Metric Learning

Dominik Gaweł
University of Warsaw
dg448617@students.mimuw.edu.pl

June 20, 2025

1 Introduction

All data processing and model training was performed in Google Colab. During experimentation, I encountered RAM limitations on the Colab free tier, which required me to force all tensors and model parameters to `float32` precision rather than `float64` to avoid out-of-memory errors. Consequently, I added explicit conversions (e.g. using `.float()` or `.to(torch.float32)`) throughout the code to ensure all data and model weights remain in `float32`. This choice is reflected in every code cell and training pipeline.

2 Hyperparameters

Below is a summary of the key hyperparameters used in all training runs. Where multiple values were tested, the chosen configuration is highlighted.

Parameter	**Tested Values**	**Chosen Value**
Number of training steps	5,000; 10,000; 15,000; 20,000	20,000
Contrastive γ	0.90; 0.98	0.90
Batch size	64	64
Learning rate	1×10^{-3}	1×10^{-3}
# of training trajectories	4,000	4,000
Precision	`float32`	`float32`

Table 1: Overview of hyperparameters tested and selected.

2.1 Temporal Distance Prediction

Normal Dataset

- **Train steps:** 20,000
- **Batch size:** 64
- **Learning rate:** 1×10^{-3}
- # of trajectories: 4,000
- Precision: `float32`

Stitching Dataset

- **Train steps:** 20,000
- **Batch size:** 64
- **Learning rate:** 1×10^{-3}
- # of trajectories: 4,000
- Precision: `float32`

2.2 Contrastive Learning

Normal Dataset

- **Train steps:** 20,000
- **Batch size:** 64
- **Learning rate:** 1×10^{-3}
- # of trajectories: 4,000
- Contrastive discount factor γ: 0.90
- Precision: `float32`

Stitching Dataset

- **Train steps:** 20,000
- **Batch size:** 64
- **Learning rate:** 1×10^{-3}
- # of trajectories: 4,000
- Contrastive discount factor γ: 0.90
- Precision: `float32`

3 Metrics Implementation

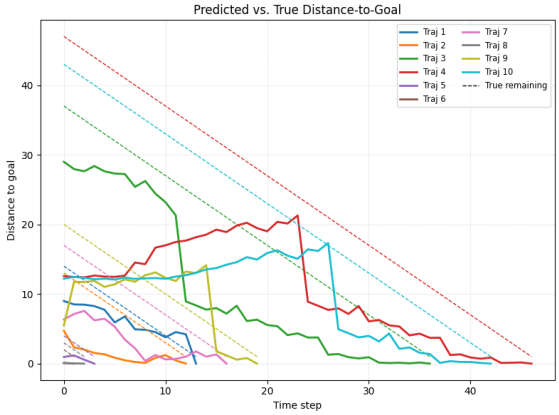
3.1 Correlation

To evaluate how well the model learns to estimate distances to the goal state, we compute the Spearman rank correlation between predicted distances and the true number of remaining steps to the goal for each trajectory.

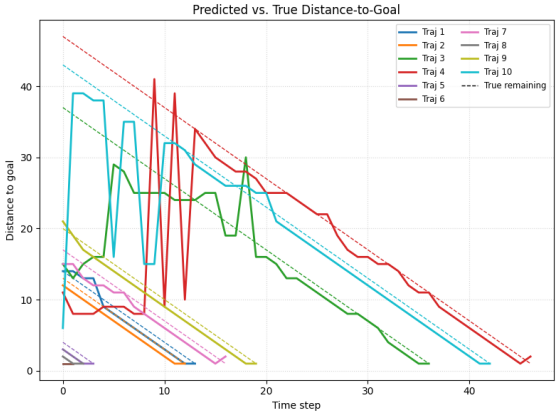
Given a trajectory, the goal state is defined as the final frame in the sequence. The model predicts a distance for each earlier state to that goal. Since the true distance to goal at time t is simply the number of steps remaining $(T - t)$, we can compute the rank correlation between predicted and actual distances.

The correlation is computed for each trajectory separately and then averaged across a sample of trajectories. We used 10 random test trajectories for evaluation. This metric is robust to scale differences and primarily evaluates monotonicity, which aligns well with the goals of representation learning in distance prediction.

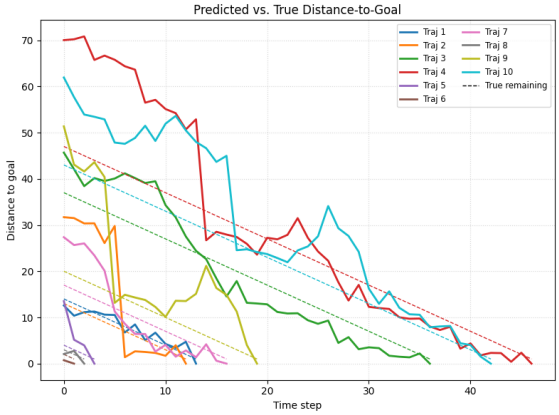
Example plots Below we present predicted distances versus the true remaining steps to the goal for selected trajectories. The solid lines represent the model's predictions, while the dashed lines represent the ground truth. High agreement between the two indicates good learning of temporal structure. All tests were run on the standard test set, regardless of which dataset the model was trained on.



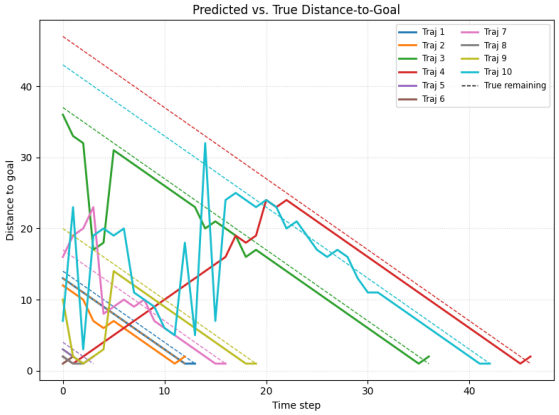
Normal dataset, Contrastive



Normal dataset, Supervised



Stitching dataset, Contrastive



Stitching dataset, Supervised

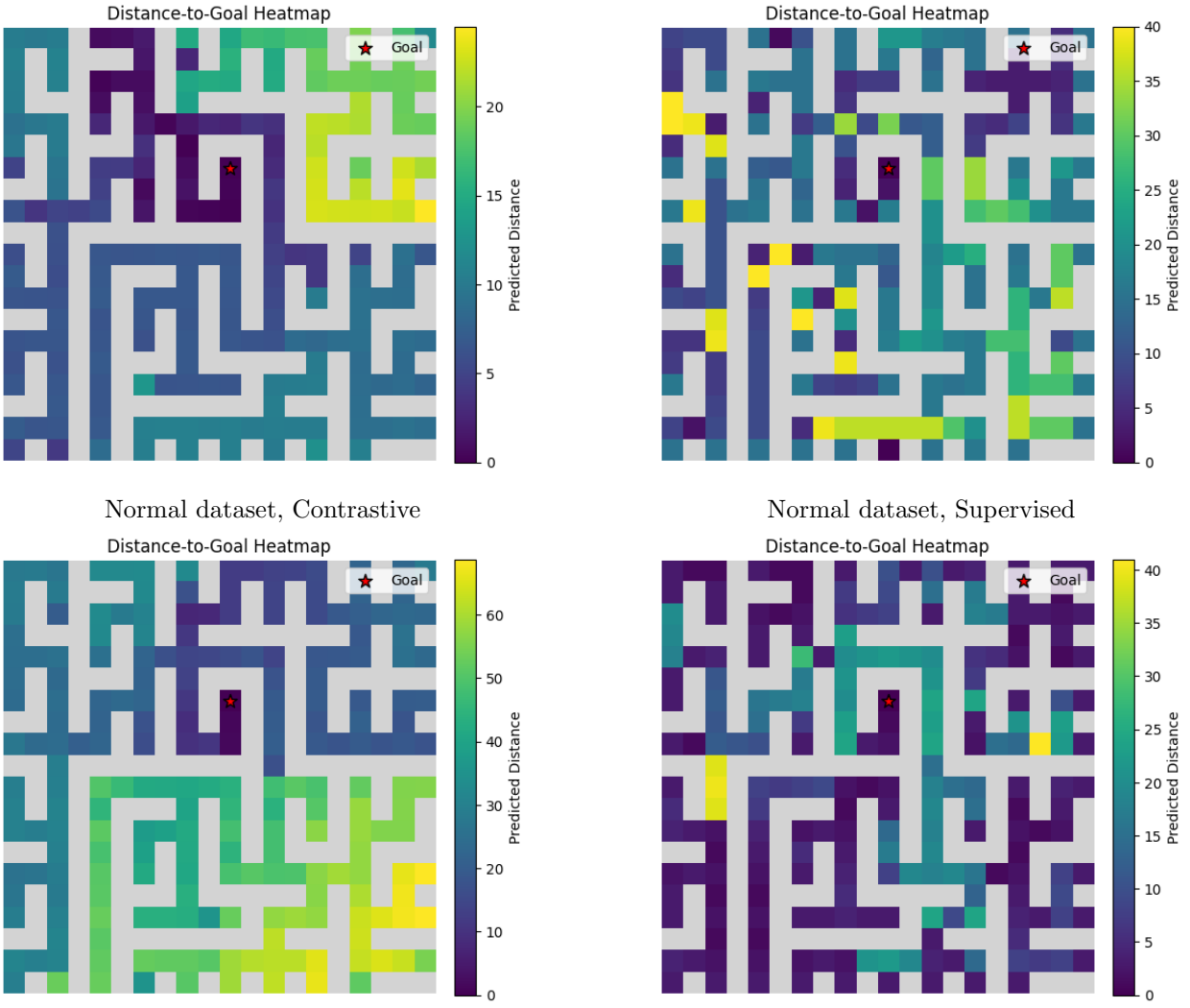
Figure 1: Predicted vs. true distance-to-goal for 10 test trajectories (solid: prediction; dashed: ground truth).

Model	**Dataset**	**Avg. Spearman Correlation**
Temporal Distance Prediction	Normal	0.807
Temporal Distance Prediction	Stitching	0.939
Contrastive Learning	Normal	0.848
Contrastive Learning	Stitching	0.470

Table 2: Average Spearman correlation over 10 test trajectories. Higher is better.

3.2 Heatmaps

The following heatmaps visualize the model's estimated distance from each navigable cell to the goal state in the maze.



Normal dataset, Contrastive

Normal dataset, Supervised

Stitching dataset, Contrastive

Stitching dataset, Supervised

Figure 2: Heatmaps of distance to goal across the maze grid.

3.3 Solved Rate

To assess the practical utility of the learned distance estimator in planning, we randomly select N start–goal pairs on unseen mazes and attempt to reach the goal by expanding states in order of increasing estimated distance. We compare two modes:

- **Search** (A-style): expand all frontier states in a priority queue.
- **Greedy** (No search): at each step, only the single best neighbor is expanded.

For each mode and each dataset (Normal vs. Stitching), we report the fraction of instances solved within the search budget and the average number of states expanded.

Model	Mode	Dataset	Solved Rate	Avg. Expanded Nodes
Contrastive	Search	Normal	1	44.7025
Contrastive	Search	Stitching	1	43.5275
Contrastive	Greedy	Normal	0.1475	9.4275
Contrastive	Greedy	Stitching	0.18	11.485
Supervised	Search	Normal	1	46.735
Supervised	Search	Stitching	1	66.56
Supervised	Greedy	Normal	0.3025	10.9275
Supervised	Greedy	Stitching	0.14	9.2775

Table 3: Solved rate and average number of expanded nodes over N test instances, for both Contrastive and Supervised models, with and without search, on Normal and Stitching datasets.

4 Results

4.1 Supervised Training

For the supervised model, the training loss decreased steadily and reached a slightly lower minimum at 20,000 steps compared to 15,000 steps, indicating modest continued improvement. On the test set, the Spearman correlation between predicted and true distances was 0.807 for the model trained on the Normal Dataset and 0.939 for the model trained on the Stitching Dataset. In heatmap visualizations, the supervised model produces sharply discretized distance contours, accurately capturing small remaining distances near the goal but showing blocky artifacts farther away

4.2 Contrastive Training

The contrastive model also continued to improve up to 20,000 steps, with its final loss marginally better than at 15,000 steps. On the test set, it achieved Spearman correlations of 0.848 (Normal Dataset) and 0.470 (Stitching Dataset), showing good generalization on the original data but struggling on stitching-augmented trajectories. Heatmaps for the contrastive model are much smoother and more continuous, reflecting a more graded embedding of distance in state space.

5 Analysis

5.1 Overall Performance

Both models reliably solve unseen mazes when used in a full search (A-style) planner, achieving 100% solved rate on both Normal and Stitching datasets. However, under a greedy policy that expands only the single best neighbor each step, performance drops sharply: supervised reaches 30.3% on Normal data and 14% on Stitching, while contrastive reaches only 14.8% and 18% respectively. Both models expand a similar number of nodes on average, within the same order of magnitude.

5.2 Supervised vs. Contrastive Comparison

- **Monotonicity (Spearman):** The supervised model achieves higher rank correlation on both datasets (Normal: 0.807 vs. 0.848; Stitching: 0.939 vs. 0.470), reflecting its stronger ability to preserve the ordering of distances.
- **Continuity:** The contrastive model produces smooth, continuous heatmaps that capture global distance gradients, whereas the supervised model's heatmaps are highly discretized and tend to concentrate on very low values.
- **Local Accuracy:** Supervised learning excels at predicting small remaining distances but struggles to generate meaningful values farther away. The contrastive model, by contrast, captures the overall shape of the distance field but lacks precision in exact distance values.

5.3 Stitching Capabilities

We evaluated both models on the original test set after training exclusively on the stitching subset (start and goal both in one quadrant).

- The supervised model maintains a high Spearman correlation (0.939) and a 100% solve rate under full search, but its greedy performance drops sharply to 14%.
- The contrastive model's correlation falls to 0.470, yet its greedy solve rate remains comparatively higher at 18%, and its heatmaps still exhibit smooth gradients.

These results suggest that supervised regression can accurately learn short range distances within a single region but produces less informative distance fields overall, whereas contrastive learning builds a more coherent global representation at the expense of precise numerical accuracy.

6 Conclusion

In this study, we evaluated two approaches to learning distance to goal in maze environments: supervised temporal distance regression and contrastive representation learning. The supervised model excels at producing precise, high correlation estimates for short range distances especially when start and goal lie in the same quadrant resulting in superior greedy policy performance on stitching augmented data. However, its discretized heatmaps reveal limited expressiveness for global structure. In contrast, the contrastive model generates smooth, continuous distance fields that support robust full search planning but lack numerical precision.