elephant plush ends up in the toy bowl
`pink_toy_in_bowl`	Pink plush ends up in the toy bowl
`banana_in_bowl`	Banana ends up in the food bowl
`blue_toy_lifted`	Blue elephant plush is lifted off the table at some point
`pink_toy_lifted`	Pink plush is lifted off the table at some point
`banana_lifted`	Banana is lifted off the table at some point
`sort_clean`	At least two objects are placed, and neither bowl holds both a toy and a food item
Multi-Step Reasoning (7 subgoals)	
`banana_off_tray`	Banana ends up released on the table, off the tray
`pink_toy_on_tray`	Pink plush ends up resting on the tray itself
`brown_toy_on_tray`	Brown plush ends up resting on the tray itself
`banana_lifted`	Banana is lifted off the tray at some point
`pink_toy_lifted`	Pink plush is lifted off the table at some point
`brown_toy_lifted`	Brown plush is lifted off the table at some point
`tray_cleared_first`	At least one toy is placed on the tray, and the banana is already off the tray when the first toy is placed
Deformable Object Manipulation (2 subgoals)	
`fold_any`	The cloth is folded any amount
`fold_large`	The folded cloth covers less than 60% of its original area on the table

TABLE IX: Task-specific evaluation subgoals and success criteria.

progress in the middle of the episode (*stuck*) from policies that are still making progress when the evaluation horizon ends (*did not finish*).

Approach	Number of Failures	Grasping	Manipulation/ Placement	Joint-Space Misalignment	Semantic	Timeout (Stuck)	Timeout (Did Not Finish)
Geometric Constraints							
$\pi_{0.5}$-DROID	20	10%	<u>30%</u>	15%	**45%**	0%	0%
Raw TAMP	19	<u>42%</u>	**47%**	5%	5%	0%	0%
DATAFARM w/o style	18	**89%**	<u>6%</u>	<u>6%</u>	0%	0%	0%
DATAFARM w/o timing	20	15%	10%	5%	5%	<u>20%</u>	**45%**
DATAFARM w/o joint space	13	0%	**46%**	**46%**	0%	0%	<u>8%</u>
DATAFARM	7	**71%**	<u>29%</u>	0%	0%	0%	0%
Human teleop (oracle)	10	0%	**100%**	0%	0%	0%	0%
Multi-Step Reasoning							
$\pi_{0.5}$-DROID	20	0%	0%	<u>15%</u>	**85%**	0%	0%
Raw TAMP	19	<u>37%</u>	**42%**	5%	0%	0%	16%
DATAFARM w/o style	20	**70%**	<u>20%</u>	0%	10%	0%	0%
DATAFARM w/o timing	20	15%	10%	0%	5%	<u>20%</u>	**50%**
DATAFARM w/o joint space	17	**35%**	<u>18%</u>	**35%**	6%	0%	6%
DATAFARM	13	<u>15%</u>	**69%**	0%	<u>15%</u>	0%	0%
Human teleop (oracle)	11	0%	**82%**	0%	<u>18%</u>	0%	0%
Semantic Reasoning							
$\pi_{0.5}$-DROID	19	0%	0%	<u>11%</u>	**89%**	0%	0%
Raw TAMP	17	6%	<u>18%</u>	12%	**53%**	0%	12%
DATAFARM w/o style	13	**62%**	8%	0%	<u>23%</u>	0%	8%
DATAFARM w/o timing	20	10%	5%	0%	<u>20%</u>	10%	**55%**
DATAFARM w/o joint space	16	<u>19%</u>	0%	**56%**	6%	0%	<u>19%</u>
DATAFARM	6	**50%**	0%	0%	**50%**	0%	0%
Human teleop (oracle)	2	0%	**50%**	0%	**50%**	0%	0%

TABLE X: Failure modes among failed rollouts. The count gives the number of failures for each approach and task. **Bold** marks the most common mode in each row; <u>underline</u> marks the runner-up.