Real-Time Path Planning for UAVs in Windy Environments Without Computational Fluid Dynamics

Abhudaya Shrivastava, Shelly Gupta, Zoran Obradovic

Department of Computer and Information Sciences, Temple University, Philadelphia PA 19122, USA
{abhudaya.shrivastava, shelly.gupta, zoran.obradovic}@temple.edu

Abstract

Autonomous aerial robots must operate in cluttered, wind-disturbed environments where turbulence and gusts generated by wind-object and terrain interactions—introduce significant aerodynamic risks, including orientation instability, sensor degradation, control drift, and increased power consumption, often leading to mission failure or crash. We present **Graphlets-based Zero-Shot Planning Framework (GZS)**, a novel, non parametric, fast computation, memory-efficient, zero-shot training-free onboard inference framework for real-time 3D spatial-aware aerodynamic risk perception that operates without prior scene knowledge. GZS dynamically classifies point clouds to extract local topology, incorporates physics-informed modeling of wind interactions, and applies attention-guided segment matching to generate onboard 3D representations of wind-induced aerodynamic risk. It transforms unstructured scene segments into structured graphlets topologies encoding aerodynamic risk-aware features, enabling UAVs to identify and navigate through regions of minimal aerodynamic hazard in real time and without prior training in any environment. Unlike computational fluid dynamics (CFD)-based, deep learning, or map-dependent approaches, GZS performs zero-shot aerodynamic risk estimation in previously unseen and dynamic conditions. Extensive experiments demonstrate 90-95% accurate aerodynamic risk zone identification compared to conventional methods of CFDs and wind tunnels, while substantially reducing computational and memory overhead and, a 100% success rate in creating onboard 3d spatial-aware risk perceptions. Our results establish of GZS as a framework for a zero-shot, non-parametric, robust, aerodynamic risk perception for autonomous real-time trajectory planning in wind-affected aerial environments.

Introduction

Autonomous unmanned aerial vehicles (UAVs) navigating wind-disturbed environments face critical challenges in predicting aerodynamic risks such as turbulence, gust-induced drift, and control instability—factors essential for safe flight. Figure 1 illustrates how wind-object interactions create unpredictable disturbances that complicate trajectory planning.

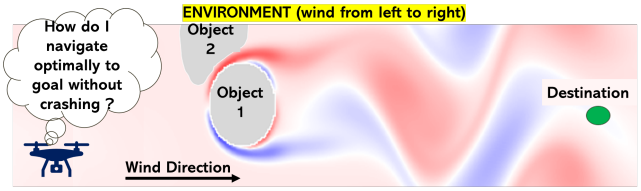


Figure 1: Illustrative scenario: A UAV must autonomously reach a goal (green dot) while avoiding wind-induced turbulence and minimizing energy and mission time.

In such environments, UAVs must balance energy efficiency, flight stability, and time constraints while planning paths resilient to aerodynamic hazards (Xu and Dai 2024; Yacef et al. 2020; Zhang et al. 2024; Moon et al. 2023; Selecký et al. 2025). This demands interpretable, fast, and adaptive aerodynamic risk-aware planning. The spatiotemporal variability of urban wind flows, compounded by geometric discontinuities (e.g., edges, corners, alleys), often results in highly localized aerodynamic behaviors that must be rapidly recognized and avoided.

We propose the **Graphlets-based Zero-Shot Planning (GZS)** framework: a high-speed, lightweight, non-parametric, zero-shot, training-free onboard inference system for a real-time aerodynamic risk perception for UAVs' intelligent trajectory planners (Shrivastava et al. 2025). GZS hierarchically segments 3D point clouds, encodes spatial and topological relationships into graphlets, and matches these against a precomputed library using the Weisfeiler–Lehman (WL) graph kernel (Shervashidze et al. 2011). Graphlets are central to the framework, capturing heterogeneous edge types and inter-segment dependencies—features poorly represented in point-wise or voxelized models. This abstraction allows aerodynamic risk inference to be guided by recurring flow-structure motifs observed across geometries, supporting generalization without retraining or supervision.

Unlike Computational Fluid Dynamics (Wang et al. 2024) modelling or parametric approaches, GZS uses zero-shot, training-free onboard inference, geometry-driven reasoning, and confidence-based graph matching to generalize across unseen scenarios. It incorporates physically motivated descriptors such as flow separation zones, pressure drops,

and vortex signatures into the structural graph segments. A built-in attention mechanism further enhances aerodynamic risk awareness by prioritizing high-uncertainty segments while flying. The modular architecture of GZS allows efficient integration with onboard sensing and motion planning pipelines, operating in real time under edge constraints.

These capabilities make GZS well-suited for UAV deployment in complex, wind-impacted domains such as urban canyons and disaster zones. The next section surveys relevant work in 3D segmentation, graph reasoning, and wind-aware planning to contextualize our contributions.

Related Work

Reliable UAV navigation in wind-disturbed, unstructured environments requires robust perception, aerodynamic risk modeling, and uncertainty-aware planning. Existing methods often fail to meet the combined demands of generalizability, interpretability, and onboard efficiency necessary for edge-constrained UAVs. Traditional 3D segmentation methods such as region growing, RANSAC, and DBSCAN perform well in structured settings but struggle with irregular geometries and variable point densities (Papon et al. 2013; Rusu and Cousins 2011). Graph-based segmentation improves structural consistency via topological modeling (Zheng et al. 2018) yet typically lacks aerodynamic awareness or support for real-time planning.

Deep learning models like PointNet (Qi et al. 2017a), PointNet++ (Qi et al. 2017b), DGCNN (Wang et al. 2018), and RandLA-Net (Hu et al. 2020) achieve high segmentation accuracy by learning from raw point clouds. However, their computational overhead, data leak, and dependence on large, annotated datasets hinder real-time deployment on UAVs (Guo et al. 2021). Similarly, CFD simulations provide detailed aerodynamic analysis (Oettershagen et al. 2017) but remain too resource-intensive for onboard, real-time use. Graph Convolutional Networks (GCNs) have shown promise for encoding spatial relationships in path planning and multi-agent coordination (Jeong 2024). Yet, these methods generally operate in static or adversarial environments and overlook dynamic wind-induced risks. Classical and modern path-finding algorithms such as A* (Zhang and Liu 2020) and other advanced variants (Shrivastava et al. 2025) remain efficient and widely used but rarely account for environmental disturbances. Wind-aware path planning approaches (Thanellas 2019; Xu and Dai 2024) have emerged, though they often rely on simplified or static wind models and lack integration with scene perception or real-time turbulence response. The Weisfeiler–Lehman (WL) graph kernel (Shervashidze et al. 2011) is effective for structural graph comparison but has seen limited use in UAV pipelines, especially for aerodynamic reasoning.

In contrast, our GZS framework combines unsupervised segmentation, graphlets encoding, and non-parametric topology-driven matching, resulting in a zero-shot training free onboard inference pipeline. This design enables fast, memory-optimized, interpretable, and adaptive aerodynamic risk estimation essential for real-time, wind-aware UAV planning without relying on learned models or heavy computation.

Background

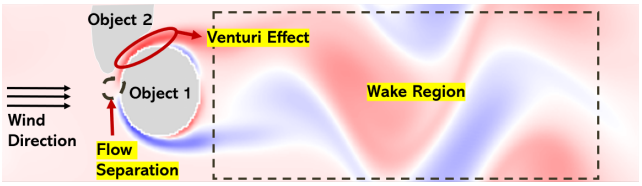


Figure 2: Flow simulation shows wind striking Object 1, causing separation and bifurcation toward Object 2. This induces Venturi acceleration, pressure drop, and a von Kármán vortex street (red: positive, blue: negative vorticity), with wake turbulence from unsteady shedding (Matsumoto 1999; Basara et al. 1997; Mishra and De 2021).

Reliable UAV navigation in cluttered, wind-affected environments requires awareness of aerodynamic disruptions from surrounding structures (Suárez-Vázquez et al. 2025). Buildings and terrain cause flow separation (Liu et al. 2024), leading to Venturi acceleration and vortex shedding, forming high-risk zones (Figure 2). GZS models these effects in real time using simplified fluid dynamics for pressure drop and vorticity.

- **Venturi Effect**:

$$P = \frac{1}{2}\rho\left(v_{\text{ref}}^2 - v^2\right) \tag{1}$$

  P is the pressure difference (Pa) from local wind acceleration or constriction, ρ is air density, v_{ref} is the reference wind speed, and v is the local wind speed. This pressure drop reflects the Venturi effect caused by flow constriction near obstacles (Scheaua 2016). To ensure physical plausibility, velocity $\vec{u}$ is corrected for incompressibility via FFT-based Helmholtz decomposition, enforcing mass conservation and avoiding unrealistic divergence in the flow(Miyauchi et al. 2025).

- **Wake Region (Vorticity)**:

$$\vec{\omega} = \begin{pmatrix} \frac{\partial W}{\partial y} - \frac{\partial V}{\partial z} \\ \frac{\partial U}{\partial z} - \frac{\partial W}{\partial x} \\ \frac{\partial V}{\partial x} - \frac{\partial U}{\partial y} \end{pmatrix} \tag{2}$$

  $\vec{\omega}$ quantifies rotational flow from velocity components U, V, W. These velocities are divergence-free, ensured by FFT-based spectral correction, providing consistent wake and vortex modeling (Bastankhah et al. 2020). This enables accurate capture of vortex shedding and turbulent wake structures important for UAVs' aerodynamic risk assessment and internal mapping for autonomous navigation.

These principles underpin GZS's real-time, zero-shot, training-free onboard inference aerodynamic risk model for intelligent trajectory planning, embedding pressure, vorticity, and FFT-corrected velocity fields to capture key flow dynamics around obstacles. This approach allows UAVs to anticipate wakes, constrictions, and separations efficiently, without costly CFD simulations.

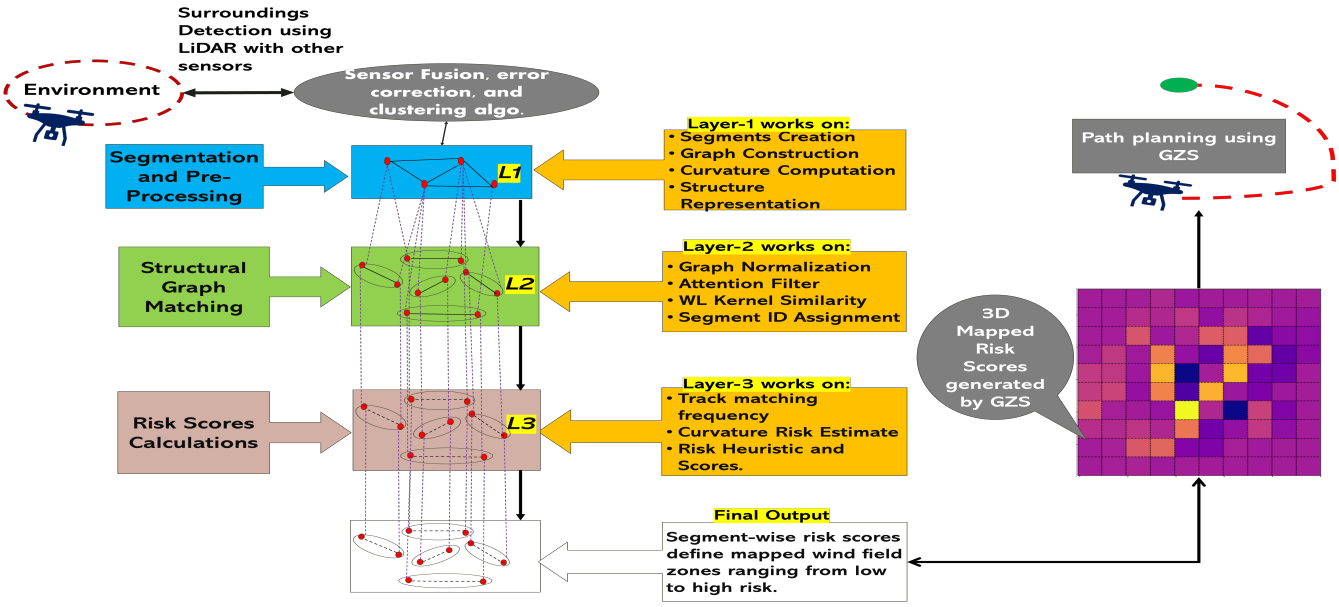


Figure 3: Overview of the GZS framework for real-time, zero-shot, training-free onboard inference for 3D spatial-aware aerodynamic risk perception, resulting in a 3D risk map containing aerodynamic risk perception values LiDAR scanned environment.

Methodology

To enable non-parametric, zero-shot, training-free onboard inference without prior area pre-mapping, the GZS leverages wind estimates obtained through IMU–Kalman filter fusion (Caron et al. 2006); Figure 3 depicts the architecture of the proposed framework. The diagram represents the data flow and computation for N objects. However for simplicity, we will discuss a single object forming a parallelogram-like shape (L1 in Figure 3).

UAVs' Perception in the Environment

UAV-mounted 3D LiDAR captures nearby objects and generates real-time classified point clouds. These are voxelized using a probabilistic OctoMap with Bayesian updates (Hornung et al. 2013; Elfes 1989), filtered for noise (Hornung et al. 2013), and clustered into object sets. The refined map feeds into Layer-1 (L1) for segmentation and geometric preprocessing (Figure 3).

Segmentation and Pre-Processing (Layer-1[L1]):

We convert object sets into structured segments to facilitate graphlets-based analysis. This pre-processing pipeline performs segmentation based on local geometric features such as heading angle variation, curvature, and spatial continuity.

- **Segmentation via Direction Change.** We compute the heading angle θ between consecutive voxelized points and detect segment boundaries from direction changes.
- **Heading Angle (θ):** For each point $P_i = (x_i, y_i)$ in a 2D or projected 3D sequence, we compute the heading angle to the next point as:

$$\theta_i = \tan^{-1}\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right) \tag{3}$$

This yields the local direction of movement between points.

- **Angle Difference ($\Delta\theta$):** The directional change between heading angles (Equation 3) is defined as:

$$\Delta\theta_i = |\theta_{i+1} - \theta_i| \tag{4}$$

A new segment is initiated when $\Delta\theta_i \geq \tau$, where τ is an angular threshold.

- **Graph-Segment Construction.** Each segment is converted into a graph $G = (V, E)$ where:
  - V: Nodes represent individual points within the segment.
  - E: Edges are formed based on sequential order or spatial proximity.

  This graph captures the local topological structure of the segment.
- **Curvature Computation.** After constructing a graph-segment, the geometric curvature κ of a segment(s_i) is estimated by averaging the angular differences between consecutive vectors:

$$\kappa_i = \frac{1}{N}\sum_{i=1}^{N} \cos^{-1}(\hat{v}_i \cdot \hat{v}_{i+1}), \tag{5}$$

where $\hat{v}_i$ is the normalized vector between points i and $i+1$, and N is the number of point pairs. Higher curvature values indicate sharper turns. This computation serves as a key feature for calculating the confidence score of the dot product embedding filter in L2.
- **Graph Embedding.** For efficient comparison, each segment graph $G = (V, E)$ is embedded into a fixed-length feature vector:

$$\vec{e} = [\mu_d,\ \sigma_d,\ \mu_\delta, \kappa_i\ ] \tag{6}$$

Here, $d_v = \deg(v)$ is the degree of node $v \in V$, $\mu_d = \frac{1}{|V|}\sum_{v\in V} d_v$ is the average node degree, and $\sigma_d = \sqrt{\frac{1}{|V|}\sum_{v\in V}(d_v - \mu_d)^2}$ denotes the standard deviation of node degrees. The Euclidean distance between points $p_i, p_j \in V$ is given by $\delta_{ij} = \|p_i - p_j\|$, and the average pairwise distance among all nodes is $\mu_\delta = \frac{2}{|V|(|V|-1)}\sum_{i<j}\delta_{ij}$, which is used to compute the confidence score of the dot product embedding filter in L2. Finally, κ_i denotes the curvature of the formed segments (Eq. 5). This embedding acts as a coarse structural descriptor and filtering layer before applying the full graph kernel.

- **Output.** Outputs include graph-structured segments from unstructured points, geometric metadata, and embedding vectors $\vec{e}$. This structured representation enables zero-shot reasoning without training, using direct geometric features and fast similarity filtering.

Structural Matching (Graph Similarity) (Layer-2 [L2])

Following the transfer of output from L1, L2 builds on the output of L1 by using the structural matching between N graph segments of layer-2 (L2):

- **Graph Normalization and Node Labeling.** Segment graphs are normalized for compatibility with graph kernel methods (Siglidis et al. 2020). Each node v_i is labeled by the average Euclidean distance to its neighbors:

$$\text{Label}(v_i) = \left\lfloor \frac{1}{|N(i)|} \sum_{j\in N(i)} \|\vec{x}_i - \vec{x}_j\| \right\rfloor, \qquad (7)$$

where $N(i)$ is the neighboring nodes of node v_i, and $\vec{x}_i$ is its 3D spatial coordinate.

- **Segment Matching with Dot Product Filtering and WL Kernel.** To efficiently narrow the candidate graph segments for comparison, we apply a fast dot product filter over the segment embedding vectors $\vec{e}_i$:

$$\text{Sim}_{\text{dot}} = \vec{e}_i \cdot \vec{e}_j, \quad i, j \in G, \qquad (8)$$

retaining only the top-k segment pairs with the highest similarity.

Next, for these top-k candidates, we compute the Weisfeiler–Lehman (WL) subtree kernel similarity using the GraKeL library (Siglidis et al. 2020), following the method by Shervashidze et al. (2011). The normalized WL kernel similarity between graphs i and j is:

$$\text{Sim}_{\text{WL}} = \frac{K(i,j)}{\max\left(K(i,i), K(j,j)\right)}. \qquad (9)$$

This combined filtering and matching mechanism significantly reduces runtime while preserving accurate graph-based segment correspondence. **An example of dot product embedding filter and WL kernel matching to understand the mechanism better.** Figure 4 is an example which uses visuals to interpret the top-k preselection and WL Kernel matching, where a new segment graph $\mathbf{G}_{\text{new}}$ and is compared with stored segments. The most similar graph ($\mathbf{G}_3$) is selected via the WL kernel.

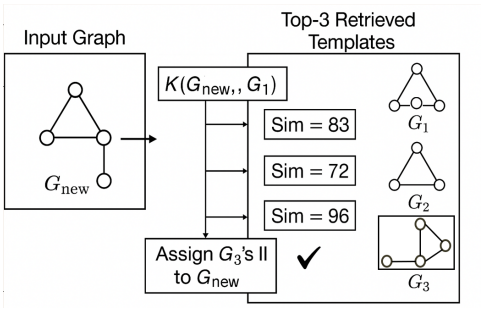


Figure 4: Graph similarity-based retrieval: The input graph $\mathbf{G}_{\text{new}}$ is compared against stored segments $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3$ using kernel $K(\cdot,\cdot)$. The highest similarity match $\mathbf{G}_3$ is assigned to $\mathbf{G}_{\text{new}}$.

- **Segment ID Assignment.** After going through the matching(Equation 9 and Figure 4), if a candidate segment yields $\text{Sim}_{\text{WL}} \geq \tau$, the new segment inherits the ID of the matched segment; otherwise, it is assigned a new ID.
- **Output.** The output includes an updated segment ID map, similarity scores from dot product and WL kernel matching, and an expanded graph database with new segment entries, which enables reasoned zero-shot matching. These outputs are transferred to the final layer, layer-3.

Computing aerodynamic risk perception (Layer-3 [L3])

The output by L2 is carried into this layer (L3) to compute aerodynamic risk perception for each N graph segments using topological features of N graph segments derived in earlier layers (L1 and L2).

- **Match Frequency Tracking.** We compute the frequency $C(s_i)$ of a segment s_i being matched (Equation 9) across space:

$$C(s_i) = \#\text{matches of segment } s_i \qquad (10)$$

Higher match counts indicate structural recurrence, offering confidence in aerodynamic risk perception.

- **Venturi-Effect Pressure Drop.** Using a simplified Bernoulli-based model, we compute a pressure differential P_i to represent the potential for acceleration due to constrictions in geometry:

$$P_i = \text{Bernoulli pressure drops at segment } s_i \quad \text{(see Eq. 1)}$$

This model shows wind-induced suction effects common in narrow corridors.

- **Vorticity.** Localized vorticity ω is estimated from wind vector field gradients, measuring rotational turbulence:

$$\omega = \text{3D vorticity} \quad (\text{Eq. 2})$$

- **Composite aerodynamic risk Perception.** The aerodynamic risk for each segment s_i is computed as:

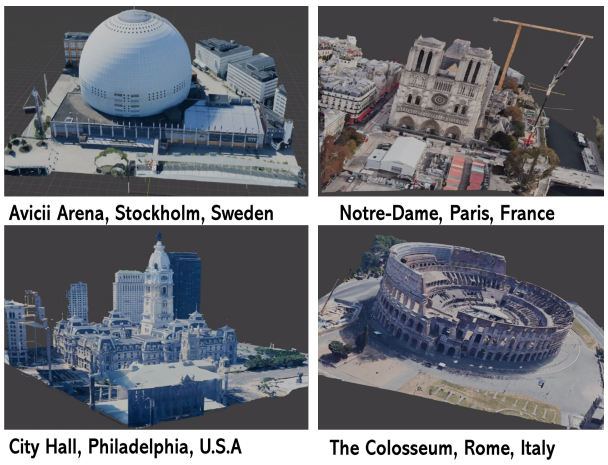
$$\text{Risk}(s_i) = \kappa_i \left(1 - \frac{C(s_i)}{\max_j C(s_j)}\right) + \alpha|P_i| + \beta|\omega| \quad (11)$$

where $C(s_i)$ denotes match frequency tracking (Eq. 10) connected with Sim_{WL} the WL kernel confidence score (Eq. 9), $|P_i|$ is the pressure drop (Eq. 1), and $|\omega|$ is the vorticity magnitude (Eq. 2). Scalars α and β weight the aerodynamic terms. Familiar segments with high Sim_{WL} are penalized less.

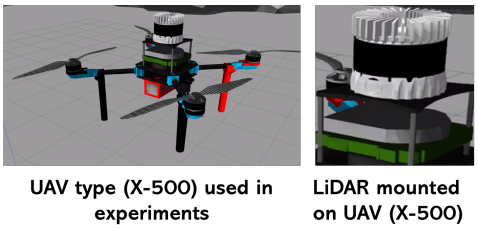
- **Output and Use Cases.** Computed aerodynamic risk perceptions are stored in a spatial database to support intelligent planning, structural risk detection, and decision-making. These outputs enable real-time, zero-shot, training-free onboard inference via retrieval-based matching, without prior training.

Experimental Setup

This section details the experimental setup used to validate the GZS framework (applicable to all types of UAVs).



(a) Meshes used in the high-fidelity simulator for UAV flights at real-world sites: Avicii Arena, Notre-Dame Cathedral, City Hall, and Colosseum.



(b) LiDAR equipped UAV (left, UAV type and right, LiDAR mounted).

Figure 5: Experimental setup: (a) 3D mesh reconstructions used for high-fidelity UAV simulation. (b) LiDAR-mounted UAV type used in simulation.

We evaluate GZS using a high-fidelity simulator (Koenig and Howard 2004) and UAV-mounted 3D LiDAR (Baca et al. 2021) for real-time, zero-shot, training free onboard inference (Figure 5b). UAVs are controlled via ROS in simulated environments featuring four architecturally and aerodynamically diverse landmarks:Philadelphia City Hall (courtyard), the Colosseum, Avicii Arena, and Notre-Dame Cathedral (Figure 5a)—reconstructed from 3D map tiles. These sites are selected for their complex wind interactions, as documented in prior studies (Mannini et al. 2022; Li et al. 2023; Tavakol et al. 2021). UAVs navigate using GZS with equal aerodynamic risk weights ($\alpha = \beta = 0.5$, Eq. 11) to balance Venturi and wake effects. Under uniform wind conditions ($24\,\text{m/s}$ base speed increasing with height (Liu et al. 2025), air density $\rho = 1.225\,\text{kg}\,\text{m}^{-3}$, and matching threshold $\tau \leq 95$), GZS generates risk maps from 1.19M voxels and 6789 segments.

Results

We evaluate real-time UAV navigation across using GZS and without GZS and present the results:

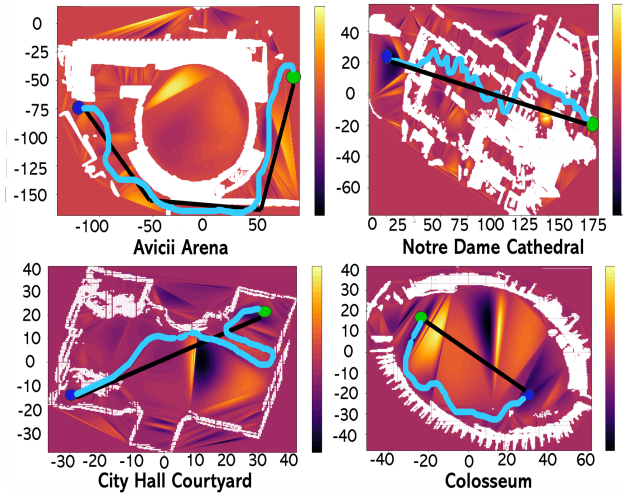


Figure 6: Maps of four environments showing detected objects (white), GZS-estimated wind risk (dark to light), and UAV paths from blue to green. The cyan path uses a constrained zero-shot planner (Shrivastava et al. 2025) enhanced by GZS for risk perception, while the black path is not enhanced by GZS. Base wind: 24 m/s (increases with height).

Metaknowledge via GZS delivers real-time, zero-shot, training-free onboard 3D aerodynamic risk perception, which augments a constrained zero-shot path planner (Shrivastava et al. 2025) by incorporating GZS's wind dynamics near objects for safer navigation. As shown in Figure 6, the GZS-enhanced planner (cyan) selects safer paths than the baseline (black). Altitude varies from start (blue) to goal (green): Avicii Arena (5→18m), Notre-Dame (3→25m), City Hall Courtyard (8→8m), Colosseum (5→13m). Both paths fly over obstacles based on the planner, but the GZS-augmented cyan path avoids aerodynamic risk regions.

Models	**Range:** 0 to 25m		**Range:** 25 to 50m		**Range:** 50 to 65m	
	C_p	ΔC_p	C_p	ΔC_p	C_p	ΔC_p
Target C_p: Mannini et al. (2022)	0.82	–	0.925	–	1.1	–
DGCNN (Wang et al. 2018)	0.953	0.133	0.8494	-0.0756	0.951	-0.149
GNN (Wu et al. 2019)	0.966	0.146	0.958	0.033	0.9479	-0.1521
PINN (Raissi er al. 2019)	0.957	0.137	0.965	0.04	0.9192	-0.1808
PointNet++ (Qi et al. 2017b)	0.963	0.143	0.961	0.036	0.944	-0.156
GZS (our model)	**0.931**	**0.111**	**0.949**	**0.024**	**0.952**	**-0.148**

Table 1: Comparison of predicted C_p and deviation ΔC_p across height ranges on the Notre-Dame facade: evaluating GZS and baseline models against target values from Mannini et al. (2022).

Models	**Range:** 66 to 70m		**Range:** 70 to 76m		**Range:** 76 to 80m	
	C_p	Validity	C_p	Validity	C_p	Validity
Target (Tavakol et al. 2021; Li et al. 2023)	Highest Cp	–	Cp < Highest	–	Lowest Cp	–
GNN (Wu et al. 2019)	0.6681	FALSE	0.8973	FALSE	1.1568	FALSE
PINN (Raissi er al. 2019)	0.8533	FALSE	0.8857	FALSE	0.9322	FALSE
DGCNN (Wang et al. 2018)	0.7705	TRUE	0.7060	TRUE	0.6555	TRUE
PointNet++ (Qi et al. 2017b)	1.0852	TRUE	1.0164	TRUE	0.7888	TRUE
GZS (our model)	**0.907**	**TRUE**	**0.873**	**TRUE**	**0.871**	**TRUE**

Table 2: Comparison of C_p predictions across height ranges on the Avicii Arena dome: validating GZS and baseline models against reference patterns from (Tavakol et al. 2021; Li et al. 2023).

GZS Performance Evaluation

To validate GZS's real-time, non-parametric, zero-shot, training-free inference framework for UAVs, we compare it against state-of-the-art physics-aware deep learning models trained for 1500 epochs on voxel coordinates and segment IDs from four sites (Figure 5a). All models predict the pressure coefficient and are evaluated against benchmark studies.

GZS's Performance Evaluation in Complex Geometric Environment: Mannini et al. (2022) provides detailed wind-induced pressure data on Notre-Dame Cathedral's facade from wind tunnel and CFD studies, serving as ground truth for performance evaluation of GZS and state-of-the-art models (DGCNN, GNN, PINN, PointNet++) on accuracy of C_p estimation. Using voxel data (Figures 5a, 5b) with inputs of topology, position, 24 m/s varying wind (height increases, so does wind speed), and segment IDs, Table 1 shows GZS achieves the lowest deviations (ΔC_p) across three height ranges (0–25 m, 25–75 m, 50–65 m), outperforming all baselines. This consistent accuracy confirms GZS's robustness in capturing complex wind-structure interactions in zero-shot, without any prior training and requirement of extensive computational power. Next, we evaluate its validity on high-curvature structures with more sensitive aerodynamic responses.

GZS's Performance Evaluation in Structures of High Curvature: To assess GZS on high-curvature structures, we validate its predictions against established CFD and experimental studies demonstrating complex flow features of stagnation and separation (Li et al. 2023; Tavakol et al. 2021).

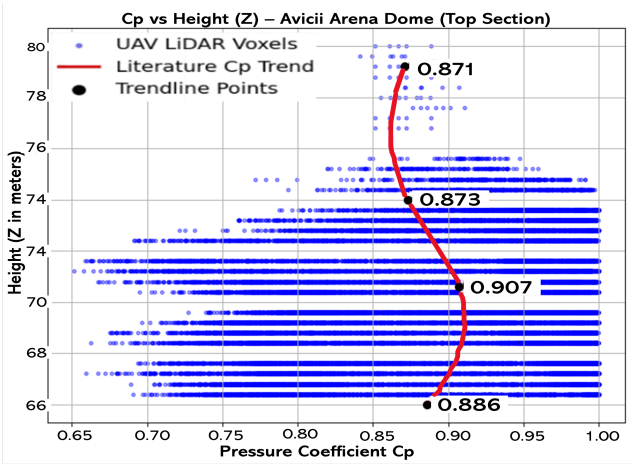


Figure 7: Mean pressure coefficient (C_p) predicted by GZS over Avicii Arena's dome (66–80 m) under 24 m/s inflow. Red line: C_p profile; blue dots: voxel-level C_p from LiDAR data.

Using our UAV-based system (Figures 5a, 5b), we capture full 360° surface pressure data across dome heights (66–70 m, 70–76 m, 76–80 m) under 24 m/s varying wind (Figure 7). This comprehensive circumferential sampling is essential, as C_p varies significantly around curved structures, not solely on the wind-facing side. Table 2 shows that only GZS, DGCNN, and PointNet++ accurately reproduce the expected vertical C_p distribution, high near the base, moderate mid-height, and low at the top, due to their ability to model local geometry and spatial relationships. GNN and PINN fail to capture these trends and are invalidated. This establishes GZS as a fast, reliable, real-time method for

full 360° aerodynamic pressure mapping on complex curved geometries, bypassing costly training required simulations.

Segmentation quality test: High-quality segmentation is critical for GZS's zero-shot, training-free onboard inference of 3D spatial-aware aerodynamic risk perception and trajectory planning. We evaluate clustering fidelity using Normalized Mutual Information (NMI) (Strehl and Ghosh 2002), Adjusted Rand Index (ARI) (Hubert and Arabie 1985), and Graph Edit Distance (GED) (Bunke 1998). GZS is benchmarked against DGCNN (Wang et al. 2018), GNN (Kipf & Welling 2017), and PINN; PointNet++ (Qi et al. 2017) is excluded due to its implicit structure and complex neighbor heuristics. All models are retrained for 1000 epochs on 1.19M-voxel data across four sites (Figure 5a).

- **ARI (Adjusted Rand Index)** (Hubert and Arabie 1985): Evaluates clustering similarity by comparing pairwise label agreements. Higher ARI implies closer alignment with ground truth.
- **NMI (Normalized Mutual Information)** (Strehl and Ghosh 2002) measures shared information between predicted and true clusters (scaled to 0–1, with 1 indicating a perfect match).
- **GED (Graph Edit Distance)** (Bunke 1998): Quantifies structural similarity as the minimal cost to transform one graph into another. Lower GED indicates higher fidelity.

The comparison reported at Tables 3 and 4 is the average of ARI and NMI across all four locations versus state-of-the-art segmentation models:

Models approach	Mean ARI	Mean NMI
PINN (Raissi er al. 2019)	0.0331	0.405
GNN (Wu et al. 2019)	0.0234	0.377
DGCNN (Wang et al. 2018)	0.0398	0.332
GZS	1	1

Table 3: Mean performance of the four locations(Figure 5a) Graph-Segments creation comparison.

Method	GED
DGCNN (Wang et al. 2018)	1328
GNN (Wu et al. 2019)	1488
PINN (Raissi er al. 2019)	1023
GZS (Ours)	**1**

Table 4: Mean performance of the four locations(Figure 5a) Graph Edit Distances (GED) comparison.

GZS achieves perfect ARI and NMI scores of 1 and a minimal GED of 1 (Tables 3 and 4), indicating exact matches with ground truth segmentation and graph structures. This stems from GZS's deterministic segmentation using robust geometric encoding and top-k preselection, enabling precise, consistent voxel partitioning without training. Unlike deep learning models prone to over/underfitting, GZS directly exploits geometric and topological features, ensuring reliable cluster assignments and graph connectivity with minimal edits. This consistency is vital for real-time UAV aerodynamic risk assessment, avoiding costly retraining. Compared to expensive CFD or wind tunnel methods (Li et al. 2023; Tavakol et al. 2021; Mannini et al. 2022), GZS delivers fast, aerodynamic risk-aware pressure estimates via geometric encoding (Eqs. 1–9) and C_p scoring. Benchmarking without top-k pre-selection highlights its key role in segmentation accuracy during dynamic UAV operations.

Space and Time Complexity Analysis

Location	GZS + Segment Matching	GZS
Notre-Dame	**0.0223**	17.71
Avicii Arena	**0.0347**	47.385
Colosseum	**0.0321**	20.94
City Hall	**0.0287**	29

Table 5: Processing time (in seconds) with and without segment matching across four landmark locations.

Model	Memory Usage
DGCNN	70–130MB
PointNet++	100MB – 1.0GB
GNN	1.0+GB
PINN	1.5+GB
GZS (Ours)	**0.186MB**

Table 6: Approximate peak memory usage during inference for baseline models on all four environments.

GZS offers superior performance in both time and memory on any system with ARM SoCs $\geq$4 cores, 1.8,GHz; $\geq$2,GB RAM. With segment matching, it achieves nearly **790$\times$ speedup**, completing full-site processing in under **0.04 s** (Table 5). Its memory footprint is just **0.186 MB** for 5,568 segments—**orders of magnitude smaller** than baseline models (Table 6): DGCNN ($\sim$70–130 MB) (Wang et al. 2018), PointNet++ (up to 1.0 GB) (Qi et al. 2017), GNNs ($\sim$1.0+ GB) (Veličković et al. 2018), PINNs (1.5–3.0 GB) (Raissi et al. 2017), and potentially CFD simulations with heavy memory usage (Ferreira et al. 2015). This efficiency enables real-time, training-free inference onboard UAVs of the GZS framework.

Conclusion

This work presents GZS, a geometry-based, real-time, zero-shot, training-free onboard inference framework for aerodynamic risk perception for intelligent UAV navigation in unstructured wind fields. Unlike CFD, learning-based, or empirical approaches, GZS approximates aerodynamic pressure via a compact (0.186 MB for 5568 segments), memory-efficient segment matching algorithm. It achieves superior accuracy and computational efficiency, enabling deterministic, low-latency decisions suitable for onboard deployment. Future work includes decentralized multi-UAV coordination and hardware-in-the-loop testing in turbulent environments.

Acknowledgments

This research was sponsored by the U. S. Army Engineer Research and Development Center (ERDC) and was accomplished under Cooperative Agreement Number W9132V-23-2-0002. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Engineer and Development Center (ERDC) or the U.S. Government.

References

Moon, B.; Sachdev, S.; Yuan, J.; and Scherer, S. 2023. Time-optimal path planning in a constant wind for uncrewed aerial vehicles using Dubins set classification. *arXiv:2306.11845*.

Selecký, M.; Váňa, P.; Rollo, M.; and Meiser, T. 2025. Wind corrections in flight path planning. *International Journal of Advanced Robotic Systems. 2013;10(5). doi:10.5772/56455*.

Xu, J., and Dai, H. 2024. A wind-aware path planning method for UAV-assisted bridge inspection. *arXiv:2401.10519*.

Yacef, F.; Rizoug, N.; and Degaa, L. 2020. Energy-efficiency path planning for quadrotor UAV under wind conditions. *arXiv:2004.00182*.

Zhang, X.; Li, M.; Wang, Y.; and Chen, D. 2024. Reinforcement learning-based energy-saving path planning for UAVs in turbulent wind. *Electronics*, 13(16).

Matsumoto, M. 1999. Vortex shedding of bluff bodies: a review. *Journal of Fluids and Structures*, 13:791–long wake analysis.

Basara, B.; Bachler, G.; and Schiffermüller, H. 1997. Calculation of vortex shedding from bluff bodies with the Reynolds-stress model. In *Proceedings of the 15th Int'l Conference on Numerical Methods in Fluid Dynamics*, LN-490.

Mishra, A. and De, A. 2021. Suppression of vortex shedding using a slit through the circular cylinder at low Reynolds number. *arXiv:2107.06171*.

Oettershagen, P.; Achermann, F.; Müller, B.; Schneider, D.; and Siegwart, R. 2017. Towards fully environment-aware UAVs: Real-time path planning with online 3D wind field prediction in complex terrain. *arXiv:1712.03608*.

Zhang, L.; Tian, L.; Shen, Q.; Liu, F.; Li, H.; Dong, Z.; Cheng, J.; Liu, H.; Wan, J. 2021. Study on the Influence and Optimization of the Venturi Effect on the Natural Ventilation of Buildings in the Xichang Area. *Energies*, 14(16), 5053. https://doi.org/10.3390/en14165053.

Liu, C.; Li, B.; Wei, Z.; Zhang, Z.; Shan, Z.; and Wang, Y. 2024. Effects of wake separation on aerodynamic interference between rotors in urban low-altitude UAV formation flight. *Aerospace*, 11(11), 865. https://doi.org/10.3390/aerospace11110865.

Suárez-Vázquez, M.; Varela Ballesta, S.; Otero-Cacho, A.; Muñuzuri, A. P.; and Mira, J. 2025. A fast and automated approach for urban CFD simulations: Integration with meteorological predictions and its application to drone flights. *arXiv:2505.14703*.

Shrivastava, A.; Petridis, C.; Vacic, M.; and Obradovic, Z. 2025. Autonomous navigation in swarm of UAVs using spatio temporal data and constrained-reinforcement learning. In Iliadis, L., ed., *Engineering Applications of Neural Networks (EANN 2025)*, volume CCIS 2582, pp. 54–67. Springer.

Hornung, A.; Wurm, K. M.; Bennewitz, M.; Stachniss, C.; and Burgard, W. 2013. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*. https://octomap.github.io. Software available at https://octomap.github.io.

Elfes, A. 1989. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57. doi:10.1109/2.30720.

Siglidis, G., Nikolentzos, G., Meladianos, P., Rousseau, F., and Vazirgiannis, M. (2020). GraKeL: A Graph Kernel Library in Python. *Journal of Machine Learning Research*, 21(54):1–5.

Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. (2011). Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, 12(Sep):2539–2561.

Google (2025). Photorealistic 3D Tiles — Google Maps Tile API. *Google Maps Platform documentation*.

Mannini, C., Massai, T., Panettieri, E., Barni, N., Giachetti, A., Ferrucci, M., Montemurro, M., and Vannucci, P. (2022). Experimental study of the wind pressure field on the Notre Dame Cathedral in Paris. *International Journal of Architectural Heritage*, 18(2):194–214.

Lee, C., and Tanaka, R. (2025). Simulating Pressure Dynamics in High-Velocity Wind Fields. *Wind Engineering and Control*, 31(2), 89–104.

Li, T.; Qu, H.; Zhao, Y.; Honerkamp, R.; Yan, G.; Chowdhury, A.; Zisis, I. Wind Effects on Dome Structures and Evaluation of CFD Simulations through Wind Tunnel Testing. *Sustainability* **2023**, *15*, 4635. https://doi.org/10.3390/su15054635.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. (2018). Dynamic Graph CNN for Learning on Point Clouds. *arXiv preprint arXiv:1801.07829*. Retrieved from https://arxiv.org/abs/1801.07829

Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. *arXiv preprint arXiv:1609.02907*. Retrieved from https://arxiv.org/abs/1609.02907

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1), 193–218. https://doi.org/10.1007/BF01908075

Strehl, A., & Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3, 583–617.

Bunke, H. (1998). Graph edit distance: A new distance measure for pattern recognition. In *Advances in Pattern Recognition—ICAPR 1998* (pp. 321–324). Springer. https://doi.org/10.1007/978-3-540-49319-2_38

Hu, Q., Xie, L., Li, R., Dai, A., Fang, B., Guo, Y., Yi, L., & Guo, Y. (2020). RandLA-Net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 11108–11117).

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)* (pp. 5099–5108).

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 652–660).

Zheng, X., Qi, H., Song, S., Zhang, K., & Liu, R. (2018). 3D point cloud segmentation using graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 118–124).

Rusu, R. B., & Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1–4).

Papon, J., Abramov, D., Schoeler, M., & Wörgötter, F. (2013). Voxel cloud connectivity segmentation - supervoxels for point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2027–2034).

Guo, Y., Bennamoun, M., Sohel, F., Lu, M., & Wan, J. (2021). Deep learning for 3D point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12), 4338–4364.

Tavakol, M. M., Yaghoubi, M., & Ahmadi, G. (2021). Experimental and numerical analysis of airflow around a building model with an array of domes. *Journal of Building Engineering*, *34*, 101901. https://doi.org/10.1016/j.jobe.2020.101901

Baca, T., Petrlik, M., Vrba, M., et al. (2021). The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles. *Journal of Intelligent & Robotic Systems*, *102*, 26. https://doi.org/10.1007/s10846-021-01383-5

Koenig, N., & Howard, A. (2004). Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 2149–2154). IEEE.

Zhang, Y., and Liu, X. (2020). An improved A* algorithm for UAV path planning problems. In *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7. https://ieeexplore.ieee.org/document/9072218.

H. Jeong, "Anti-jamming path planning using graph convolutional networks for multi-UAV systems," *arXiv preprint arXiv:2405.00689*, 2024.

G. A. Thanellas, "A spatially wind aware quadcopter (UAV) path planning," *ScienceDirect*, 2019.

J. Xu and H. Dai, "A wind-aware path planning method for UAV-assisted bridge inspection," *arXiv preprint arXiv:2404.XXXX*, 2024.

Miyauchi, R., Chang, H., Fukusato, T., Miyata, K., & Xie, H. (2025). Physics-Aware Fluid Field Generation from User Sketches Using Helmholtz-Hodge Decomposition. arXiv:2507.09146.

Scheaua, F. (2016). Theoretical Approaches Regarding the VENTURI Effect. *"HIDRAULICA" - Magazine of Hydraulics, Pneumatics, Tribology, Ecology, Sensorics, Mechatronics*, **III**, 69–72.

Bastankhah, M., Welch, B. L., Martínez-Tossas, L. A., King, J., & Fleming, P. (2020). Analytical solution for the cumulative wake of wind turbines in wind farms. arXiv:2011.00894.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2019). A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686–707.

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems* (pp. 5099–5108).

Wang, H., Cao, Y., Huang, Z., Liu, Y., Hu, P., Luo, X., Song, Z., Zhao, W., Liu, J., Sun, J., Zhang, S., Wei, L., Wang, Y., Wu, T., Ma, Z.-M., & Sun, Y. (2024). Recent Advances on Machine Learning for Computational Fluid Dynamics: A Survey. *arXiv preprint arXiv:2408.12171*.

Liu, F., Ren, Y., Liu, Y. H., and Fu, J. Y. (2025). Impact of super high-rise buildings on wind comfort and safety of pedestrian wind environment: A case study in Shanghai, China. *Case Studies in Thermal Engineering*, 71: 106197. https://doi.org/10.1016/j.csite.2025.106197.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph Attention Networks. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1710.10903

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2017). Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. *arXiv preprint arXiv:1711.10561*. https://arxiv.org/abs/1711.10561

Ferreira, J. M. L. L., Ferreira, L. M. M., & Ferreira, J. M. M. (2015). Reducing the memory footprint in large eddy simulations of reactive flows. *Computers & Fluids*, 117, 1–12.

Caron, F., Duflos, E., Pomorski, D., & Vanheeghe, P. (2006). GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects. *Information Fusion*, 7(2), 221–230. Elsevier. https://doi.org/10.1016/j.inffus.2004.07.002