VISTA-MPC

Vision-Language Informed Socially Attentive Model Predictive Control for Robot Navigation

Jiaxi Huo

University of Technology Sydney

January 29, 2026

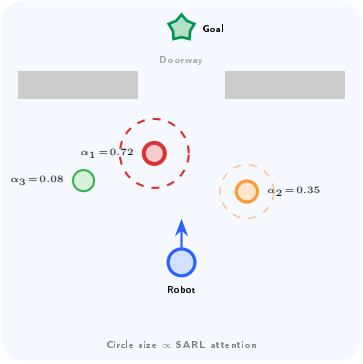
Outline

The Challenge of Social Robot Navigation

Problem: Robots navigating in human-populated environments must respect *implicit social norms*, not just avoid collisions.

Key challenges:

- Variable crowd density and scene structure
- Not all pedestrians are equally important
- Context-dependent social behavior (doorway $\neq$ open space)
- Real-time control under uncertainty
- Need for *interpretable* decisions



Our Insight: Combine **learned social attention** (SARL), **scene understanding** (VLM), and **optimal control** (MPC) in a principled hierarchy.

Limitations of Existing Approaches

Reactive Methods (DWA, RVO2)

- Treat all pedestrians equally
- No scene context understanding
- Myopic — no lookahead

End-to-End RL

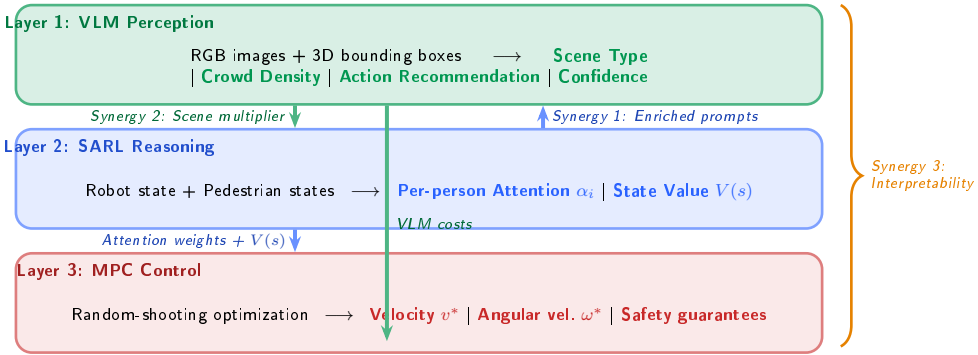
- Black-box decisions
- Sim-to-real transfer gap
- No safety guarantees

VISTA-MPC + SARL (Ours)

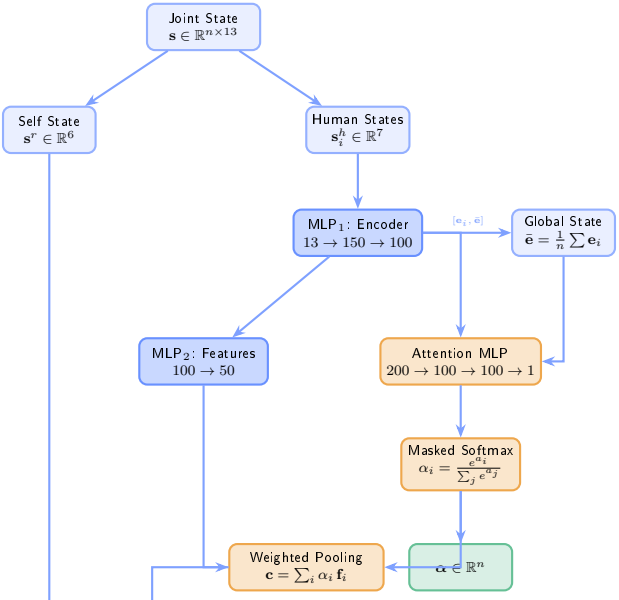
- **Attention-weighted** social reasoning
- **Scene-adaptive** via VLM
- **Predictive** MPC horizon (3 s)
- **Interpretable** explanations
- **Safety guarantees** via hard constraints
- **Graceful fallback** when data stale

Key Idea: Use RL for *what to attend to*, VLM for *scene understanding*, and MPC for *safe optimal control*.

Three-Layer Hierarchical Architecture



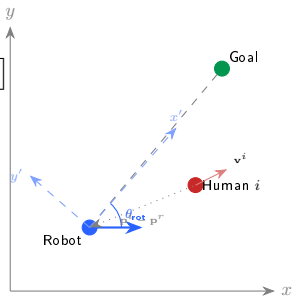
SARL Value Network Architecture



SARL: State Representation & Coordinate Transform

Joint State Vector (per human i):

$$\mathbf{s}_i = \big[\underbrace{d_g,\ v_{\text{pref}},\ \theta,\ r}_{\text{self state}},\ \underbrace{v_x^r,\ v_y^r}_{\text{velocity}},\ \underbrace{p_x^i,\ p_y^i,\ v_x^i,\ v_y^i,\ r^i,\ d_a,\ r_\Sigma}_{\text{human state}}\big]$$



Goal-Oriented Rotation:

$$\theta_{\text{rot}} = \text{atan2}(g_y - p_y^r,\ g_x - p_x^r)$$

$$\mathbf{R} = \begin{bmatrix} \cos\theta_{\text{rot}} & \sin\theta_{\text{rot}} \\ -\sin\theta_{\text{rot}} & \cos\theta_{\text{rot}} \end{bmatrix}$$

Applied to all velocity and position vectors:

$$\mathbf{v}_{\text{rot}} = \mathbf{R}\,\mathbf{v}_{\text{world}}, \quad \mathbf{p}_{\text{rot}}^i = \mathbf{R}\,(\mathbf{p}_{\text{world}}^i - \mathbf{p}^r)$$

Why Rotate?

Goal-centric frame makes the policy *translation and rotation invariant* — the same network weights generalize across positions.

Self-Attention Mechanism

Stage 1: Encode each human–robot pair:

$$\mathbf{e}_i = \mathrm{MLP}_1(\mathbf{s}_i) \in \mathbb{R}^{100}$$

Stage 2: Compute global context:

$$\bar{\mathbf{e}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{e}_i$$

Stage 3: Attention scores with global state:

$$a_i = \mathrm{MLP}_{\mathrm{attn}}\big([\mathbf{e}_i,\ \bar{\mathbf{e}}]\big) \in \mathbb{R}$$

Stage 4: Masked softmax normalization:

$$\alpha_i = \frac{\exp(a_i) \cdot \mathbb{1}[\mathbf{s}_i^h \neq \mathbf{0}]}{\sum_{j=1}^{n} \exp(a_j) \cdot \mathbb{1}[\mathbf{s}_j^h \neq \mathbf{0}]}$$

Stage 5: Weighted feature aggregation:

Interpretation

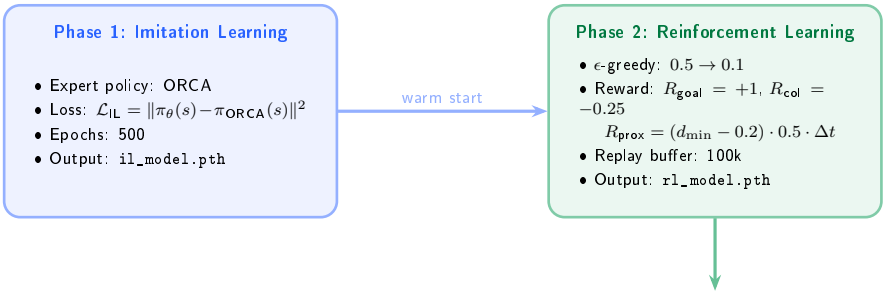
- α_i captures *behavioral importance* of pedestrian i
- Not just proximity — considers velocity, heading, relative position
- Zero-masking handles variable crowd sizes
- Global state $\bar{\mathbf{e}}$ provides crowd context to each attention score

Value Function

$$V(s) = \mathrm{MLP}_3\big([\mathbf{s}^r,\ \mathbf{c}]\big) \in \mathbb{R}$$

- Encodes long-term navigation quality

Two-Phase Training Pipeline



Simulation: Custom Gym environment with RVO2 humans.
Circle-crossing scenario, configurable crowd size (5–20 humans).
Discount factor $\gamma = 0.99$, batch size 64, target network updated every 500 steps.

Model Predictive Control: Problem Formulation

Unicycle Dynamics

$$
\begin{aligned}
x_{k+1} &= x_k + v_k \cos(\theta_k) \cdot \Delta t \\
y_{k+1} &= y_k + v_k \sin(\theta_k) \cdot \Delta t \\
\theta_{k+1} &= \theta_k + \omega_k \cdot \Delta t
\end{aligned}
$$

MPC Optimization

$$
(v^*, \omega^*) = \underset{v_{0:N-1},\, \omega_{0:N-1}}{\arg\min} \sum_{k=0}^{N-1} \underbrace{J_k^{\text{stage}}}_{\text{stage cost}} + \underbrace{J_N^{\text{term}}}_{\text{terminal cost}}
$$

Subject to:

- $0 \le v_k \le v_{\max}$
- $|\omega_k| \le \omega_{\max}$
- $d_{\text{obs}}(\mathbf{x}_k) \ge d_{\text{hard}}$ (safety)
- Unicycle dynamics (above)

$N = 15$ steps, $\Delta t = 0.2$ s (horizon = 3.0 s), 100 random rollouts, 10 Hz control rate

MPC Cost Function: Full Breakdown

$$J_k^{\text{stage}} = \underbrace{J_k^{\text{goal}}}_{\text{Goal}} + \underbrace{J_k^{\text{social}}}_{\text{SARL Social}} + \underbrace{J_k^{\text{obs}}}_{\text{Obstacle}} + \underbrace{J_k^{\text{smooth}}}_{\text{Smoothness}} + \underbrace{J_k^{\text{vlm}}}_{\text{VLM}}$$

Goal Cost:

$$J_k^{\text{goal}} = w_g \left\| (x_k - g_x,\ y_k - g_y) \right\|^2$$

SARL Attention-Weighted Social Cost:

$$J_k^{\text{social}} = \sum_{i=1}^{n} \frac{w_s \cdot n \cdot \alpha_i \cdot m_{\text{attn}}}{d(\mathbf{x}_k, \mathbf{p}_i) + \epsilon}$$

where α_i = SARL attention, m_{attn} = scene multiplier

Smoothness Cost:

$$J_k^{\text{smooth}} = w_{\text{sm}} \left[(v_k - v_{k-1})^2 + (\omega_k - \omega_{k-1})^2 \right]$$

Obstacle Cost:

$$J_k^{\text{obs}} = \begin{cases} +\infty & d < d_{\text{hard}} \\ 1000 & d_{\text{hard}} \le d < d_{\min} \\ \dfrac{w_o}{d + 0.1} & d \ge d_{\min} \end{cases}$$

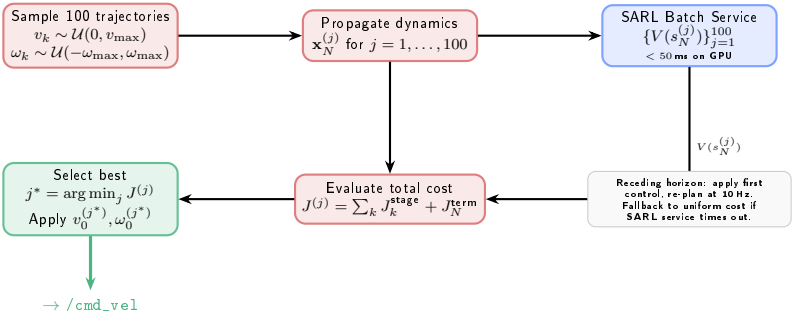
SARL Terminal Cost:

$$\boxed{J_N^{\text{term}} = -w_t \cdot m_{\text{term}} \cdot V(s_N)}$$

Negated: higher $V(s) \Rightarrow$ lower cost

$\Rightarrow$ MPC seeks high-value terminal states

Random-Shooting MPC with SARL Batch Evaluation



Vision-Language Model: Scene Understanding

VLM Input (Prompt Assembly):

- RGB camera image
- 3D bounding boxes of detected pedestrians
- Robot position and goal
- **SARL attention weights** per person (Synergy 1)

VLM Output (Structured):

- **Scene type**: doorway, corridor, crossing, queue, open_space
- **Crowd density**: empty, sparse, medium, dense
- **Recommended action**: go_ahead, slow_down, yield, wait
- **Confidence**: $[0, 1]$
- **Min. personal distance**: adaptive threshold

Trigger Conditions:

- New person detection

Example VLM Prompt (with SARL)

```
Robot at (2.1, 3.4), goal (8.0, 1.0)
Pedestrians (3):
1. ID=001, Pos=(3.5, 4.1)
   Vel=(0.2, -0.1) m/s
   SARL_ATTN=0.72
   RL_RISK=HIGH
2. ID=002, Pos=(5.2, 2.8)
   Vel=(0.0, 0.3) m/s
   SARL_ATTN=0.20
   RL_RISK=LOW
3. ID=003, Pos=(1.0, 5.5)
   Vel=(-0.1, 0.0) m/s
   SARL_ATTN=0.08
   RL_RISK=LOW
```

VLM Additional Cost Terms for MPC

The VLM translator converts semantic outputs into four MPC-compatible cost terms:

1. **Directional Cost** ($w_{\text{vlm-dir}}$):
   - Penalizes heading deviation from VLM-recommended direction
   - E.g., "pass on the left"
2. **Action Cost** ($w_{\text{vlm-act}}$):
   - Penalizes velocity when VLM says "stop and wait" or "yield"
3. **Scene Cost** ($w_{\text{vlm-scene}}$):
   - Scene-specific penalties (doorway narrowness, crossing density)
4. **Personal Space Cost** ($w_{\text{vlm-pers}}$):
   - Distance violation when closer than VLM-recommended minimum

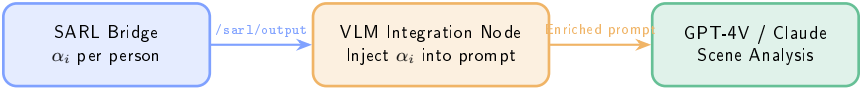
VLM Cost in MPC

$$
\begin{aligned}
J_k^{\text{vlm}} &= w_{\text{dir}} \cdot C_{\text{dir}}(\theta_k) \\
&+ w_{\text{act}} \cdot C_{\text{act}}(v_k) \\
&+ w_{\text{scene}} \cdot C_{\text{scene}}(\mathbf{x}_k) \\
&+ w_{\text{pers}} \cdot C_{\text{pers}}(d_{\min}^k)
\end{aligned}
$$

Enable/Disable VLM

VLM can be toggled via launch argument `enable_vlm`. When disabled, MPC uses only SARL attention + standard costs. System remains fully functional.

Synergy 1: SARL Attention Enriches VLM Prompts



Without SARL enrichment: "Person_001: distance=2.5m, motion=walking" $\longrightarrow$ VLM treats all equally by distance

With SARL enrichment: "Person_001: distance=2.5m, SARL_ATTENTION=0.72, RL_RISK=HIGH" $\longrightarrow$ VLM understands *behavioral* importance

Result: VLM recommendations are informed by what the RL policy has *learned* matters, not just geometric proximity.

Synergy 2: VLM Scene Context Conditions SARL Weights

VLM provides **scene_type** and **crowd_density**.
These modulate how much MPC trusts SARL via `getSceneMultiplier()`:

$$J_k^{\text{social}} = \sum_i \frac{w_s \cdot n \cdot \alpha_i \cdot \boldsymbol{m_{\text{attn}}}}{d_i + \epsilon}$$

$$J_N^{\text{term}} = -w_t \cdot \boldsymbol{m_{\text{term}}} \cdot V(s_N)$$

Intuition:

- **Doorway** (constrained): Amplify SARL → trust learned social behavior more
- **Open space**: Reduce SARL → simple distance-based cost suffices

Scene Multiplier Matrix

Scene	m_{attn}	m_{term}
Doorway (dense)	2.5	1.2
Corridor (dense)	2.0	1.0
Corridor (medium)	1.5	0.7
Crossing (medium)	1.5	0.7
Crossing (sparse)	1.0	0.5
Open space (sparse)	0.5	0.3

Synergy 3: Combined Interpretability

The `NavigationExplanation` message merges VLM and SARL outputs into a *single unified explanation*:

Contents:

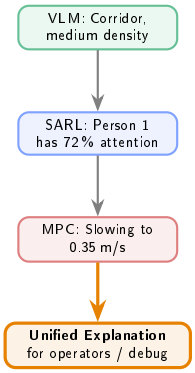
- VLM: scene type, crowd density, recommended action, confidence
- SARL: per-person attention weights, distances, motion labels
- Control: applied $v_{\max}$, w_{social}, scene multiplier
- **Human-readable narrative**

Published to: `/navigation/explanation`

Example Explanation

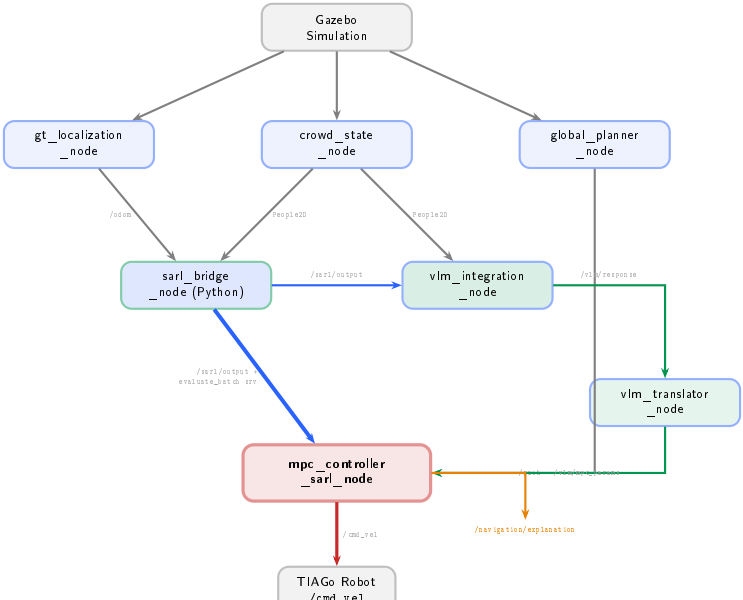
```
Scene: corridor (medium) | SARL active:
V(s)=0.35, attention_mult=1.5
Focus: person_001 (w=0.72, HIGH)
Action: slow_down | v_max=0.35 m/s
```

Why Interpretability Matters



Enables trust calibration, debugging, and human oversight of autonomous navigation decisions.

ROS 2 Node Architecture



SARL Bridge Node: Real-Time Inference

Dual Interface Design:

Interface 1: Topic Publisher (10 Hz)

- Subscribes: `/odom`, `/person_info`
- Publishes: `/sarl/output`
- Content: $\{\alpha_i\}$, $V(s)$, person IDs
- RViz markers: colored spheres
  (green→yellow→red $\propto$ attention)

Interface 2: Batch Service

- Service: `/sarl/evaluate_batch`
- Input: 100 terminal states from MPC
- Output: $\{V(s_N^{(j)})\}_{j=1}^{100}$
- Latency: < 50 ms with PyTorch

Implementation Details:

- Loads `rl_model.pth` from CrowdNav
- Replicates `ValueNetwork` architecture exactly
- Goal-oriented `rotate()` coordinate transform
- Unicycle → holonomic velocity conversion:

$$v_x = v\cos(\theta), \quad v_y = v\sin(\theta)$$

Staleness Handling:

- SARL data timestamped at publication
- MPC checks: if age > 0.5 s $\Rightarrow$ fallback to uniform social cost
- Ensures safety even if SARL node crashes

RViz Attention Markers:

- Sphere radius: $0.2 + \alpha_i \times 0.6$ m

Social Contract Helper: Rule-Based Safety Layer

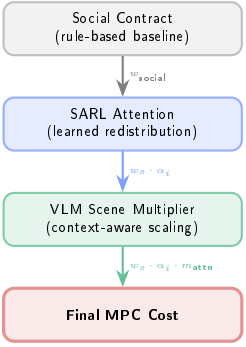
Purpose: Proximity-based heuristic that adjusts MPC parameters *before* optimization, providing an additional safety layer on top of SARL attention.

Adjustable Parameters:

- $v_{\max}$: reduced when people nearby
- w_{social}: increased when person in path
- $d_{\min}^{\text{person}}$: minimum distance threshold
- `person_in_front`: boolean flag for blocking detection

Integration with SARL:

- Social contract sets *baseline* w_{social}
- SARL attention *redistributes* this weight across pedestrians
- VLM scene multiplier *scales* the combined result



Key Configuration Parameters

MPC Parameters (`mpc_sarl.yaml`)

Parameter	Value	Unit
Δt	0.2	s
N (horizon)	15	steps
Rollouts	100	—
Control rate	10	Hz
w_{goal}	2.0	—
w_{social}	1.0	—
w_{obstacle}	3.0	—
w_{smooth}	0.1	—
$w_{\text{sarl-attn}}$	1.0	—
$w_{\text{sarl-term}}$	0.5	—
Staleness timeout	0.5	s
d_{hard}	0.2	m
$d_{\min}^{\text{obs}}$	0.3	m

SARL Bridge (`sarl_bridge.yaml`)

Parameter	Value
Model path	`rl_model.pth`
MLP_1 dims	[150, 100]
MLP_2 dims	[100, 50]
MLP_3 dims	[150,100,100,1]
Attn dims	[100, 100, 1]
Global state	True
Publish rate	10 Hz

Launch Command

```
ros2 launch social_mpc_nav
  mpc_sarl_vlm.launch.py
  enable_vlm:=true
  log_mpc_to_csv:=true
  goal_x:=10.0 goal_y:=5.0
```

VISTA-MPC + SARL: Complete Algorithm

Require: Trained SARL model, VLM API, MPC config
Initialize ROS 2 nodes, load SARL weights
while navigation active (at 10 Hz) **do**
  Get robot state (x, y, θ, v) from odometry
  Get pedestrian states $\{\mathbf{p}_i, \mathbf{v}_i\}$ from tracker
  // SARL Inference (10 Hz)
  Rotate states to goal-centric frame
  Compute $\{\alpha_i\}, V(s)$ via ValueNetwork
  Publish `/sarl/output`
  // VLM (event-triggered)
  if trigger condition met **then**
    Assemble prompt with SARL α_i *Syn. 1*
    Query VLM $\rightarrow$ scene type, density, action
  end if
  // MPC Optimization
  Get scene multiplier $m_{\text{attn}}, m_{\text{term}}$ *Syn. 2*
  Sample 100 random trajectories
  Call `evaluate_batch` $\rightarrow \{V(s_N^{(j)})\}$
  Evaluate $J^{(j)}$ with all cost terms
  $j^* \leftarrow \arg\min_j J^{(j)}$
  Apply $v_0^{(j^*)}, \omega_0^{(j^*)}$ to `/cmd_vel`
  Publish explanation *Syn. 3*

Key Properties:

- **Anytime:** Best trajectory available at any point
- **Graceful degradation:** VLM timeout $\rightarrow$ SARL-only; SARL stale $\rightarrow$ uniform cost
- **Safety:** Hard distance constraints always enforced
- **Interpretable:** Every decision is explainable

Computation Budget

SARL inference: $\sim$5 ms
Batch evaluation: $<$50 ms
MPC optimization: $\sim$10 ms
VLM query: 1–3 s (async)
Total: $<$100 ms / cycle

Key Contributions

1. **Hierarchical VLM–SARL–MPC integration**
   First framework combining learned social attention, vision-language scene understanding, and model predictive control in a principled three-layer hierarchy.

2. **Three synergy mechanisms**
   - **S1** SARL attention enriches VLM prompts for better scene analysis
   - **S2** VLM scene context adaptively scales SARL cost contributions
   - **S3** Combined interpretability for human-readable navigation explanations

3. **SARL terminal value for MPC**
   Learned value function $V(s_N)$ provides long-horizon social reasoning beyond MPC's finite horizon (3 s), evaluated in batch for 100 rollouts.

4. **Robust real-time system**
   ROS 2 implementation with graceful fallback (SARL stale $\rightarrow$ uniform cost), hard safety constraints, and full CSV logging for experiment analysis.

Advantages of the Hybrid Approach

Property	Pure RL	Pure MPC	VISTA-MPC+SARL
Social attention	✓	×	✓
Scene understanding	×	×	✓ (VLM)
Safety guarantees	×	✓	✓
Interpretability	×	∼	✓
Long-horizon reasoning	✓ ($V(s)$)	∼ (finite N)	✓ (both)
Scene adaptation	Fixed policy	Manual tuning	Automatic
Graceful fallback	×	✓	✓

VISTA-MPC + SARL combines the **social intelligence** of deep RL, the **semantic reasoning** of VLMs, and the **safety & optimality** of MPC.

Future Work

Short-Term

- Real robot deployment on TIAGo platform
- Systematic ablation study:
  VLM-only vs SARL-only vs combined
- Human-subject evaluation of social compliance
- Benchmark on standard SocNav datasets

Long-Term

- End-to-end fine-tuning of SARL with VLM feedback
- Pedestrian trajectory prediction integration
- Multi-robot coordination with shared attention
- CasADi-based gradient MPC replacing random shooting
- Smaller on-device VLMs for edge deployment

Thank You!

Questions?

Code: `github.com/huojiaxi/RL-Social-Nav`

Appendix: Message Definitions

SARLOutput.msg

```
time stamp
float32[] attention_weights
string[] person_names
float32 state_value
float32 robot_x
float32 robot_y
float32 robot_yaw
float32 inference_time_ms
bool is_valid
```

EvaluateSARLBatch.srv

```
# Request
float32[] robot_states
float32 goal_x
float32 goal_y
float32[] people_states
int32 num_rollouts
int32 num_people
--
# Response
float32[] values
float32 inference_time_ms
bool success
```

NavigationExplanation

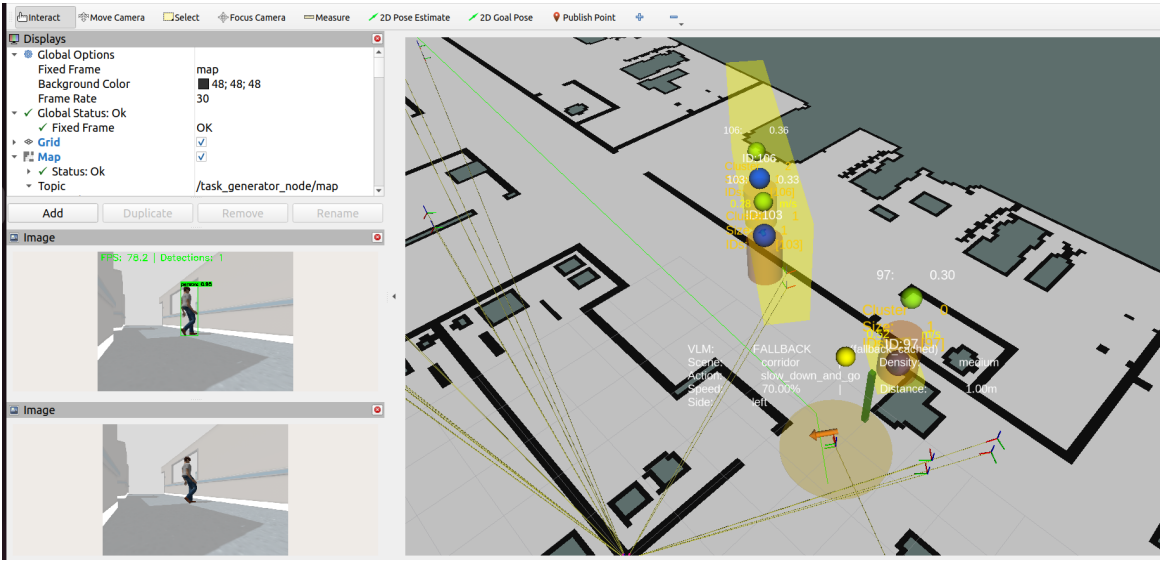
```
string scene_type
string crowd_density
string recommended_action
float32 vlm_confidence
string[] person_names
float32[] attention_weights
float32[] person_distances
string[] person_motions
float32 state_value
float32 applied_v_max
float32 applied_w_social
float32 sarl_scene_mult
string explanation_text
```

Appendix: SARL Network Dimensions

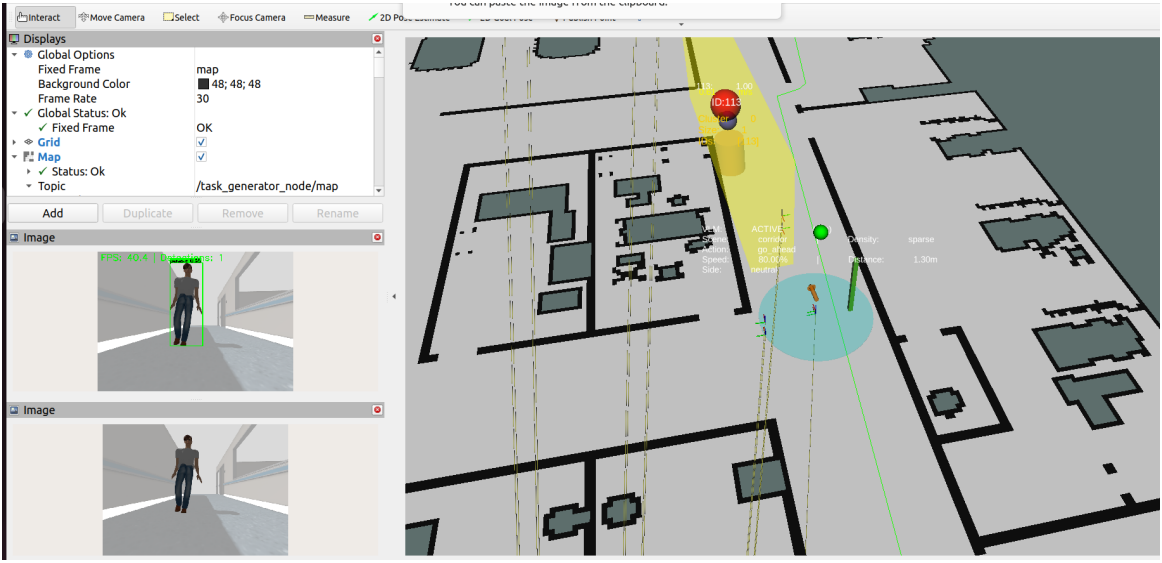
Component	Layer	Input	Output	Activation
MLP_1 (Encoder)	Linear + ReLU	13	150	ReLU
	Linear + ReLU	150	100	ReLU
Attention MLP	Linear + ReLU	200	100	ReLU
	Linear + ReLU	100	100	ReLU
	Linear	100	1	—
MLP_2 (Features)	Linear + ReLU	100	50	ReLU
MLP_3 (Value Head)	Linear + ReLU	56	150	ReLU
	Linear + ReLU	150	100	ReLU
	Linear + ReLU	100	100	ReLU
	Linear	100	1	—

Total parameters: $\sim$78k. Inference time: $\sim$5 ms (CPU), $<$1 ms (GPU).
Input dimension 200 for attention = 100 (MLP_1 output) + 100 (global state).
Input dimension 56 for MLP_3 = 6 (self state) + 50 (weighted features).

VISTA-MPC with SARL



VISTA-MPC with SARL



VISTA-MPC with SARL