Spatially-Enhanced Recurrent Memory and Perception Pretraining for Long-Range Mapless Navigation from Egocentric Vision

Anonymous Authors

***Abstract*—Perception pretraining can supply reusable geometric features for robot navigation, but long-range mapless navigation still requires memory and control under partial observability. We study this interface through reinforcement learning (RL) for mapless goal navigation from stereo depth: a pretrained perception encoder supplies egocentric features, while state-conditioned attention and Spatially-Enhanced Recurrent Units (SRUs) form the local planning memory and action interface. The original system combines perception pretraining on synthetic depth, structured depth-noise modeling, attention, SRU memory, and PPO-based RL. In unseen simulated mazes, pillars, stairs, and pit environments, replacing LSTM memory with SRU memory increases overall success from 63.5% to 78.9% under the same policy stack. The same pretrained policy transfers zero-shot to indoor, terrace, and forest environments on a Unitree B2W robot, including dead-end recovery and traversals beyond the training start-to-goal distance. We also summarize a DeFM-style pretrained depth frontend that preserves the attention-SRU planner while replacing the task-specific VAE encoder.**

I. Introduction

Perception pretraining is an increasingly practical way to reuse large-scale geometric data before training a closed-loop robot policy. For long-range mapless navigation from egocentric stereo depth, however, perception alone does not solve partial observability: the policy must retain and register spatial evidence after it leaves the camera view. We therefore study a compact reinforcement learning (RL) stack in which pretrained depth features feed state-conditioned attention and Spatially-Enhanced Recurrent Unit (SRU) memory.

We frame the SRU navigation system around two complementary roles: perception pretraining supplies robust egocentric features, while SRU memory supports local planning over changing viewpoints. The original system uses task-specific VAE/noise pretraining to cover the sim-to-real depth gap; a DeFM-style variant asks whether broader self-supervised depth features can improve the same planner without changing its attention or memory modules.

In summary, our main contributions are as follows:

- **Spatial Mapping with SRU Memory:** We show that standard recurrent units struggle to register observations from changing egocentric viewpoints, while SRUs better learn implicit spatial transformations for mapless navigation.
- **Attention-SRU Policy for Long-Range Navigation:** We combine state-conditioned attention and SRU memory in a compact RL policy that operates directly on front-facing stereo depth features.
- **Pretrained Perception for Transfer:** We analyze both task-specific VAE/noise pretraining and a DeFM-style depth frontend [17], showing how sensor-specific noise modeling supports sim-to-real transfer and how broader depth pretraining is associated with higher success in out-of-distribution simulation without changing the downstream planner.

II. Problem Setting

We consider goal-conditioned navigation as a partially observable decision problem. At each time step, the robot receives an egocentric visual observation I_t, proprioceptive state q_t, previous action a_{t-1}, and relative goal p_t expressed in the robot frame. The policy outputs a high-level velocity command a_t for the locomotion controller:

$$a_t = \pi_\theta(I_t, q_t, a_{t-1}, p_t, h_{t-1}), \tag{1}$$

where h_{t-1} denotes the recurrent memory state. The policy has no prior map of the test environment and does not build an explicit occupancy map during execution. A state-estimation and localization module provides velocities, projected gravity, and the relative goal; exteroception comes from the front-facing stereo camera.

This setting exposes a planning-memory tension. A narrow camera observes only the next few meters, while a successful route may involve entering a corridor, discovering a blockage, backtracking, and selecting an alternative branch. In the original experiments, the depth camera has a maximum range of 10 m, while simulated evaluation includes routes up to 120 m when the episode horizon is extended. Solving such tasks from the current image alone is unlikely; the policy needs a latent belief that carries spatially organized information across ego-motion.

III. Method

The policy in Eq. (1) and Fig. 1 is composed of a visual encoder, a state-conditioned attention module, an SRU memory module, and an MLP action head. The computational path is summarized in Eq. (2):

$$\begin{aligned} F_t &= \phi_\theta(I_t), \\ z_t &= \psi_\theta(F_t, q_t, a_{t-1}, p_t), \\ h_t &= \mathrm{SRU}_\theta(z_t, h_{t-1}), \\ a_t &= \rho_\theta(h_t). \end{aligned} \tag{2}$$

Although the reported implementation uses depth images, the recurrent and attention components operate on encoded

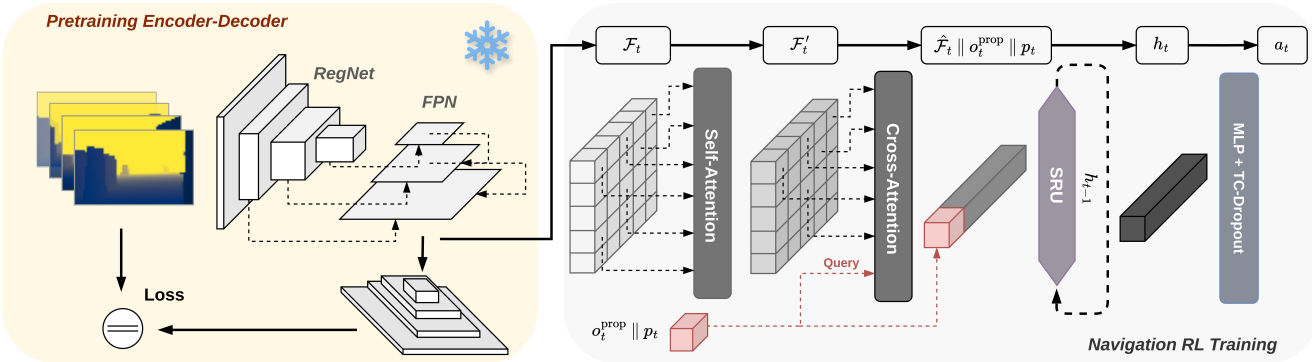


Fig. 1. Attention-based recurrent navigation policy. The relative goal and proprioceptive state form a query over encoded egocentric visual features through cross-attention. SRU memory fuses the resulting compact representation and the action head decodes it into velocity commands.

egocentric visual features. This gives a direct perception-pretraining interface for RL: the visual frontend can be a task-specific VAE/noise encoder or a DeFM-style pretrained depth encoder, while the attention and SRU modules keep the same planning role during policy optimization.

A. Perception Frontend

The visual frontend is the encoder ϕ_θ in Eq. (2), which converts each egocentric depth image into a spatial feature map for attention and recurrent memory. In the original SRU experiments, this frontend is a RegNet backbone [19] with a Feature Pyramid Network [14]. Before RL optimization, it is pretrained as part of a variational autoencoder on TartanAir [21], with the structured depth-noise model described in Sec. IV. This stage broadens the set of geometric layouts seen by the encoder beyond the images collected during policy rollouts.

The related DeFM navigation evaluation [17] keeps the downstream attention-SRU planner unchanged and replaces the VAE frontend. It uses a compact RegNet/FPN depth encoder whose representation is pretrained through DINO-style self-distillation [3, 16] on diverse depth data. Its dense features feed the same cross-attention, SRU memory, and action head. This separates the representation question from the planner: the VAE/noise baseline tests task-specific sim-to-real preparation, while the DeFM-style frontend tests whether broader depth pretraining improves the same mapless planner in out-of-distribution scenes.

B. State-Conditioned Attention

The attention module has two stages. First, self-attention enriches the spatial feature map $F_t \in \mathbb{R}^{C\times H\times W}$ by mixing context across visual tokens. Let F'_t denote this self-attended feature map. Second, cross-attention uses the robot state and goal as queries. The query is derived from linear velocity v_t, angular velocity ω_t, projected gravity n_t, previous action a_{t-1}, and relative goal p_t:

$$\begin{aligned} Q_t &= W_q[v_t, \omega_t, n_t, a_{t-1}, p_t], \\ K_t, V_t &= W_k F'_t,\ W_v F'_t, \\ z_t &= \mathrm{softmax}\Big(Q_t K_t^\top / \sqrt{d}\Big)\, V_t. \end{aligned} \tag{3}$$

The output z_t is therefore a compact, state-conditioned visual feature rather than a generic image descriptor.

This design is useful for recurrent planning because memory capacity is finite. Passing dense visual features at every step can burden the recurrent state with irrelevant details. Eq. (3) gives the policy a learned way to emphasize free space, obstacles, and terrain structure according to the current motion state and goal. In the SRU ablation, removing attention drops success to 50.5%, using a goal-guided transformer attention variant reaches 68.4%, and the two-stage spatial attention reaches 78.9% with SRU memory.

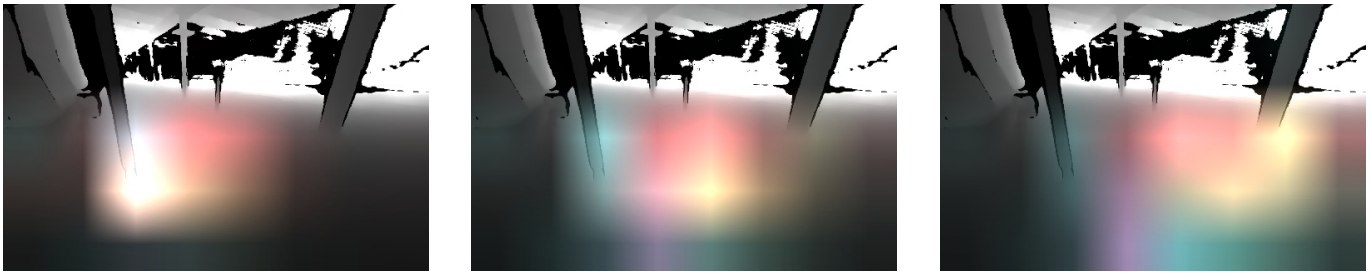
Fig. 2 illustrates this behavior on a real depth input. The attention heads shift when the robot is manually commanded to turn left, move straight, or turn right. These attention maps are not supervised labels; they emerge from end-to-end navigation training and provide a useful diagnostic for the attention stack. The recurrent unit receives a filtered visual summary that changes with the action context, which can better support local planning decisions such as passing between pillars or avoiding an obstacle during a turn.

C. Spatial Recurrent Memory

Standard recurrent units such as GRU and LSTM [5, 11] can store temporal information, but they are not explicitly shaped to register observations from changing egocentric frames. Even long-sequence models such as S4 and Mamba [8, 6] do not directly solve this spatial-registration issue in the controlled memorization task from the original paper. SRUs modify the recurrent candidate update by introducing a learned spatial modulation term. For the SRU-LSTM variant used as the final model, the update is:

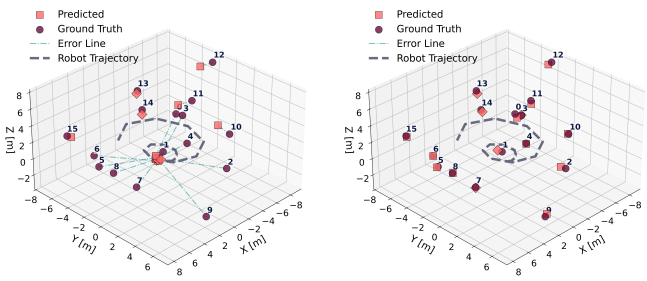
$$\begin{aligned} s_t &= W_{xs} x_t + b_s, \\ g_t &= \tanh(s_t \odot (W_{xg} x_t + W_{hg} h_{t-1} + b_g)), \\ r_t &= i_t \odot \Big(1 - (1 - f_t)^2\Big) + (1 - i_t) \odot f_t^2, \\ c_t &= r_t \odot c_{t-1} + (1 - r_t) \odot g_t. \end{aligned} \tag{4}$$

In Eq. (4), x_t denotes the recurrent input derived from the attention output z_t, s_t modulates the candidate memory update, and r_t is a refined gate adapted from improved recurrent



turning left moving straight turning right

Fig. 2. State-dependent cross-attention over the same real depth scene. When the commanded motion changes, attention shifts toward the left, central, or right spatial region, indicating that the visual memory input is conditioned by robot state rather than pooled uniformly.



LSTM SRU

Fig. 3. Spatial memory comparison in a controlled ego-motion task. Both models can recall temporal labels, but SRU memory better aligns earlier landmark observations into the final robot frame.

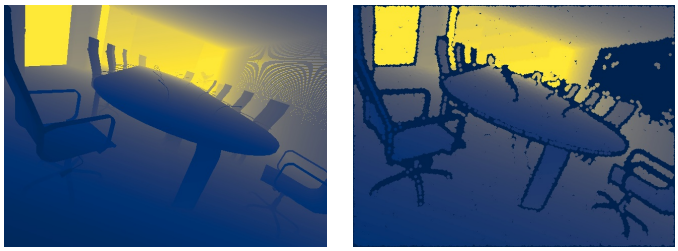
gating [7]. The modification is lightweight: it keeps the recurrent structure and adds a path that helps the candidate state account for spatial alignment.

The effect is visible in a controlled spatial-temporal memorization task, shown in Fig. 3. A robot follows a spiral path, observes landmarks from varying viewpoints, and must report their coordinates in the final robot frame. The baseline recurrent models recall category labels accurately, but their spatial errors grow for earlier observations. SRUs maintain lower spatial error, indicating better learned registration across ego-motion. This controlled result motivates the use of SRU memory for navigation, where obstacles and branch decisions may leave the camera view before they become relevant.

IV. Perception Pretraining and Noise Modeling

From a perception-pretraining perspective, the visual encoder is a natural place to use additional data: real robot trajectories are expensive, but geometry-rich synthetic and self-supervised depth data can be used before RL policy optimization. Depth images collected in simulation are also cleaner than stereo depth observed on a robot, especially near object boundaries, thin structures, and invalid matches. To narrow this gap, the original SRU system starts from a task-specific variational autoencoder (VAE) pretrained on large-scale synthetic depth, together with a structured depth-noise model used during both perception pretraining and RL. This VAE frontend is the main baseline below; the DeFM-style frontend in Sec. III-A uses the same downstream planner and replaces the encoder representation.

For stereo depth, the model works in disparity space. Given depth z, focal length f, and stereo baseline b, disparity is $u = fb/z$. Three corruptions are applied in batch: edge noise drops



clean synthetic depth augmented depth

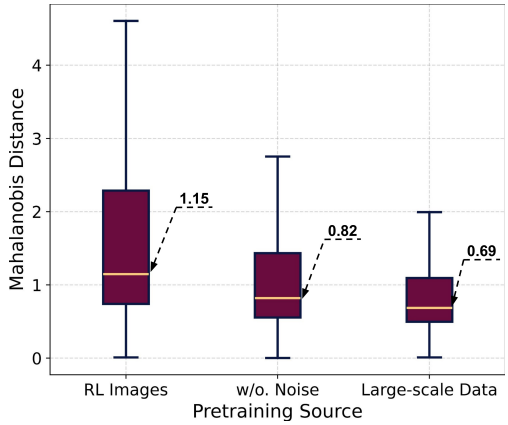


Fig. 4. Perception pretraining with structured depth-noise augmentation. Large-scale synthetic pretraining reduces the feature-distribution gap to real depth inputs, and adding the proposed noise model further lowers the median Mahalanobis distance.

or perturbs pixels near sharp disparity changes; filling noise models invalid or interpolated stereo matches; and rounding noise quantizes disparity before converting back to depth. The corruptions target common RGB-D and stereo artifacts [9, 1, 2]. Fig. 4 illustrates the clean and corrupted depth images.

The perception-pretraining analysis compares latent features from real ZEDX depth images with latent features from simulated RL rollout images. Table I summarizes the median Mahalanobis distance reported in the original study. With an encoder trained only on RL images, the median distance is 1.15. Large-scale synthetic perception pretraining reduces it to 0.82. Adding the structured depth-noise model reduces it further to 0.69. Qualitatively, the encoder with large-scale perception pretraining and noise augmentation reconstructs real ZEDX depth inputs with fewer artifacts than an encoder trained only on RL images. These results do not prove full domain invariance, but they indicate that perception pretraining and

TABLE I
FEATURE GAP AFTER PERCEPTION PRETRAINING. LOWER MEDIAN MAHALANOBIS DISTANCE IS BETTER.

Encoder source	Median distance
RL images only	1.15
Synthetic depth, no noise model	0.82
Synthetic depth with structured noise	**0.69**

TABLE II
NAVIGATION SUCCESS RATE FOR RECURRENT UNITS UNDER THE SAME ENCODER, ATTENTION, REGULARIZATION, AND RL SETUP.

Model	Maze	Pillar	Stair	Pit	Overall
GRU	68.1	73.6	35.7	66.7	61.0
LSTM	70.3	78.2	33.1	72.7	63.5
SRU-GRU	73.1	78.8	74.1	74.8	75.2
SRU-LSTM	75.9	76.7	79.3	74.1	76.5
SRU-Ours	**76.0**	**81.0**	**82.8**	**75.6**	**78.9**

sensor-specific noise modeling better align simulated and real perception features before downstream RL.

V. RL TRAINING AND SIMULATION EVALUATION

The policy is trained end-to-end in IsaacLab [15] with PPO [20] and an asymmetric actor-critic setup [18]. The actor observes noisy egocentric depth, proprioception, previous action, and relative goal. The critic uses cleaner observations during training. The high-level navigation policy runs at 5 Hz and outputs linear and angular velocity commands to a low-level locomotion controller running at 50 Hz [13].

The reward combines sparse task progress with penalties for collisions, unstable body tilt, and overly aggressive actions. Two regularizers stabilize recurrent RL: deep mutual learning, where two policies exchange policy-distribution information during PPO optimization [23], and temporally consistent dropout, where the dropout mask is held fixed across recurrent time steps [10]. The original ablation reports that DML raises SRU success from 65.7% to 78.9%, while temporally consistent dropout raises the final SRU policy from 77.2% to 78.9%. These numbers suggest that RL optimization support is important for realizing the spatial-memory advantage, especially because spatial registration can learn more slowly than simple temporal recall.

The simulated evaluation set includes unseen mazes, randomly generated pillars, stairs, and negative obstacles such as pits. These environments test different forms of memory: remembering a dead-end branch, navigating around clutter, handling 3D terrain, and avoiding hazards that may disappear from the camera view during turning. Policies are evaluated on 4800 episodes across 120 randomly generated environments distinct from the training set.

Table II isolates the memory unit. Standard GRU and LSTM reach 61.0% and 63.5% overall success. Adding the SRU modification improves success to 75.2% with GRU and 76.5% with LSTM; the refined SRU reaches 78.9%. The largest gap appears in stair environments, where the standard recurrent policies struggle with occlusion and 3D structure.

Table III compares the complete policy against two planning baselines. GTRL uses stacked historical observations with goal-guided transformer attention; EMHP uses explicit mapping and historical path memory. The SRU policy outperforms both in the reported environments. Replacing GTRL's stacked history with SRU memory raises success from 38.2% to 66.3%, and the full policy reaches 78.3% in that baseline comparison. The attention ablation in the lower block shows the same trend: without attention, success is 50.5%, while spatial attention with SRU reaches 78.9%.

TABLE III
BASELINE AND ATTENTION COMPARISONS FROM THE ORIGINAL SRU STUDY. SR DENOTES SUCCESS RATE.

Baseline comparison	SR (%)
GTRL, stacked observations [12]	38.2
EMHP, explicit mapping/path [13]	60.4
GTRL* with SRU memory	66.3
Full policy with SRU memory	**78.3**
Attention comparison	**SR (%)**
Without attention [22]	50.5
GoT attention with SRU	68.4
Spatial attention with SRU	**78.9**

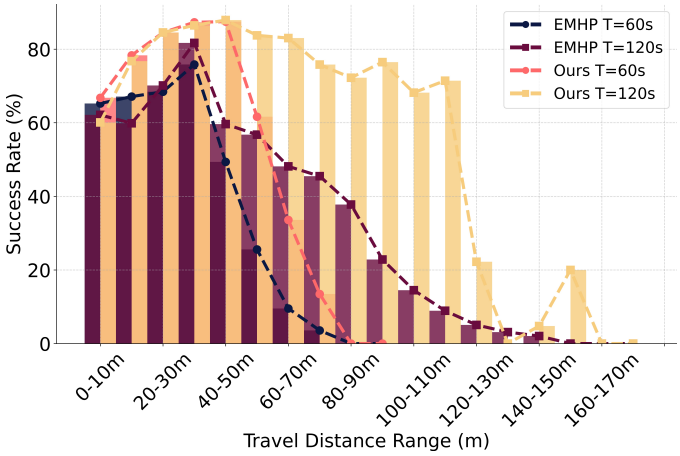


Fig. 5. Success rate by travel distance. The SRU policy maintains higher success for long routes, including over 70% success up to 120 m when the episode horizon is extended, although training start-to-goal distances are at most 30 m.

Fig. 5 focuses on long-range behavior. With the same 60 s episode horizon used in training, EMHP drops sharply after 40 m, while the SRU policy stays above 80% up to 50 m. With a 120 s horizon, SRU remains above 70% up to 120 m. This is evidence that the learned recurrent memory can generalize beyond the local perception radius and beyond the training start-to-goal distance, while still being a local mapless navigation method rather than a global navigation system.

Table IV summarizes the DeFM navigation evaluation: replacing the VAE encoder with a DeFM-style pretrained depth encoder preserves the downstream planner while achieving higher success across industrial, cable, and downtown scenes. This is consistent with the modular view that broader depth perception pretraining can improve out-of-distribution navigation without changing the attention-SRU RL policy.

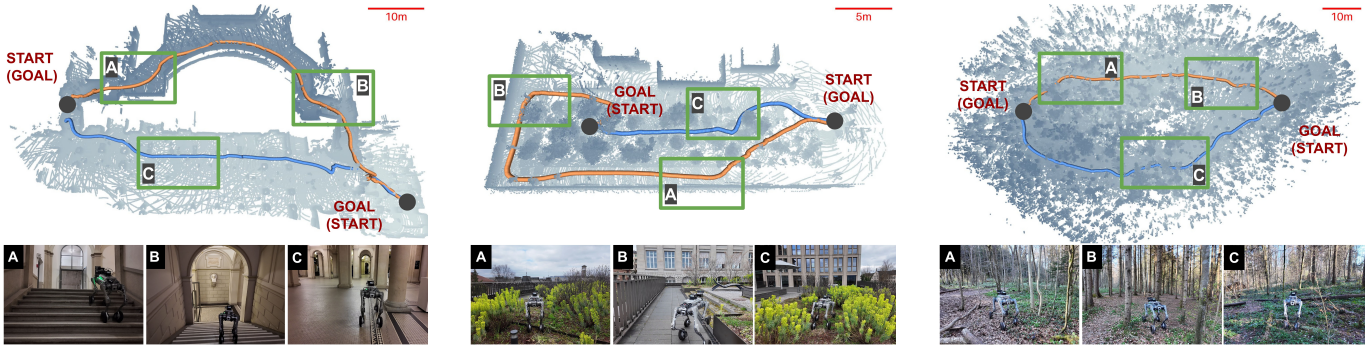


Fig. 6. Zero-shot real-world navigation examples. The same pretrained SRU policy navigates a university main hall, an outdoor terrace, and a forest environment without real-world fine-tuning.

TABLE IV
CONTROLLED PERCEPTION-FRONTEND SWAP FROM THE DEFM NAVIGATION EVALUATION [17]. THE DOWNSTREAM ATTENTION-SRU RL PLANNER IS FIXED; VALUES ARE SUCCESS RATES (%) IN OUT-OF-DISTRIBUTION SIMULATION SCENES.

Frontend	Industrial	Cable	Downtown	Avg.
VAE/noise	89.67	83.00	73.67	82.11
DeFM-style	**92.67**	**84.33**	**79.00**	**85.33**

VI. REAL-WORLD TRANSFER

Real-world experiments deploy the same pretrained encoder and RL navigation policy on a Unitree B2W legged-wheel robot with a ZEDX front-facing stereo depth sensor and an NVIDIA Jetson AGX Orin. The system is deployed without real-world fine-tuning. The robot receives no prior environment map; it uses stereo depth as exteroception and a LiDAR-inertial state-estimation module for velocities, projected gravity, and relative goal [4].

The office dead-end experiment directly tests recurrent spatial memory. Several passageways are blocked so that the robot must enter dead ends, backtrack, and continue searching. The SRU policy reaches the goal and adapts when a blocked area is later opened. The LSTM baseline repeatedly loops between dead-end areas and fails to reach the goal, matching the simulated finding that standard recurrent memory can preserve temporal context without reliably aligning spatial observations.

The broader real-world tests cover a university main hall, outdoor terrace, and forest, as shown in Fig. 6. These environments include walls, stairs, vegetation, trees, bushes, and uneven terrain. The same pretrained RL policy is used across these settings. In the forest setting, reported goal distances exceed 70 m and the traversed path exceeds 100 m, despite the maximum start-to-goal distance of 30 m used during RL training. The point-cloud trajectories in the original paper are generated for visualization and can contain drift because the navigation policy itself does not maintain a dedicated mapping or loop-closure module.

VII. DISCUSSION

The results suggest three practical points for planning with pretrained perception and RL. First, perception pretraining is useful even when the downstream planner is compact and learned end-to-end: synthetic perception pretraining with structured depth noise better aligns real depth features, and the DeFM simulation frontend swap achieves higher out-of-distribution success without changing the attention-SRU policy architecture. Second, attention provides a lightweight interface between representation and RL-based planning by reducing the visual information carried through recurrent state. Third, spatial structure inside the recurrent update helps use observations after they leave the field of view.

The scope should be stated carefully. The experiments support local long-range mapless navigation, where "long-range" means planning beyond the approximately 10 m local perception radius and 30 m training start-to-goal distance. The real-world evidence comes from the original VAE/noise-pretrained policy; the DeFM result is a simulation frontend study motivating follow-up real-world testing. Kilometer-scale navigation would need additional mechanisms such as global mapping, retrieval, or hierarchical memory. Other egocentric modalities, higher-resolution DeFM-style encoders, or end-to-end adapter tuning require separate experiments.

VIII. CONCLUSION

We present SRU-based mapless navigation as a case study in how pretrained depth perception can support robot planning from goal specifications and egocentric sensory streams. The original results show that long-range navigation can be improved by combining perception pretraining, PPO-based RL, state-conditioned attention, and spatial recurrent memory. In the depth-camera setting studied here, synthetic perception pretraining with structured noise better aligns simulated and real visual features; cross-attention selects navigation-relevant cues; and SRU memory improves alignment across ego-motion. Together, these components improve simulated success over standard recurrent units and support zero-shot transfer to indoor, terrace, and forest environments. The DeFM-style frontend points toward broader depth pretraining as a useful upgrade for more diverse scenes, while longer-duration navigation will likely require pairing SRU-style local memory with hierarchical or global memory modules.

References

[1] Jonathan T. Barron and Jitendra Malik. Intrinsic scene properties from a single RGB-D image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 17–24, 2013.

[2] Jeannette Bohg, Javier Romero, Alexander Herzog, and Stefan Schaal. Robot arm pose estimation through pixel-wise part classification. In *IEEE International Conference on Robotics and Automation*, pages 3143–3150, 2014.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[4] Kenny Chen, Ryan Nemiroff, and Brett T Lopez. Direct LiDAR-inertial odometry: Lightweight LIO with continuous-time motion correction. *IEEE International Conference on Robotics and Automation*, pages 3983–3989, 2023.

[5] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014.

[6] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[7] Albert Gu, Caglar Gulcehre, Thomas Paine, Matt Hoffman, and Razvan Pascanu. Improving the gating mechanism of recurrent neural networks. In *International Conference on Machine Learning*, pages 3800–3809, 2020.

[8] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

[9] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. A benchmark for RGB-D visual odometry, 3d reconstruction and SLAM. In *IEEE International Conference on Robotics and Automation*, pages 1524–1531, 2014.

[10] Matthew Hausknecht and Nolan Wagener. Consistent dropout for policy gradient reinforcement learning. *arXiv preprint arXiv:2202.11818*, 2022.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[12] Wenhui Huang, Yanxin Zhou, Xiangkun He, and Chen Lv. Goal-guided transformer-enabled reinforcement learning for efficient autonomous navigation. *IEEE Transactions on Intelligent Transportation Systems*, 25(2):1832–1845, 2023.

[13] Joonho Lee, Marko Bjelonic, Alexander Reske, Lorenz Wellhausen, Takahiro Miki, and Marco Hutter. Learning robust autonomous navigation and locomotion for wheeled-legged robots. *Science Robotics*, 9(89):eadi9641, 2024.

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[15] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8 (6):3740–3747, 2023.

[16] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[17] Manthan Patel, Jonas Frey, Mayank Mittal, Fan Yang, Alexander Hansson, Amir Bar, Cesar Cadena, and Marco Hutter. DeFM: Learning foundation representations from depth for robotics. *arXiv preprint arXiv:2601.18923*, 2026. doi: 10.48550/arXiv.2601.18923. URL https://arxiv.org/abs/2601.18923.

[18] Lerrel Pinto, James Davidson, and Abhinav Gupta. Asymmetric actor critic for image-based robot learning. In *Robotics: Science and Systems*, 2017.

[19] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020.

[20] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017.

[21] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4909–4916, 2020.

[22] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. DD-PPO: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *International Conference on Learning Representations*, 2020.

[23] Zhengpeng Xie, Jiahang Cao, Changwei Wang, Fan Yang, Marco Hutter, Qiang Zhang, Jianxiong Zhang, and Renjing Xu. Representation convergence: Mutual distillation is secretly a form of regularization. *arXiv preprint arXiv:2501.02481*, 2025.