REGRESSION ACCURACY IS NOT ENOUGH: DESTINATION-AGNOSTIC GRAPH EMBEDDINGS FOR POTENTIAL-FIELD NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Potential-field navigation using destination-conditioned Graph Neural Networks (GNNs) requires a new forward pass whenever the target goal changes. To eliminate this per-goal computational bottleneck, we propose a *destination-agnostic* embedding framework. By executing a single forward pass per map, our model assigns a D-dimensional vector z_i to each free cell such that the distance $\|z_i - z_{\text{goal}}\|$ approximates the shortest path to an arbitrary goal, enabling instantaneous inference for any goal without retraining or additional forward passes. We train the network using a combined objective of pairwise distance regression and an order-preserving ranking loss. Model performance is evaluated on both distance-regression accuracy and task-level navigation success rate via greedy descent. Through extensive hyperparameter sweeps over ranking-loss weights (from 1000 down to 1) and anchor/pair sampling densities, we reveal that regression accuracy (Mean Squared Error, MSE) and navigation success are surprisingly decoupled: while MSE improves monotonically toward the smallest weight, completion rate and ranking correctness peak at an intermediate weight. Thus, the configuration that minimizes regression error fails to maximize navigability. We trace this phenomenon to specific map topographies—including cases where completion rate remains unchanged despite substantial MSE improvements—and attribute it to structural limitations of symmetric embeddings in greedy goal-reaching. Our findings demonstrate that regression error alone is an unreliable proxy for downstream planning success, highlighting the need for future evaluation protocols to report task-level navigability alongside embedding accuracy.

1 INTRODUCTION

Potential-field methods are a classical and effective approach to grid and graph navigation (Khatib, 1986): they assign every traversable cell a scalar value that decreases monotonically toward the goal, allowing an agent to follow the local gradient. However, when the field is generated by a *destination-conditioned* GNN—which takes the goal as an input channel to predict a goal-specific field—every change of destination necessitates a new forward pass over the entire map. This per-goal computational cost presents the main practical bottleneck when querying a single map against many goals, such as in repeated route queries, multi-agent navigation with heterogeneous targets, or interactive replanning.

To address this limitation, we study a *destination-agnostic* alternative. Our GNN performs a single forward pass per map and assigns every free cell a D-dimensional embedding vector $\boldsymbol{z}_i$ such that $\|\boldsymbol{z}_i - \boldsymbol{z}_{\text{goal}}\|$ approximates the shortest-path distance to an arbitrary goal cell. Because these embeddings are independent of the destination, the same forward pass supports any downstream goal without retraining; at query time, only an $O(1)$ vector subtraction and norm computation are required. This framework reformulates field regression as a graph-embedding task, trained with a combination of pairwise distance regression and an order-preserving ranking loss between candidate targets.

The central empirical puzzle reported in this paper is that *regression accuracy* (how closely $\|\boldsymbol{z}_i - \boldsymbol{z}_{\text{goal}}\|$ matches the true hop distance) and *downstream navigability* (whether greedy descent

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

reaches the goal) are only loosely coupled and can move in opposite directions. Through extensive hyperparameter sweeps over the ranking-loss weight and sampling density, we consistently observe configurations where lower regression error leaves the completion rate unchanged or even degraded—a significant and reproducible effect rather than a minor artifact. We argue that this behavior stems from a structural property of the objective: greedy descent requires the field to be *locally* monotonic along the entire path, a substantially stricter requirement than achieving a low *average* regression error.

Contributions.

- We propose a destination-agnostic embedding GNN with an anchor-based regression and ranking objective, enabling reuse across arbitrary goals from a single forward pass per map (Section 4).
- We provide empirical evidence that regression accuracy and greedy-navigation success are decoupled: MSE is minimized at one extreme of the ranking-loss weight sweep, whereas completion rate and ranking correctness peak at an interior value (Sections 6–7).
- We present a comprehensive ranking-weight and anchor-density ablation study that serves as a practical recipe for stable training with this loss class (Section 6).
- We analyze failure cases on maps where the decoupling is most pronounced and explain why symmetric embeddings are structurally limited for greedy goal-reaching (Section 7).

2 Related Work

Destination-conditioned field regression. The baseline we replace is a GNN that takes the goal as an input channel and regresses a per-goal potential field in a single forward pass, requiring a new pass per goal change – part of a broader line of work using neural networks to predict planning-relevant quantities (heuristics, values, potential fields) in place of, or to accelerate, classical search.

GNN architectures for spatial reasoning. We build on attention-based message passing (Veličković et al., 2018; Brody et al., 2022) over an 8-connected grid graph, distinct from the 4-connected graph defining ground-truth shortest-path labels (Section 3): connectivity for message passing is a modeling choice, not part of the label definition.

Position-aware, anchor-based, and goal-conditioned embeddings. Sampling anchor nodes and computing distances to them is a proven technique for injecting global position information into otherwise permutation-equivariant message passing (You et al., 2019). While our anchor sampling scheme (Section 4) is architecturally similar, it uses anchors strictly as a training-time sampling mechanism for arbitrary goals selected at inference time, evaluating downstream navigability alongside regression accuracy. More broadly, factorizing distance or value functions via separate state and goal embeddings to support arbitrary goals without retraining draws from Universal Value Function Approximators (Schaul et al., 2015). Our embedding serves as a supervised, planning-focused instantiation of this paradigm, where $\|\boldsymbol{z}_i - \boldsymbol{z}_{\text{goal}}\|$ functions as a learned distance-to-goal estimator.

Quasimetric embeddings and ranking/metric-learning losses. A separate line of work argues symmetric embeddings are structurally limited for goal-reaching, since optimal-cost distances need not be symmetric, proposing explicit quasimetric parameterizations instead (Wang et al., 2023). Our grid is undirected, so ground-truth distances *are* symmetric; we argue in Section 7 that even so, low *average* error does not guarantee the *local, everywhere* monotonicity greedy descent needs – related but distinct from the failure mode motivating quasimetric embeddings. Our ranking loss itself is a fixed-margin hinge on pairwise order, in the family of triplet/contrastive losses used broadly in metric learning (Schroff et al., 2015); its gradient has fixed magnitude when a margin is violated regardless of violation size, whose interaction with a separately-weighted regression term we characterize empirically (Section 4.3).

3 Problem Setup

Map and graphs. A map is represented as an occupancy grid with a set of free cells V. We define two distinct graphs over V: an 8-connected message-passing graph G_{mp} (diagonal neighbors without

corner-cutting) that specifies the GNN's receptive field, and a 4-connected labeling graph G_{bfs} that defines the ground-truth shortest-path distances. These two connectivities are independent and do not need to match.

Destination-agnostic embedding. A single forward pass of the GNN f_θ processes the map to yield $f_\theta(\text{map}) = \{\boldsymbol{z}_i \in \mathbb{R}^D\}_{i \in V}$, generating one D-dimensional embedding per free cell without relying on a specific target goal. The only node-level inputs provided to the network are normalized row and column coordinates, ensuring that no destination information is observed during feature extraction.

Potential field. For an arbitrary goal cell $g \in V$ specified after the forward pass, the predicted potential field value at cell i is computed as:

$$\text{field}_{\text{pred}}(i) = \|\boldsymbol{z}_i - \boldsymbol{z}_g\|_2. \tag{1}$$

An agent navigates by greedily selecting the G_{mp} neighbor (8-connected, matching both the GNN's receptive field and the map reachability guarantee; Section 5.1) that minimizes $\text{field}_{\text{pred}}$. The navigation process terminates successfully when the agent reaches g, or fails if it encounters a local minimum or exceeds the maximum step budget.

Two evaluation axes. We keep separate throughout: (a) *regression accuracy*, how close $\text{field}_{\text{pred}}$ is to true BFS(Breadth-First Search) distance on average, and (b) *navigability*, the completion rate of greedy descent from a fixed set of start cells. An oracle completion rate on the *true* field is near 100% on essentially every test map, so completion-rate shortfalls are attributable to embedding error, not the greedy procedure itself.

4 METHOD

4.1 ARCHITECTURE

The embedding network (`EmbeddingGNN`) is structured as follows:

$$\begin{aligned} &\text{input_proj (Linear)} \;\rightarrow \big[\text{GATv2Conv} \rightarrow \text{GroupNorm} \rightarrow \text{ReLU} \rightarrow \text{Dropout}\big]^{\times \text{num_layers}} \\ &\qquad\qquad \text{(residual)} \;\rightarrow \text{output_proj (Linear)}, \end{aligned}$$

where GroupNorm (Wu & He, 2018) is applied between layers. Our default configuration uses hidden_dim $= 64$, embed_dim $= 16$ (i.e., $D = 16$), num_layers $= 4$, 4 attention heads, and a dropout rate of 0.1. Message passing is performed over G_{mp}. The model receives only two node-level input features—the normalized spatial coordinates (`row_norm`, `col_norm`)—and never observes destination information during execution.

4.2 ANCHOR SAMPLING

Because a single forward pass generates node embeddings for the entire map, supervision is provided by sampling pairs of cells rather than conditioning on a fixed target goal. In the default `grid` mode, an evenly spaced set of `num_anchors` positions is cached for each map. For every anchor, `pairs_per_anchor` target cells are resampled in each epoch. These targets supply regression labels (Section 4.3) and, when paired with a second target from the same anchor, provide ranking labels. Since the forward pass is executed only once per map regardless of `num_anchors` or `pairs_per_anchor`, increasing these parameters grows indexing and loss computation costs without increasing neural network forward-pass runtime—a property we leverage in Section 6.

4.3 TRAINING OBJECTIVE

The total training loss combines a pairwise distance-regression term and an order-preserving ranking term:

$$\mathcal{L} = \text{mse_weight} \cdot \mathcal{L}_{\text{reg}} + \text{rank_weight} \cdot \mathcal{L}_{\text{rank}}. \tag{2}$$

Regression term. $\mathcal{L}_{\text{reg}}$ measures the mean squared error between the predicted embedding distance $\|\boldsymbol{z}_{\text{anchor}} - \boldsymbol{z}_{\text{target}}\|$ and the ground-truth BFS distance in G_{bfs}, defined as hop_dist $\cdot$ level_step. Because regression operates on the *raw* (unnormalized) hop-count scale, initial loss magnitudes typically range in the hundreds to thousands.

Ranking term. $\mathcal{L}_{\text{rank}}$ enforces pairwise order preservation using a fixed-margin hinge loss. For two targets t_1, t_2 sampled from the same anchor where t_1 is ground-truth closer than t_2, the predicted embedding distance to t_1 must be smaller than that to t_2 by at least rank_margin (default 0.5):

$$\mathcal{L}_{\text{rank}} = \text{relu}\big(\text{rank_margin} - \text{sign}(\Delta_{\text{true}}) \cdot \Delta_{\text{pred}}\big), \tag{3}$$

where Δ_{true} and Δ_{pred} represent the ground-truth and predicted distance differences between t_1 and t_2, respectively. When the margin is violated, the loss gradient maintains a *fixed* magnitude (± 1) regardless of violation severity. Consequently, rank_weight directly scales the gradient update magnitude for every violated pair; setting an excessively large value (e.g., 1000) can destabilize embedding training rather than simply saturating loss values (confirmed in Section 6).

Hyperparameters and discarded alternatives. This work builds on an earlier precursor adapted from the Water Sink Model (Jeon & Jung, 2019), which regresses a scalar water-level field toward a single fixed destination per map. Initial experiments regressing destination-specific water levels failed to generalize to unseen test maps—achieving a markedly low completion rate—which motivated the destination-agnostic formulation presented here.

Unless specified otherwise, default training parameters are: mse_weight $= 1.0$, rank_margin $= 0.5$, level_step $= 1.0$, hidden_dim $= 64$, embed_dim $= 16$, num_layers $= 4$, Adam optimizer (Kingma & Ba, 2015) with learning rate 5×10^{-4}, 200 epochs, random seed 42, and anchor_mode $=$ `grid`. Because the fixed-magnitude ranking gradient is scaled relative to a raw-scale regression target, rank_weight required values one to three orders of magnitude larger than conventional settings, prompting our hyperparameter sweep starting at 1000.

Prior to establishing this ranking loss, we evaluated an alternative approach: a GAT architecture trained with a hinge penalty on *adjacent* cells along anchor-to-target BFS paths. This formulation required each step toward the target to strictly decrease predicted distance—intuitively aligning with greedy descent requirements. However, this approach achieved only $\sim 65\%$ completion (compared to 92–93% for our primary model) and was discarded. While not a controlled ablation due to differing hyperparameters, this result foreshadows a core insight of this paper: enforcing local monotonicity *directly* can perform worse than encouraging it *indirectly* (Section 8).

5 Experimental Setup

5.1 Data

All experiments are conducted using $1{,}000$ training maps and a disjoint test set of 50 held-out maps. Both datasets are generated using the same script, `grid_wmap_lgenerator.py`. All results reported in Sections 6–7 are evaluated over this fixed test set using `evaluate_embedding.py`.

Generation procedure. The generator script reads a plain-text configuration file (`gen.list`)—which specifies output directories, filename templates, and generation rules—and writes individual `.gwlmap` files. Each generation rule is defined by five parameters: $(m, n, \text{rate}\%, \text{blob_count}, \text{count})$, as detailed in Table 1. The `blob_count` parameter partitions obstacles into non-overlapping, convex disk-shaped clusters; compared to uniform random scattering, this clustering prevents isolating free space and significantly improves map generation success at high obstacle densities.

Table 1: Map generation configuration rules and dataset split details.

Split	Resolution ($m \times n$)	Obstacle Density (rate%)	`blob_count`	Map Count (`count`)
Train (Rule 1)	30×30	20%	4	500
Train (Rule 2)	30×30	30%	5	500
Test	30×30	25%	4	50

As shown in Table 1, all $1{,}050$ maps utilize convex blob placement, and the test density (25%) strictly interpolates within the training range (20%–30%).

Map file structure and visualization. Figure 1 illustrates representative maps (`train_map_0001` and `test_map_0001`) stored in `.gwlmap` format. Table 2 summarizes how each component stored in a `.gwlmap` file is utilized during training and evaluation phases.

Table 2: `.gwlmap` file components and their usage across training and test phases.

Component	File Representation	Phase-Specific Usage
Obstacles	Dark regions	Defines spatial connectivity (G_{mp}, G_{bfs}) for all maps
Ground-Truth Field	Color gradient	Stored for both splits; target distance reference (Section 3)
Goal Cell	Star marker (Test only)	**Training:** Discarded (supervised via anchor pairs; Section 4.2) **Test:** Marked and used as the fixed evaluation target (g)

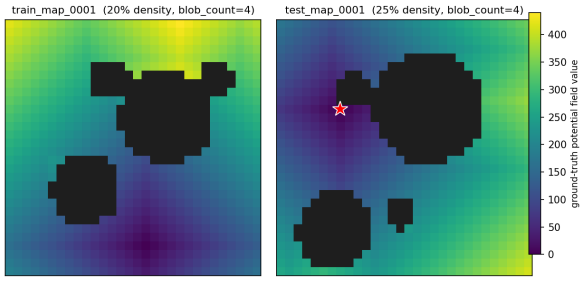

Figure 1: `train_map_0001` and `test_map_0001`, representing the first map of each split, shown on a shared color scale for direct comparison. Color intensity indicates ground-truth potential field values (goal $= 0$; brighter indicates greater distance), while dark regions represent obstacles. The star marks the fixed goal used for test-time evaluation; the training map goal is unmarked because training relies exclusively on anchor pairs. Both maps are 30×30 with blob-placed obstacles at their respective split densities (Section 5.1).

Goal placement, reachability, and usage. Every map is assigned a randomly placed goal cell that is accepted only if *every* non-obstacle cell can reach it via G_{mp} (8-connected path, matching the greedy-descent agent). A diagonal step is disallowed only when *both* adjacent orthogonal cells are obstacles. Maps and goal locations are jointly resampled up to $5{,}000$ times until this full-reachability constraint is satisfied. This strict connectivity requirement ensures that an oracle agent operating on the ground-truth field achieves near-100% completion across maps (Section 3).

Furthermore, this corner-cutting rule guarantees that G_{mp}-reachability implies 4-connected (G_{bfs}) reachability: every valid diagonal edge features at least one open orthogonal neighbor, creating a two-step 4-connected bypass. Consequently, all G_{mp}-reachable cells possess well-defined BFS hop distances without undefined labels.

Goal usage differs between phases: training discards the map-level goal and relies entirely on anchor-target pairs (Section 4.2), whereas test-time evaluation uses the goal directly as the target g described in Section 3. All $1{,}050$ maps are 30×30 grids with densities within $[20\%, 30\%]$. Evaluating out-of-distribution generalization—such as larger grid dimensions or densities outside this range—is reserved for future work.

5.2 Baselines

Destination-conditioned field regression (primary baseline). This model takes the goal cell as an explicit input channel and requires a new forward pass whenever the destination changes. It uses the same GAT backbone and a mono-weighted hinge loss as our approach. We evaluate this baseline over a sweep of loss weights across three random seeds per setting on the same 50 test maps (Table 3, row 9; best setting: `mono_weight`$= 3$).

Table 3: Effect of ranking-loss weight and anchor/pair sampling density, averaged over 50 held-out test maps. Rows 1–3 set anchors×pairs to 8×32; rows 4–7 use 16×64; row 8 increases sampling to 32×128 at rank_weight$= 10$. Bold values mark the best result per column. The decoupling across rows 5 and 7 represents the central finding of this paper (Section 7.1). Oracle completion on the true field is near 100% across maps and is omitted. Row 9 shows the destination-conditioned baseline (Section 5.2).

#	rank_wt	anch.×pairs	MSE↓	Max err↓	Compl.%↑	Viol.%↓
1	1000	8×32	1861.51	125.06	86.02	7.17
2	300	8×32	1101.11	99.25	92.70	6.82
3	100	8×32	987.58	94.27	92.30	6.76
4	100	16×64	887.92	90.90	93.03	5.90
5	10	16×64	774.42	89.29	**93.07**	5.77
6	3	16×64	769.77	89.43	92.41	5.90
7	1	16×64	764.15	89.52	92.71	5.86
8	10	32×128	**746.27**	**89.24**	92.66	**5.71**
9	goal-cond. baseline[†]		578.39	75.85	81.33	5.41

Classical non-learned embeddings (secondary comparison). Non-learned baseline techniques, such as landmark MDS (de Silva & Tenenbaum, 2004) applied to shortest-path distance matrices, are left for future work.

5.3 Metrics

For each test map, we compute the following metrics:

- **MSE**: The mean squared error between predicted and ground-truth BFS distances evaluated over sampled cell pairs.
- **Mean Max Error**: The average maximum per-map prediction error.
- **Completion Rate**: The proportion of greedy-descent navigation runs that successfully reach the target goal.
- **Ranking Violation Rate**: The fraction of same-anchor cell pairs whose relative distance ordering is incorrectly predicted.
- **Oracle Completion Rate**: The success rate achieved by greedy descent on the ground-truth field (near 100% across maps), serving as an empirical upper bound throughout Section 7.

6 Results

Table 3 summarizes the complete hyperparameter sweep across six ranking-loss weights (from 1000 down to 1), averaged over the 50 held-out test maps.

[†]Destination-conditioned GAT regression baseline (Section 5.2) at its best setting (mono_weight$= 3$), averaged over 3 random seeds. Unlike rows 1–8, it requires a new forward pass for each destination goal.

At its best setting, row 9 achieves a *lower* MSE than all destination-agnostic configurations, yet reaches only 81.33% completion compared to 92–93% for rows 4–8. This striking gap provides compelling evidence that regression accuracy does not directly predict downstream navigability. Furthermore, this disconnect is not an artifact of our embedding architecture, as it occurs even in a model trained and evaluated on a per-goal basis.

6.1 Effect of the Ranking-Loss Weight

At 8×32 density (rows 1–3), setting rank_weight $= 1000$ yields worse performance across *all* metrics compared to 300. This supports our hypothesis that a fixed-magnitude hinge gradient can destabilize training when weighted excessively (Section 4.3). Lowering the weight to 100 further reduces MSE and max error while completion rate and ranking violations remain plateaued.

At 16×64 density (rows 4–7), MSE decreases monotonically across the entire sweep, reaching its minimum at rank_weight $= 1$. Conversely, completion rate and ranking violation rates peak at an *intermediate* weight of 10, deteriorate at 3, and only partially recover at 1. This non-monotonic trend contrasts sharply with MSE's monotonic trajectory. Crucially, the configuration that optimizes regression accuracy and the one that optimizes navigability are *distinct parameter settings*, rather than opposing ends of a standard trade-off.

6.2 Effect of Anchor/Pair Density

Comparing rows 3 and 4 at a fixed rank_weight of 100, a $4\times$ increase in sampling density ($8 \times 32 \rightarrow 16 \times 64$) improves performance across all four metrics, most noticeably reducing ranking violations ($6.76\% \rightarrow 5.90\%$). Because forward-pass computation is independent of sampling density (Section 4.2), increasing density offers a computationally efficient performance gain. This suggests that low pair counts (128 pairs/step) produce noisy ranking-loss gradients. However, effective loss weight scaling is closely tied to sampling density, preventing rank_weight from acting as an isolated hyperparameter.

Increasing density by an additional $4\times$ (row 8, 32×128, rank_weight $= 10$) reveals that this benefit eventually plateaus. While MSE and ranking violations continue to improve ($774.42 \rightarrow 746.27$ and $5.77\% \rightarrow 5.71\%$), completion rate slightly degrades ($93.07\% \rightarrow 92.66\%$). This indicates that pushing sampling density too far re-introduces the underlying tension between regression accuracy and navigability.

6.3 Comparison to the Destination-Conditioned Baseline

Row 9 of Table 3 shows the destination-conditioned baseline at its optimal setting (Section 5.2). Despite achieving superior MSE, its completion rate falls significantly short of our proposed framework. Additionally, the two approaches differ fundamentally in computational efficiency at inference time: querying G distinct goals on a single map requires G separate forward passes for the baseline, but exactly 1 forward pass for our destination-agnostic embedding. This computational advantage scales linearly with the number of target queries per map, validating the efficiency argument established in Section 1.

7 Analysis: When Does Regression Accuracy Predict Navigability?

7.1 The MSE–Completion-Rate Decoupling

Across the ranking-loss weight sweep at a fixed sampling density of 16×64, MSE decreases monotonically ($100 \rightarrow 10 \rightarrow 3 \rightarrow 1$) and reaches its minimum at the smallest weight. In contrast, completion rate and ranking-violation rate trace a non-monotonic trajectory that peaks at an intermediate weight of rank_weight $= 10$ (Table 3, rows 4–7). Selecting rank_weight purely to minimize MSE—as a conventional regression pipeline would—yields a configuration with inferior navigability (row 7) compared to a navigability-aware selection (row 5). This provides direct evidence that minimizing mean regression error does not align with optimizing navigability, even though the two metrics move together during earlier parameter regimes (rows 1–4).

Greedy descent requires the potential field to be strictly monotonic along at least one path at *every* visited cell. An averaged regression loss can be reduced by improving predictions on *typical* cell pairs, while simultaneously degrading local gradients along specific navigation trajectories. Consistent with this observation, increasing the pair sampling *density* (Section 6.2) improves navigability alongside MSE reductions. Conversely, varying rank_weight alone under a fixed high density induces divergence between the metrics. This suggests that the ranking loss's pairwise-order supervision provides local structural guarantees that greedy descent relies on more effectively than standard regression loss.

Table 4: Per-map completion rate (%) for four representative maps (`test_map_NNNN`, abbreviated) whose performance runs counter to the aggregate MSE trend across configurations (Table 3, rows 3–7).

	8×32	16×64			
Map (rank_wt→)	100	100	10	3	1
`0018`	65.73	89.66	81.24	79.91	81.24
`0022`	99.85	83.95	83.95	84.24	83.95
`0044`	88.02	76.78	90.38	74.70	74.70
`0041`	73.89	74.93	73.44	75.96	74.93

Table 5: MSE values for three of the four flagged maps (`test_map_NNNN`, abbreviated). Map `0041` represents the dataset's worst MSE outlier across all checked configurations (64–71% higher than the second-worst map, `test_map_0029`). Map `0022`'s completion rate remains flat (0.29-point variation) despite substantial shifts in MSE. "—" indicates unmeasured values.

	8×32	16×64			
Map (rank_wt→)	100	100	10	3	1
`0041`	3210	3473	3634	3725.54	3711.97
`0022`	—	656	517	464.30	474.02
`0044`	1304	1207	1133	1198.31	1159.70

7.2 Failure-Case Studies

Tables 4 and 5 detail four specific maps whose performance trajectories most clearly demonstrate this decoupling across rows 3–7 of Table 3.

Map `test_map_0022` provides the clearest illustration of decoupling: its completion rate stays within a tight 0.29-point range while its MSE varies substantially ($656 \rightarrow 517 \rightarrow 464.30 \rightarrow 474.02$). Its lowest MSE occurs at rank_weight $= 3$, a setting where the *aggregate* corpus MSE has not yet reached its minimum. Thus, MSE improvements on this map yield virtually no benefit to downstream navigability.

Map `test_map_0041` remains the worst MSE outlier across all configurations. Its MSE degrades through rank_weight $= 3$ even as the 50-map corpus average improves. Furthermore, its completion rate remains locked within a narrow 2.5-point band (73.44%–75.96%), echoing `0022`'s pattern of substantial MSE fluctuations with negligible impact on navigation. Structural analysis—including BFS graph diameter, free-cell counts, obstacle density, and 4-connected articulation points—rules out global topological anomalies. Specifically, `0041` lies within 0.6σ of the corpus mean on all global metrics (diameter 430 vs. 447 ± 56; 2 articulation points vs. 1.18 ± 1.41; 674 free cells and 25.1% density).

Visual inspection (Appendix, Figure 2) reveals that `0041`'s goal cell is located in a narrow corridor between two adjacent obstacle clusters. Quantifying local goal crowding (obstacle ratio within a 7×7 window centered on the goal) yields a value of 0.29 for `0041`, higher than `0018` (0.17) or `0044` (0.16). While local crowding likely contributes to navigation failures, it does not represent the corpus's most extreme case and correlates only weakly with MSE overall ($r = 0.25$ across all 50 maps, using the destination-conditioned baseline's `mse_only` values from Figure 2). Finally, maps `test_map_0018` and `0044` display the same qualitative decoupling patterns as the aggregate benchmark, but at larger amplitudes (Table 4).

7.3 Why Symmetric Embeddings Struggle with Greedy Descent

Because ground-truth distances on our undirected grid graph are symmetric, an asymmetric model formulation is not strictly necessary to represent ground-truth labels—distinguishing our setting from scenarios that necessitate quasimetric embeddings (Wang et al., 2023). Instead, the failure mode is structural: while predicted distances $\|\boldsymbol{z}_i - \boldsymbol{z}_g\|_2$ satisfy the triangle inequality by construction, they are not guaranteed to maintain *local* consistency with shortest-path structures. Specifically,

greedy descent requires that every cell along a path to g has at least one G_{mp} neighbor (8-connected) with a strictly smaller predicted distance. While the ground-truth BFS field satisfies this constraint by design, no such local property is guaranteed for learned embeddings.

A global regression loss over randomly sampled pairs regularizes the overall embedding space, but provides only sparse supervision for local monotonicity. Consequently, achieving a low average MSE remains compatible with localized gradient violations in sparsely sampled regions (such as near obstacles, as observed in `test_map_0041`). The ranking loss enforces relative ordering between same-anchor targets, providing supervision closer to local monotonicity. This explains why varying rank_weight primary governs the decoupling phenomenon (Section 7.1), reinforcing the argument that standard parametric distance metrics can fail to guarantee task-level goal reachability even when structurally expressive.

8 Discussion and Limitations

Our experiments are conducted on synthetic grid-world maps using a single greedy-descent agent, rather than a full path planner (e.g., A*) or a non-greedy policy capable of recovering from local field imperfections. Consequently, our generalization findings are restricted to the training map-size and layout distribution (Section 5). This scope affects both the practical reach of our map-reuse argument (Section 1) and the extent to which our comparison against the destination-conditioned baseline (Section 6.3) extends to broader settings.

Furthermore, while the structural explanation in Section 7.3 is plausible, it remains an empirical hypothesis rather than a proven theorem. We note that enforcing local monotonicity directly during training is not unexplored: an earlier GAT variant that explicitly penalized local monotonicity violations achieved only $\sim 65\%$ completion, compared to 92–93% for our primary model. This outcome suggests that the most direct optimization signal does not automatically provide the optimal inductive bias for greedy navigation. Promising directions for future work include using the ranking-violation rate as a *diagnostic metric* rather than a direct loss term, as well as evaluating embeddings with explicitly quasimetric parameterizations.

9 Conclusion

We presented a destination-agnostic graph embedding approach for potential-field navigation that enables a single per-map forward pass to be reused across arbitrary target goals. Our model is trained using a joint objective combining pairwise distance regression and an order-preserving ranking loss. Across a comprehensive hyperparameter sweep over ranking-loss weights ($1000 \to 1$) and anchor/pair sampling densities, we demonstrated that regression accuracy and greedy-navigation success are optimized by *different* parameter settings. Specifically, MSE decreases monotonically and reaches its optimum at the smallest loss weight, whereas completion rate and ranking-violation rate peak at an intermediate weight.

We traced this decoupling phenomenon to specific map instances—including one where completion rate remains within a tight 0.29-point range despite substantial shifts in MSE. This behavior reflects a fundamental structural gap between the global alignment enforced by an averaged regression loss and the local, everywhere monotonicity required by greedy descent. Our findings offer a broader lesson for learned planning representations: regression error alone is an unreliable proxy for downstream task performance, underscoring that future evaluation protocols in this domain should consistently report task-level navigability alongside embedding accuracy.

AI Use Statement

We utilized generative AI tools to refine the grammar, clarity, and readability of the manuscript text, assist in literature retrieval and related work discovery, aggregate results across hyperparameter sweeps, and aid in Python GUI development and map visualization. All AI-assisted text, tables, and figures were thoroughly reviewed and edited by the authors, who take full responsibility for the final content of this manuscript.

Reproducibility statement

Architecture and training-objective details are given in Section 4 (fixed hyperparameters in Section 4.3); data composition and evaluation protocol are given in Section 5; the full hyperparameter sweep is given in Table 3, with remaining experimental gaps listed in the Appendix.

Acknowledgments

References

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations (ICLR)*, 2022.

Vin de Silva and Joshua B. Tenenbaum. Sparse multidimensional scaling using landmark points. Technical report, Stanford University, 2004.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Gi-Yoon Jeon and Jin-Woo Jung. Water sink model for robot motion planning. *Sensors*, 19(6):1269, 2019. doi: 10.3390/s19061269.

Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37, pp. 1312–1320. PMLR, 2015.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, 2015.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *International Conference on Machine Learning (ICML)*, 2023.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pp. 7134–7143. PMLR, 2019.

A Appendix

A.1 Visual Inspection of Flagged Test Maps

Figure 2 shows the four maps discussed in Section 7.2: obstacles in dark gray, the goal marked with a star, background colored by ground-truth BFS hop level from the goal. `test_map_0022` (open goal, no nearby obstacles) is included for contrast with the three flagged maps.

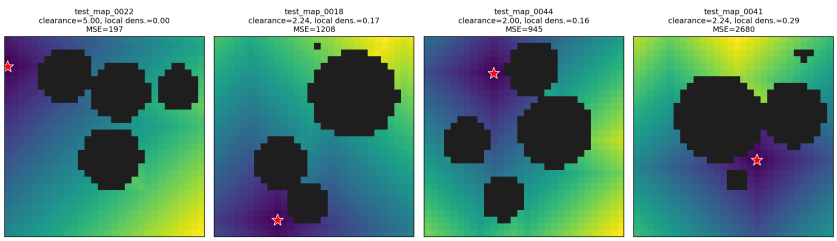

Figure 2: The four maps of Section 7.2. Clearance is the goal's Euclidean distance to the nearest obstacle; local density is the obstacle fraction within a 7×7 window centered on the goal; MSE is from the destination-conditioned baseline's `mse_only` configuration (Section 5.2), used here only as a fixed, single-run per-map reference value for the visual comparison – it is not the main method's MSE and is not the source of Table 5.

`test_map_0044`'s MSE and `test_map_0041`'s completion-rate trajectories are now tabulated in full (Tables 4–5), and the 32×128 anchor-density configuration is now included as row 8 of Table 3 (Section 6.2).

A.2 Implementation and Training Details

Four scripts, built on PyTorch Geometric (Fey & Lenssen, 2019) (its `GATv2Conv` layer, `DataLoader`, and disjoint-union graph batching), implement the destination-agnostic pipeline used throughout the paper. `model_embedding.py` defines `EmbeddingGNN` exactly as in Section 4.1. `dataset_embedding.py` builds, per map, the two graphs of Section 3 directly from the `.gwlmap` obstacle grid – an 8-connected graph for message passing (a diagonal edge is dropped only when *both* orthogonal flanking cells are obstacles, matching Section 5.1) and a 4-connected graph for labels – and implements anchor sampling (Section 4.2): the default `grid` mode places anchors at evenly-spaced points snapped to the nearest free cell, computes each anchor's BFS row once per map and caches it, so only the `pairs_per_anchor` targets are re-drawn every epoch; a `random` anchor mode also exists but was not used for the results in this paper. `train_embedding.py` runs Adam (lr 5×10^{-4}) on the loss of Section 4.3, batching `batch_size`$= 4$ maps per step as one disjoint-union graph, and checkpoints the epoch with the lowest *training* loss (no held-out validation split is used for model selection). `evaluate_embedding.py` performs the single per-map forward pass central to this paper's argument, rescales embedding distances to each map's own `level_step` (training fixes `level_step`$= 1$), and computes all four metrics of Section 5 restricted to cells with a defined ground-truth label.

Every row of Table 3 (rows 1–8) is produced by the same command with only `--rank_weight`, `--num_anchors`, and `--pairs_per_anchor` varied per row (Section 4.3's shared defaults otherwise):

```
python train_embedding.py --map_dir ./data/train
--epochs 200 --batch_size 4 --lr 5e-4 --hidden_dim
64 --embed_dim 16 --num_layers 4 --mse_weight 1.0
--rank_margin 0.5 --level_step 1.0 --anchor_mode grid
--seed 42 --rank_weight {1000,300,100,10,3,1} --num_anchors
{8,16,32} --pairs_per_anchor {32,64,128} --save_path
./checkpoints_embedding/<run_name>/best_model.pt
```

followed by `evaluate_embedding.py --checkpoint .../best_model.pt --eval_dir ./data/test` for each checkpoint against the fixed 50-map test set.