SocialNav: Training Human-Inspired Foundation Model for Socially-Aware Embodied Navigation

Supplementary Material

Content

A. Supplementary Materials for the SocNav Dataset and Benchmark

A.1. Construction of Trajectories in D_{sim}

The simulated subset D_{sim} in the SocNav Dataset is designed to provide both *standard trajectories* and *recovery trajectories* in structured social scenes. This section details the construction pipeline and provides schematic visualizations.

A.1.1. Standard Trajectory Generation

For each scene, we first construct a navigation graph over the legal road network (sidewalks, crosswalks, plazas, etc.):

1. **Road network annotation.** For SocCity, we derive the traversable occupancy map M_{occ} by mapping the semantic ground annotations inherent to the 3D city assets. For other 3D scenes (SocialGS, MatterPort3D, InteriorGS), we manually annotate M_{occ}. We ensure a unified definition of traversability across all datasets (refer to Sec.A.2.2 for details). Subsequently, guided by the M_{occ} of each scene, we manually annotate the road network along the centerlines of the traversable paths.
2. **Trajectory Generation.** We randomly sample a start position $\boldsymbol{s}$ and a goal position $\boldsymbol{g}$ on the traversable road network, ensuring that the geodesic distance $d(\boldsymbol{s}, \boldsymbol{g})$ along the network exceeds a minimum threshold $\ell_{\text{min}} = 50\,\text{m}$. We then employ an A* planner to compute the shortest path, which serves as the standard trajectory $\tau^{\star}$. These trajectories represent ideal navigation behavior that consistently maintains a safe distance from untraversable areas.

A.1.2. Recovery Trajectory Generation

While standard trajectories provide an ideal reference, training a policy exclusively on such expert data can lead to brittleness. The agent may fail to recover from even small deviations, a classic problem of covariate shift in imitation learning. To address this and equip the agent with robust recovery capabilities, we augment the standard expert demonstrations with a set of *recovery trajectories*. These are designed to simulate scenarios where the agent starts in a suboptimal or dangerous state and must execute a corrective maneuver to rejoin the ideal path.

The generation process for these recovery trajectories is as follows:

1. **Define Convergence Point.** For each standard trajectory $\tau^{\star}$ starting at $\boldsymbol{s}$, we first identify a convergence point $\boldsymbol{p}_{\text{conv}}$ on $\tau^{\star}$, located 5 meters ahead of the start point $\boldsymbol{s}$.
2. **Sample Recovery Start.** We then generate a corresponding recovery start point $\boldsymbol{s}_{\text{rec}}$ by applying a random lateral offset (either left or right) to the standard start point $\boldsymbol{s}$. To construct a challenging recovery task, the initial orientation at $\boldsymbol{s}_{\text{rec}}$ is deliberately misaligned:
   - If $\boldsymbol{s}_{\text{rec}}$ is offset to the **left**, the initial heading is set to a random angle within $[-90^\circ, -45^\circ]$ relative to the forward direction of the standard path.
   - If $\boldsymbol{s}_{\text{rec}}$ is offset to the **right**, the initial heading is set to

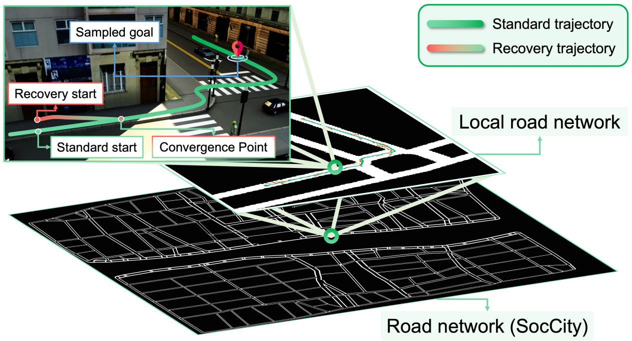


Figure 5. **Standard and recovery trajectories in D_{sim}.** Visual examples from SocCity scenes. Green curves denote standard expert trajectories obtained by A* planning on the navigation graph, while the other curves depict locally sampled recovery trajectories originating from intermediate points. The background shows the semantic occupancy map $\mathcal{M}_{\text{occ}}$ with walkable regions (white) and non-traversable regions (black).

an angle within $[45°, 90°]$.
This forces the agent to start by facing away from the correct path, requiring an immediate and decisive turn.

3. **Generate Recovery Trajectory.** A dense recovery path τ^{rec} is generated by linearly interpolating between the recovery start state $\boldsymbol{s}_{\text{rec}}$ and the convergence point $\boldsymbol{p}_{\text{conv}}$ with a fixed spatial step of approximately 5 cm. To simulate natural human micro-corrections and avoid trivial straight-line paths, we perturb each interpolated point $\boldsymbol{q}_t$ with small, zero-mean Gaussian noise:

$$\boldsymbol{q}'_t = \boldsymbol{q}_t + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, (0.01\,\mathrm{m})^2\mathbf{I}). \tag{7}$$

   The final perturbed path $\tau^{\text{rec}'}$ is formed by concatenating τ^{rec} with the path from $\boldsymbol{p}_{\text{conv}}$ to $\boldsymbol{g}$. It is kept only if all its points remain in cells of $\mathcal{M}_{\text{occ}}$ with sufficient clearance from obstacles.

These systematically constructed recovery trajectories mimic plausible failure scenarios, such as starting from the wrong orientation or position, and provide explicit supervision on how to execute safe and efficient corrective actions. This enriches the training data far more effectively than simple random noise, significantly improving the policy's robustness in challenging situations.

A.1.3. Visualization of Standard and Recovery Trajectories

Figure 5 illustrates typical standard and recovery trajectories on $\mathcal{M}_{\text{occ}}$, highlighting the diversity introduced by local recovery paths.

A.2. Annotation of Socially Traversable Regions

Socially traversable regions provide the supervision signal for training the SocialNav Brain to predict socially compliant polygons. We describe the annotation pipeline and the corresponding guidelines.

A.2.1. Annotation Pipeline

1. **Internet-scale data collection.** We gather first-person Internet videos and images depicting pedestrians moving in outdoor scenes such as streets, campuses, and parks.
2. **Automatic filtering.** A vision-language model is used to filter out frames with unsuitable viewpoints (e.g., too close to walls, heavily occluded, or dominated by sky/ground) and to discard low-quality images.
3. **Manual polygon annotation.** Human annotators draw coarse polygons on the remaining frames to delineate socially traversable regions. We create one or more polygons to cover all regions that is legally and socially allowed to walk.

A.2.2. Annotation Guidelines

- **Socially Traversable Regions.** These are defined as outdoor areas where pedestrians are permitted to walk by both legal regulations and common social norms. Annotators were instructed to label surfaces such as sidewalks, pedestrian-only streets, marked crosswalks, public plazas, and accessible outdoor staircases. For indoor scenes, all non-obstacle areas are considered traversable.
- **Non-Traversable Regions.** This category encompasses all areas that are unsafe, illegal, or socially unacceptable for pedestrian traffic. Key examples include motor vehicle lanes, bike lanes, bus lanes, green belts, ornamental lawns, flower beds, water bodies, etc.
- **Annotation Protocol and Polygon Standards.** Annotators draw coarse, low-vertex polygons to cover the full extent of the walkable surface; pixel-perfect alignment with curbs or markings is not required. Annotation focuses on the walkable region directly connected to the camera's viewpoint. A single polygon can span multiple connected surfaces (e.g., a sidewalk leading into a plaza). Isolated regions that cannot be reached without crossing non-traversable areas are ignored, except for pedestrian safety islands that are visibly reachable via crosswalks.

A.2.3. Qualitative Analysis of Social Traversability Prediction

Figure 6 presents qualitative comparisons of predicted socially traversable regions across six unseen test scenes. Each column corresponds to a different scene, while the four rows show: (1) Qwen2.5-VL predictions in SocCity, (2) SocialNav predictions in SocCity, (3) Qwen2.5-VL predictions in the real world, and (4) SocialNav predictions in the real world. Green polygons denote model-predicted socially traversable regions.

Across both simulated and real scenes, the base Qwen2.5-VL-3B model often produces polygons that are coarse, spatially misaligned, or leak into socially invalid

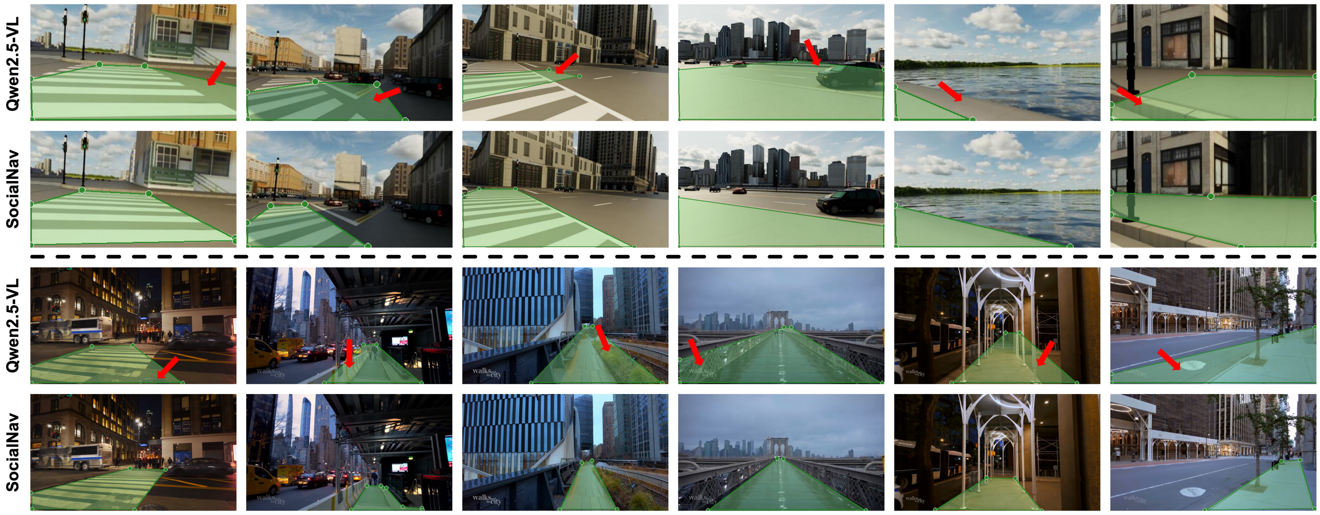


Figure 6. **Predicted socially traversable regions on unseen scenes.** Green polygons denote predicted socially traversable regions, and red arrows highlight areas incorrectly classified as traversable. SocialNav yields more semantically aligned polygons in both domains.

areas such as curb edges, grass, or vehicle lanes. In contrast, the SocialNav Brain, trained with our large-scale traversability annotations, provides predictions that are more structured and socially consistent: polygons more tightly follow sidewalks, pedestrian paths, and crosswalks, and avoid visually similar but non-walkable regions. The improvements hold across all six diverse unseen scenes, indicating that fine-tuning not only enhances accuracy but also significantly strengthens cross-domain generalization.

A.3. Navigation Chain-of-Thought Construction

To construct the Cognitive Activation Dataset (CAD) for navigation, we elicit chain-of-thought (CoT) style explanations using a instruction prompt. The goal is to obtain, for each navigation step, (i) a structured reasoning trace that explains the agent's decision in terms of scene layout, social norms, and future consequences, and (ii) a final discrete action selected from a predefined action space. This subsection details the prompt design and the corresponding input–output specification.

A.3.1. Prompt Design

Task and Role Description. We explicitly cast the brain model as the *Thinking Module* of a professional navigation system. Its responsibility is to produce a logically coherent CoT and a final movement decision.

Then the prompt enforces a strict three-stage reasoning protocol:

1. **Global situation analysis**: jointly parse all input information, including the robot state, the target location, and visual observations.
2. **Chain-of-thought generation**: A **CoT segment**, enclosed between `[CoT]` and `[\CoT]`, containing the full reasoning process which includes structured, logically tight reasoning process that evaluates the consequences of different candidate actions under social and geometric constraints.
3. **Final decision**: A **Decision segment**, enclosed between `[Decision]` and `[\Decision]`, containing exactly one chosen action from the predefined action space.

Prompt Template. Figure 7 presents the complete English-language prompt template used to generate the navigation CoTs for our CAD. The resulting CoT and Decision segments are stored alongside the visual observations and trajectory states, forming rich supervision for the Brain Module to acquire socially aware navigation reasoning.

A.3.2. Qualitative Example of Navigation CoT

Figure 8 provides a qualitative example of the chain-of-thought (CoT) generated by our Brain Module for a navigation decision in an unseen urban intersection. The CoT showcases the model's ability to perform complex, multi-stage reasoning by integrating its state, goal, and rich perceptual understanding.

In this example, the model correctly identifies a conflict between the long-term goal vector (to the forward-right) and the immediate safety and social constraints. Instead of pursuing a direct path, the model decomposes the problem, recognizing that the sanctioned crosswalk directly ahead, aided by a green pedestrian signal, is the necessary intermediate step. It explicitly evaluates actions against a learned hierarchy of navigational rules—1) Safety, 2) Social Compliance, and 3) Task Efficiency—to invalidate unsafe actions

Role and task description:
You are now the *Thinking Module* of a professional navigation AI system. Your single core task is to generate, for every movement of a quadruped robot dog, a comprehensive, in-depth, and logically rigorous *Chain of Thought (CoT)*, and then output a clear movement decision. This CoT must integrate all available environmental information and explain why the final decision is the optimal choice under the current situation.

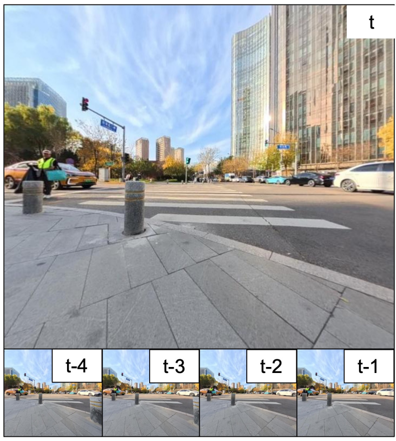
Workflow (you *must* strictly follow these three steps, without omission or reordering):
1. Comprehensive analysis: parse all input information, including the robot's own state, the goal position, and environmental perception.
2. CoT generation: based on your analysis, build a structured, logically sound reasoning process and evaluate all plausible alternatives.
3. Final decision: select one and only one action from the candidate action space, and output it in the specified format.

Input information:
1. Scene information:
a. Format: [[history_coord_1], [history_coord_2], [history_coord_3], [history_coord_4], [current_coord], [goal_coord]] together with the corresponding observation images in the same order.
b. Interpretation: this sequence represents the robot dog's past and current coordinates, followed by the target coordinate.
c. Coordinate system: the positive Y direction corresponds to forward, and the positive X direction corresponds to right.

2. Candidate action space:
-- Go Straight
-- Stay Still
-- Move Backward
-- Move Forward-Left
-- Move Forward-Right
-- Move Left
-- Move Right

Output requirements:
You must generate a *single*, continuous block of text (no meaningless spaces, blank lines, or extra line breaks). Absolutely no additional explanations, headings, greetings, or apologies are allowed.
You must strictly follow the format below:
[CoT]... [\CoT][Decision]... [\Decision]

Figure 7. **Navigation Chain-of-Thought Prompt Template.** The full prompt used to construct the Cognitive Activation Dataset.



[CoT]Ego-state Analysis: The historical trajectory [[0.12, -4.12], [0.11, -3.29], [0.05, -2.05], [0.02, -0.97], [0.0, 0.0]] indicates a consistent forward motion pattern. The global target is located at [22.66, 47.713], which is in the forward-right quadrant relative to the robot's current pose and heading. The initial path plan must reconcile this vector with immediate environmental constraints. Environmental Perception and Scene Understanding: The robot is situated at an urban intersection corner. The static environment is composed of a sidewalk, bollards, and a multi-lane road. Critically, a marked crosswalk (zebra crossing) is located directly ahead. The traversable area is thus partitioned; direct diagonal movement is obstructed by the road, a high-risk zone. The only sanctioned path to cross this obstacle is the designated crosswalk. Dynamic analysis identifies a green pedestrian traffic signal for the crosswalk ahead, indicating a temporary safe corridor and that opposing vehicle traffic is expected to yield. Navigation Rules and Social Norms: The navigational hierarchy is clear: 1) Safety: Prohibits entering active traffic lanes outside designated crossings. 2) Social Compliance: Mandates adherence to traffic signals and the use of crosswalks. 3) Task Efficiency: Requires making progress toward the target. Action Evaluation and Synthesis: The evaluation of the action space under these rules leads to a clear conclusion. Move Forward-Right or Move Right would constitute a critical safety violation by entering the road directly. Stay Still is safe but inefficient, failing to utilize the safe crossing window afforded by the green light. Move Left or Turn Around are directionally incorrect and thus inefficient. The Move Forward action is the uniquely optimal choice. It is the first step in a decomposed, multi-stage plan to reach the forward-right target. This action adheres to all rules: it is safe, socially compliant, and efficient. Therefore, initiating the crossing by moving forward is the only logical action.[\CoT]
[Decision]Go Straight[\Decision]

Figure 8. **Example of a navigation chain-of-thought (CoT) in an unseen crossing scenario.** Given the historical observations (left) and a goal in the forward-right direction, the Brain Module generates a structured CoT (right). The CoT demonstrates hierarchical decision-making, prioritizing Safety (no jaywalking) and Social Compliance (using the crosswalk) over direct path efficiency.

(e.g., `Move Forward-Right`) and inefficient ones (e.g., `Stay Still`). The final decision to `Go Straight` is thus not a simple reactive choice but the output of a deliberate plan to safely and compliantly progress toward the goal.

A.4. Details of the SocNav Benchmark

The SocNav Benchmark provides a closed-loop evaluation platform for jointly assessing navigation performance and social compliance.

A.4.1. Metric Definitions

Standard point-goal Navigation metrics.

- **MAOE.** Following CityWalker [27], we compute the Average Orientation Error (AOE) at each prediction step k as the mean angular difference between the predicted action $\hat{a}_{i_k}$ and the ground-truth action a_{i_k} over all samples:

$$\mathrm{AOE}(k) = \frac{1}{n}\sum_{i=1}^{n}\theta_{i_k}, \quad \theta_{i_k} = \arccos\left(\frac{\langle \hat{a}_{i_k}, a_{i_k}\rangle}{\|\hat{a}_{i_k}\|_2\,\|a_{i_k}\|_2}\right). \tag{8}$$

  The Maximum Average Orientation Error (MAOE) aggregates over predicted horizons:

$$\mathrm{MAOE} = \frac{1}{n}\sum_{i=1}^{n}\max_{k}\theta_{i_k}, \tag{9}$$

  which emphasizes the worst-case orientation mismatch along future steps.
- **Success Rate (SR).** The fraction of episodes in which the agent reaches within 3 m of the target with fewer than three collisions.
- **Route Completion (RC).** The ratio between the geodesic distance from start to the final position and the geodesic distance from start to goal.
- **Success weighted by Path Length (SPL).** The SPL metric, as in prior navigation work, jointly captures success and path efficiency.

Social compliance metrics: DCR and TCR. The Distance Compliance Rate (DCR) measures the proportion of distance spent in socially traversable regions:

$$\mathrm{DCR} = \begin{cases} \dfrac{d_{\text{compliant}}}{d_{\text{actual}}}, & \text{if } s = 1, \\ 0, & \text{otherwise,} \end{cases} \tag{10}$$

where $s \in \{0, 1\}$ indicates success, $d_{\text{compliant}}$ is the distance traveled in regions labeled as socially traversable, and d_{actual} is the total traveled distance.

The Time Compliance Rate (TCR) is defined analogously by replacing distance with duration spent in compliant regions. Both metrics are computed only for successful episodes and are averaged over all tasks.

A.4.2. Pedestrian Placement and Behavior Model

To simulate realistic social environments, we populate the evaluated scenes with dynamic pedestrians:

Table 5. **Model Architecture and Parameters.**

Brain Module (Qwen2.5-VL-3B)	
Layers / Hidden / MLP	36 / 2048 / 11008
Attention heads (KV)	16 (2)
Activation / Norm	SiLU / RMSNorm
Max context	128k (SW: 32k)
RoPE scaling	multi-segment (16/24/24)
Vocab size	151,673
Action Expert (Diffusion Transformer)	
Action dim / Chunk size	2 / 5
Flow-matching steps	5

- **Placement and Density.** To ensure realistic positioning, pedestrians are spawned within a specific margin, between 50 cm and 100 cm from the inner boundary of the socially traversable regions. We maintain a pedestrian density of not exceeding 6 individuals per 100 meters of walkable path.
- **Behavior model.** The behavior of each pedestrian is governed by a skeletal animation system, which supports both walking and running actions. Each pedestrian selects a random goal on the road network and moves along the shortest path at a speed sampled from a truncated normal distribution (mean 1.0 m/s, standard deviation 0.2 m/s).
- **Interaction with the robot.** Pedestrians do not explicitly cooperate with the robot; they follow their own routes. Consequently, the robot must proactively adapt its trajectory to maintain social distances and avoid intruding into non-traversable regions.

B. Supplementary Materials for the SocialNav Foundation Model

The **SocialNav** foundation model adopts the hierarchical Brain–Action design described in the main paper. Here we summarize key architectural details.

B.1. Model Architecture and Parameters

Table 5 summarizes the architectural parameters of the **Brain Module** (Qwen2.5-VL-3B) and the **Action Expert** (Diffusion Transformer). The table includes structural details necessary for reproducibility.

B.2. Training Details

This section provides concise supplementary training information for the three-stage pipeline introduced in the main paper. Table 6 summarizes both the trainable components and the datasets used in each stage, while some shared optimization settings remain unchanged across the entire pipeline.

Table 6. **Trainable Components and Datasets Across Stage 1–3.** Checkmarks indicate modules updated during each stage.

Stage	Datasets	Trainable Modules			
		VLM Brain	Vision Encoder	Waypoint Encoder	Action Expert
Pre-training	D_{video}, D_{sim}, D_{cog}	✓	✓	✓	✓
Fine-tuning	D_{real}	×	×	×	✓
SAFE-GRPO	D_{sim} (SocCity)	×	×	×	✓

Table 7. **Hyperparameter Settings.**

Settings	**Value**
Optimizer	AdamW
Adam betas (β_1, β_2)	$(0.9, 0.95)$
Weight decay	0.1
LR scheduler	Cosine decay
Precision	BF16
Flash Attention 2	Enabled
Gradient accumulation	1
Gradient checkpointing	Enabled

Hyperparameter settings are summarized in Table 7.

B.3. SAFE-GRPO Reward Design

Social Compliance Reward $\mathcal{R}_{\text{social}}$: This is the primary incentive for respecting both physical safety and social norms. From $\mathcal{M}_{\text{occ}} \in \{0,1\}^{H \times W}$, we compute a Distance Transform (DT) map $D(\boldsymbol{x})$, which assigns each traversable location $\boldsymbol{x}$ its Euclidean distance to the nearest non-traversable cell. This DT map encodes both collision avoidance and social distancing principles—higher values indicate safer, more normatively acceptable regions.

Let $\{\boldsymbol{x}_t\}_{t=1}^{T}$ be the predicted trajectory in world coordinates, and let $\bar{d}_{\text{pred}} = \frac{1}{T}\sum_{t=1}^{T} D(\boldsymbol{x}_t)$ denotes the average obstacle-free clearance along the path. Similarly, we compute $\bar{d}_{\text{gt}}$ for the expert trajectory. The social reward is then formulated as:

$$\mathcal{R}_{\text{social}} = \beta \cdot \sigma\left(\frac{\bar{d}_{\text{pred}} - \bar{d}_{\text{gt}}}{\alpha}\right), \tag{11}$$

where $\sigma(\cdot)$ denotes the sigmoid function, and hyperparameters $\alpha = 0.5$, $\beta = 2.0$ control sensitivity and scaling. This formulation rewards trajectories that maintain comparable or greater clearance than the expert.

Expert Trajectory Similarity Reward ($\mathcal{R}_{\text{expert}}$): We measure similarity in both spatial proximity and directional consistency. Given predicted trajectory $\boldsymbol{p}$ and expert $\boldsymbol{g}$ in world coordinates:

$$\mathcal{R}_{\text{expert}} = w_d \cdot r_{\text{dist}} + w_\theta \cdot r_{\text{dir}}, \tag{12}$$

where:

$$r_{\text{dist}} = \exp\left(-\frac{1}{T}\sum_{t=1}^{T} \|\boldsymbol{p}_t - \boldsymbol{g}_t\| \Big/ \tau_d\right), \tag{13}$$

$$r_{\text{dir}} = \frac{1}{2}\left(\cos(\Delta\theta_{\text{avg}}) + 1\right) \in [0, 1], \tag{14}$$

with $w_d = 0.7$, $w_\theta = 0.3$, $\tau_d = 1.0\,\text{m}$. Here, $\Delta\theta_{\text{avg}}$ is the average angular difference between consecutive displacement vectors.

Trajectory Smoothness Reward ($\mathcal{R}_{\text{smooth}}$): We encourage consistent step lengths by penalizing high variance in inter-step distances:

$$\mathcal{R}_{\text{smooth}} = \exp\left(-\frac{\operatorname{std}(\{\|\Delta \mathbf{x}_t\|\}_{t=2}^{T})}{\alpha_s}\right), \tag{15}$$

where $\alpha_s = 0.8$, and $\operatorname{std}(\cdot)$ computes the standard deviation of step magnitudes. A lower variance yields higher reward, promoting natural gait-like movement.

Path Efficiency Reward ($\mathcal{R}_{\text{efficiency}}$): To encourage forward progress without excessive detours, we compare the agent's net advancement to that of the expert:

$$\mathcal{R}_{\text{efficiency}} = \beta_l \cdot \sigma\left(\frac{\|\boldsymbol{x}_T - \boldsymbol{x}_0\|_2 - \|\boldsymbol{x}_T^{\text{gt}} - \boldsymbol{x}_0^{\text{gt}}\|_2}{\alpha_l}\right), \tag{16}$$

where $\alpha_l = 5.0$, $\beta_l = 2.0$. By combining these reward components, our design ensures that the agent learns to navigate not only effectively, but also in a manner that is predictable, respectful, and aligned with human expectations within shared environments.

B.4. Ablation on SAFE-GRPO Reward Funtions

We ablate the $\mathcal{R}_{\text{social}}$ defined in Eq. (4) of the main paper. For the variant, we set the weight of the $\mathcal{R}_{\text{social}}$ to zero while keeping all other training configurations unchanged.

The results in Table 8 show that $\mathcal{R}_{\text{social}}$ is crucial for high DCR and TCR; without it, the agent tends to take shorter but socially risky shortcuts.

C. Additional Experimental Results

C.1. Extended Open-Loop Results on the CityWalker Benchmark

We provide an extended comparison on the CityWalker open-loop benchmark. The model variants marked with asterisk (*) denotes the models that were exclusively trained on the D_{real} dataset.

Compared to the official CityWalker model, the retrained CityWalker* shows consistent improvements across all scenarios, confirming that our D_{real} provides a more comprehensive navigation motion priors. Similarly, GNM*, ViNT*, and NoMaD* benefit from retraining.

Table 8. **Ablation on SAFE-GRPO reward components on the SocNav Benchmark.** We deactivate each reward term in turn by setting its weight to zero. Checkmarks indicate that the reward is used.

Variant	**Reward terms**				**SocNav Benchmark**				
	$\mathcal{R}_{\text{social}}$	$\mathcal{R}_{\text{expert}}$	$\mathcal{R}_{\text{smooth}}$	$\mathcal{R}_{\text{effciency}}$	SR↑	RC↑	SPL↑	DCR↑	TCR↑
w/o $\mathcal{R}_{\text{social}}$	–	✓	✓	✓	84.7	90.3	**78.5**	61.4	62.1
Full SAFE-GRPO	✓	✓	✓	✓	**86.1**	**91.2**	77.4	**82.5**	**82.9**

Table 9. **Open-Loop Evaluation on CityWalker Benchmark** [27]. We evaluate MAOE metric in each critical scenario for all methods. Percentages under scenarios indicate their data proportions. The ”Mean” column shows scenario means averaged over six scenarios; ”All” shows sample means over all data samples.

Method	**Mean**	**Turn** 8%	**Crossing** 12%	**Detour** 12%	**Proximity** 6%	**Crowd** 7%	**Other** 55%	**All** 100%
GNM* [37]	15.2	29.5	13.6	11.9	13.6	12.3	10.4	11.5
ViNT* [38]	15.8	30.0	14.6	12.3	14.0	12.7	11.0	12.2
NoMaD* [43]	17.8	33.4	16.5	14.8	16.2	13.6	12.2	12.3
CityWalker* [27]	14.2	25.0	12.8	12.5	13.2	11.5	10.0	11.2
SocialNav*	11.7	21.5	10.9	10.4	9.8	8.9	8.7	9.4
SocialNav (Full)	**10.2**	**20.1**	**8.8**	**8.4**	**8.9**	**7.6**	**7.2**	**7.8**

SocialNav (Full) achieves the lowest MAOE in every scenario and in the overall mean. The largest relative gains are observed in the *Turn* and *Crossing* categories, where understanding of social layout and high-level semantics is particularly important. This trend supports our claim that integrating the Brain Module with the flow-based Action Expert leads to trajectories that more closely match human-operated behaviors.

C.2. Additional Real-World Deployment Visualizations

To further illustrate the real-world performance of SocialNav, we visualize third-person view from the Unitree Go2 deployments described in Table 3.

The visualizations highlight that SocialNav can be executed in real-time on a cloud server with an NVIDIA A10 GPU, maintaining over 5 Hz control frequency while preferring socially acceptable walkways even when shorter but socially inappropriate shortcuts exist.

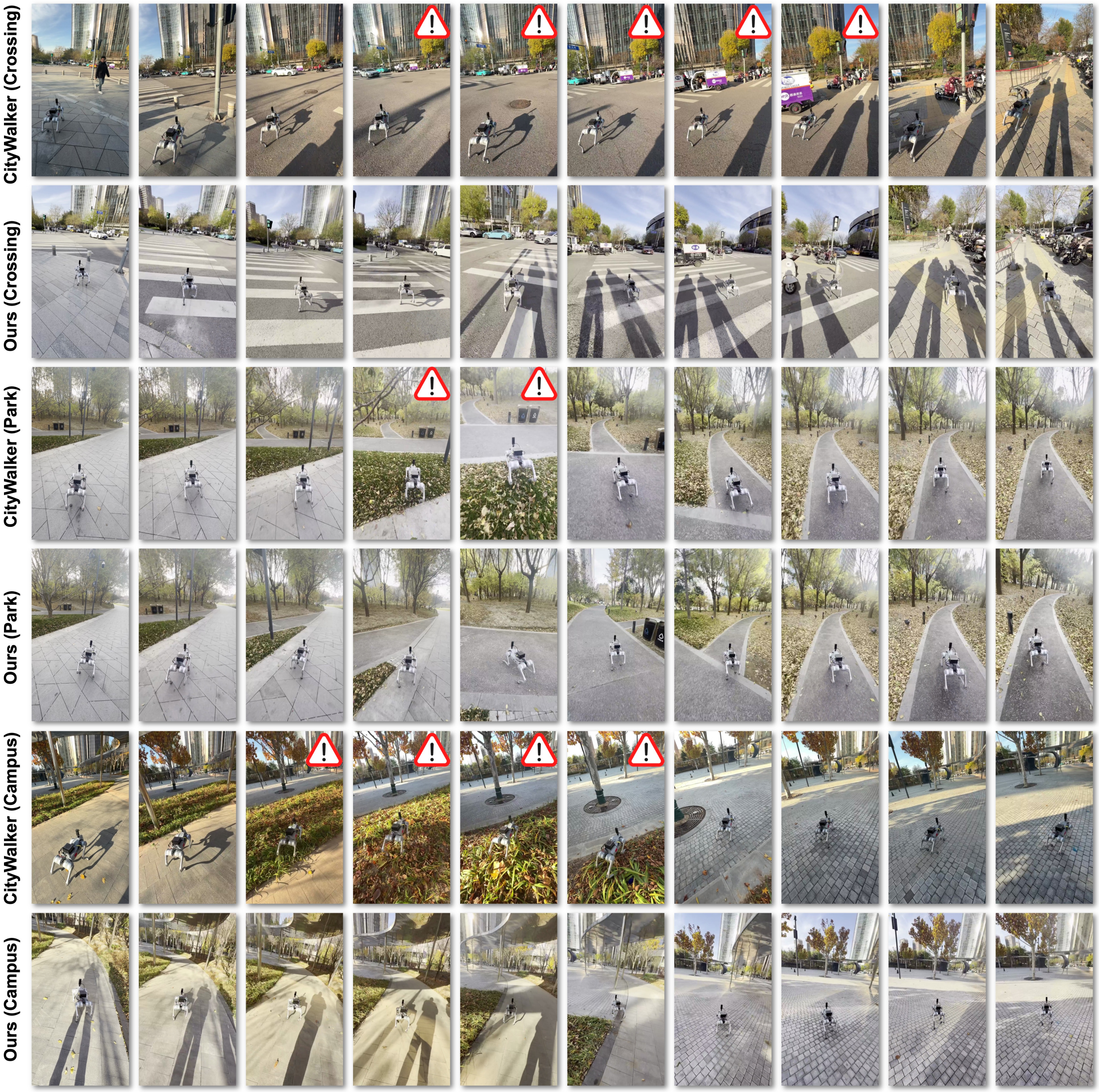


Figure 9. **Real-world deployment visualizations.** Third-person views of the Unitree Go2 robot navigating in street, park, and campus environments. SocialNav successfully follows sidewalks, avoids stepping onto lawns or driveways, respects pedestrian flows.