

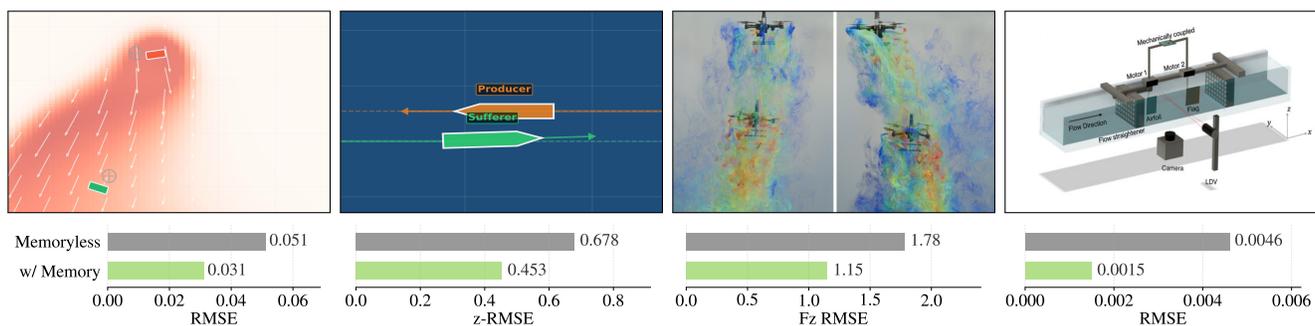
# Wake Up to the Past: Using Memory to Model Fluid Wake Effects on Robots

Luca Vendruscolo\*

Eduardo Sebastián

Amanda Prorok

Ajay Shankar



**Fig. 1:** Comparing models with and without memory: we study the influence of modeling temporal context in the prediction of fluid wake effects in four domains, including aerial and aquatic scenarios. In dynamic regimes, data-driven predictors must support history and learn the transport delay from the medium. Illustrations of the experimental domains in the third and fourth column are extracted from [1] and [2].

**Abstract**—Autonomous aerial and aquatic robots that attain mobility by perturbing their medium, such as multicopters and torpedoes, produce wake effects that act as disturbances for adjacent robots. Wake effects are hard to model and predict due to the chaotic spatio-temporal dynamics of the fluid, entangled with the physical geometry of the robots and their complex motion patterns. Data-driven approaches using neural networks typically learn a memory-less function that maps the current states of the two robots to a force observed by the “sufferer” robot. Such models often perform poorly in agile scenarios: since the wake effect has a finite propagation time, the disturbance observed by a sufferer robot is some function of relative states in the past. In this work, we present an empirical study of the properties a wake-effect predictor must satisfy to accurately model the interactions between two robots mediated by a fluid. We explore seven data-driven models designed to capture the spatio-temporal evolution of fluid wake effects in four different media. This allows us to introspect the models and analyze the reasons why certain features enable improved accuracy in prediction across predictors and fluids. As experimental validation, we develop a planar rectilinear gantry for two spinning monicopters to test in real-world data with feedback control. The conclusion is that support of history of previous states as input and transport delay prediction substantially helps to learn an accurate wake-effect predictor—[sites.google.com/view/wake-up-to-the-past](https://sites.google.com/view/wake-up-to-the-past)

## I. INTRODUCTION

Robots that fly or swim generate momentum by pushing against the air or water around them. This interaction perturbs the medium and produces fluid wake effects—such as the downwash from a multicopter or the trailing

vortex of a torpedo—that act as significant disturbances for adjacent robots [3]. Accurate modeling and compensating for these effects is thus crucial for safe and reliable operation of a fleet in close proximity [4], [5]. While humans can manually correct for these forces during specific maneuvers like mid-flight refueling or ship docking, robot deployments with unknown motion patterns require predictors that are accurate and computationally lean, as real-time low-latency disturbance compensation is crucial.

The primary challenge in modeling these effects lies in the chaotic spatio-temporal dynamics of the fluid, which are inextricably linked to the physical geometry and complex motion of the robots [3], [6]. Existing observation schemes typically include current velocity and relative velocity as variables that inform the evolution of the interaction between the robots [7], [8]. However, instantaneous information is insufficient for capturing the *mechanical* aspects of the medium, in particular, transport delays in the propagation of wake effects. Because wake effects have a finite propagation time, the disturbance currently affecting a sufferer robot is in fact a function of the “wake source” robot’s relative state at a point in the past. Without temporal context, the sufferer is unable to correlate the current relative state with the current disturbance. Consequently, memory-less models perform poorly in agile scenarios. In spite of this dependency, there is a dearth of analyses that incorporate history and recurrence in wake-effect prediction models for robots.

The goal of this paper is to fill this gap and shed light on the properties a data-driven predictor must satisfy to guarantee accurate modeling of wake effects at a low computational cost, such that the predictor can be deployed in real-time

Authors are with the Department of Computer Science and Technology, University of Cambridge, UK. \*Author for correspondence; e-mails: {lv407, es2121, asp45, as3233}@cst.cam.ac.uk.

feedback. Specifically, our contributions are:

- An empirical study of wake effects across seven data-driven architectures and four domains. The architectures range from simple history-based MLPs to advanced sequential models such as Mamba [9], to determine which temporal mechanisms best capture fluid dynamics. The domains include 2D particle-based simulations to datasets from computational fluid dynamics simulators with aerial and aquatic robots. We identify two key enablers for accurate predictions: a representation of past states in the input, and a module that explicitly predicts transport delay.
- The design of a planar rectilinear gantry for two tethered monocopters, on which we collect real-world interaction data under time-varying wake source thrust and validate the prediction accuracy of each architecture.

**Related Work.** Modeling inter-robot wake interaction effects has seen substantial interest in recent years, particularly in the aerial domain. Multirotor downwash, for instance, can exhibit high variability in turbulence depending on the flight regime [10], and has non-trivial effects on neighboring robots [3]. These effects have been studied extensively using numerical models [1], [10], [11] as well as some empirical validation [8], [12]. Computational and numeric fluid modeling has also similarly been applied for aquatic surface and sub-surface vehicles [13], [14].

Of particular interest is the ability to incorporate a given model into a real-time feedback control and planning framework, i.e., the ability to *reject* the disturbance. To this end, recent work has modeled these effects on multirotors as forces and torques acting on a sufferer, and has developed methods to predict them using numerical models [8], and neural networks with differential [15]–[18] and geometric priors [1], [7]. In aquatic domains, distributed networks of sensors have been integrated with hydrodynamic models to supply the information needed to predict the forces suffered by underwater vehicles [19], [20]. Early research even considered wake-effect predictors for control compensation in aquatic manipulation [21], [22]. From a biological perspective, studies on fish schooling [5] have shown the importance of wake-effect prediction not only for stability, but also for energy efficiency in robot motion. Inspired by them, current methods for aquatic robots explore computational fluid simulators and pressure-based analytical methods for soft robots that mimic the mechanical structure of the animals [23]–[25]. Similar approaches are found in the field of bio-inspired flapping-wing robots [26]–[29].

While the aforementioned data-driven approaches have shown impressive results, we observe that the regimes where such models typically tend to perform well are slow-moving and near-steady-state, laminar flow. This assumption admits modeling the observed effect at a given time as a function of the relative state *at that time*. However, wake effects have a finite propagation time in any medium. Thus, in highly dynamic settings, the impact on a sufferer is a result of some relative state in the past, and the instantaneous state is a poor

approximation for modeling. This leads to the premise of this work, which we will analyze in the remainder of the paper.

## II. PRELIMINARIES

Consider two robots, a wake source and a sufferer, moving in a fluid medium. The wake source’s actuation at some time  $t$  perturbs the medium, creating a wake effect (e.g., downwash or trailing vortices). This perturbation propagates through the fluid at a finite speed and eventually reaches the sufferer at time  $t + \Delta t$ , which we model as a disturbance force  $F_{db} \in \mathbb{R}^3$ . The objective of a wake-effect predictor is to learn a mapping  $\Phi$  that *predicts* the instantaneous disturbance force  $F_{db}(t + \Delta t)$  experienced by the sufferer.

We assume that the wake source and sufferer can cooperate, and thus, wake source can communicate unobservable information (such as its thrust vector) to the sufferer when necessary. Similar to prior work, we also assume that the sufferer has a measure of  $F_{db}$  through disturbance predictors [7], direct sensing [8], [30], or simulators [1]. For the most generalizable setting, we collect all relevant data into a time-indexed dataset comprised of the full state of the two robots and  $F_{db}$ . The full state information is used to build an observation vector  $o(t) \in \mathbb{R}^d$ , which is then preprocessed and/or concatenated with past data (depending on the model architecture) to form an input vector  $O(t)$ . We train a predictor over this dataset using a supervised loss to learn the mapping  $\Phi$  with a neural network  $\hat{F}_{db}(t) = f(O(t), \theta)$ , where  $\theta$  denotes the learnable parameters of the model and  $\hat{\cdot}$  denotes predicted force. We use a weighted mean-squared error (MSE) loss of the form

$$\mathcal{L} = w_x \text{MSE}(\hat{f}_x, f_x) + w_y \text{MSE}(\hat{f}_y, f_y) + w_z \text{MSE}(\hat{f}_z, f_z),$$

where  $f_x, f_y$  are the horizontal force components of  $F_{db}$  parallel to the sufferer’s geometric plane and  $f_z$  is the perpendicular component, expressed in the sufferer’s frame of reference. A higher weight (e.g.,  $10\times$ ) is typically applied to the horizontal components to compensate for their smaller magnitudes relative to vertical forces.

The goal of this work is to study the properties that  $O$  and  $f$  must satisfy in order to make accurate predictions of  $F_{db}$ . In particular, we test the key hypothesis that encoding temporal context and modeling propagation delay  $\Delta t$  are essential in highly agile and dynamic settings. The specific observation vector  $O$  varies across domains, but generally encodes relative position and velocity between robots, actuation inputs (e.g., thrust or rudder commands), and domain-specific kinematic quantities. The dimensionality of the observations range from 3 (fish schooling) to 14 (3D multirotor computational fluid dynamics).

## III. MODEL ARCHITECTURES

The modeling of fluid wake effects presents a significant challenge due to the chaotic spatio-temporal dynamics of the interactions between robots and medium. We investigate a diverse range of data-driven predictors categorized by their mechanism for handling temporal context. In this section, we describe each of the models, detailing their relevant

TABLE I: Summary of model architectures. Parameter counts refer to trainable weights.

Architecture	Parameters	Temporal Mechanism	Optimized Hyperparameters	Category
Agile MLP [1]	18K	Instantaneous input features	#layers, hidden dimensions	● Memoryless
History MLP [32]	49K	Flattened window of past snapshots	#snapshots, hidden dimensions	● Explicit History
Delay Embedding [33]	35K	Learned Gaussian kernel over history	$\mu_0, \sigma_0$ , selector dimensions	● Explicit History
GRU [34]	71K	Learned hidden state	hidden dimensions, dropout	● Recurrent
TCN [35]	73K	Causal dilated 1D convolutions	#layers, kernel size, dropout	● Recurrent
Mamba [9]	9K	Selective state-space model	$d_{\text{model}}, d_{\text{state}}$ , convolution width	● Recurrent
RC (ESN) [36]	2K*	Hidden state from fixed random reservoir	reservoir size, spectral radius, leak rate	● Recurrent
Cross-Attention [37]	16K	Weighted combination of processed tokens	embed dimensions, #heads	● Attention-based

\*RC (ESN) uses additional 1,008K frozen weights in the reservoir.

hyperparameters and the rationale behind their architecture (and summarized in Table I). The goal is to build a common framework to make results comparable across domains. We remark that the specific values of the hyperparameters of all models (per model and for each domain) have been found through Bayesian optimization using Optuna [31], keeping the total number of trainable parameters below 100K to ensure the models are computationally lean and, potentially, implementable on board a robot.

#### A. Memoryless Baseline

Agile MLP [1] is a memory-less feedforward network that maps the  $d$ -dimensional input equivariant feature vector directly to a 3D force prediction in the equivariant frame. It assumes the fluid wake is a function of the instantaneous relative state (geometry, velocity, thrust). The architecture is a single MLP with ReLU activations. It has recently been shown that this parameterization outperforms other existing memory-less models in pair-wise downwash prediction in quadrotors [7], [15]–[17], justifying it as a primary baseline.

#### B. Explicit History Models

These models explicitly provide a history of past snapshots to capture temporal changes without recurrence.

1) *History MLP*: The History MLP concatenates multiple past snapshots into a single flattened input vector. By observing a sliding window of recent history, the model can infer which wake patterns generated in the past are arriving at the sufferer now. It utilizes evenly-spaced snapshots covering a tunable time window. The architecture mirrors Agile MLP but with an input layer sized for the flattened history.

2) *Delay Embedding*: The Delay Embedding model learns to pick an snapshot at time  $\tau$  rather than processing a full window of snapshots. Since the wake travels at a finite speed, this model uses a selector to identify the specific past state that caused the wake effect. A selector MLP maps past snapshots to Gaussian attention parameters (delay center  $\mu$  and kernel width  $\sigma$ ) to weigh historical observations. In practice, it employs a “delay gap” to mask the most recent snapshots, forcing the model to rely on past observations.

#### C. Recurrent Models

These models maintain an internal hidden state to summarize the history of observations, rather than explicitly holding an explicit time-series in the input.

1) *Gated Recurrent Unit (GRU)*: GRUs process observations sequentially, maintaining a compressed summary of the history in a hidden state. Learned gates (reset and update) allow the model to selectively remember stable wake patterns or rapidly update the state during agile maneuvers. The architecture includes a linear input projection layer, a GRU, and an output projection layer.

2) *Temporal Convolutional Network*: The Temporal Convolutional Network (TCN) processes trajectories using causal dilated 1D convolutions. By using exponentially increasing dilations, the TCN builds a wide receptive field that can detect temporal patterns at multiple timescales simultaneously. Its architecture stacks multiple layers to cover a receptive field well beyond the typical transport delays.

3) *Mamba*: Mamba is a selective state-space model that makes the recurrence parameters dependent on the input. It allows the model to selectively retain or discard information at every timestep, providing a learned counterpart to fixed-recurrence systems. The model dimension and state dimension are tuned per domain via Optuna.

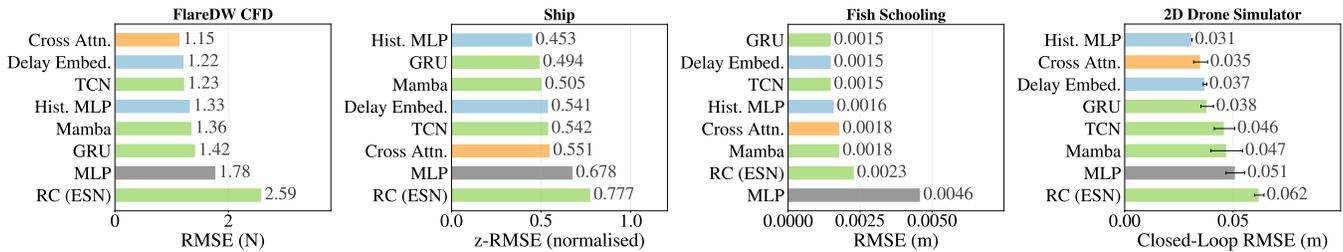
4) *Reservoir Computing with Echo State Networks*: Reservoir Computing with Echo State Networks (RC-ESN) uses a large, fixed, randomly wired recurrent reservoir and only learns the readout layer. Inspired by cognitive models of the brain, a high-dimensional random dynamical system naturally creates “echoes” of past inputs, allowing a linear readout to pick the most predictive temporal features. The architecture uses 1000 reservoir neurons with a sparse connectivity recurrent matrix. Only the readout linear output layer is trained, although it is important to remark that the reservoir involves a set of frozen weights that contribute to computation, even if they are not learned.

#### D. Cross-Attention

The Cross-Attention model allows the sufferer to selectively query relevant moments from the wake source’s past trajectory. By dynamically weighing the relative states, the model accounts for the motion of both robots during the interaction. In practice, the network is based on multi-head cross-attention with fixed sinusoidal positional encodings.

## IV. EVALUATIONS

We now evaluate each of the temporal-context models, as well as memory-less baselines on four different evaluation domains: a CFD dataset for two quadrotors [1], a model



**Fig. 2:** Performance comparison of each of the methods on the four domains. **(left)** A CFD dataset for two quadrotors [1]. **(center left)** A numerical model of ship-ship interactions [38]. **(center right)** A dataset that captures hydrodynamic interactions between two fish swimming in line [2]. **(right)** Closed-loop tracking on a custom 2D downwash simulator.

for ship-to-ship interactions in encounter and overtaking maneuvers [38], a dataset that captures hydrodynamic interactions between two fish swimming in line [2], and a custom 2D multicopter downwash simulator. In the first three domains, we focus our study on the accuracy of the models in predicting the wake-effect disturbance. In the 2D simulator, we additionally consider the closed-loop trajectory tracking performance when the predictions of the model are incorporated into the control loop. All the results are averaged over 3 seeds per model. Finally, we evaluate the models on our real-world experimentation setup using two tethered monocoverters on a gantry.

Figure 1 presents a glimpse of the conclusions of the study: we already see that models that incorporate temporal context outperform those that do not. The bar plots for each of the four domains show the performance of the memory-less MLP baseline, compared against the best performing memory-based architecture in each domain. We now analyze the results for all methods and domains in detail<sup>1</sup>.

### A. Agile quadrotors interactions

We begin our analysis on the influence of temporal context using datasets from prior work, FlareDW [1], which utilizes a custom high-fidelity CFD simulator to model interactions between two P600 quadrotors (3 kg mass, 0.6 m rotor-to-rotor distance). It enables us to explore agile, non-stationary downwash disturbances at velocities ranging from 0.5 m/s to 4.0 m/s, well beyond the low-speed regimes ( $< 0.5$  m/s) typically addressed in prior work. Data collection, recorded at 200 Hz, encompasses four distinct maneuvers: Fly Below, Fly Above, Swapping (both moving), and Fast Swapping (up to 8 m/s relative velocity). The resulting dataset, derived from 30 minutes of simulated flight, incorporates absolute velocities, thrust vectors and relative states to capture the distorted, time-delayed force profile observed at higher speeds.

Figure 2 shows the performance of each of the models on this dataset. We report root mean squared error (RMSE) in predicting downwash force, measured in Newtons. All temporal models excluding RC outperform the baseline MLP. We hypothesize that RC struggles to retain information from the wide variety of scenarios in the FlareDW database using only the 2K trainable parameters. Cross-Attention and Delay Embedding outperform the other methods by providing the

predictor with a direct estimate of the wake source’s state at the time of wake creation. History MLP, despite access to history, can struggle when the true delay falls between sliding windows. The sequential models (GRU, TCN, Mamba) are competitive but the relevant past state is diluted through recurrent processing, yielding noisier estimates.

### B. Ship encounter and overtaking

Next, we evaluate wake effects in maritime environments using a numerical model of ship-to-ship interaction forces [38]. This model specifically investigates sway forces and yaw moments experienced by vessels during encounter and overtaking maneuvers in restricted channels. Data are generated using a discrete vortex distribution numerical technique and slender body theory, which is particularly suited for high-speed, fine-form craft. Crucially, this numerical framework has been empirically validated in prior research [39], showing reasonably good agreement with experimental results. The dataset focuses on the influence of water depth-to-draught ratio, lateral separation distance, and relative ship speeds. The models are trained to predict the sway force and yaw moment experienced by the sufferer ship. Unlike the original formulation [38], which assumes fixed parallel paths, we introduce time-varying lateral targets for the sufferer that are not included in the observation vector. This creates a temporal ambiguity that a memory-less model cannot resolve: from a single observation, a model without temporal context cannot distinguish whether the sufferer is veering due to hydrodynamic interaction forces or because it is tracking a new lateral target. Only models with access to temporal context can disambiguate by observing the sufferer’s trajectory over time.

Figure 2 shows the performance of each of the models in this domain. As expected, we observe that models that incorporate history generally outperform the baseline MLP. However, in contrast to the results from FlareDW (Section IV-A), here we observe that History MLP performs the best. Unlike the downwash domains where the transport delay is roughly constant, the ship encounter produces a continuously varying delay as the vessels approach and recede, making it harder for explicit-delay models to learn a single representative lag. A simple sliding window suffices, as the relative observation features already encode the encounter phase.

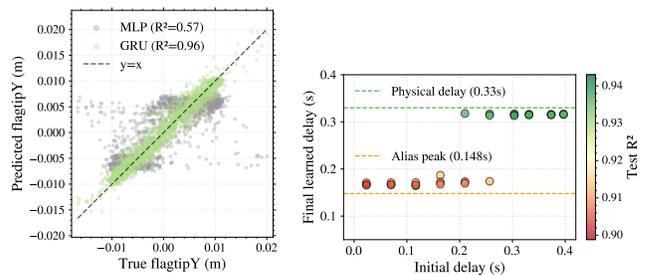
<sup>1</sup>We provide more materials on the [website](#).

### C. Fish Schooling

Next, we use data from an experimental setup for fish schooling that uses a robotic platform designed to replicate a simplified two-fish subsystem [2]. The system consists of an actively pitching NACA 0012 airfoil positioned upstream and a compliant Mylar flag located 5.4 cm downstream. These components interact via two distinct pathways: a hydrodynamic pathway mediated by vortex shedding from the airfoil, and an electromechanical pathway where the flag’s leading rib is actuated to follow the airfoil’s motion with a controlled time delay. We use only the single-pathway (hydrodynamic) condition, in which the flag responds passively, to isolate wake-mediated coupling. Data collection includes high-resolution camera tracking of component movements at 60 Hz, downsampled to 30 Hz for training, and laser Doppler velocimetry (LDV) to record streamwise flow velocity at an intermediate point between the bodies. The input features are airfoil pitching angle, LDV velocity, and flag front displacement; the prediction target is flag tip displacement. Causal interactions and associated time lags are then disentangled using an information-theoretic approach based on transfer entropy, which identified a causal delay of  $\sim 0.33$  s, which subsequently informs our 0.5 s history window. This is an interesting domain since the disturbances are sinusoidal, and follow a nearly constant period and transport delay.

The performance of each of the models is shown in Figure 2. Similar to the prior two domains, models that incorporate temporal context are the best-performing. To investigate further, in Figure 3 we additionally report the coefficient of determination for the best (GRU) and the worst performer (baseline MLP). Although the baseline MLP achieves a small RMSE, we observe that it does so with an  $R^2$  of only 0.57. In addition, Figure 3 also shows that the memory-less MLP has two correlation modes (clusters) due to the phase ambiguity described above. We attribute this to two factors. First, the wake vortices take  $\approx 0.33$  s to travel from the airfoil to the flag, so the MLP correlates the current foil angle with a flag response that was caused by a past state it cannot observe. Second, the sinusoidal pitching introduces a phase ambiguity: the same foil angle occurs twice per cycle, once in ascent and the again in descent, producing different wake structures each time. Without temporal context, the MLP cannot distinguish these phases. Furthermore, the airfoil occasionally makes motions that deviate from the standard oscillation, producing non-periodic wake structures that a memory-less model cannot anticipate. History-aware models resolve these limitations by observing the trajectory of the foil angle over time, recovering the phase and detecting transient deviations. Recurrent networks, including GRUs, are inherently designed to process sequential data, making them effective at recognizing repeating patterns over time. Figure 3 shows that GRU, with  $R^2 = 0.96$ , presents a more uniform spread about  $y = x$ , which indicates an unbiased modeling of the wake effect.

Compared to the two previous scenarios, models that explicitly predict transport delay, namely Delay Embedding



**Fig. 3:** Fish schooling analysis. **(left)** Scatter plot of true vs. predicted disturbances. The memoryless MLP ( $R^2 = 0.57$ ) exhibits two distinct modes, failing to distinguish periodic phases, while the GRU achieves accurate, unbiased predictions. **(right)** Convergence basin of the Delay Embedding model. Depending on the random initialization of  $\mu$ , the model converges to either the true physical delay (0.33 s, higher  $R^2$ ) or an alias peak (0.148 s, lower  $R^2$ ).

and Cross-Attention, are not as effective in this domain. We investigate this by analyzing the Delay Embedding model, which exposes its learned delay through the  $\mu$  parameter. Figure 3 compares the final predicted delay across different random initializations of  $\mu$ . Because the disturbance is nearly sinusoidal, shifted copies of the signal at multiples of the half-period correlate almost as well as the true delay, creating alias correlation peaks that act as local minima during optimization. As a result, the model converges to the true physical delay only when  $\mu$  is initialized in its basin of attraction; otherwise, it settles on an alias peak with lower  $R^2$ . This sensitivity is specific to domains with constant-frequency oscillations. In domains with richer temporal structure, such as FlareDW and the 2D downwash simulator, the Delay Embedding model reliably converges to the true physical delay regardless of the initialization of  $\mu$ .

### D. 2D Downwash Simulator

We now introduce a custom 2D multicopter downwash simulation that models the velocity field generated by a wake source multicopter. The simulator models the multicopters as planar rigid bodies (XZ plane) that tilt and produce 2D thrust vectors. The velocity field  $\mathbf{v} = (v_x, v_z)$  evolves on a 2D Eulerian grid with cell sizes  $\Delta x, \Delta z$  through four operations applied sequentially at timestep  $\Delta t$ , similar to what is done in [40], [41]. First, the wake source at position  $(p_x, p_z)$  with pitch  $\theta$  injects momentum using a Gaussian spatial profile,

$$\Delta \mathbf{v}_{ij} = S \sqrt{\frac{u}{u_{\text{hover}}}} \exp\left(-\frac{r_{ij}^2}{2\sigma^2}\right) \Delta t \begin{pmatrix} \sin \theta \\ -\cos \theta \end{pmatrix},$$

where  $r_{ij}^2 = (x_i - p_x)^2 + (z_j - p_z)^2$ ,  $S$  is the injection strength,  $\sigma$  is the spatial width,  $u$  and  $u_{\text{hover}}$  denote the wake source’s thrust and the thrust at hover, and  $(\sin \theta, -\cos \theta)^\top$  is a unit vector opposite to the body-frame thrust vector. Second, the field advects downward at a constant speed  $w_a = 6$  m/s using an upwind finite-difference scheme, followed by a semi-Lagrangian horizontal advection that traces each cell back to its upstream source position  $x_{\text{src}} = i - v_x[i, j] \cdot \Delta t / \Delta x$ . Third, diffusion is modeled as a Gaussian blur with standard deviation  $\ell_d = \sqrt{2\kappa\Delta t}$ , equivalent to solving  $\partial \mathbf{v} / \partial t = \kappa \nabla^2 \mathbf{v}$  for one timestep, with

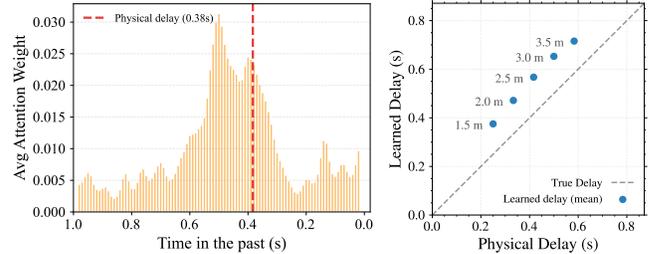
$\kappa = 0.3 \text{ m}^2/\text{s}$  the diffusivity parameter. Finally, energy dissipation is captured by an exponential decay,  $\mathbf{v} \leftarrow \mathbf{v} \cdot e^{-\lambda \Delta t}$ . The downward advection produces an emergent transport delay of  $\Delta z/w_a \approx 0.38 \text{ s}$  for a default vertical separation of  $\Delta z = 2.3 \text{ m}$ . The disturbance force at the sufferer is computed as quadratic drag,  $F_{db} = \frac{1}{2} \rho C_d \mathbf{v} \|\mathbf{v}\|$ , evaluated at the sufferer’s position via bilinear interpolation. Each multicopter uses a Linear Quadratic Regulator (LQR) for flight, and the predicted interaction forces are incorporated as feedforward thrust and pitch corrections. This allows us to evaluate closed-loop tracking performance under three conditions: a baseline with no compensation, using an oracle with access to the true forces, and using predictions from each of the trained models. The observation is an 8D vector comprising relative states and thrust components, and the output is the 2D disturbance force, all recorded at 100 Hz.

Figure 2 shows the tracking performance of the sufferer when following random pre-planned trajectories. We observe again that models that incorporate temporal context outperform the baseline MLP. To further validate that the models learn the correct temporal structure, Figure 4 visualizes the average attention weights of the Cross-Attention model across all episodes. Since the vertical separation between the two multicopters is constant, the transport delay is fixed across all episodes and attention weights can be meaningfully averaged. The attention profile reveals a clear peak at 0.5 s in the past, close to the true physical transport delay  $\Delta z/w_a$  at 0.38 s. Furthermore, the distribution of weights is concentrated in a window around the predicted delay, with negligible weight assigned to more recent or distant time steps. We also compare the true vs learned delay in the Delay Embedding model for five different  $\Delta z$  (Figure 4, right). The model clearly learns to attend to the past states of the wake-source, albeit with an added  $\approx 0.12 \text{ s}$  shift across all tests (similar to Cross-Attention). We hypothesize that this shift is a mechanism learned in both models for conservativeness and in-distribution generalization: looking slightly beyond the true transport delay can reveal more information about relative motion of the source.

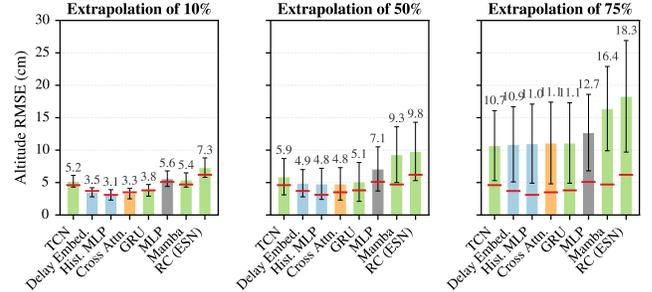
Finally, we leverage the customization capabilities of the simulator to test how well the different models generalize to fluid conditions out of the training distribution. We train all models with wake speed, diffusivity, and vertical separation sampled uniformly at  $\pm 10\%$  of their nominal values ( $w_a = 6.0 \text{ m/s}$ ,  $\kappa = 0.30 \text{ m}^2/\text{s}$ ,  $\Delta z = 2.3 \text{ m}$ ) and evaluate closed-loop RMSE under increasingly perturbed conditions:  $\pm 10\%$ ,  $\pm 50\%$ , and  $\pm 75\%$ . As shown in Figure 5, performance retains the same relative ordering across conditions. Therefore, the performance gains achieved by the memory-based models do not come from overfitting to the training dataset—“just” due to a greater representational capacity—but because memory-based models are better suited than memory-less models to capture the underlying wake-effect dynamics.

### E. Real tethered monicopters

Finally, we perform validations on real hardware using a custom in-house rectilinear 2D gantry mechanism. The



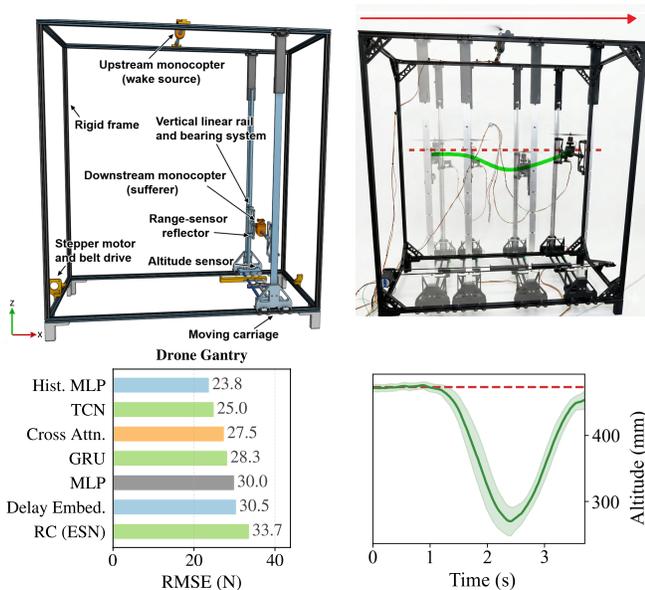
**Fig. 4:** (left) Average attention weights of the Cross-Attention model over all episodes in the 2D downwash simulator. The profile shows a clear peak near the true physical delay (red dashed line at 0.38 s), demonstrating that the model attends to the correct time lag. (right) Learned versus physical delay for the Delay Embedding model across five values of  $\Delta z$ .



**Fig. 5:** Out-of-distribution closed-loop ablation on the 2D downwash simulator. Models are trained with physics parameters sampled at  $\pm 10\%$  of their nominal values and evaluated at  $\pm 10\%$ ,  $\pm 50\%$ , and  $\pm 75\%$  perturbation levels. Red lines denote the baseline closed-loop RMSE from a 3-seed evaluation trained and evaluated at the nominal (unperturbed) conditions. Degradation preserves the relative ranking across conditions.

system, shown in Figure 6, tethers two monicopters to enable constrained motion in the XZ plane (similar to the 2D simulation framework above). The motion in the vertical axis is supported by two SBR12 linear rails with metal bearings that have very low static and dynamic friction. The lateral movement is effected using a Nema 23 stepper motor with a belt drive system. Each monicopter is built using an off-the-shelf brushless DC outrunner motor, which is driven by a commercial off-the-shelf brushless electronic speed controller. The feedback control and measurement is implemented on a Raspberry Pi 5, which interfaces with an Arduino UNO (ATmega328P) microcontroller to generate PWM drive signals for the ESCs. A VL53L0X Time-of-Flight sensor mounted at the base measures the altitude, while the lateral position is measured using open-loop step-per step counting with a calibrated constant of 4200 steps/m. We use a Kalman filter to estimate position and velocity at 32 Hz during each episode. Acceleration is obtained offline via a Rauch-Tung-Striebel smoother, which runs a forward-backward pass over the recorded data to produce near-zero-lag estimates. The total round-trip communication and sensing latency is empirically determined to be  $\approx 18 \text{ ms}$ .

During each episode, both the lateral velocity of the carriage and the RPM of the wake source monicopters are varied continuously, creating a rapidly changing downwash field. We do not model the response time of the motor and the ESC to a new PWM command. Thus, it is likely that the commands sent to the wake source sometimes change faster than the mechanical spool rate of the system. This



**Fig. 6:** The tethered monocopters gantry system. **(top left)** The wake source can move freely in the X axis, while the sufferer can move freely in the XZ plane. The sensors interface with an Arduino UNO and a Raspberry Pi 5 to implement feedback control with/without disturbance rejection. **(top right)** Four overlay snapshots from an episode where the sufferer attempts to track a reference trajectory (dashed red) and deviates significantly (green curve) due to the time-varying turbulence caused by the wake source. **(bottom left)** Memory-based predictors achieve better open-loop prediction performance than the memory-less baseline, matching numerical and simulated evaluations in Figure 2. **(bottom right)** Mean altitude trajectory across  $n=5$  repeated PID-only episodes on the same randomization seed. The dashed red line indicates the target altitude; the solid green curve shows the mean position; the shaded region denotes  $\pm 1\sigma$ . The characteristic downwash-induced dip is clearly visible.

makes the instantaneous RPM (and therefore the downwash) a function of a sliding window of some last  $n$  PWM commands. This effect compounds with the wake transport delay, further increasing the temporal context required for accurate prediction. A memory-less model observing only the current PWM command cannot recover either lag, whereas temporal models can infer the true motor state from the recent history of commands. This also explains why Delay Embedding performs very close to the baseline MLP: it attends to a single past snapshot at the learned delay, which cannot capture the moving-average nature of the motor response.

## V. DISCUSSION

Our empirical results establish that models with temporal context consistently and substantially outperform memory-less baselines in modeling fluid wake effects. While prior work often assumes steady-state or laminar conditions where instantaneous observations suffice, our study demonstrates that in agile regimes the finite propagation time of disturbances makes memory-less approximations inadequate.

The consistent performance of memory-based architectures over the memory-less baseline confirms that the sufferer robot’s state is inextricably linked to the wake source’s past actions. However, the optimal temporal mechanism can vary depending on the physical characteristics of the medium and the motion patterns involved. A key finding is the efficacy of explicit delay prediction. In FlareDW and the 2D Simulator,

Cross-Attention and Delay Embedding excel by providing a direct estimate of the wake source’s state at the time of wake creation, giving the predictor an unambiguous causal signal. Explicit-delay models struggle when the delay is not clearly identifiable: in Fish Schooling, periodicity creates alias peaks, and in Ship Encounter the changing inter-vessel distance makes the delay non-stationary. Recurrent models like GRUs offer a robust alternative by implicitly encoding the relevant history in a hidden state. RC (ESN), despite its large reservoir, is constrained by its low trainable capacity and succeeds only when the temporal structure is simple.

A key **trade-off** is evident in the computational cost. While temporal models provide the accuracy needed for safe proximity flight, they require more data and compute for training and real-time inference. History MLP is the simplest temporal model and performs consistently well across all domains, offering a practical default for adding wake-effect compensation to any downwash problem with a lower computational footprint than the sequential alternatives.

Consequently, we summarize the findings of the paper in two complementary features that an accurate wake-effect predictor must have: *(i)* some explicit representation of temporal context in its input, and, *(ii)* a module that explicitly predicts transport delay. A **limitation** we observe in this study is the absence of a clear ‘winner’ among the models with temporal context across domains. The combination of both mechanisms is an interesting line for further analysis.

## VI. CONCLUSION

This paper has demonstrated that capturing the spatio-temporal evolution of fluid wake effects is essential for maintaining the stability of autonomous aerial and aquatic robots. Our core findings have revealed that the inclusion of temporal context significantly improves force prediction accuracy compared to memory-less baselines, as these models have successfully accounted for the finite propagation time of fluid disturbances. We have shown that data-driven models are capable of learning physical transport delays directly from data without explicit physics programming. While no single neural architecture has outperformed all others across every experimental domain, the necessity for models to support the history of previous states and include a transport delay predictor has emerged as a universal requirement. Our evaluations have confirmed that the benefits of temporal context are most pronounced in agile motion regimes where current state features are insufficient to encode fluid dynamics.

Future work will focus on the real-world validation of these models within full closed-loop controller integrations and extending the framework to handle interactions between more than two agents. This latter focus is especially critical for swarm autonomy, where emergent interactions are typically more complex than the sum of their individual parts.

## REFERENCES

- [1] P. Kharitenko, Y. Fan, X. Liu, and Y. Wang, “A spatiotemporal downwash modeling for agile close-proximity multirotor flight,” in

- IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2025, pp. 807–812.
- [2] R. Das, S. D. Peterson, and M. Porfiri, “Disentangling coexisting sensory pathways of interaction in schooling fish,” *Flow*, 2025.
  - [3] J. Gielis, A. Shankar, R. Kortvelesy, and A. Prorok, “Modeling aggregate downwash forces for dense multirotor flight,” in *International Symposium on Experimental Robotics*. Springer, 2023, pp. 393–404.
  - [4] A. Shankar, H. Woo, and A. Prorok, “Docking multirotors in close proximity using learnt downwash models,” in *International Symposium on Experimental Robotics*. Springer, 2023, pp. 427–437.
  - [5] L. Li, X. Zheng, R. Mao, and G. Xie, “Energy saving of schooling robotic fish in three-dimensional formations,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1694–1699, 2021.
  - [6] C. Brouzet, C. Raufaste, and M. Argentina, “Undulatory underwater swimming: linking vortex dynamics, thrust and wake structure with a biorobotic fish,” *Journal of Fluid Mechanics*, vol. 1015, p. A53, 2025.
  - [7] H. Smith, A. Shankar, J. Gielis, J. Blumenkamp, and A. Prorok, “So (2)-equivariant downwash models for close proximity flight,” *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1174–1181, 2023.
  - [8] A. Kiran, N. Ayanian, and K. Breuer, “Influence of static and dynamic downwash interactions on multi-quadrotor systems,” in *Robotics: Science and Systems*, 2025.
  - [9] A. Gu and T. Dao, “Mamba: Linear-time sequence modeling with selective state spaces,” in *Conference on Language Modeling*, 2024.
  - [10] H. Zhang, Y. Lan, N. Shen, J. Wu, T. Wang, J. Han, and S. Wen, “Numerical analysis of downwash flow field from quadrotor unmanned aerial vehicles,” *International Journal on Precision Agricultural Aviation*, vol. 3, no. 4, pp. 1–7, 2020.
  - [11] D. Shukla and N. Komerath, “Low reynolds number multirotor aerodynamic wake interactions,” *Experiments in Fluids*, vol. 60, no. 4, p. 77, 2019.
  - [12] K. P. Jain, T. Fortmuller, J. Byun, S. A. Mäkiharju, and M. W. Mueller, “Modeling of aerodynamic disturbances for proximity flight of multirotors,” in *IEEE International Conference on Unmanned Aircraft Systems*, 2019, pp. 1261–1269.
  - [13] S. Tavakoli, P. Shaghaghi, S. Mancini, F. De Luca, and A. Dashtimanesh, “Wake waves of a planing boat: An experimental model,” *Physics of Fluids*, vol. 34, no. 3, 2022.
  - [14] S. Gaggero, S. Brizzolara, et al., “Exact modeling of trailing vorticity in panel method for marine propeller,” *International Conference on Marine Research and Transportation*, vol. 98, 2007.
  - [15] J. Li, L. Han, H. Yu, Y. Lin, Q. Li, and Z. Ren, “Nonlinear mpc for quadrotors in close-proximity flight with neural network downwash prediction,” in *IEEE Conference on Decision and Control*, 2023, pp. 2122–2128.
  - [16] G. Shi, W. Hönig, Y. Yue, and S.-J. Chung, “Neural-swarm: Decentralized close-proximity multirotor control using learned interactions,” in *IEEE International Conference on Robotics and Automation*, 2020, pp. 3241–3247.
  - [17] G. Shi, W. Hönig, X. Shi, Y. Yue, and S.-J. Chung, “Neural-swarm2: Planning and control of heterogeneous multirotor swarms using learned interactions,” *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 1063–1079, 2021.
  - [18] K. Y. Chee, P.-A. Hsieh, G. J. Pappas, and M. A. Hsieh, “Flying quadrotors in tight formations using learning-based model predictive control,” in *IEEE International Conference on Robotics and Automation*, 2025, pp. 10951–10957.
  - [19] M. Krieg, K. Nelson, and K. Mohseni, “Distributed sensing for fluid disturbance compensation and motion control of intelligent robots,” *Nature machine intelligence*, vol. 1, no. 5, pp. 216–224, 2019.
  - [20] K. Nelson and K. Mohseni, “Hydrodynamic force decoupling using a distributed sensory system,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3235–3242, 2020.
  - [21] B. Lévesque and M. J. Richard, “Dynamic analysis of a manipulator in a fluid environment,” *The International journal of robotics research*, vol. 13, no. 3, pp. 221–231, 1994.
  - [22] T. W. McLain and S. M. Rock, “Development and experimental validation of an underwater manipulator hydrodynamic model,” *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 748–759, 1998.
  - [23] Y. Xu and K. Mohseni, “Fish lateral line inspired hydrodynamic force estimation for autonomous underwater vehicle control,” in *IEEE Conference on Decision and Control*, 2013, pp. 6156–6161.
  - [24] —, “Fish lateral line inspired hydrodynamic feedforward control for autonomous underwater vehicles,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 3565–3870.
  - [25] R. Pramanik, R. Verstappen, and P. Onck, “Computational fluid–structure interaction in biology and soft robots: A review,” *Physics of Fluids*, vol. 36, no. 10, 2024.
  - [26] S. Armanini, M. Karásek, G. De Croon, and C. De Visser, “On-board/offboard sensor fusion for high-fidelity flapping-wing robot flight data,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 8, pp. 2121–2132, 2017.
  - [27] D. F. Kurtulus, “Ability to forecast unsteady aerodynamic forces of flapping airfoils by artificial neural network,” *Neural Computing and Applications*, vol. 18, no. 4, pp. 359–368, 2009.
  - [28] C. T. Orłowski and A. R. Girard, “Modeling and simulation of non-linear dynamics of flapping wing micro air vehicles,” *AIAA journal*, vol. 49, no. 5, pp. 969–981, 2011.
  - [29] X. Cai, D. Kolomenskiy, T. Nakata, and H. Liu, “A CFD data-driven aerodynamic model for fast and precise prediction of flapping aerodynamics in various flight velocities,” *Journal of Fluid Mechanics*, vol. 915, p. A114, 2021.
  - [30] E. Davis and P. E. Pounds, “Direct sensing of thrust and velocity for a quadrotor rotor array,” *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1360–1366, 2017.
  - [31] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” in *ACM International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2623–2631.
  - [32] H. Bourlard and C. J. Wellekens, “Links between markov models and multilayer perceptrons,” *Advances in Neural Information Processing Systems*, vol. 1, 1988.
  - [33] G. Yang, “Wide feedforward or recurrent neural networks of any architecture are gaussian processes,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.
  - [34] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.
  - [35] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv preprint arXiv:1803.01271*, 2018.
  - [36] H. Jaeger, “Adaptive nonlinear system identification with echo state networks,” *Advances in Neural Information Processing Systems*, vol. 15, 2002.
  - [37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
  - [38] K. S. Varyani, R. C. McGregor, P. Krishnakutty, and A. Thavalingam, “New empirical and generic models to predict interaction forces for several ships in encounter and overtaking manoeuvres in a channel,” *International Shipbuilding Progress*, vol. 49, no. 4, pp. 237–262, 2002.
  - [39] V. KS, “Identification of trends in extremes of sway-yaw interference for several ships meeting in restricted waters,” *Schiffahrts-Verag Hansa*, vol. 49, pp. 174–191, 2002.
  - [40] K. Chang, S. Chen, M. Wang, X. Xue, and Y. Lan, “Numerical simulation and verification of rotor downwash flow field of plant protection uav at different rotor speeds,” *Frontiers in plant science*, vol. 13, p. 1087636, 2023.
  - [41] C. He and J. Zhao, “Modeling rotor wake dynamics with viscous vortex particle method,” *AIAA journal*, vol. 47, no. 4, pp. 902–915, 2009.