arXiv:2610.29845v1 [cs.RO] 21 Oct 2026

RangeNav: Task-Aligned LiDAR Range-Image Pretraining for Robot Navigation

Samuel Foo Enze[1], Manthan Patel[1], Fan Yang[1], Linus Kramer[1], Pascal Roth[1], Marco Hutter[1], Cesar Cadena[1]

***Abstract*— Autonomous mobile robots operating in dense crowds and cluttered sites require 360° perception to react to surrounding hazards. While panoramic LiDAR offers full-surround coverage, learning representations within reinforcement learning (RL) is sample-inefficient under sparse rewards, and existing 3D pretrained point-cloud encoders are too memory-heavy for massively parallel RL simulation. We present RangeNav, a compact (183k parameters) LiDAR range-image encoder pretrained entirely on synthetic data. RangeNav uses anisotropic convolutions and circular padding to handle the extreme aspect ratio and azimuthal seam of range images. We pretrain the network using a navigation-aligned objective combining range reconstruction, semantic segmentation, and bird's-eye-view (BEV) prediction, then freeze it for policy learning. In simulation, adding semantic and BEV supervision increases success rates over a reconstruction-only baseline by an average of 32.03 and 23.73 percentage points (pp) in procedurally generated training and out-of-distribution realistic test environments, respectively. Zero-shot real-world deployments on a Unitree B2-W quadruped demonstrate robust navigation through dense crowds, low-light settings, and construction zones where limited-field-of-view depth policies struggle. Code and models will be open-sourced.**

I. Introduction

Autonomous navigation enables mobile robots to operate in challenging real-world domains, from negotiating dense pedestrian crowds in last-mile delivery to inspecting unstructured industrial and construction sites. A major driver of recent progress in learning-based navigation has been the use of pretrained visual representations. Rather than requiring the policy to learn perception from scratch, a frozen or lightly fine-tuned encoder supplies rich geometric and semantic features. This approach has proven effective for RGB-based navigation [1], [2] and depth image-based policies [3], [4].

However, both RGB and depth cameras inherit limitations. RGB features degrade under poor lighting and appearance shifts, and depth cameras return noisy or missing values beyond a few meters, in sunlight, and on reflective or textureless surfaces. Crucially, both are limited to a narrow field of view (FoV). Obstacles approaching from the side, pedestrians overtaking from behind, and routes outside the current view are simply invisible until it is too late to react. LiDAR addresses these weaknesses with 360° coverage, metric range up to tens of meters, and invariance to illumination, making it the natural sensor for the crowded, outdoor, and industrial settings that motivate this work. Yet pretrained representations for LiDAR-based navigation remain underexplored. Although point-cloud foundation models (FMs) such as Sonata [5] and Utonia [6] learn strong geometric and semantic features, their computational cost limits their use in massively parallel reinforcement learning (RL), and their effectiveness as pretraining sources for compact navigation encoders remains an open question. A navigation-ready LiDAR encoder should therefore be *lightweight* enough for large-scale RL, *pretrained* so the policy does not have to learn perception from sparse rewards, and *generalizable* across scenes and to the real world.

To this end, we present a compact LiDAR range-image

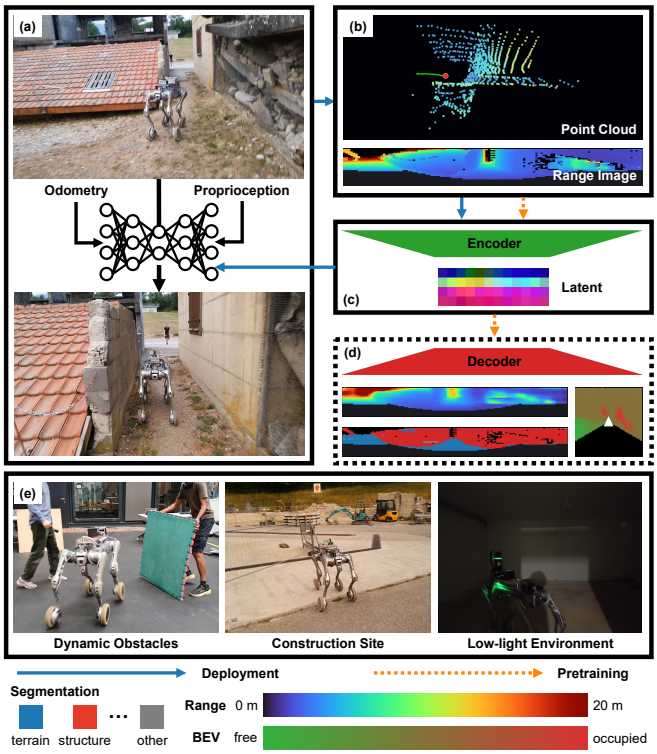


Fig. 1: System overview. (a) A mobile robot observes the environment with LiDAR, odometry, and proprioception. (b) The LiDAR scan is projected into a panoramic range image. (c) Our pretrained encoder compresses the range image into a spatial latent for the navigation policy. (d) During pretraining, task-aligned decoders supervise range reconstruction, semantic segmentation, and bird's-eye-view (BEV) prediction. (e) The frozen encoder transfers to real-world navigation in dynamic environments, construction sites, and low-light environments without real-world training.

[1] Robotic Systems Lab (RSL), ETH Zurich, Switzerland

encoder pretrained specifically for navigation. Exploiting the fixed scan pattern of spinning LiDARs, we spherically project each scan onto a 2D *range image* [7]. This representation is cheap to compute and batches predictably across parallel environments, but its geometry differs from a perspective image: it has many more azimuthal columns than elevation rows, and the image wraps around at the $0^\circ/360^\circ$ seam. We account for both with a fully convolutional encoder that uses anisotropic kernels and circular convolutions along the azimuth. We pretrain the encoder on range images rendered from TartanGround [8] with three navigation-aligned objectives: range reconstruction, semantic segmentation, and BEV prediction, which preserve object semantics, obstacle boundaries, and top-down traversability information. The frozen latent representation then feeds an RL navigation policy using spatially-enhanced recurrent units (SRUs) [3] (Fig. 1).

In simulation, our pretraining recipe improves navigation success rate (SR) over a reconstruction-only baseline by an average of 32.03 and 23.73 percentage points (pp) in procedurally generated training and out-of-distribution test environments, respectively. Compared with the depth-based policy of [3], our LiDAR policy achieves comparable average SR, shorter navigation times, and higher success weighted by path length (SPL), and surpasses it in crowded and densely cluttered environments, highlighting the benefits of a wider FoV. Deployed zero-shot on a Unitree B2-W robot, the policy navigates obstacle-rich environments, unlit spaces, dense crowds, and cluttered construction sites, settings in which limited-FoV depth policies struggle.

In summary, our contributions are:

1) A navigation-aligned pretraining recipe combining range reconstruction, semantic segmentation, and BEV prediction that significantly improves downstream navigation.
2) A lightweight (183k parameters) LiDAR range-image encoder with anisotropic and circular convolutions, suitable for large-scale RL.
3) Extensive simulation and zero-shot real-world evaluation demonstrating the advantages of LiDAR-based navigation policies over limited-FoV depth-based policies.

II. Related Work

a) Pretrained perception for navigation: Recent methods decouple perception learning from downstream control, reducing the perceptual burden on policy learning caused by sparse rewards [3], [9], [10]. Several works build upon RGB vision foundation models [11]–[13], keeping encoders frozen or lightly fine-tuned for point-goal navigation [1], [2], image-goal navigation [9], [14], and visual affordance prediction [15]–[17]. Concurrently, representations learned directly from depth images [3], [4], [18] have demonstrated robust zero-shot sim-to-real transfer for navigation. While these camera-based representations highlight the benefits of pretrained perception, they are typically designed for perspective projections with dense, isotropic pixel grids. Adapting them to panoramic LiDAR range images requires accounting for the large horizontal-to-vertical aspect ratio, structured beam sparsity, and azimuthal wraparound.

b) LiDAR representations: Most general-purpose LiDAR encoders operate directly on irregular 3D geometry. Seminal architectures like PointNet++ [19] and MinkowskiEngine [20] consume raw point sets or sparse spatial tensors, while recent FMs such as Sonata [5] and Utonia [6] pretrain Point Transformers [21] across large-scale 3D datasets. Lightweight variants like LitePT [22] optimize hierarchical point processing to reduce overhead. However, point-based and sparse-tensor operations incur prohibitive memory and runtime costs across thousands of parallel RL environments (Sec. IV). Projection-based approaches circumvent these costs by structuring scans into standard arrays. Notably, RangeNet++ [7] showed that spherical projection allows efficient 2D convolutions, and RangeFormer [23] further scaled range representations using vision transformers. However, these systems are optimized and benchmarked for semantic perception, and pretrained range encoders remain largely unexplored for RL navigation.

c) Navigation-aligned representation learning: Effective pretraining preserves task-relevant information. Prior work guides visual navigation encoders with auxiliary supervision, including action and temporal-distance predictions [24], dynamics and progress targets [25], depth prediction and loop-closure classification [26], and auxiliary occupancy prediction for stereo cameras [27]. While these studies show the benefit of task-aligned supervision for camera inputs, combining complementary geometric and semantic objectives for LiDAR navigation has received little attention.

III. Method

Our approach consists of three stages. We first project synthetic depth observations into LiDAR range images to generate a large pretraining dataset. We then pretrain a deterministic autoencoder with reconstruction, semantic segmentation, and BEV prediction objectives. Finally, we train a recurrent actor–critic navigation policy utilizing the spatial latents obtained from the frozen encoder.

A. Range-Image Generation

We use TartanGround [8] to generate synthetic LiDAR range images. For each pose, six depth images with known intrinsics and extrinsics are back-projected and transformed into a common sensor frame. The aggregated point cloud is mapped into the azimuth and elevation bins of a target LiDAR's spherical range image (Fig. 2(a)), with bin collisions resolved by minimum pooling. Projecting 3D point clouds into 2D range images yields fixed-shape tensors for efficient batching across thousands of parallel RL environments at lower computational cost than 3D point-cloud encoders (Table III).

B. Range Preprocessing and Augmentation

Let r denote a clean metric range image. During encoder pretraining, we apply random circular horizontal rolls and

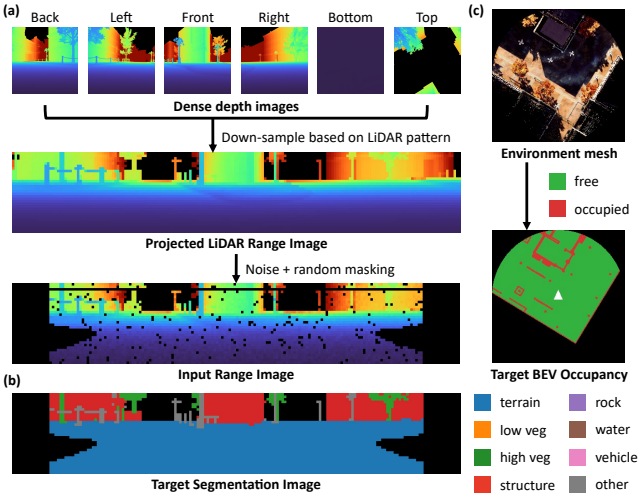


Fig. 2: Data processing for encoder pretraining. (a) Six dense depth views are fused in 3D and projected onto the target spherical LiDAR grid, then corrupted with noise and random masking to form the encoder input. (b) A similar approach generates the semantic target from segmented images. (c) The privileged BEV occupancy and ground-height target is rendered from the scene mesh.

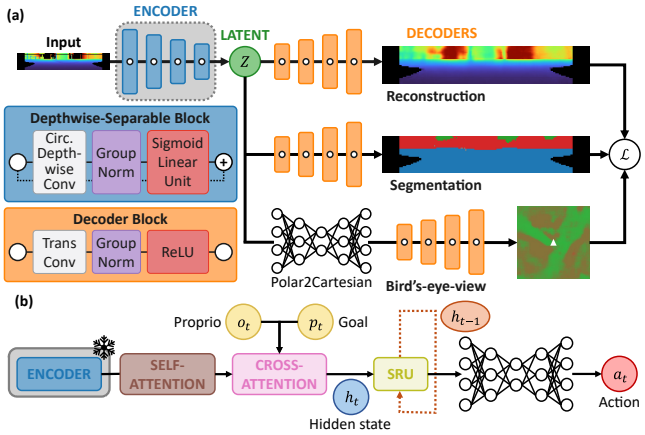


Fig. 3: Pretraining and policy architecture. (a) The encoder compresses the range image into a spatial latent that three decoder heads read in parallel: range reconstruction, per-return semantic segmentation, and privileged BEV occupancy and ground height. A fully connected bottleneck precedes the BEV deconvolutions, mapping the polar latent onto the Cartesian grid. Insets detail the depthwise-separable encoder block and the transposed-convolution decoder block. All heads are discarded after pretraining. (b) The frozen encoder feeds attention over noisy proprioception and the relative goal into the recurrent SRU policy that predicts the action.

corrupt valid ranges with scan-ring bias, distance- and edge-dependent Gaussian noise, random dropout, and contiguous masking. The corresponding clean, rolled scan remains the reconstruction target. These augmentations simulate arbitrary sensor yaw, measurement noise, missing returns, and occlusion (Fig. 2(a)). For a clean measurement r_{ij}, the pre-dropout corrupted measurement is

$$\tilde{r}_{ij} = \mathrm{clip}(r_{ij} + b_i + \eta_{ij}, d_{\min}, d_{\max}), \tag{1}$$

where $b_i \sim \mathcal{N}(0, \sigma^2_{\mathrm{ring}})$ is shared by scan ring i, and the variance of η_{ij} grows with range and local edge magnitude. The dropout probability likewise increases with range.

Before encoding, invalid returns in r and $\tilde{r}$ are filled with the maximum working range and all ranges are clipped to the LiDAR's working interval. Finally, we log-transform the range images as $x_{ij} = \log(1 + r_{ij})$ and $\tilde{x}_{ij} = \log(1 + \tilde{r}_{ij})$. The logarithm emphasizes nearby geometry while compressing distant returns. We use $\tilde{x}$ as the encoder input and x as its corresponding clean reconstruction target.

C. Task-Aligned Pretraining

We pretrain the encoder on range images with geometric and semantic targets, allowing the pretraining dataset to scale through scene rendering without requiring action sequences or goal-conditioned rollouts. Within the encoder (Fig. 3(a)), circular convolutions preserve continuity across the 360° azimuthal seam [28], while depthwise-separable residual blocks reduce computational cost and anisotropic convolutions provide wider horizontal receptive fields suited to the panoramic range-image grid [28], [29]. The encoder produces the latent z_L and applies layer normalization [30] independently at each spatial location.

Three decoder heads condition this bottleneck. The range decoder predicts the clean log-transformed range image $\hat{x}$, the semantic head predicts a semantic label for each valid return, and the BEV head predicts top-down occupancy and ground height. The combined objective is

$$\begin{aligned}\mathcal{L}_{\mathrm{pre}} = \lambda_{\mathrm{recon}}\mathcal{L}_{\mathrm{recon}} + \lambda_{\mathrm{grad}}\mathcal{L}_{\mathrm{grad}} \\ + \lambda_{\mathrm{seg}}\mathcal{L}_{\mathrm{seg}} + \lambda_{\mathrm{BEV}}\mathcal{L}_{\mathrm{BEV}}.\end{aligned} \tag{2}$$

The λ coefficients weight their corresponding objectives. The range decoder is supervised jointly by $\mathcal{L}_{\mathrm{recon}}$, an edge-weighted masked Huber loss on its prediction $\hat{x}$, and $\mathcal{L}_{\mathrm{grad}}$, which matches the spatial gradients of $\hat{x}$ and the clean target x. More precisely, let e_{ij} denote the target range discontinuity around pixel (i, j) and $\bar{e}$ its clamped mean. With valid mask m_{ij} and edge weight $w_{ij} = 1 + \alpha_e e_{ij}/\bar{e}$, the reconstruction term is

$$\mathcal{L}_{\mathrm{recon}} = \frac{\sum_{ij} m_{ij} w_{ij}\, \mathrm{Huber}(\hat{x}_{ij}, x_{ij})}{\sum_{ij} m_{ij} w_{ij}}. \tag{3}$$

The gradient term applies the same robust penalty to adjacent horizontal and vertical differences in $\hat{x}$ and x wherever both target pixels are valid. Let $\mathcal{D} = \{(0,1),(1,0)\}$ denote the horizontal and vertical offsets and $\Delta_d x_{ij} = x_{i+d_i,j+d_j} - x_{ij}$. Then

$$\mathcal{L}_{\mathrm{grad}} = \frac{\sum_{d\in\mathcal{D}} \sum_{ij} m_{ij} m_{i+d_i,j+d_j}\, \mathrm{Huber}(\Delta_d \hat{x}_{ij}, \Delta_d x_{ij})}{\sum_{d\in\mathcal{D}} \sum_{ij} m_{ij} m_{i+d_i,j+d_j}}. \tag{4}$$

Semantic labels are projected with the same spherical bins and the raw TartanGround labels are collapsed into eight classes: terrain, low vegetation, high vegetation, structure,

rock, water, vehicle, and other (Fig. 2(b)). On valid, non-self-occluded returns,

$$\mathcal{L}_{\text{seg}} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{Lovász}}. \tag{5}$$

Here, $\mathcal{L}_{\text{CE}}$ is the mean pixel-wise cross-entropy loss, and $\mathcal{L}_{\text{Lovász}}$ is the Lovász–Softmax loss, a differentiable surrogate for the mean intersection-over-union (mIoU) segmentation metric [31]. The privileged BEV target contains free/blocked occupancy and ground height on a Cartesian grid, supervised from scene-level semantic point clouds in visible and occluded cells (Fig. 2(c)). For known cells,

$$\mathcal{L}_{\text{BEV}} = \mathcal{L}^{\text{occ}}_{\text{CE}} + \mathcal{L}^{\text{occ}}_{\text{Lovász}} + \mathcal{L}^{\text{height}}_{1}. \tag{6}$$

The term $\mathcal{L}^{\text{height}}_{1}$ is the mean absolute error between the predicted and target ground heights over known cells.

All three decoder heads are used only during pretraining; only the encoder is retained for the RL policy.

D. Recurrent Navigation Policy

The pretrained LiDAR encoder is frozen during policy learning (Fig. 3(b)). Its spatial latent at time t, $z_{L,t}$, is combined with noisy proprioception and the relative goal before entering the recurrent memory:

$$h_t = \text{SRU}(z_{L,t}, o^{\text{prop}}_t, g_t, h_{t-1}). \tag{7}$$

Here, o^{prop}_t is the proprioceptive observation, g_t is the relative goal, and h_t and h_{t-1} are the current and previous hidden states, respectively. The actor predicts the control action from h_t, while an asymmetric critic can use privileged geometric observations during training.

We train the actor–critic with Mutual Distillation Policy Optimization (MDPO) [32] in massively parallel Isaac Lab environments. Randomized terrain, dynamics, sensing, actuation, delays, and disturbances expose the policy to transfer-relevant variation. Episodes end on goal reach, timeout, unsafe contact, excessive roll/pitch, or a fall. Following SRU [3], the reward combines sparse task rewards for reaching the goal with dense shaping rewards for goal progress, smooth motion, and path efficiency.

IV. Experiments

We first outline the implementation details (Sec. IV-A) and evaluation protocol (Sec. IV-B). Next, our experiments evaluate three central questions: (i) how do our pretraining objectives affect navigation SR (Sec. IV-C), (ii) how does panoramic LiDAR compare with front-facing depth (Sec. IV-D), and (iii) how does our range-image encoder compare with general-purpose point-cloud encoders (Sec. IV-E). We then ablate the design choices of our encoder (Sec. IV-F), and finally perform several real-world deployments (Sec. IV-G).

A. Implementation Details

For encoder pretraining, six TartanGround depth views are fused and sampled to simulate a 32-beam RoboSense Helios LiDAR with a 70° vertical field of view (VFoV) and a 360° horizontal field of view (HFoV). We use a 0.4° horizontal resolution and a range of $[0.2, 20]$ m. After adding noise to the resulting 32×900 range image, we apply minimum pooling along the azimuth to reduce it to 32×180, preserving the closest returns while eliminating horizontal redundancy. For the noise model in Sec. III-B, we use $\sigma_0 = 0.01$, range factor 0.002, edge factor 0.03, base dropout 0.005, range-dependent dropout factor 0.02, and $\sigma_{\text{ring}} = 0.01$.

The encoder maps an input in $\mathbb{R}^{1\times32\times180}$ to a spatial latent in $\mathbb{R}^{64\times4\times12}$. The pretraining dataset contains around 144,000 frames. We jointly optimize the encoder and all enabled decoder heads for 5,000 epochs using the Adam optimizer with a batch size of 1,024 and a learning rate of 6×10^{-4}. The loss weights are set to $\lambda_{\text{recon}} = 1$, $\lambda_{\text{grad}} = 0.3$, $\lambda_{\text{seg}} = 2$, and $\lambda_{\text{BEV}} = 2$ when the corresponding heads are enabled, with edge-weight coefficient $\alpha_e = 0.5$. The BEV target covers 40×40 m with a 256×256 grid at 0.15625 m per cell.

For the RL training stage, we train the actor–critic with MDPO in 1,920 parallel environments. The policy has a one-layer 512-unit recurrent state followed by $[512, 256, 128]$ actor and critic multilayer perceptrons (MLPs). The policies in Table I are trained for 15,000 iterations, whereas Tables V and IV use a shorter budget of 5,000 training iterations for ablations. Maximum episode duration ranges from 60 to 120 s across environments.

To increase the robustness of the encoder and RL policy, we randomly mask parts of the range image to simulate occlusions by different robot-body and payload patterns. We also add stick noise to simulate returns from thin poles, bars, and ring-like structures near the sensor. These patterns model persistent structured artifacts that are not captured by independent Gaussian noise or dropout.

B. Evaluation Protocol

Fig. 4 shows representative simulation environments. Success denotes reaching the goal within 0.5 m before termination. Every evaluation uses 1,024 fixed start–goal scenarios per environment. The base mixture comprises procedurally generated mazes, pillar fields, stairs, and pits, exposing the policy to corridor selection, isolated obstacles, elevation changes, and negative obstacles. Only these environments are used for RL training. The held-out TartanGround scenes [8] comprise AbandonedCable, Gascola, ModernCityDowntown, and Market, which provide more diverse layouts and realistic object arrangements. We additionally use two targeted held-out settings. The side-wall evaluation places the robot approximately 0.10 m from walls to test navigation from cluttered starts with limited local clearance. The dynamic-obstacle evaluation introduces 13 pedestrians moving along random paths between spawn and goal locations. For the comparison with a front-facing depth camera policy (Sec. IV-D), we retrain the policy from [3] under comparable RL settings.

C. Task-Aligned Pretraining Objectives

We compare three cumulative LiDAR pretraining objectives: reconstruction alone, adding semantic segmentation,

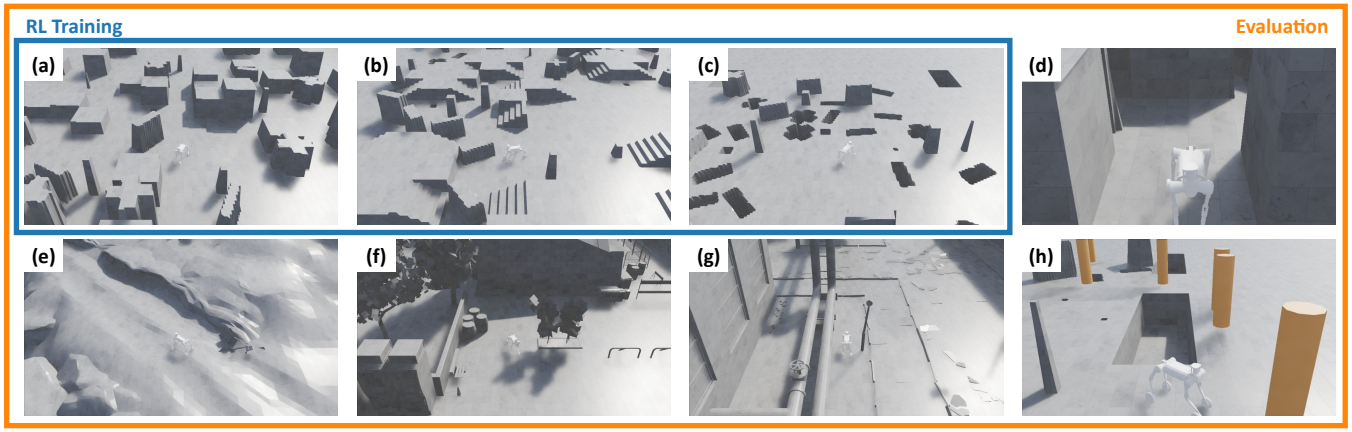


Fig. 4: Simulation environments. (a) Maze, (b) stairs, (c) pits, (d) side wall, (e) Gascola, (f) ModernCityDowntown (City), (g) AbandonedCable (Cable), and (h) dynamic obstacles, where the orange cylinders are moving pedestrians. For the results in Table I, the environments in the blue box (a)–(c) are examples of the environments used for RL training, whereas all environments are used for evaluation.

and adding BEV prediction. Checks in Table I mark each encoder's pretraining heads.

The encoder supervised by segmentation and BEV objectives is the strongest LiDAR variant in every column. Notably, it improves substantially over reconstruction on Pits (26.76% to 92.09%) and AbandonedCable (51.46% to 81.45%), environments with many obstacles below the sensor height, where the LiDAR beams are sparser. A reconstruction objective can be dominated by the many easier returns without ensuring that the latent preserves these relatively small but safety-critical structures. Semantic supervision encourages the encoder to distinguish geometrically similar returns with different navigation implications. BEV supervision then provides a further gain by organizing those features around free space and terrain layout. This improvement in navigation performance persists outside the RL training mixture as well, indicating that the benefit transfers across layouts. Together, the results suggest that semantic and BEV supervision addresses the challenges of sparse LiDAR data and produces a more navigation-aligned LiDAR latent than reconstruction alone.

D. Panoramic LiDAR versus Front-Facing Depth

Next, we compare our LiDAR policy with the dense front-facing depth policy to evaluate whether our encoder provides effective navigation features from sparse LiDAR observations and whether panoramic coverage offers advantages over denser observations restricted to the robot's forward view.

Across the base mixture and TartanGround environments, our LiDAR policy achieves higher SR than the depth policy in five of the eight environments (Table I). Depth achieves higher SR on stairs, pillars, and Cable. Its advantage on stairs and Cable may reflect denser observations of step geometry and thin or low-lying obstacles that sparse LiDAR beams can miss. LiDAR performs better in the remaining environments, including pits with negative obstacles and mazes with corridor layouts, with the largest gains in Gascola and City, which feature uneven outdoor terrain and urban street layouts. Overall, our LiDAR policy achieves slightly higher average SR on both evaluation sets despite its sparser observations, suggesting that our encoder provides features useful for navigation.

Front-facing depth also creates a partial-observability problem when safe navigation depends on geometry outside the robot's current viewing direction. This limitation is particularly evident in two evaluation settings (Table I). In the side-wall setting, the robot starts closer to walls. As it turns, the robot may collide with surrounding obstacles outside of the frontal depth view, leading to the depth policy failing more frequently. Panoramic LiDAR retains this lateral geometry and enables the policy to account for the available clearance before moving sideways or turning, improving SR from 79.98% to 88.87%. In the dynamic-obstacle setting, pedestrians can approach from either side and enter the robot's path before appearing in the frontal view. By observing these agents earlier, the LiDAR policy can react and avoid them more reliably, increasing SR from 77.64% to 79.49% (Table I).

Beyond success rate, panoramic context improves route efficiency when multiple passages are locally plausible. This effect is particularly evident in the maze setting, where LiDAR can reveal lateral openings and surrounding corridor structure without requiring the robot to reorient or enter an unproductive passage. Across the base-mixture environments, the LiDAR policy consequently achieves an SPL score 0.58 pp higher than the depth policy and reaches the goal 1.38 s faster on average (Table II). The representative trajectories in Fig. 5 illustrate this behavior: the LiDAR policy recognizes a dead end earlier and avoids repeatedly revisiting clutter, producing a more direct route.

E. Comparison with General-Purpose Point-Cloud Encoders

Next, we compare our pretraining objectives with recent cross-domain and general-purpose point-cloud encoders Sonata [5] and Utonia [6].

Firstly, we measure the computational efficiency of the

TABLE I: Navigation SR (%) (↑) comparing depth and LiDAR policies trained using different pretraining objectives.

	Pretraining heads			Base mixture					TartanGround					Side-wall	Dynamic
Input	Recon	Seg	BEV	Maze	Pillars	Stairs	Pits	Average	Cable	Gascola	City	Market	Average	-	-
Depth		—		95.61	**99.02**	**95.70**	91.11	95.36	**88.57**	47.27	83.11	72.85	72.95	79.98	77.64
LiDAR	✔	✘	✘	78.81	80.08	68.07	26.76	63.43	51.46	29.98	46.39	69.92	49.44	61.52	63.18
LiDAR	✔	✔	✘	95.70	96.09	80.08	82.13	88.50	77.83	43.07	80.76	68.07	67.43	85.94	78.03
LiDAR	✔	✔	✔	**96.88**	98.54	94.34	**92.09**	**95.46**	81.45	**51.56**	**86.13**	**73.54**	**73.17**	**88.87**	**79.49**

TABLE II: LiDAR and depth navigation efficiency.

	SPL (%, ↑)		Time to goal (s, ↓)	
Split	LiDAR	Depth	LiDAR	Depth
Maze	**90.15**	89.37	**15.87**	17.94
Pillars	**93.71**	92.50	**11.89**	12.90
Stairs	**75.38**	75.33	**16.41**	17.82
Pits	**81.19**	80.91	**16.60**	17.60
All	**85.11**	84.53	**15.19**	16.57

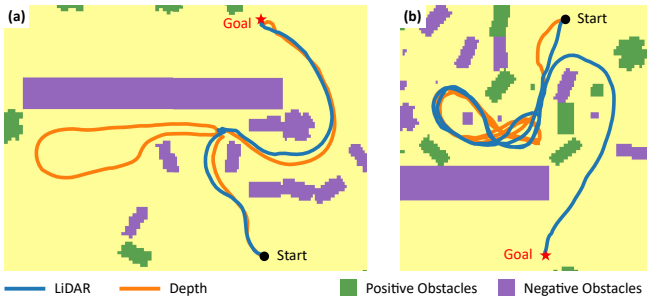


Fig. 5: Representative LiDAR (blue) and depth (orange) policy trajectories. (a) The depth policy detours into a dead end before recovering, whereas the LiDAR policy recognizes it earlier and takes a direct route. (b) The depth policy repeatedly revisits the same local clutter, whereas the LiDAR policy escapes and reaches the goal.

encoders by comparing the maximum number of parallel RL environments that fit in GPU memory and the resulting training throughput on a single NVIDIA RTX 4090 (Table III). The point-cloud encoders are large, accommodate relatively few parallel environments, and reach only 19–75 steps/s during RL training, making them impractical for large-scale training. In contrast, our range-image encoder achieves 1,469 steps/s with 1,920 parallel environments, supporting efficient, massively parallel RL training.

Nevertheless, in order to still compare with these point-cloud encoders, we distill their features into our range-image encoder architecture on the same TartanGround pretraining data. The distillation uses cosine similarity between the student and teacher features. For each teacher, we report its best-performing configuration: XYZ coordinates as input for Sonata, and XYZ coordinates, RGB color, and surface normals for Utonia. We then compare the downstream RL SR against our policy, which uses the task-aligned encoder. Table IV summarizes the results.

Evaluating on the base-mixture episodes, our pretraining objective exceeds the best Sonata- and Utonia-distilled variants by 7.32 pp and 11.40 pp, respectively (Table IV). Thus,

TABLE III: Encoder scalability for parallel RL.

Encoder	Representation	Params	Max RL envs (↑)	Throughput (steps/s, ↑)
Utonia [6]	Point transformer v3	137M	48	19
Sonata [5]	Point transformer v3	108M	64	30
LitePT-S [22]	Hybrid point hierarchy	12.7M	96	75
Ours	Fixed range image	**183k**	**1,920**	**1,469**

TABLE IV: Navigation SR (%) (↑) with the same encoder architecture pretrained using our pretraining objectives or features distilled from point-cloud encoders.

Pretraining source	SR (%) (↑)
Sonata [5]	85.06
Utonia [6]	80.98
Ours	**92.38**

our pretraining objective produces stronger downstream features than distillation from these point-cloud encoders. This suggests that the transferable 3D features learned by general-purpose encoders like Sonata and Utonia are not necessarily the best suited for navigation.

F. Encoder Architecture and Ablations

We next examine how our encoder's channel width, stem kernel size, and architecture affect navigation performance. All variants retain the same range-image representation and are evaluated on the base mixture (Table V).

Firstly, to test model size, we uniformly scale our encoder's six channel widths $(32/48/64/96/128/160)$ by a common factor. This scaling does not improve navigation SR.

Secondly, for kernel anisotropy, the 3×9 stem performs best, highlighting the importance of appropriately anisotropic kernels for LiDAR range images, whose azimuthal (width) dimension has many more pixels than their elevation (height) dimension.

Finally, we measure the effect of model type by comparing our encoder with the RegNet feature pyramid network (FPN) architecture [3], [33]. The best-performing RegNet-FPN variant obtains a lower reconstruction loss of 0.02809 than our encoder's 0.04685, yet our substantially smaller encoder obtains a much higher SR. Pixel-level range reconstruction is therefore not a reliable proxy for navigation performance.

G. Real-World Deployments

Transferring a navigation policy trained exclusively in simulation requires robustness to real-world sensor noise and occlusion, unstructured geometry, and dynamic obstacles. We evaluate whether our perception and navigation pipeline can handle these conditions without real-world fine-tuning.

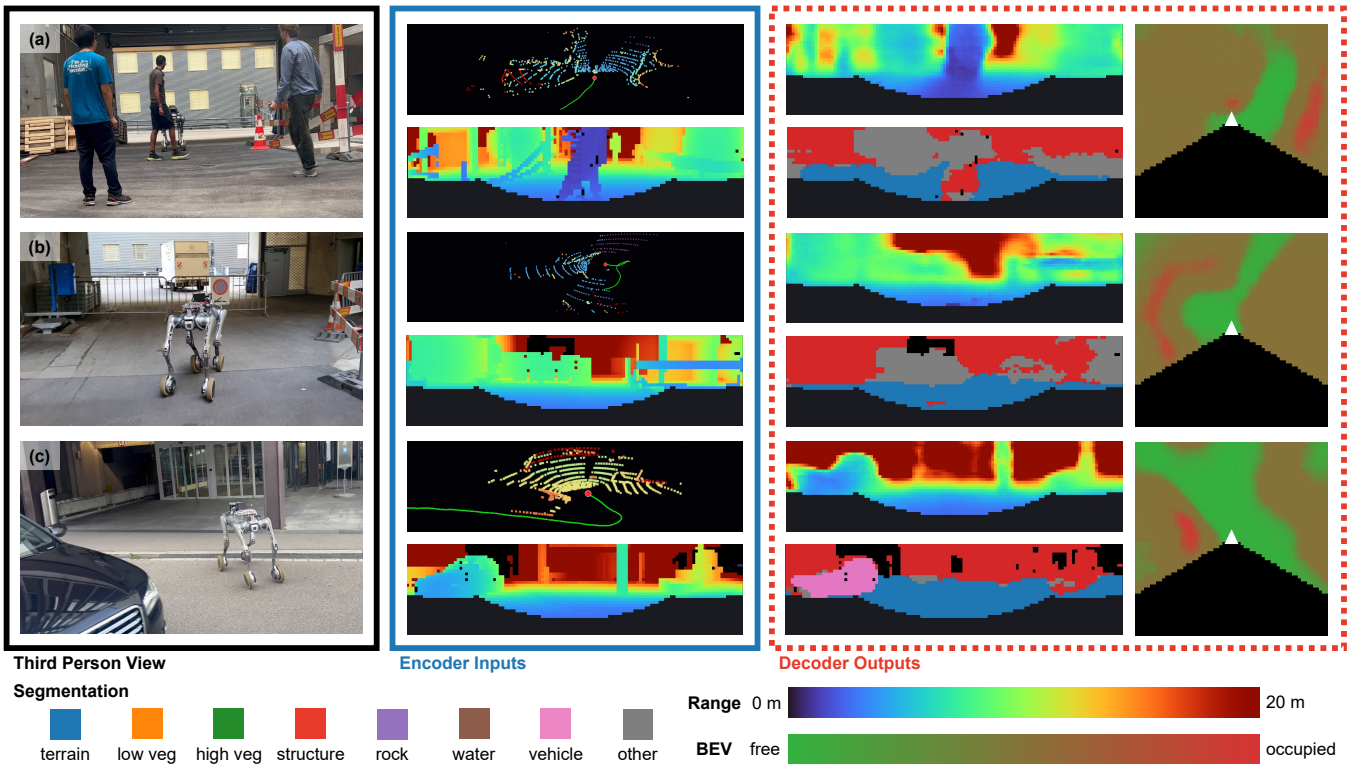


Fig. 6: Perception examples from physical deployment with (a) pedestrians, (b) a fence, and (c) vehicles. Each row shows the scene photograph, the input point cloud and its projected range image (blue box, top and bottom), and the decoder outputs (red dashed box): range reconstruction (left, top), semantic segmentation (left, bottom), and BEV prediction (right).

TABLE V: Range-image encoder ablations and architectural comparisons.

Ablation	Channel width	Stem kernel	Params	SR (%) (↑)
Ours	×1	3 × 9	183k	**92.38**
Model size	×1.5	3 × 9	301k	87.40
	×2	3 × 9	462k	91.53
	×3	3 × 9	916k	91.24
Kernel anisotropy	×1	3 × 3	182k	85.96
	×1	3 × 5	183k	84.67
	×1	3 × 13	183k	84.77
	×1	5 × 5	184k	80.00
	×1	5 × 9	184k	87.48
Architecture	*RegNet-FPN* [3]		5.8M	70.46

Our hardware platform is a Unitree B2-W equipped with a RoboSense Helios 32-beam LiDAR and an IMU. BIEVR-LIO [34] estimates the robot pose from the LiDAR and inertial measurements. The full stack runs onboard an NVIDIA Jetson AGX Orin. Because a rear-mounted payload box occludes part of the LiDAR scan, we mask the rear 120° sector before projecting each scan into a range image, as indicated by the black regions in Fig. 6. The encoder then maps this range image to a latent fed into the RL policy. Fig. 6 visualizes the learned representation through the reconstruction, semantic segmentation, and BEV prediction decoder outputs. These outputs show accurate geometry reconstruction, segmentation of obstacles such as fences and cars, and BEV occupancy prediction.

We evaluate the system across diverse real-world environments that challenge both sensing and navigation. In low-light and dark scenes, LiDAR provides geometric observations independently of ambient illumination. Dynamic crowds test whether the navigation policy can respond online to pedestrians whose positions and trajectories continually change. In construction sites with dense clutter, narrow passages, and irregular geometry, panoramic LiDAR reveals obstacles and openings outside a front-facing depth camera's field of view, while the recurrent policy retains temporal context to avoid revisiting dead ends. Representative deployments are shown in Figs. 1 and 6. Successful rollouts across these settings show that the LiDAR-based system can navigate under limited illumination while its mapless policy balances reactive pedestrian avoidance with longer-horizon progress, continuing through temporarily obstructed passages once they clear while avoiding repeated attempts at impassable routes. Because the same perception–policy interface is used in simulation and on the robot, these rollouts also demonstrate zero-shot sim-to-real transfer.

H. Limitations

RangeNav remains limited by sensor visibility and sensor-specific scan geometry. Sparse LiDAR sampling, range-image projection and downsampling, and sensor occlusions can discard fine structure or leave thin and low-lying obstacles unobserved. The rear payload mask further creates a substantial blind sector on the deployed platform. Moreover, the fixed-size range image is tied to the channel count, field of view, and angular resolution of the target LiDAR. Deployment with a different scan pattern may therefore

require explicit resampling before encoding or pretraining the encoder from scratch with sensor-specific input dimensions. Dynamic obstacles are also currently handled reactively, without explicit agent tracking or motion prediction, making sudden interactions and dense crowds potential failure cases. Future work will address cross-sensor adaptation and explicit motion prediction.

V. Conclusion

We present a compact, task-aligned LiDAR range-image encoder for RL navigation. Using synthetic range images with segmentation and BEV labels, we pretrain an architecture tailored to the spatial structure of LiDAR observations. Adding semantic segmentation and BEV prediction objectives substantially improves downstream navigation performance over range reconstruction alone. The resulting encoder supports efficient policy learning, and its task-aligned features yield better navigation performance than those distilled from substantially larger general-purpose point-cloud encoders. Compared with a front-facing depth policy, our LiDAR policy achieves greater path efficiency, reaches goals faster, and performs better in cluttered and dynamic-obstacle settings. Finally, deployment on a physical robot demonstrates zero-shot transfer in dark, crowded, and construction environments. Taken together, these findings establish task-aligned range-image pretraining as a practical foundation for LiDAR navigation.

Acknowledgements

This research was supported by the Swiss National Science Foundation (SNSF) under Project No. 227617 and Grant 200021E_229503, the Luxembourg National Research Fund (Ref. 18990533), and armasuisse Science and Technology.

The author(s) used generative AI to assist with drafting and editing the manuscript. All content was reviewed by the authors, who take full responsibility for it.

References

[1] W. Cai, J. Peng, Y. Yang, Y. Zhang, M. Wei, H. Wang, Y. Chen, T. Wang, and J. Pang, "NavDP: Learning sim-to-real navigation diffusion policy with privileged information guidance," *arXiv preprint arXiv:2505.08712*, 2025.

[2] Y. Inglin, J. Frey, C. Chen, and M. Hutter, "Less is more: Scalable visual navigation from limited data," *arXiv preprint arXiv:2601.17815*, 2026.

[3] F. Yang, P. Frivik, D. Hoeller, C. Wang, C. Cadena, and M. Hutter, "Spatially-enhanced recurrent memory for long-range mapless navigation via end-to-end reinforcement learning," *The International Journal of Robotics Research*, 2025.

[4] M. Patel, J. Frey, M. Mittal, F. Yang, A. Hansson, A. Bar, C. Cadena, and M. Hutter, "DeFM: Learning foundation representations from depth for robotics," *arXiv preprint arXiv:2601.18923*, 2026.

[5] X. Wu, D. DeTone, D. Frost, T. Shen, C. Xie, N. Yang, J. Engel, R. Newcombe, H. Zhao, and J. Straub, "Sonata: Self-supervised learning of reliable point representations," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 22 193–22 204.

[6] Y. Zhang, X. Wu, Y. Yang, X. Fan, H. Li, Y. Zhang, Z. Huang, N. Wang, and H. Zhao, "Utonia: Toward one encoder for all point clouds," *arXiv preprint arXiv:2603.03283*, 2026.

[7] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and accurate LiDAR semantic segmentation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

[8] M. Patel, F. Yang, Y. Qiu, C. Cadena, S. A. Scherer, M. Hutter, and W. Wang, "TartanGround: A large-scale dataset for ground robot perception and navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2025, pp. 20 524–20 531.

[9] A. Majumdar, K. Yadav, S. Arnaud, J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, T. Wu, J. Vakil *et al.*, "Where are we in the search for an artificial visual cortex for embodied intelligence?" *Advances in Neural Information Processing Systems*, vol. 36, pp. 655–677, 2023.

[10] A. Stooke, K. Lee, P. Abbeel, and M. Laskin, "Decoupling representation learning from reinforcement learning," in *International Conference on Machine Learning*, 2021.

[11] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "DINOv2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[12] M. Ranzinger, G. Heinrich, J. Kautz, and P. Molchanov, "AM-RADIO: Agglomerative vision foundation model reduce all domains into one," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2024, pp. 12 490–12 500.

[13] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "SAM 2: Segment anything in images and videos," in *International Conference on Learning Representations*, 2025.

[14] G. Zhou, H. Pan, Y. LeCun, and L. Pinto, "DINO-WM: World models on pre-trained visual features enable zero-shot planning," *arXiv preprint arXiv:2411.04983*, 2024.

[15] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter, "Fast traversability estimation for wild visual navigation," *arXiv preprint arXiv:2305.08510*, 2023.

[16] M. Schmittle, R. Baijal, N. Hatch, R. Scalise, M. Guaman Castro, S. Talia, K. Khetarpal, B. Boots, and S. Srinivasa, "Long range navigator (LRN): Extending robot planning horizons beyond metric maps," in *Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 305, 2025, pp. 3185–3199.

[17] H. Shah, E. Tevere, D. Atha, M. Kaufmann, S. Khattak, M. Patel, M. Hutter, J. Frey, and P. Spieler, "WildOS: Open-vocabulary object search in the wild," *arXiv preprint arXiv:2602.19308*, 2026.

[18] F. Yang, C. Wang, C. Cadena, and M. Hutter, "iPlanner: Imperative path planning," *arXiv preprint arXiv:2302.11434*, 2023.

[19] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017.

[20] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal ConvNets: Minkowski convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[21] X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao, "Point transformer v3: Simpler, faster, stronger," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2024, pp. 4840–4851.

[22] Y. Yue, D. Robert, J. Wang, S. Hong, J. D. Wegner, C. Rupprecht, and K. Schindler, "LitePT: Lighter yet stronger point transformer," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026, pp. 24 173–24 184.

[23] L. Kong, Y. Liu, R. Chen, Y. Ma, X. Zhu, Y. Li, Y. Hou, Y. Qiao, and Z. Liu, "Rethinking range view representation for LiDAR segmentation," in *IEEE/CVF International Conference on Computer Vision*, 2023, pp. 228–240.

[24] J. Ye, D. Batra, E. Wijmans, and A. Das, "Auxiliary tasks speed up learning point goal navigation," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 155. PMLR, 2021, pp. 498–516. [Online]. Available: https://proceedings.mlr.press/v155/ye21a.html

[25] S. S. Desai and S. Lee, "Auxiliary tasks for efficient learning of point-goal navigation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 717–725.

[26] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, D. Kumaran, and R. Hadsell, "Learning to navigate in complex environments," in *International Conference on Learning Representations*, 2017.

[27] H. Li, T. Padir, and H. Jiang, "StereoNavNet: Learning to navigate using stereo cameras with auxiliary occupancy voxels," *arXiv preprint arXiv:2403.12039*, 2024.

[28] B. Yang, P. Pfreundschuh, R. Siegwart, M. Hutter, P. Moghadam, and V. Patil, "TULIP: Transformer for upsampling of LiDAR point clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 354–15 364.

[29] B. Wu, A. Wan, X. Yue, and K. Keutzer, "SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation

from 3D LiDAR point cloud," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 1887–1893.

[30] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[31] M. Berman, A. R. Triki, and M. B. Blaschko, "The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4413–4421.

[32] Z. Xie, J. Cao, C. Wang, F. Yang, M. Hutter, Q. Zhang, J. Zhang, and R. Xu, "Representation convergence: Mutual distillation is secretly a form of regularization," *arXiv preprint arXiv:2501.02481*, 2025.

[33] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 428–10 436.

[34] P. Pfreundschuh, T. Tuna, C. Le Gentil, R. Siegwart, C. Cadena, and H. Oleynikova, "BIEVR-LIO: Robust LiDAR-inertial odometry through bump-image-enhanced voxel maps," in *Proceedings of Robotics: Science and Systems*, 2026.