In: AIAA Aviation Forum and Ascend 2025, 21-25 July 2025, Las Vegas, USA, Paper number AIAA 2025-3397
DOI:10.2514/6.2025-3397

Instance-Specific Heuristic Learning for Scalable UAV Path Planning

Elie Thellier*, Adolfo Perrusquía†, and Antonios Tsourdos‡
Cranfield University, Bedford, MK43 0AL, United Kingdom

Efficient and scalable path planning is critical for autonomous UAVs navigating complex, obstacle-dense environments. Traditional heuristic search algorithms like A* and Focal Search often face challenges with scalability and adaptability in such scenarios. We present a framework that leverages Transformer-based heuristic learning to predict Path Probability Maps (PPM), which are probabilistic grids that indicate the likelihood of each cell being part of an optimal path from start to goal. This significantly enhances search efficiency and solution quality by guiding the search towards high-probability regions. Trained on diverse motion planning datasets and tailored for UAV-specific challenges, our method reduces computational overhead while maintaining near-optimal path quality. Empirical results demonstrate the framework's effectiveness, solving over 50% of scenarios optimally and reducing node expansions by a factor of 2. Additionally, the framework exhibits robust scalability across varying instance sizes, highlighting the potential of instance-dependent heuristic learning to transform UAV path planning for real-time applications.

I. Introduction

Path planning plays a vital role in enabling autonomous UAVs to navigate efficiently through challenging environments. In real-world applications such as disaster response, environmental monitoring, or urban deliveries, UAVs must generate precise and efficient paths to achieve their objectives while navigating dense and dynamic obstacles. The ability to find paths that minimize travel time, conserve energy, and ensure safety is critical for maximizing operational success.

In scenarios like disaster recovery, UAVs can rapidly locate survivors or assess damage by minimizing unnecessary search and computational overhead [1]. Similarly, optimizing flight paths in urban environments or rugged terrains can significantly extend mission duration by reducing energy consumption [2]. However, achieving such efficiency in real-time remains challenging, especially in large-scale environments.

Conventional search-based approaches, such as Dijkstra's algorithm and A* [3], have long been employed for UAV path planning. While effective in structured settings, these methods often fall short in scalability and adaptability when faced with complex, irregular obstacle distributions or highly dynamic conditions. Their reliance on fixed, instance-independent heuristics—such as straight-line or Manhattan distances—can lead to inefficient routes and excessive computational costs.

To address these challenges, modern methods incorporate learning-based approaches to create adaptive, context-aware heuristics. Deep learning, in particular, has shown promise for generating instance-specific guidance that enhances search efficiency without compromising path quality. Predictive models trained on diverse datasets can capture environmental complexity, enabling UAVs to navigate a wide range of scenarios with minimal reconfiguration.

This work focuses on leveraging a Transformer-based framework to predict Path Probability Maps (PPM), a powerful heuristic representation that assigns a likelihood score to each region, indicating its probability of being part of an optimal path. By integrating PPM into traditional search algorithms, our approach reduces computational overhead while maintaining near-optimal performance. This method is particularly suited for UAV path planning, where scalability, adaptability, and real-time operation are paramount.

A. Related Work

Recent advances in heuristic learning have significantly improved the performance of path planning methods. These approaches enhance traditional search algorithms by incorporating adaptive, instance-specific heuristics predicted using

*MSc Graduate in Applied Artificial Intelligence, Cranfield University.
†Lecturer in Reinforcement Learning for Engineering, Cranfield University.
‡Head of the Autonomous and Cyber-Physical Systems Centre, Cranfield University.

Published by AIAA. This is the Author Accepted Manuscript issued with: Creative Commons Attribution License (CC:BY 4.0).
The final published version (version of record) is available online at DOI:10.2514/6.2025-3397. Please refer to any applicable publisher terms of use.

machine learning models. Wang et al. [4] categorize heuristic learning into three main strategies: generating path probability maps, cost-to-go regression, and self-supervised learning.

The path probability map (PPM) approach assigns a probability to each region in the environment, representing its likelihood of being part of an optimal path between a start and goal. Traditional search methods, like A*, can then be guided by these probabilities to focus computations on high-likelihood regions. For instance, Kulvicius et al. [5] utilized Fully Convolutional Neural Networks (FCNNs) to produce PPMs, enabling efficient multi-agent path planning. Their method achieved paths within 10% of optimal length in the vast majority of cases, demonstrating a notable runtime improvement over standard A* for complex scenarios. However, scalability to larger maps remained a limitation. Extending this work, Bian et al. [6] introduced an end-to-end framework using FCNNs to predict both shortest and lowest-cost paths, achieving significant performance gains in runtime and path quality. Despite these advances, their approach was confined to static environments with fixed start and end points. Ferreira et al. [7] further demonstrated the potential of neural encoders in reducing dimensionality, enabling faster planning in dynamic settings. Their work decreased execution time across multiple algorithms by over 50%, showcasing the adaptability of neural-guided pathfinding.

Other learning-based strategies, such as cost-to-go regression and self-supervised learning, focus on different aspects of heuristic improvement. Cost-to-go regression predicts the estimated cost of reaching the goal from any given state and uses it as a heuristic function in traditional algorithms. Takahashi et al. [8], for example, employed deep neural networks to refine A*'s heuristics, leading to fewer node expansions and better scalability. Meanwhile, self-supervised learning methods such as Neural A* [9] and imperative learning-based A* (iA*) [10] reformulate path planning as a differentiable process, enabling continuous refinement of heuristics based on observed outcomes. These methods reduce reliance on extensive labeled data and have shown promise in dynamic and adaptive environments.

Our approach is rooted in the path probability map methodology and draws inspiration from the work of Kirilenko et al. [11], who introduced TransPath to address the inefficiencies of static heuristics in search-based algorithms. TransPath leverages attention-based neural networks to predict PPMs, guiding search efforts to high-probability regions while maintaining computational efficiency. This framework achieved near-optimal solutions with significantly fewer node expansions in diverse environments. Building on this foundation, our work adapts PPM-based methods to the specific challenges of UAV navigation, emphasizing scalability and real-time performance in complex, obstacle-rich scenarios.

B. Original Contributions

This study builds upon the principles of instance-dependent heuristic learning, extending their application to UAV path planning in complex, obstacle-dense environments. Our contributions address the limitations of existing methods in scalability, adaptability, and computational efficiency:

1) Augmented Dataset for UAV Navigation: We develop a specialized dataset with a custom size distribution tailored to UAV applications. This dataset enhances scalability and ensures the framework can generalize effectively across diverse environment sizes.
2) Instance-Specific Heuristic Prediction via Path Probability Maps (PPMs): We focus on leveraging PPMs to guide search algorithms, enabling more accurate and efficient heuristic estimation. This method reduces unnecessary expansions while maintaining high path quality, optimizing performance for UAV-specific challenges.
3) Comprehensive Empirical Validation: We demonstrate the scalability and efficiency of our framework through extensive experiments. The results show its ability to generalize across variable map sizes and its suitability for real-time UAV navigation by consistently achieving near-optimal paths with significantly reduced computational effort.

This paper is structured as follows: We first review foundational concepts and related work. Then, we detail the methods and framework introduced in this research. The subsequent sections describe experimental setups, results, and analyses. Finally, we conclude with a summary of our findings and potential avenues for future research in UAV path planning.

II. Preliminaries

This section outlines the technical definitions and formulations, adapted to the context of UAV path planning and the use of Path Probability Maps (PPM).

A. Path Planning Problem

The objective of path planning for an autonomous UAV in a complex, obstacle-dense environment is to find a safe and efficient path from a start position to a goal position while avoiding obstacles. We model the environment as a 2D grid map, where each cell represents a discrete region that can either be free or occupied by an obstacle. The UAV is constrained to move between adjacent free cells, with no diagonal movement allowed. Each move from one free cell to another incurs a unit movement cost.

A path $\pi(start, goal)$ is defined as a sequence of adjacent cells starting from the $start$ position and ending at the $goal$ position. A path is considered valid if all cells in the path are free of obstacles. The cost of a path is the sum of the movement costs between consecutive cells.

The optimal path π^* is the path that minimizes the total movement cost among all valid paths. A sub-optimal path π^w is a path whose cost is at most w times that of the optimal path, where $w > 1$ represents the sub-optimality factor.

In this work, we aim to solve the path planning problem by using heuristic-based search algorithms. Our approach leverages Path Probability Maps (PPM) to predict the likelihood that a grid cell is part of the optimal path. This predictive map, generated through Transformer-based heuristic learning, helps guide the search algorithm towards high-probability regions, reducing computational effort while maintaining near-optimal path quality.

Thus, solving the path planning problem in this context involves utilizing the PPM to efficiently predict path likelihoods, guiding the UAV to find either the optimal or a bounded sub-optimal valid path for a given environment.

B. Focal Search Algorithm

The Focal Search (FS) algorithm, introduced by Pearl and Kim [12], is an improvement upon the traditional A* algorithm, designed to enhance search efficiency in path planning. While A* expands nodes based on the f-value, defined as $f(n) = g(n) + h(n)$, where $g(n)$ is the cost to reach node n from the start and $h(n)$ is the heuristic estimate to reach the goal from n, Focal Search introduces a more focused approach to node expansion.

In A*, nodes are expanded based on the lowest f-value, with each node being evaluated independently. This can lead to the expansion of a large number of nodes, especially in large, complex environments. To address this, Focal Search maintains a secondary list of nodes, called FOCAL, which includes only the nodes whose f-values are within a factor ϵ of the smallest f-value found during the search. This reduces the number of nodes expanded at any given time, thereby improving search efficiency for a trade off in path optimality.

1. Focal Search Process

The Focal Search algorithm works by iteratively expanding nodes from the FOCAL list until either the goal is reached, and the path is reconstructed, or no path can be found. As presented in Algorithm **??**, the steps involved in the process are as follows:

1) Node Selection: At each iteration, Focal Search selects the node with the smallest f-value from the FOCAL list for expansion.
2) FOCAL List Management: Nodes are added to the FOCAL list if their f-values fall within the range defined by the smallest f-value encountered so far, adjusted by a factor ϵ. The value of ϵ determines how many nodes are kept in the search frontier. A smaller ϵ restricts the search to a narrower set of candidates, while a larger ϵ allows for broader exploration.
3) Secondary Heuristic: A secondary heuristic, h_{FOCAL}, is introduced to further prioritize nodes within the FOCAL list which improves the focus of the search by considering nodes that are more likely to lead to the goal efficiently.

The primary advantage of Focal Search over A* is that it reduces the number of node expansions, making the algorithm more efficient, especially in large search spaces. This is achieved by limiting the search to the most promising areas, as defined by the FOCAL list. The introduction of the parameter ϵ provides a flexible trade-off between efficiency and completeness, allowing for faster solutions while maintaining bounded sub-optimality. A value of $\epsilon = 1.25$ is used in this work. Additionally, the secondary heuristic h_{FOCAL} prioritizes nodes based on their likelihood of leading to an optimal path, which further refines the search process. In this study, the Path Probability Map (PPM) is employed as a key component for h_{FOCAL}, guiding the algorithm by predicting the probability of each cell being part of the optimal path, thus enhancing the efficiency and scalability of the search process.

Algorithm 1 Focal Search Algorithm

Input: map, $start$, $goal$, h, h_{FOCAL}, ϵ
Output: Valid Path π
Initialize OPEN, CLOSED, and FOCAL
while OPEN $\neq \emptyset$ **do**
  sort FOCAL by h_{FOCAL}
  $n \leftarrow$ node in FOCAL or OPEN with lowest f-value
  remove n from OPEN and FOCAL, add to CLOSED
  if f_{min} has changed **then**
    update FOCAL using ϵ
  end if
  if $n = goal$ **then**
    reconstruct and **return** path π
  end if
  for n' in validNeighbors(map, n) **do**
    if $n' \in$ CLOSED **then**
      continue
    end if
    if $g(n') > g(n) + cost(n, n')$ **then**
      $f(n') \leftarrow g(n') + h(n', goal)$
      update or insert n' in OPEN
      if $f(n') \leq \epsilon \cdot f_{min}$ **then**
        update or insert n' in FOCAL
      end if
    end if
  end for
end while
return failure

C. Heuristics

Heuristics play a crucial role in guiding the search process during path planning. A heuristic function provides an estimate of the cost to reach the goal from a given node. In pathfinding algorithms like Focal Search, heuristics help prioritize which nodes to expand by predicting their potential contribution to finding an optimal path.

A good heuristic helps the algorithm make efficient progress towards the goal while maintaining the possibility of finding an optimal solution. For a heuristic to be admissible, it must never overestimate the true cost to reach the goal. A consistent heuristic also satisfies the triangle inequality, ensuring that the heuristic value between any two nodes is always less than or equal to the sum of the heuristic values of the individual segments between them. Both admissibility and consistency help the algorithm remain efficient and optimal.

1. Primary Heuristic: Manhattan Distance

In grid-based path planning, a common instance-independent heuristic is the Manhattan distance. This heuristic calculates the sum of the horizontal and vertical distances between a node and the goal, effectively estimating the minimum number of steps required if there were no obstacles. While easy to compute, the Manhattan distance does not account for obstacles in the environment, and as a result, it can lead to over-expansions, especially in cluttered or obstacle-dense environments.

2. Secondary Heuristic: Path Probability Map (PPM)

To improve the accuracy and efficiency of pathfinding, we introduce the Path Probability Map (PPM), a secondary, instance-dependent heuristic used in the Focal Search algorithm. The PPM, illustrated in Figure 1 (b), represents the likelihood that each grid cell is part of the optimal path between the start and goal. It is generated by predicting the areas that are most likely to contribute to the shortest path, allowing the search algorithm to focus on the most promising regions.

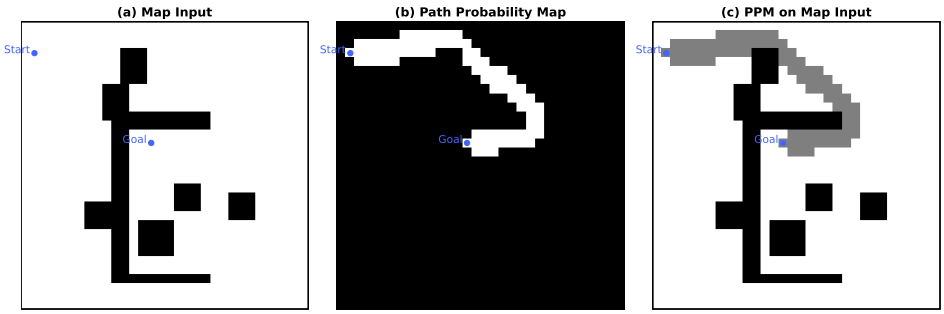


Fig. 1 Illustration of the Path Probability Map.

III. Methods

A. Overview

The search framework is composed of a neural network designed to predict the Path Probability Map (PPM) heuristic, and the Focal Search algorithm. Two neural networks are trained using standard supervised learning on two distinct datasets: Motion Planning (MP) dataset and Tiled-MP dataset. This dual-training approach is of interest when undergoing evaluation to assess the framework's scalability capabilities.

B. Datasets

This study uses two publicly available path planning datasets with obstacle annotations:

1) MP Dataset, introduced by Bhardwaj et al. in [13],
2) Tiled-MP Dataset, presented by Yonetani et al. in [9].

1. Augmented MP Dataset

The MP dataset is widely used for evaluating motion planning algorithms and has been applied in numerous studies, including [14–17]. This dataset contains eight distinct types of environments with different obstacle layouts, as illustrated in Figure 2. These environments represent a variety of challenges that a UAV can encounter, such as maze-like obstacles, gaps, and bug traps, each requiring different strategies for efficient pathfinding.

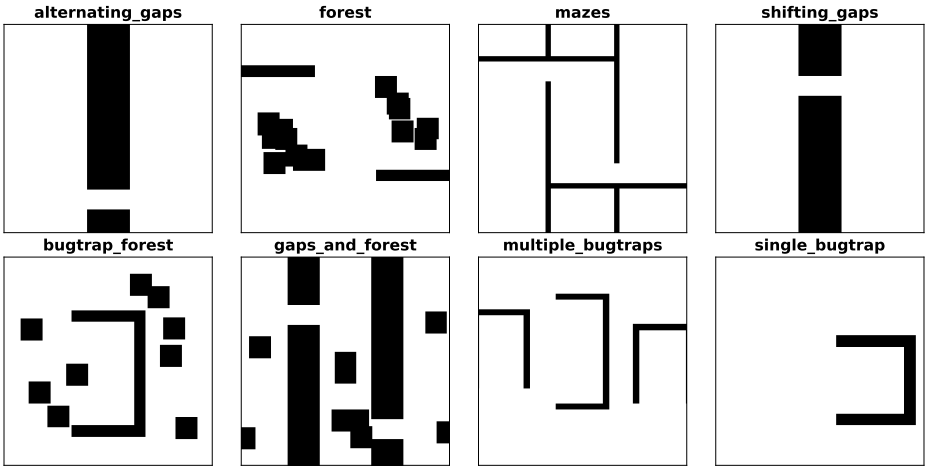


Fig. 2 MP dataset: Type of map.

The dataset contains 1,000 maps per environment type, each with the same obstacle type but different layouts. These maps are resized to 32x32 pixels and divided into training, validation, and test sets in an 8:1:1 ratio. To expand the training set, four augmentation techniques were applied (mirroring and rotation), resulting in 32,000 training samples.

For each map, 10 instances were created by randomly placing start and goal points at least 30 pixels apart, yielding 400,000 total instances: 320,000 for training, 40,000 for validation, and 40,000 for testing.

2. Tiled-MP Dataset

The Tiled-MP dataset extends the MP dataset by combining smaller maps into larger ones, increasing the variety of obstacle structures and map sizes. Maps of 32x32, 64x64, 96x96, 128x128, and 256x256 pixels are created by tiling 1, 4, 9, 16, and 64 smaller maps, respectively (see Figure 3).

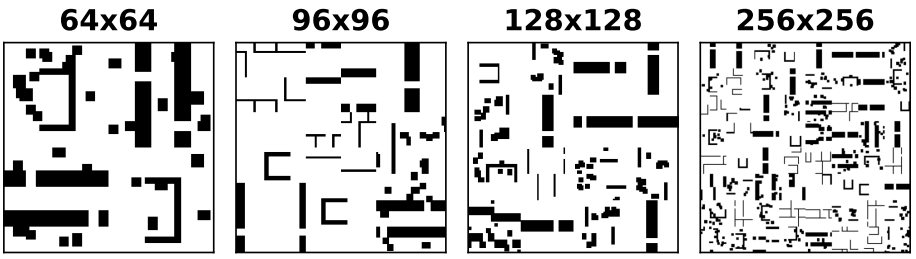


Fig. 3 Tiled-MP dataset: Different map sizes.

For each map, five instances were generated by placing the start and goal points half the map size apart. This resulted in the distribution shown in Table 1.

Table 1 Size distribution in the Tiled-MP dataset.

Size	Training	Validation	Test
32x32	160,000	20,000	312
64x64	40,000	5,000	312
96x96	17,777	2,222	312
128x128	10,000	1,250	312
256x256	2,500	312	312

3. Ground Truth Map Generation

For both datasets, Path Probability Maps (PPM) are generated to indicate the likelihood of each cell being part of an optimal path. This is done using the Theta* algorithm, which computes paths as a series of waypoints. These waypoints are connected using Bresenham's line algorithm, and the path is thickened to ensure robustness, as shown in Figure 4.

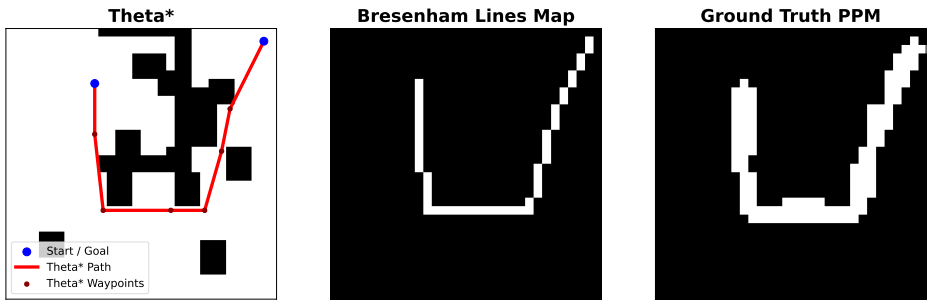


Fig. 4 Generating ground truth path probability maps.

4. Instance Complexity

The Instance Complexity (IC) metric measures the difficulty of a pathfinding instance. It is the ratio of the length of the optimal path to the Manhattan distance heuristic. An IC value close to 1 suggests an easy instance, while higher values indicate more complex instances due to obstacles.

$$IC = \frac{h^*(\text{start})}{h(\text{start})} \tag{1}$$

where $h^*(\text{start})$ is the shortest path and $h(\text{start})$ is the Manhattan distance. Figure 5 illustrates different instance complexities.

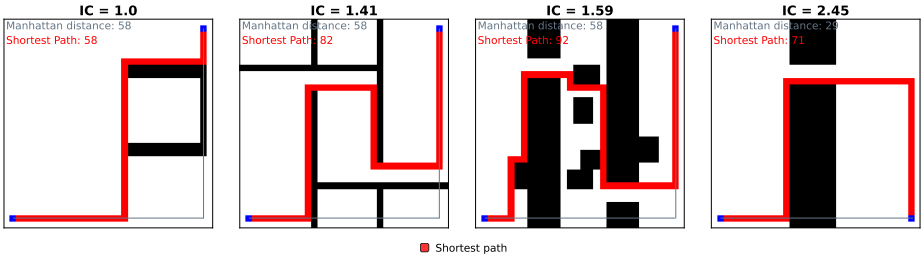


Fig. 5 Illustration of instance complexity.

C. Model Design

1. Architecture

The architecture, TransAE (Transformer Autoencoder), for learning the Path Probability Map (PPM), adapted from [11], consists of a convolutional encoder, spatial transformer blocks, and a convolutional decoder, as shown in Figure 6. The instance, composed of a binary image of the map input and a matrix marking the start and goal positions, is given to the encoder to capture local features such as obstacles and pathways. The spatial transformer blocks, based on self-attention, model global relationships, linking start and goal positions. The decoder reconstructs the transformed feature maps into the final PPM output.

With 871,745 parameters and a trained model size of 3,400 KB, TransAE is efficient and lightweight, making it well-suited for constrained environments.

2. Training Setup

Two training setups were used for the two PPM models trained either on MP or Tiled-MP dataset. For the MP TransAE model, training was performed for 50 epochs using the Adam optimizer [18], with a batch size of 64 and a peak learning rate of 4×10^{-4}. The Tiled-MP TransAE model was trained with the same optimizer, but for 20 epochs and with a batch size of 1 due to the varying input sizes. The OneCycleLR scheduler was used in both setups to adjust the learning rate.

The performance was assessed using a Masked MSE loss function, which computes the mean squared error (MSE) over the free cells of the map, ignoring obstacles. This provides a more accurate evaluation for predicting PPM values.

IV. Experiments and Results

A. Planner and Baseline

To evaluate the proposed heuristic combinations, we developed the **FS+MAN+PPM** planner, which integrates PPM with the Focal Search algorithm, using Manhattan distance as the primary heuristic and the predicted PPM as a secondary heuristic. The parameters for the Focal Search algorithm were set with $\epsilon = 1.25$ to ensure a balance between path length and computational cost.

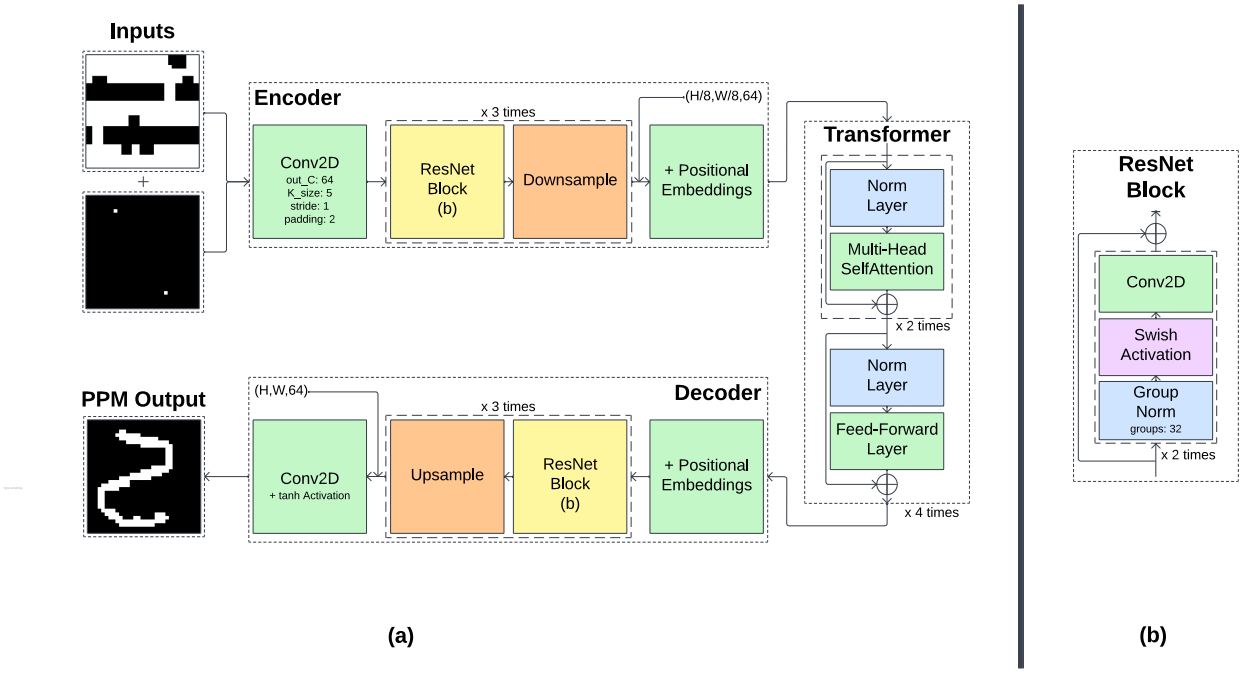


Fig. 6 Overview of the neural network architecture. Adapted from [11].

The performance of this planner is compared against the following baseline: A*+Manhattan: The standard A* algorithm using the Manhattan distance heuristic, which guarantees optimal paths but does not account for obstacles.

B. Evaluation Metrics

To compare the performance of the proposed planners, we use standard metrics that assess path optimality, efficiency, and computational performance:

1. Optimality Ratio (OR)

The OR (2) measures the percentage of instances where the planner finds the optimal path. A higher OR indicates better performance.

$$\text{OR} = 100 \times \frac{N_{opt}}{N} \quad (\%) \tag{2}$$

Where N_{opt} is the number of optimal paths and N is the total number of instances.

2. Path Length Ratio (PLR)

The PLR (3) evaluates how close a planner's path cost is to the optimal cost. A lower PLR is preferred for more efficient paths.

$$\text{PLR} = 100 \times \frac{\text{cost}(\pi)}{\text{cost}(\pi^*)} \quad (\%) \tag{3}$$

Where π is the planner's path and π^* is the optimal path.

3. Expansions Ratio (ER)

The ER (4) compares the number of node expansions to A* with the Manhattan heuristic. A lower ER indicates better efficiency.

$$\text{ER} = 100 \times \frac{E}{E_{Man}} \quad (\%) \tag{4}$$

Where E is the number of node expansions and E_{Man} is the expansions with the Manhattan heuristic.

4. Timing Metrics

- Search Time: The time required for the planner to solve the problem, typically correlated with node expansions.
- Inference Time: The fixed time taken by the neural network to generate heuristic maps.

C. Evaluation Procedure

To evaluate the planner across two test sets: MP (for efficiency) and Tiled-MP (for scalability), we follow these steps:

1) Model and Dataset Selection: Choose the trained TransAE model from either the MP or Tiled-MP dataset, and select a test dataset (MP or Tiled-MP).
2) Instance Evaluation:
    - For each test instance, record the ground truth path length, expansions, and search time using a baseline planner.
    - Compute the instance complexity (IC) and retain instances where $IC \geq 1.05$, eliminating trivial instances.
    - Use the TransAE model to predict the PPM map and record the inference time.
    - Combine the predictions and run the search algorithm, recording the resulting path length, expansions, and search time.
3) Metric Calculation: Compute Optimality Ratio (OR), Path Length Ratio (PLR), and Expansions Ratio (ER) to evaluate planner performance.
4) Performance Analysis: Perform statistical analysis to calculate the mean and standard deviation of the metrics for varying instance complexities.

D. Results on MP test set

The experimental results on the MP test set, shown in Table 2, highlight the comparative performance of the FS+MAN+PPM planner and the baseline A*+MAN planner. The A*+MAN planner, as expected, served as a benchmark with its standard heuristic guaranteeing optimal paths and full exploration.

In contrast, the FS+MAN+PPM planner demonstrated a distinct trade-off between solution quality and computational efficiency. When trained on the Tiled-MP dataset, FS+MAN+PPM achieved the highest performance on the test set, surpassing the same planner trained on the MP dataset in both efficiency and accuracy. Specifically, FS+MAN+PPM with Tiled-MP training exhibited a higher overall optimality ratio, slightly improved path length consistency, and reduced search effort compared to its MP-trained counterpart. This indicates that the diversity and structure of the Tiled-MP dataset enable the model to generalize better to the MP test scenarios.

While A*+MAN consistently achieved the best solution quality, the computational cost was significantly higher due to exhaustive node expansions. The FS+MAN+PPM planner, leveraging learned heuristics, drastically reduced the number of expanded nodes while maintaining near-optimal paths. This efficiency highlights the planner's scalability, particularly when trained on datasets that capture a broader range of problem complexities, such as Tiled-MP.

Table 2 Experimental results on the MP test set.

Planners	Train DS	OR (%)	PLR (%)	ER (%)
A*+MAN		100	100	100
FS+MAN+PPM	MP	44.75	103.17 ± 4.36	48.14 ± 27.23
	Tiled-MP	**54.30**	**102.52** ± **3.90**	**47.96** ± **26.48**

E. Results on Tiled-MP test set

The results on the Tiled-MP test set, presented in Table 3, highlight the challenges of solving larger, more complex problem instances. The A*+MAN planner, serving as a baseline, provided optimal paths while exploring the full search space, but at a significant computational cost. This exhaustive search guarantees high solution quality but becomes less practical as problem complexity increases.

The FS+MAN+PPM planner demonstrated its ability to address these challenges by significantly reducing computational effort while maintaining competitive performance. When trained on the Tiled-MP dataset, the planner achieved notable improvements over its MP-trained counterpart. Specifically, FS+MAN+PPM with Tiled-MP training achieved a much higher optimality ratio and lower search effort, making it better suited for handling the increased complexity of Tiled-MP scenarios. This performance underscores the importance of training data diversity and structure in enabling the model to generalize effectively.

A key takeaway from these results is the scalability of the FS+MAN+PPM approach. The Tiled-MP dataset, with its diverse and larger problem instances, enabled the model to learn heuristics that are more adaptable to complex environments. This adaptability was particularly evident in the reduced number of expanded nodes during search, highlighting the planner's ability to handle problem instances with greater computational efficiency.

Overall, the FS+MAN+PPM planner trained on the Tiled-MP dataset demonstrated strong scalability, balancing computational efficiency with path quality. This makes it a compelling choice for UAV path planning in environments where complexity and size pose significant challenges to traditional exhaustive search methods.

Table 3 Experimental results on the Tiled-MP test set.

Planners	**Train DS**	**OR (%) ↑**	**PLR (%) ↓**	**ER (%) ↓**
A*+MAN		100	100	100
FS+PPM+MAN	MP	12.99	132.38 ± 27.76	104.78 ± 103.37
	Tiled-MP	**39.64**	**106.92 ± 10.90**	**49.54 ± 42.69**

F. Runtime Analysis

This section evaluates the runtime performance of the FS+MAN+PPM planner, focusing on the balance between search and inference times for the PPM model trained on the Tiled-MP dataset.

On the MP dataset, the average inference time for the PPM model was 13.7 ms, measured on an NVIDIA GeForce GTX 1050 Max-Q Design GPU, while the average search time was 27.31 ms. On the Tiled-MP dataset, the average inference time increased slightly to 15.2 ms due to greater variability in problem instance sizes. Despite this, the planner demonstrated scalability, as the PPM model predicts outputs in constant time regardless of input size. This property is crucial since search times grow exponentially with problem complexity.

A comparison of inference and search contributions to the total runtime shows that search time dominates as instance complexity increases. While inference time remains nearly constant, search time escalates with longer paths, becoming the primary runtime factor. For paths exceeding 30 nodes, search time accounts for the majority of the total runtime. For paths longer than 170 nodes, inference time drops below 10% of the total runtime and converges to about 2.5% for very long paths.

This analysis underscores the scalability of PPM as a heuristic predictor in the FS planner. The constant inference time ensures that the total runtime primarily depends on the search process, enabling the planner to efficiently handle increasingly complex instances without significant overhead.

V. Conclusion

This research focused on addressing the challenges of scalable and efficient path planning for autonomous UAVs by integrating instance-specific heuristic learning with traditional search algorithms. Our framework employed transformer-based autoencoders (TransAE) to predict Path Probability Maps (PPM) as heuristics, enhancing the performance of the Forward Search (FS) algorithm in environments represented by known datasets.

Experiments conducted on the Motion Planning (MP) and Tiled-MP datasets demonstrated the effectiveness of this approach. By leveraging PPM predictions, the FS algorithm achieved significant reductions in node expansions while

maintaining high-quality paths. The framework exhibited strong scalability, effectively handling increasingly complex instances without substantial runtime overhead.

While the results are promising, certain limitations remain. The reliability of PPM predictions may decrease for highly complex environments within the datasets, leading to suboptimal paths in some cases. Moreover, the performance variability across problem instances suggests the need for further refinement of the heuristic prediction model to ensure consistent efficiency.

Future improvements include optimizing the TransAE model, extending to 3D environments, integrating safety constraints, exploring self-supervised learning, enhancing PPM for complex cases, testing alternative planners and heuristics, and validating in real-world scenarios.

References

[1] Leon, F., and Atanasiu, G. M., *Probabilistic Path Finding Method for Post-Disaster Risk Estimation*, Springer International Publishing, Cham, 2014, pp. 61–76. https://doi.org/10.1007/978-3-319-00458-7_5, URL https://doi.org/10.1007/978-3-319-00458-7_5.

[2] Na, Y., Li, Y., Chen, D., Yao, Y., Li, T., Liu, H., and Wang, K., "Optimal Energy Consumption Path Planning for Unmanned Aerial Vehicles Based on Improved Particle Swarm Optimization," *Sustainability*, Vol. 15, No. 16, 2023. https://doi.org/10.3390/su151612101, URL https://doi.org/10.3390/su151612101.

[3] Hart, P. E., Nilsson, N. J., and Raphael, B., "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, 1968, pp. 100–107. URL https://doi.org/10.1109/TSSC.1968.300136.

[4] Wang, J., Zhang, T., Ma, N., Li, Z., Ma, H., Meng, F., and Meng, M. Q.-H., "A survey of learning-based robot motion planning," *IET Cyber-Systems and Robotics*, Vol. 3, No. 4, 2021, pp. 302–314. URL https://doi.org/10.1049/csy2.12020.

[5] Kulvicius, T., Herzog, S., Lüddecke, T., Tamosiunaite, M., and Wörgötter, F., "One-Shot Multi-Path Planning for Robotic Applications Using Fully Convolutional Networks," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1460–1466. https://doi.org/10.1109/ICRA40945.2020.9196719, URL https://doi.org/10.1109/ICRA40945.2020.9196719.

[6] Bian, T., Xing, Y., and Zolotas, A., "End-to-End One-Shot Path-Planning Algorithm for an Autonomous Vehicle Based on a Convolutional Neural Network Considering Traversability Cost," *Sensors*, Vol. 22, No. 24, 2022. https://doi.org/10.3390/s22249682, URL https://doi.org/10.3390/s22249682.

[7] Ferreira, J., Júnior, A. A., Galvão, Y. M., Barros, P., Fernandes, S. M. M., and Fernandes, B. J., "Performance Improvement of Path Planning algorithms with Deep Learning Encoder Model," *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, IEEE, 2020, pp. 1–6. URL https://doi.org/10.48550/arXiv.2008.02254.

[8] Takahashi, T., Sun, H., Tian, D., and Wang, Y., "Learning heuristic functions for mobile robot path planning using deep neural networks," *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 29, 2019, pp. 764–772. URL https://doi.org/10.1609/icaps.v29i1.3545.

[9] Yonetani, R., Taniai, T., Barekatain, M., Nishimura, M., and Kanezaki, A., "Path Planning using Neural A* Search," *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 139, edited by M. Meila and T. Zhang, PMLR, 2021, pp. 12029–12039. URL https://doi.org/10.48550/arXiv.2009.07476.

[10] Chen, X., Yang, F., and Wang, C., "iAΘ: Imperative Learning-based AΘ Search for Pathfinding," *arXiv preprint arXiv:2403.15870*, 2024. URL https://doi.org/10.48550/arXiv.2403.15870.

[11] Kirilenko, D., Andreychuk, A., Panov, A., and Yakovlev, K., "Transpath: Learning heuristics for grid-based pathfinding via transformers," *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37, 2023, pp. 12436–12443. URL https://doi.org/10.1609/aaai.v37i10.26465.

[12] Pearl, J., and Kim, J. H., "Studies in semi-admissible heuristics," *IEEE transactions on pattern analysis and machine intelligence*, , No. 4, 1982, pp. 392–399. URL https://doi.org/10.1109/tpami.1982.4767270.

[13] Bhardwaj, M., Choudhury, S., and Scherer, S., "Learning Heuristic Search via Imitation," , 2017. URL https://doi.org/10.48550/arXiv.1707.03034.

[14] Wang, J., Chi, W., Li, C., Wang, C., and Meng, M. Q.-H., "Neural RRT*: Learning-Based Optimal Path Planning," *IEEE Transactions on Automation Science and Engineering*, Vol. 17, No. 4, 2020, pp. 1748–1758. https://doi.org/10.1109/TASE.2020.2976560, URL https://doi.org/10.48550/arXiv.2309.14595.

[15] Zhang, T., Wang, J., and Meng, M. Q.-H., "Generative Adversarial Network Based Heuristics for Sampling-Based Path Planning," *IEEE/CAA Journal of Automatica Sinica*, Vol. 9, No. 1, 2022, pp. 64–74. https://doi.org/10.1109/JAS.2021.1004275, URL https://doi.org/10.1109/JAS.2021.1004275.

[16] Ma, N., Wang, J., Liu, J., and Meng, M. Q.-H., "Conditional Generative Adversarial Networks for Optimal Path Planning," *IEEE Transactions on Cognitive and Developmental Systems*, Vol. 14, No. 2, 2022, pp. 662–671. https://doi.org/10.1109/TCDS.2021.3063273, URL https://doi.org/10.1109/TCDS.2021.3063273.

[17] Ariki, Y., and Narihira, T., "Fully Convolutional Search Heuristic Learning for Rapid Path Planners," *ArXiv*, Vol. abs/1908.03343, 2019. URL https://doi.org/10.48550/arXiv.1908.03343.

[18] Kingma, D. P., and Ba, J., "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. https://doi.org/10.48550/arXiv.1412.6980, URL https://doi.org/10.48550/arXiv.1412.6980.



https://dspace.org/

Instance-specific heuristic learning for scalable UAV path planning

Thellier, Elie; Perrusquía, Adolfo; Tsourdos, Antonios

2025-07-21
American Institute of Aeronautics and Astronautics (AIAA)
https://doi.org/10.2514/6.2025-3397
Thellier E, Perrusquía A, Tsourdos A. (2025) Instance-specific heuristic learning for scalable UAV path planning. In: AIAA Aviation Forum and Ascend 2025, 21-25 July 2025, Las Vegas, USA, Paper number AIAA 2025-3397

Downloaded from a DSpace repository. https://dspace.org/