2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Learning to Predict the Wind for Safe Aerial Vehicle Planning

Florian Achermann[1], Nicholas R.J. Lawrance[1], René Ranftl[2],
Alexey Dosovitskiy[2], Jen Jen Chung[1], Roland Siegwart[1]

***Abstract*— Obtaining an accurate estimate of the local wind remains a significant challenge for small unmanned aerial vehicles (UAVs). Small UAVs often operate at low altitudes near terrain, where the wind environment can be more complex than at higher altitudes. Combined with their relatively low mass, this makes small UAVs particularly susceptible to wind. In this paper we present an approach for predicting high-resolution wind fields based on a terrain elevation model and known inflow conditions. Our approach uses a deep convolutional neural network (CNN) to generate 3D wind estimates. We show that our approach produces wind estimates with lower prediction error than existing methods, and that inference can be performed on an on-board computer in less than two seconds. By providing the wind estimate to a sampling-based planner we show that the improved estimates allow the planner to generate safer paths in strong wind scenarios than with alternative wind estimation techniques.**

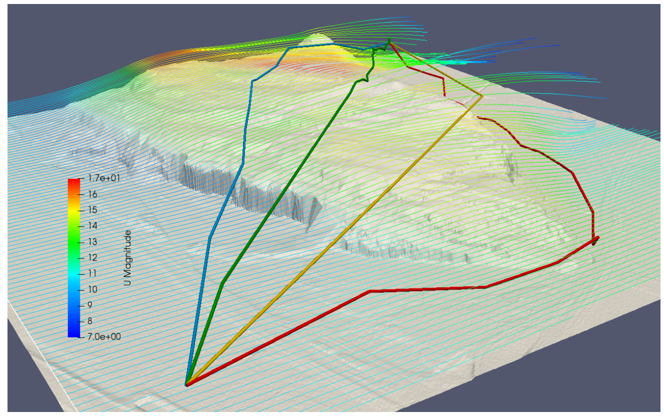

Fig. 1. Paths generated by a sampling-based planner to minimize air-relative distance using wind fields estimated by different approaches. Our wind prediction approach allows the planner to find a safe trajectory to the goal (red path), while baseline methods (green, yellow, blue) lead to generating infeasible paths that would cause the UAV to crash due to its airspeed limitations. This highlights the value of accurate wind prediction for autonomous UAV navigation. Green, yellow, and blue paths were obtained by predicting zero wind everywhere, replicating inflow conditions, and linearly interpolating between the vertical edges of the volume, respectively.

I. INTRODUCTION

Small UAVs provide the capacity to conduct inspection and monitoring tasks in challenging environments. In particular, fixed-wing UAVs enable long endurance flights (compared to multi-rotor UAVs) and can therefore be used in areas that are inaccessible to humans and potentially beyond visual line of sight. Under such circumstances, robust platform autonomy is key to mission success, and this relies on a combination of situational awareness and fast reactive (if not pre-emptive) planning. Further, the robust autonomy required for long-duration flights must also be executed on-board the aircraft, since external communication can be unreliable and have limited bandwidth, and the aircraft must be able to react to unforeseen environmental changes.

Unfortunately, small UAVs are particularly susceptible to wind because naturally-occurring wind speeds can often approach or exceed the airspeed of the UAV. For example, in mountainous areas, the yearly average wind speed can exceed 10 m/s [1], a speed similar to the normal cruise speed of a small UAV [2]. Further, small aircraft have correspondingly low rotational inertia properties, so roll or pitch moments caused by spatial variations in wind at a similar scale to the aircraft size can result in significant rotational moments. These conditions mean that small UAVs often cannot be flown safely in high wind speeds or turbulent wind conditions without risking the safety of the aircraft [3].

Currently, one of the primary sources of data for wind predictions used in UAV flight planning is from large-scale numerical weather prediction (NWP) models. These models generally incorporate observations made using satellite and ground-based sensors with forward simulations based on flow transport equations (such as Navier–Stokes) to solve for flow estimates on a discretized grid [4]. The smaller scale effects of terrain and local thermal variations are often neglected or reduced because the simulation resolution is generally of order 1 km or larger. This limitation often means that NWP predictions are not sufficiently accurate or reliable for use in small UAV operations [3].

Estimating the wind at a resolution relevant to UAV flight planning remains a challenging task, especially because true wind is a spatially and temporally varying three-dimensional vector field. The processes that drive variations in wind range from large-scale atmospheric forces such as pressure differences and Coriolis forces down to smaller-scale influences such as thermal differences and terrain. Furthermore, air is transparent to many remote sensing modalities, while in-situ measurements provide limited value since they often arrive too late for the platform to react. Previous work has shown that (in simulation), if a small UAV could accurately predict the wind, it would allow planning methods that avoid dangerous flows or even make use of favourable wind conditions to improve endurance [5]–[7].

In this paper, we present a learning-based method for predicting the steady (time-averaged) wind flow at meter-scale resolutions from terrain elevation data and upstream wind conditions suitable for inference on an on-board CPU. To generate sufficient training data, we leverage existing

*This work was supported by Intel Network on Intelligent Systems.

[1]Autonomous Systems Lab, ETH Zürich {florian.achermann, nicholas.lawrance, jenjen.chung}@mavt.ethz.ch
[2]Intel Labs {rene.ranftl, alexey.dosovitskiy}@intel.com

(and computationally expensive) Reynolds-averaged Navier–Stokes (RANS) computational fluid dynamics (CFD) estimation techniques to solve for the wind field given any terrain model and inflow condition. Our generated wind flow data, along with the terrain and inflow data, are then used to train a deep CNN, which can then be queried online by an on-board flight controller to generate flight plans that avoid dangerous wind speeds and exploit the flow to improve flight efficiency.

Validation studies on the accuracy of our CNN wind estimates demonstrate a mean absolute prediction error of 0.44 m/s on unobserved terrains and inflows. Furthermore, this error drops to 0.33 m/s when only considering winds within our targeted flight altitudes of 30 m - 500 m. We also tested the efficacy of our wind prediction in a flight planning pipeline. We use a sampling-based planner with a cost function based on air relative distance (equivalent to travel time) to plan paths using the predicted wind field (Fig. 1). Our results show that the planner generates feasible (safe) paths with a higher likelihood and achieves a more accurate cost estimate when using our wind predictions as compared to using existing wind prediction techniques.

II. RELATED WORK

A. Computationally-efficient fluid flow estimation

A range of applications require timely or low computational-cost estimates of fluid flows. One such application is computer graphics [8], [9]. The goal here is typically to calculate the temporal variation of a fluid (including liquids and gases) in response to disturbances in a way that results in stable and visually-realistic solutions. This can be achieved with particle-based methods that simulate larger scale flow by modeling individual particles and solving pressure and flow based on the Navier–Stokes equations across finite volumes in the region of interest. Such methods can solve and render flows with thousands of particles in real time on a standard desktop computer [10]. Recently, traditional particle-based fluid simulation approaches are being combined with data-driven methods to further speed up the simulation. In this line of work, both random forests [11] and deep networks [12], [13] have been used to increase the speed of fluid solutions by multiple orders of magnitude. However, the particle-based methods used in graphics are typically optimized for visual appearance, not physical accuracy, and therefore cannot readily be used for UAV planning applications. Moreover, most particle-based methods assume a closed volume and often a constant number of particles, rather than the flow-through type cases we expect in terrain wind flow.

Steady (time-averaged) flow is commonly used for design and analysis, such as estimating drag for automobile and aircraft design. Umetani and Bickel [14] demonstrated an approach to estimate drag, flow velocities, and pressure around 3D bodies using a Gaussian process based on a novel parameterized shape representation. Baqué et al. [15] use a similar shape parameterization approach combined with a geodesic CNN for shape optimization in aerodynamics applications. Of most relevance to the current paper is the work by Guo et al. [16], who use a CNN to predict time-averaged laminar fluid flow around shapes to speed up industrial aerodynamics applications. They use Lattice Boltzmann Methods to generate training data on two-dimensional shapes of medium complexity such as cars, as well as low-resolution three-dimensional geometric primitives. In contrast, we use RANS CFD solutions on complex and diverse 3D terrain geometries to generate training data, as RANS solutions have demonstrated high-quality predictive performance compared to other methods for complex terrain flow models [17]. We validate our CFD setup against a microscale flow model benchmark [17], [18] to ensure that the generated training data is realistic. Additionally, we propose a CNN architecture that is conditioned on inflow conditions and is capable of predicting high resolution wind fields (64^3) that are required for accurate planning.

B. Wind estimation

A number of research and industrial applications require wind estimates at lateral resolutions higher than 1.1 km, which is the resolution typically provided by numerical weather prediction [19]. One such application is wind turbine 'micro-siting' – identifying and evaluating installation sites based on power potential and safety. CFD remains one of the most popular techniques for turbine siting applications, including both steady (time-averaged) and transient methods such as large eddy simulation (LES) [20]. Modelers also use wind tunnels and simpler numerical methods to analyze energy potential and characterize turbulence [21]. We drew on publications from turbine siting to inform the CFD methods used in this paper, and we view this as a potential application area for the work presented here.

Industrial aerodynamics, particularly for analysis of wind around built environments, is also relevant in terms of desired quantities and scale. Traditional methods used databases and parametric models [22], but CFD has increased in popularity with increased computational capacity [23]. We selected RANS CFD methods for this project due to their maturity in commercial and open source CFD solvers, proven capabilities and low computational cost relative to LES methods.

C. UAV planning in wind

There is also a body of work that deals with gathering in-situ measurements of the wind using UAVs. Applications include monitoring tornado genesis [24]–[27] and cloud evolution [28], [29], as well as autonomous soaring for long endurance flight [30]–[32]. These works tend to focus on the information gathering aspect of the problem since precise wind sampling is constrained along the UAV trajectory. However, in many cases, data from external sensors such as radar [24] or weather balloons [33] are also incorporated into the wind field estimate to enable informative planning.

In our work, we develop a wind prediction pipeline that does not rely on external sensor data and can be executed entirely on-board the UAV. Our system only requires knowledge of the terrain beneath the flight operation region and any coarse weather (wind speed) predictions provided for example by NWP.

III. CFD WIND DATA

In this work, we are interested in predicting the flow over natural terrain. In particular, we approach this as a machine learning problem of training a model offline using representative data and performing online inference during flight. To obtain suitable training data, we used a CFD solver to generate a large dataset of fluid flow solutions over a set of real terrain samples.

A. Terrain models

The terrains used to generate our training data were collected from the Swiss geodata service, which provides access for Swiss researchers to high-resolution (2 m lateral, 0.5 m vertical) elevation data across all of Switzerland[1]. We manually sampled 370 patches of terrain, each measuring approximately 1.2 km square. We favoured selecting terrain patches that contained one side with near-constant elevation, as this allows us to simulate a formed boundary layer flowing into the region from that edge. Each terrain sample was collected as a geographically aligned GEOTIFF elevation file, which was then converted to an STL representation. An example of a terrain patch along with the corresponding CFD flow solution (represented by the streamlines colored by velocity magnitude) is illustrated in Fig. 1.

B. CFD solutions

We are primarily interested in time-averaged estimates as predicting the dynamic wind needs good knowledge of the initial conditions, which is not available on-board on the plane. We elected to use a RANS solver, namely the popular $k-\epsilon$ two-equation turbulence closure [34]. To solve the flow, we used the open source solver OpenFOAM [35]. We created an automated pipeline that ingests terrain patches as STL files and outputs flow solutions over the terrain.

For each terrain, we use the OpenFOAM SnappyHexMesh utility to generate a mesh around the terrain and solve using the steady simpleFoam solver. Wind enters through one face of the domain, perpendicular to the face. The input wind speed U, turbulent kinetic energy k and turbulence dissipation rate ϵ across the inflow face vary with height z, defined using a standard logarithmic boundary layer profile:

$$U = \frac{U^*}{\kappa} \log\left(\frac{z - z_0}{z_0}\right), \quad k \ = \frac{(U^*)^2}{C_\mu^{1/2}}, \quad \epsilon = \frac{(U^*)^3}{\kappa\,(z - z_0)},$$

where the friction velocity U^* is

$$U^* = \kappa U_{ref} \left[\log\left(\frac{(Z_{ref} + z_0)}{z_0}\right)\right]^{-1}. \tag{1}$$

We use standard values for flow constants ($\kappa = 0.4$, $C_\mu = 0.09$), and a fixed reference height of $Z_{ref} = 10$ m. All simulations use a constant surface roughness height z_0 of 0.01 m, corresponding to grass with few trees [36]. Future work will explore developing more complex estimates of surface roughness by inferring from geolocated aerial imagery.

For each terrain, we solved with reference speeds U_{ref} from 1 m/s to 15 m/s in 1 m/s increments. Although we simulated 15 wind speeds for each terrain, not all simulations converged and we discarded solutions that did not reach our specified solution tolerance. On average, each solution took approximately 4 hours to converge on a single processor. Solutions were resampled from the irregular CFD mesh onto a regular 64^3 grid after convergence. In total, we generated 3318 converged CFD solutions from 370 terrain patches.

[1]geodata4edu.ch, Geodata © swisstopo

IV. WIND PREDICTION

Given the training data provided by CFD simulation, we now develop a model that can approximate these in a computationally efficient manner. We chose to use a CNN as a function approximator, since CNNs are highly expressive while allowing for efficient inference even on mobile platforms.

For our model, the input consists of four volumetric channels, each with a spatial resolution of 64^3. The input channels are the inflow conditions ($U_{x,in}$, $U_{y,in}$, $U_{z,in}$) and a terrain model T represented as a binary occupancy grid. The inflow conditions are only specified at the input face, but to maintain consistent volumes we replicate the same wind across the entire volumetric input domain before feeding it to the network. The network outputs three channels of dimension 64^3 representing the predicted velocity ($U_{x,out}$, $U_{y,out}$, $U_{z,out}$).

To increase the size of the training dataset, we augmented the data by rotating each CFD solution around the vertical axis in the cardinal directions, and flipping the lateral dimensions. This resulted in an eight-fold increase in the number of training samples to 26,544. We partitioned the resulting dataset into three parts: 19,728 samples from 290 terrains for training, 5,120 samples (64 terrains) for validation, and 1,704 samples (16 terrains) for testing. The data was split across terrains, meaning no terrain that was seen during training is in the validation or test set.

A. Network architecture

We use an encoder-decoder CNN based on U-net [37]. Our final network architecture is illustrated in Fig. 2, and contains 5.14×10^6 trainable parameters. We store all floating points numbers in single precision (32-bit), and a single forward-pass inference requires 1.07×10^{10} floating point operations. The encoder consists of a sequence of 3D convolutional and max-pooling layers, followed by a fully connected layer. The decoder applies upsampling and 3D convolutional layers to generate the three output channels at the original resolution. We found that upsampling with nearest-neighbor interpolation combined with stride-1 convolution resulted in smoother outputs with lower error than standard transpose convolution layers, which tended to introduce artifacts in the output channels [38]. The network utilizes skip connections to preserve high-resolution feature information from the encoder to aid in the decoding to higher resolution. On a single CPU core one forward-pass inference requires 2.5 GB RAM and is finished on average in 1.6 s. We experimented with training the network both with L^1 and mean squared error (MSE) loss functions, and found that extrema values were better predicted when training with MSE loss than with

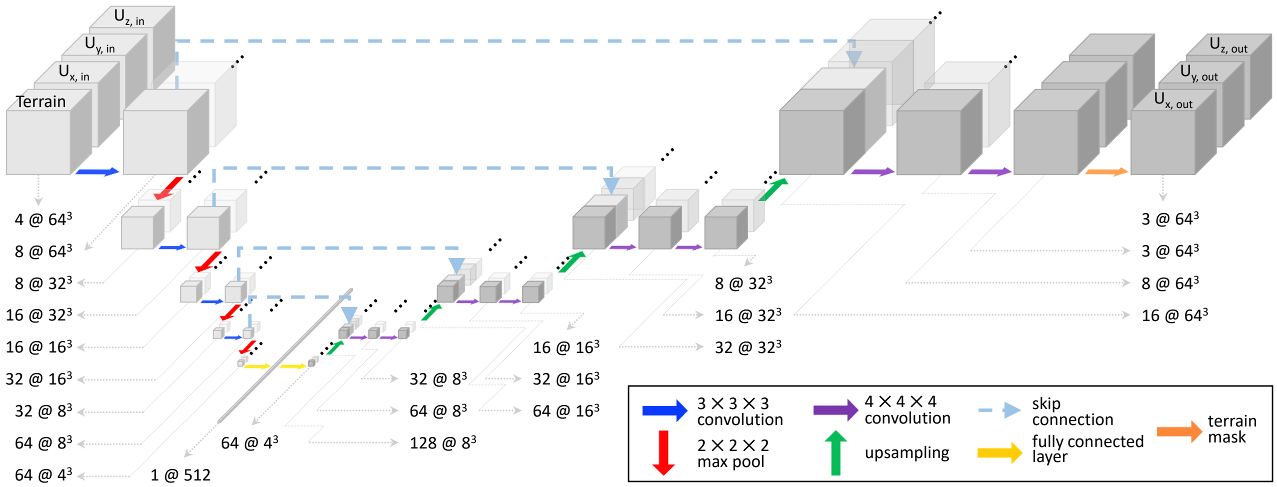

Fig. 2. CNN architecture. We use a 3D encoder-decoder with skip-connections.

L^1 loss. Such extrema often occur close to ridges or steep slopes and can be critical for safe navigation.

Removing the skip connections or the fully connected layers resulted in increases in MSE loss on the test set of 37% and 25% respectively. We also explored alternative minimum code sizes in the fully connected layer, and found that 512 resulted in the best performance compared to 256 (+4.6% MSE test loss), 1024 (+2.4%), 2048 (+10.0%) or 4096 (+12.6%). Using trilinear interpolation instead of nearest neighbor reduced visual artifacts in the output, but increased error slightly (+6.9%).

V. EXPERIMENTS

We now evaluate the proposed approach experimentally. We start by validating the CFD pipeline that was used to generate our training data. We then evaluate the accuracy of the network predictions. Finally, we demonstrate the value of the predicted wind fields for planning safe UAV flight paths.

A. *Verifying the CFD solution*

To verify that our CFD pipeline generates realistic flow estimates, we compared predictions from our CFD pipeline against a microscale flow model verification benchmark. The verification tests described in [17], [18] provide benchmarks of prediction quality from a range of microscale prediction models, including multiple CFD solver schemes, against in-situ measurements.

To validate our CFD setup, we used our CFD pipeline to generate a flow solution on the Bolund hill case and then evaluated the result against the published data. We attempted to keep the solver setup as similar as possible to that used to generate the training dataset, however, we had to make minor adjustments to meet the specifications of the benchmark: we changed the domain shape to be 700×500 m, rather than the 1.2km^2 patches used in our pipeline, and we applied the specified surface roughness on the hill and surrounding areas.

We ran one inflow case (case 2 out of 4 in [18]) with incoming wind at bearing $270°$, as this is the same setup as our pipeline (single incoming wind face). Using the wind speed-up error defined in Eq. 15 of [17], our method showed a mean absolute error of 10.3%. The mean error across all models (and all cases) in [17] was 15.8% and the mean for all RANS 2-equation models was 13.6%. The best performing model (a RANS model) had a mean absolute speed-up error of 10.2%. Thus, our approach is competitive with current methods, validating our CFD setup.

B. *Wind flow prediction*

We compare the accuracy of the network predictions against three baselines. The first is the trivial approach of predicting zero wind everywhere. The second baseline assumes known inflow conditions and replicates them across the entire domain. This is essentially the same as the input to our wind-predicting CNN. The final baseline simulates using an accurate meteorological-scale wind model. In Switzerland, the high-resolution NWP model has a lateral resolution of 1.2 km [19]. We simulate having perfect (no-noise) observations of each vertical edge of a cube enclosing a terrain patch (with 1.2 km side length) by collecting the ground truth wind values from the CFD solution, and then using a trilinear interpolation to estimate the wind inside the region.

Table I shows prediction error metrics for each method, averaged across the whole domain (excluding points inside terrain) and across all samples in the test set. We report MSE and mean absolute error for the U_x and U_z components of the flow, as well as the complete flow vector $\mathbf{U}$. The U_x component represents lateral flow accuracy since the error in the U_y component is almost identical to U_x due to the terrain rotation used in data augmentation.

The quantitative evaluation shows that the network is able to predict the wind with an average absolute error of less than 0.5 m/s for the full wind vector. Both the interpolation and the inflow replication baselines also perform surprisingly well. However, their absolute errors exceed the error of the network prediction by a factor of 1.7 and 2.2, respectively. The relatively good performance of these simple baselines can be attributed to the fact that the flow far away from the

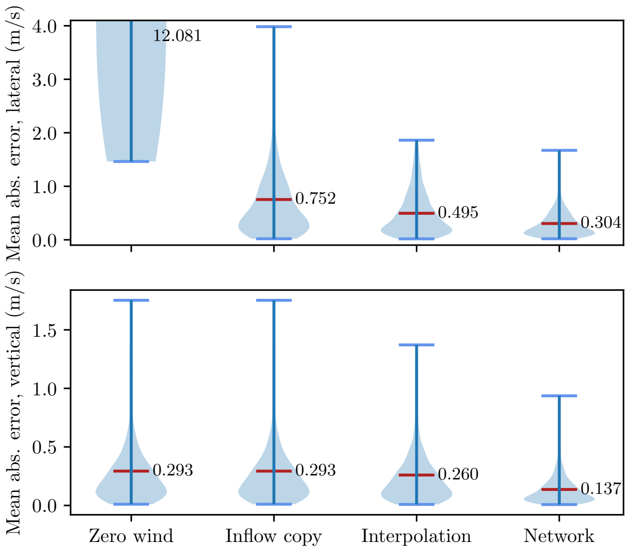

Fig. 3. Distribution of mean absolute prediction error for target altitudes (between 30 and 500m above terrain). We show error for the lateral flow (top) and vertical flow (bottom). Labeled red bars indicate mean values.

terrain is generally smooth and easy to predict, with low amounts of vertical or cross-wind flow.

For our application, the UAV would generally maintain an altitude of at least 30 m above the terrain, thus the prediction accuracy of regions very close to the terrain is not as critical. To highlight the predictive performance of the models in the flight region, we analyze the estimation error for altitudes between 30 m and 500 m above the terrain. Figure 3 illustrates the distributions of mean absolute error for lateral and vertical flow in this target altitude range. The network performs well on lateral predictions, with a mean error of 0.3 m/s, outperforming the closest baseline by a factor of 1.6. The advantage of learning is highlighted even more significantly in the vertical errors, where interpolation does not produce significantly better results than either zero wind or copying the inflow (which both estimate no vertical flow). Here the network outperforms the best baseline by a factor of 1.9. As shown by the distribution plots, the improvement is not only in the average error, but also in the maximal error, which can be particularly important for safe path planning.

To understand where the network has the poorest prediction performance, we visualize the magnitude of the estimation error as a volumetric plot for a terrain sample from the test set in Figure 4. This example illustrates one of the poorer performing predictions, and shows that the

TABLE I
WIND PREDICTION ERROR

Method	MSE loss $(m/s)^2$			Mean abs. error (m/s)		
	U_x	U_z	$\mathbf{U}$	U_x	U_z	$\mathbf{U}$
Zero wind	2.392	0.070	1.617	6.008	0.312	11.872
Inflow copy	0.115	0.070	0.100	0.490	0.312	0.962
Interpolation	0.059	0.040	0.061	0.372	0.281	0.743
Network (ours)	0.012	0.017	0.014	0.238	0.152	0.436

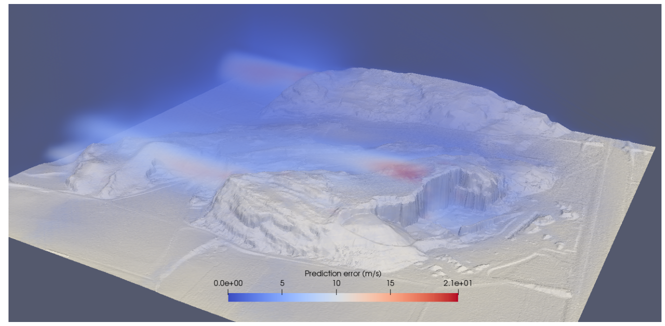

Fig. 4. Prediction error between CFD solution and network prediction for a member of the test set. Note that flow enters from the right. Areas behind complex geometry have the highest magnitude errors.

network prediction is worst in the regions behind prominent terrain features. These areas also contain the most complex flow, where the flow detaches and can form recirculation zones. These regions are also difficult to properly resolve for CFD solvers, as they usually contain a high amount of turbulent flow. We think that increasing the number and variety of training terrain samples may help with improving generalization across terrain variations. In future work we will be exploring how to improve the prediction performance in these regions and developing uncertainty measures, potentially based on learning from the turbulence estimate outputs of the CFD solver.

Figure 5 compares the qualitative performance of different methods, visualized as two-dimensional slices of the predicted three-dimensional volumes. While lateral flow is qualitatively similar for the network prediction and the interpolated result, the vertical flow is predicted much better by the network, even in the challenging case shown at the bottom.

C. Results: UAV path planning

We demonstrate that wind estimates from our network allow a basic planner to more reliably generate feasible plans. We use RRT* [39], a probabilistically complete and asymptotically optimal sampling-based planner. We selected RRT* because it provides anytime behaviour (monotonically improving path cost with execution time) after finding an initial solution. The planner searches for a valid path between specified start and goal positions that minimizes total cost along the path and avoids collision with the terrain.

We define the cost function as the air-relative distance based on the wind estimate which is also a proxy for travel time with a constant airspeed:

$$C = \int \frac{1}{\|\vec{a} + \vec{w}\|_2} ds, \tag{2}$$

where $\vec{a}$ and $\vec{w}$ are the airspeed and wind vectors respectively.

In our model, the aircraft airspeed is set at 15 m/s, and we impose vertical rate limits of 3.8 m/s. We provide the planner with the wind estimates from each prediction method, the terrain model, and start and goal locations. The planner then returns the lowest-cost path found within a fixed time budget (20 seconds). We evaluate the true cost of each path using the

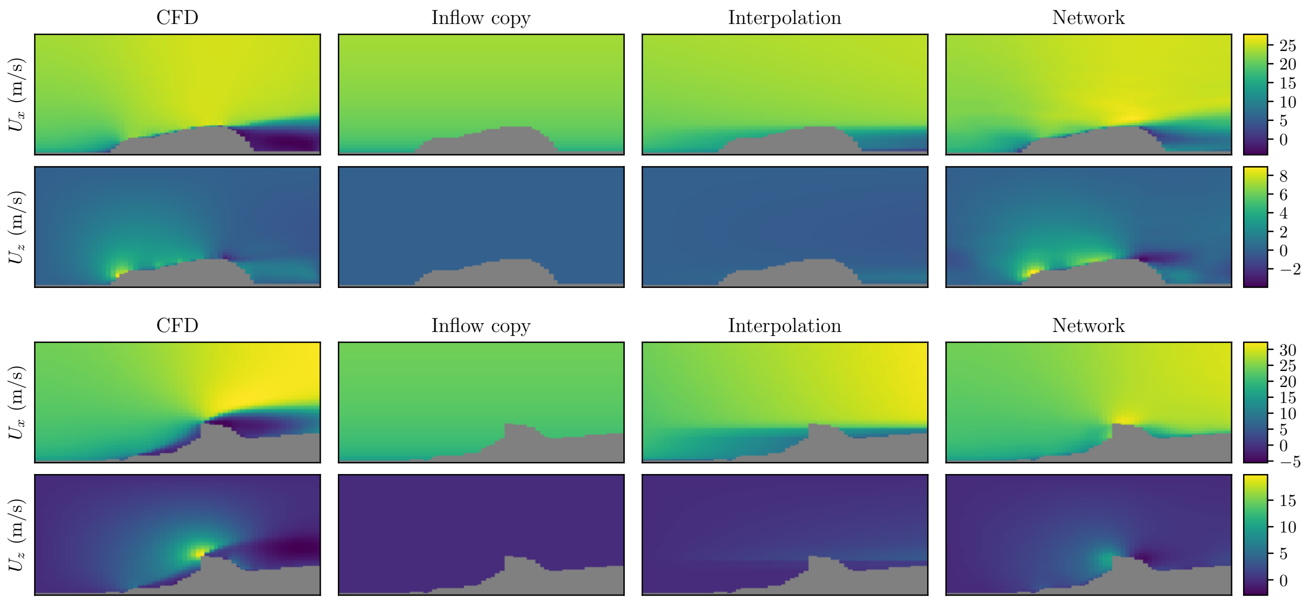

Fig. 5. Qualitative results of wind prediction. All predictions are three-dimensional, but we show two-dimensional vertical $(x-z)$ slices for visualization purposes. Flow enters from the left. We show predictions of the network, two baselines, and the CFD ground truth for both lateral and vertical flow. **Top:** A typical case. Both the lateral and the vertical flow are predicted well by the network. **Bottom:** A challenging case. Geographic features of this type were not represented well in the training set, and the region behind the sheer bluff is not as well modelled.

TABLE II
WIND-AWARE PATH PLANNING PERFORMANCE

Case	Prediction method	Valid	Planned cost [s]	True cost [s]
1	Network	8/10	329.6	386.0
	Interpolation	0/10	-	-
	Inflow	0/10	-	-
	Zero Wind	0/10	-	-
2	Network	5/10	299.2	478.6
	Interpolation	1/10	336.5	488.2
	Inflow	1/10	356.8	495.9
	Zero Wind	0/10	-	-
3	Network	0/10	-	-
	Interpolation	0/10	-	-
	Inflow	0/10	-	-
	Zero Wind	0/10	-	-
4	Network	10/10	38.2	38.5
	Interpolation	10/10	39.3	38.2
	Inflow	10/10	39.2	38.0
	Zero Wind	10/10	68.3	37.7
5	Network	10/10	213.9	213.4
	Interpolation	10/10	196.2	247.0
	Inflow	10/10	197.5	234.3
	Zero Wind	10/10	64.2	242.2

same planner with the wind field generated by the CFD. We do not quantify path risk (the likelihood of collision with the terrain), but rather classify a path as invalid if any segment cannot be completed, either because the headwind is higher than the airspeed, or because the vertical speed is higher than the vertical rate limit.

We evaluate the performance for five different cases with varying terrains and wind speeds. For each of these cases a feasible path exists. Table II summarizes the results over ten runs of each scenario and setting. Planned cost is the cost estimated by the planner using the predicted wind, averaged over feasible plans. True cost is the average cost of the feasible planned paths evaluated with the true (CFD) wind field. Over two out of five cases, using the network prediction for planning significantly increased the chance to find a feasible path compared to the best baseline (0% to 80% and 10% to 50%). However, in some configurations, such as in Case 3, the network prediction is not good enough to result in feasible paths. When finding a feasible path, the network wind prediction typically provided a better estimate for the true cost of the planned path and a lower overall true cost than the baselines.

VI. CONCLUSIONS

We have proposed a deep-learning-based approach for meter-scale wind prediction from a terrain profile. The approach can operate on a single CPU core, and therefore could be deployed on-board a UAV. To showcase the use of the predicted wind fields for UAV flight control, we integrated the wind predictions with a trajectory planner and demonstrated that safety of the planned trajectories can be significantly increased by taking the predicted wind into account. This work could be extended in multiple ways. In reality, wind fields vary over time, and future work will consider temporal variation (transient flow) and local turbulence in the wind field. Online wind measurements that are performed on-board the aircraft could be incorporated in the wind prediction pipeline to further increase its accuracy. Furthermore, a terrain model, incorporating both geometry and terrain roughness, could be extracted online from the visual stream recorded by the UAV. Finally, alternative network structures may improve prediction performance and will be explored in future work.

REFERENCES

[1] T. Schlegel, M. Geissmann, M. Hertach, and D. Kröpfli, "Windatlas Schweiz: Jahresmittel der modellierten windgeschwindigkeit und windrichtung," Federal Department of Environment, Transport, Energy and Communications (UVEK), Tech. Rep. COO.2207.110.2.1073455, 2016.

[2] T. J. Stastny, A. Dash, and R. Siegwart, "Nonlinear MPC for fixed-wing UAV trajectory tracking: Implementation and flight experiments," in *AIAA Guidance, Navigation, and Control Conference*, 2017, p. 1512.

[3] P. Oettershagen, A. Melzer, T. Mantel, K. Rudin, T. Stastny, B. Wawrzacz, T. Hinzmann, S. Leutenegger, K. Alexis, and R. Siegwart, "Design of small hand-launched solar-powered UAVs: From concept study to a multi-day world endurance record flight," *Journal of Field Robotics*, vol. 34, no. 7, pp. 1352–1377, 2017.

[4] I. M. Navon, *Data Assimilation for Numerical Weather Prediction: A Review*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 21–65.

[5] A. Chakrabarty and J. Langelaan, "UAV flight path planning in time varying complex wind-fields," in *2013 American Control Conference*, June 2013, pp. 2568–2574.

[6] L. Wirth, P. Oettershagen, J. Ambühl, and R. Siegwart, "Meteorological path planning using dynamic programming for a solar-powered UAV," in *2015 IEEE Aerospace Conference*, March 2015, pp. 1–11.

[7] J. J. Chung, N. R. Lawrance, and S. Sukkarieh, "Learning to soar: Resource-constrained exploration in reinforcement learning," *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 158–172, 2015.

[8] Y. R. Fei, C. Batty, E. Grinspun, and C. Zheng, "A multi-scale model for simulating liquid-fabric interactions," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, p. 51, 2018.

[9] Y. Xie, E. Franz, M. Chu, and N. Thuerey, "tempoGAN: A temporally coherent, volumetric GAN for super-resolution fluid flow," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, p. 95, 2018.

[10] M. Müller, D. Charypar, and M. Gross, "Particle-based fluid simulation for interactive applications," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '03. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2003, pp. 154–159. [Online]. Available: http://dl.acm.org/citation.cfm?id=846276.846298

[11] L. Ladický, S. Jeong, B. Solenthaler, M. Pollefeys, and M. Gross, "Data-driven fluid simulations using regression forests," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 199:1–199:9, Oct. 2015. [Online]. Available: http://doi.acm.org/10.1145/2816795.2818129

[12] B. Kim, V. C. Azevedo, N. Thuerey, T. Kim, M. Gross, and B. Solenthaler, "Deep fluids: A generative network for parameterized fluid simulations," *arXiv preprint arXiv:1806.02071*, 2018.

[13] S. Wiewel, M. Becher, and N. Thuerey, "Latent-space physics: Towards learning the temporal evolution of fluid flow," *arXiv preprint arXiv:1802.10123*, 2018.

[14] N. Umetani and B. Bickel, "Learning three-dimensional flow for interactive aerodynamic design," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 89:1–89:10, Jul. 2018. [Online]. Available: http://doi.acm.org/10.1145/3197517.3201325

[15] P. Baqué, E. Remelli, F. Fleuret, and P. Fua, "Geodesic convolutional shape optimization," *arXiv preprint arXiv:1802.04016*, 2018.

[16] X. Guo, W. Li, and F. Iorio, "Convolutional neural networks for steady flow approximation," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 481–490. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939738

[17] A. Bechmann, N. N. Sørensen, J. Berg, J. Mann, and P.-E. Réthoré, "The Bolund experiment, part II: blind comparison of microscale flow models," *Boundary-Layer Meteorology*, vol. 141, no. 2, p. 245, 2011.

[18] J. Berg, J. Mann, A. Bechmann, M. Courtney, and H. E. Jørgensen, "The Bolund experiment, part I: flow over a steep, three-dimensional hill," *Boundary-layer meteorology*, vol. 141, no. 2, p. 219, 2011.

[19] M. Baldauf, A. Seifert, J. Förstner, D. Majewski, M. Raschendorfer, and T. Reinhardt, "Operational convective-scale numerical weather prediction with the COSMO model: Description and sensitivities," *Monthly Weather Review*, vol. 139, no. 12, pp. 3887–3905, 2011. [Online]. Available: https://doi.org/10.1175/MWR-D-10-05013.1

[20] T. Uchida and Y. Ohya, "Micro-siting technique for wind turbine generators by using large-eddy simulation," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 96, no. 10, pp. 2121 – 2138, 2008, 4th International Symposium on Computational Wind Engineering (CWE2006). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167610508000664

[21] J. Palma, F. Castro, L. Ribeiro, A. Rodrigues, and A. Pinto, "Linear and nonlinear models in wind resource assessment and wind turbine micro-siting in complex terrain," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 96, no. 12, pp. 2308 – 2326, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167610508001037

[22] T. Whalen, E. Simiu, G. Harris, J. Lin, and D. Surry, "The use of aerodynamic databases for the effective estimation of wind effects in main wind-force resisting systems:: application to low buildings," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 77-78, pp. 685 – 693, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167610598001834

[23] Y. Tominaga, A. Mochida, R. Yoshie, H. Kataoka, T. Nozu, M. Yoshikawa, and T. Shirasawa, "AIJ guidelines for practical applications of CFD to pedestrian wind environment around buildings," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 96, no. 10, pp. 1749 – 1761, 2008, 4th International Symposium on Computational Wind Engineering (CWE2006). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167610508000445

[24] J. S. Elston, J. Roadman, M. Stachura, B. Argrow, A. Houston, and E. Frew, "The tempest unmanned aircraft system for in situ observations of tornadic supercells: design and vortex2 flight results," *Journal of Field Robotics*, vol. 28, no. 4, pp. 461–483, 2011.

[25] J. Elston and E. W. Frew, "Unmanned aircraft guidance for penetration of pretornadic storms," *Journal of guidance, control, and dynamics*, vol. 33, no. 1, pp. 99–107, 2010.

[26] E. W. Frew, J. Elston, B. Argrow, A. Houston, and E. Rasmussen, "Sampling severe local storms and related phenomena: Using unmanned aircraft systems," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 85–95, 2012.

[27] J. Roadman, J. Elston, B. Argrow, and E. Frew, "Mission performance of the tempest unmanned aircraft system in supercell storms," *Journal of Aircraft*, vol. 49, no. 6, pp. 1821–1830, 2012.

[28] A. Renzaglia, C. Reymann, and S. Lacroix, "Monitoring the evolution of clouds with UAVs," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 278–283.

[29] C. Reymann, A. Renzaglia, F. Lamraoui, M. Bronz, and S. Lacroix, "Adaptive sampling of cumulus clouds with UAVs," *Autonomous robots*, vol. 42, no. 2, pp. 491–512, 2018.

[30] J. W. Langelaan, N. Alley, and J. Neidhoefer, "Wind field estimation for small unmanned aerial vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 1016–1030, 2011.

[31] N. R. J. Lawrance and S. Sukkarieh, "Path planning for autonomous soaring flight in dynamic wind fields," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 2499–2505.

[32] I. Guilliard, R. J. Rogahn, J. Piavis, and A. Kolobov, "Autonomous thermalling as a partially observable markov decision process," in *Robotics: Science and Systems*, 2018.

[33] J. W. Langelaan, J. Spletzer, C. Montella, and J. Grenestedt, "Wind field estimation for autonomous dynamic soaring," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 16–22.

[34] B. E. Launder and B. Sharma, "Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc," *Letters in heat and mass transfer*, vol. 1, no. 2, pp. 131–137, 1974.

[35] H. Jasak, A. Jemcov, v. Tuković *et al.*, "OpenFOAM: A C++ library for complex physics simulations," in *International workshop on coupled methods in numerical dynamics*. IUC Dubrovnik, Croatia, 2007, pp. 1–20.

[36] F. V. Hansen, "Surface roughness lengths," Army Research Lab White Sands Missile Range NM, Tech. Rep. ARL-TR-61, 1993.

[37] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[38] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016. [Online]. Available: http://distill.pub/2016/deconv-checkerboard

[39] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.