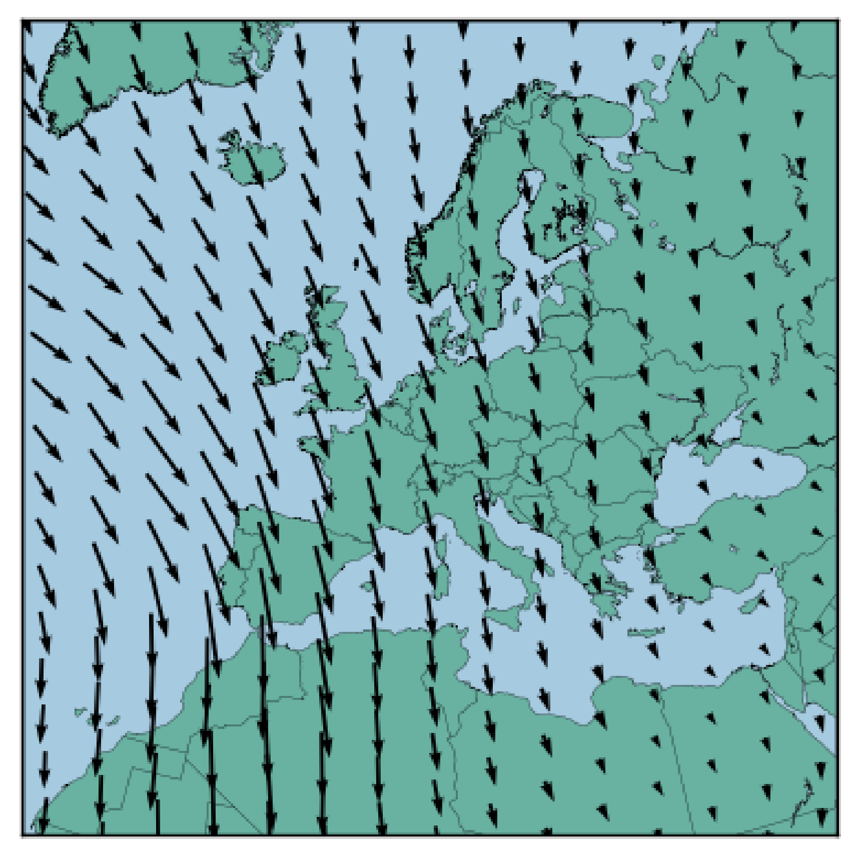
Wind Field Nowcasting and Forecasting

Applying Diffusion Probabilistic Models with Aircraft-Derived Data

Matthijs Slobbe



TU Delft

Wind Field Nowcasting and Forecasting

Applying Diffusion Probabilistic Models with Aircraft-Derived Data

Thesis report

by

Matthijs Slobbe

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on July 1st, 2025, at 10:00

Thesis committee:	
Chair:	Prof.dr.ir Jacco Hoekstra
Supervisors:	Dr. Junzi Sun
	Dhr. Evert Westerveld
External examiner:	Dr. Alessandro Bombelli
Place:	Faculty of Aerospace Engineering, Delft
Project Duration:	September, 2024 - July, 2025
Student number:	5051096

Faculty of Aerospace Engineering · Delft University of Technology

TU Delft Delft University of Technology

Preface

Before the reader lies my thesis, written to obtain a Master of Science in Aerospace Engineering at Delft University of Technology. This project was done in collaboration with Air Traffic Control The Netherlands (LVNL) as a part of the Centre of Excellence, funded by the Knowledge and Development Centre Mainport Schiphol. Not only that, this document is also a testament to my journey as a student over the last 6 years. It doesn't cease to amaze me how much one can develop, personally and professionally, in such a short period.

Thank you, Junzi, for giving me the opportunity to take on this project and the freedom to explore it in a way that allowed me to go where my curiosity led me. Your supervision has been invaluable, and I thank you for always being available for me at a moment's notice whenever I needed it. Your absolute faith in my abilities has given me the courage to keep going during the hard times.

Evert, thank you for going out of your way to find a supervisor for my unique project and deciding to supervise me yourself when all else failed. Your knowledge of aviation and air traffic management, together with your anecdotes and jokes, has inspired me to love this area of study so much more than I imagined I could. Thanks to you, I have never walked away from one of our conversations not feeling energised, and I am grateful for it.

There are a few others whom I'd like to mention: Jan, Christiaan, Jens, and Ferdinand. Your ability to spread your enthusiasm for air traffic management has made my time at LVNL much more than it otherwise would have been. I look forward to our future encounters.

And to my fellow students at iLabs, I have thoroughly enjoyed our coffee machine chats and aircraft spotting out the window. Thank you for supporting me when I paced back and forth in front of my computer screen or threw pens across the room during the struggles of getting the models to work in the last 10 months. I cannot imagine having had to sit alone in a room during my project.

Finally, I would like to thank the three most important people in life, my parents, Annemiek and Aart, and my brother, Quinten. You are always there for me when I need it most, even now after I've moved to the other side of the world to follow my dreams. You have no idea how much you are still with me, albeit more spiritually than physically.

Contents

Part I

Scientific Article

Wind Field Nowcasting and Forecasting using Denoising Diffusion Probabilistic Models with Aircraft-Derived Data

Matthijs Slobbe
Under the supervision of Junzi Sun (TU Delft), Evert Westerveld (LVNL), and Jacco Hoekstra (TU Delft)

Section Control & Simulation, Department Control & Operations, Faculty of Aerospace Engineering
Delft University of Technology, Delft, The Netherlands

***Abstract*—As with many aspects of modern life, wind nowcasting and forecasting are integral parts of aviation and Air Traffic Management (ATM). This study investigates the use of a Denoising Diffusion Probabilistic Model (DDPM) for nowcasting (inpainting) and forecasting (image-to-video) of wind fields using aircraft-derived meteorological data. The DDPM, implemented with a U-Net backbone, demonstrated strong performance in nowcasting tasks, outperforming previous models such as the Lagrangian transportation-based Meteo-Particle (MP) model and Physically Inspired Neural Network (PINN) approach with a 29% improvement in magnitude error and a 62% reduction in directional error. The nowcasting model achieved a magnitude error of 2.03 m/s and a directional error of 4.2°, based on 190 test samples from late 2024. A key contribution lies in the DDPMs ability to produce more consistent and lower-variance predictions than prior methods. The RMSE improved on the PINN results by 29%, to 3.99 m/s. Despite these successes, forecasting proved significantly more challenging, with no meaningful results achieved. The study used ECMWF CERRA reanalysis data for training and evaluated model performance with simulated aircraft tracks on known wind fields and with real aircraft-derived data from the The Royal Netherlands Meteorological Institute (KNMI)'s EMADDC dataset split into model input and validation subsets. High computational demands restricted testing capabilities, and uncertainty quantification and severe weather conditions remain challenging for the model.**

***Index Terms*—wind nowcasting, diffusion model, meteorology, wind forecasting**

I. Introduction

Accurate weather forecasting is vital to aviation flight planning and safety. Wind speed and direction play a critical role among the many factors involved. They influence decisions such as selecting optimal flight trajectories and determining active runway usage at airports. Wind data also supports strategic decision-making by Air Traffic Controllers (ATCos), helping to maintain safe airspace by balancing traffic capacity and demand [1]. This becomes especially important during extreme weather conditions, significantly affecting air traffic capacity. As noted by Marinescu et al., wind is a significant source of uncertainty in both current and future Air Traffic Management (ATM) systems [2].

Consequently, there is a high demand for *accurate wind nowcasting and forecasting*, allowing Air Navigation Service Providers (ANSPs) and airlines alike to *safely and efficiently* conduct day-to-day operations.

Accurate weather forecasting is particularly challenging. Complications arise due to the non-linear dynamic nature of the atmosphere, introducing significant uncertainty. Variations in measurement methods and numerical solvers, each with strengths and weaknesses, further contribute to the complexity [3]. State-of-the-art Numerical Weather Prediction (NWP) models are sensitive to initial conditions, which impact the prediction, especially as the forecast horizon increases. The current approach is probabilistic, where an ensemble of initial conditions slightly perturbed from measurement data is used to quantify the uncertainty in the model [4].

Advancements in data assimilation techniques (nowcasting) such as 4DVAR increase the accuracy of initial conditions, and hence forecasts [5]. Data assimilation is also used to improve representations of previous states of the atmosphere, so-called reanalyses. Large, spatially and temporally dense, reanalysis datasets exist, for example, the European Centre for Medium-Range Weather Forecasts (ECMWF) ERA5 dataset (1950-present) [6]. Despite this, local real-time forecasts remain difficult. This is precisely what airlines and ANSPs need to make well-informed decisions[1].

Local real-time, accurate forecasts require local real-time, accurate data. To this end, meteorological conditions derived from aircraft surveillance data can be used to initialize assimilation procedures, which can then be used for forecasting. Sun et al. have identified two main approaches: ground-based trajectory observations and interrogated aircraft data [7]. Both have been successfully implemented for wind field assimilation procedures [8, 9]. These nowcasting procedures are analogous to image inpainting, a typical application of Deep Learning (DL).

DL has also been applied to weather forecasting. Despite success in predicting global weather patterns, extreme weather conditions remain problematic [10, 11]. Unlike NWP, DL models lack uncertainty handling [11]. Although there is great potential for combined DL and NWP approaches.

Using everyday images as an initial condition to create semantically accurate videos has been accomplished. Diffusion class DL models obtain the best results [12]. This technique is hypothesized to accurately display the evolution of a wind field in time (i.e., forecast), just as everyday images can be extrapolated into the future in the form of a video. In this case, each successive frame is a future state of the wind field.

[1] This follows from conversations with Air Traffic Control the Netherlands (LVNL).

This study seizes the opportunity to utilize aircraft surveillance data as a partially available wind field, in combination with a state-of-the-art DL architecture, a Denoising Diffusion Probabilistic Model (DDPM). Posed as an inpainting problem for nowcasting (assimilation), followed by an image-to-video forecast, the objective is to improve real-time wind field estimation. The structure of this work is as follows. First, Section II presents related work and necessary background information. In Section III, the data sources for training and inference will be described. Next, Section IV explains the principles of the model architecture. Section V describes the error metrics and the training and validation procedures for nowcasting and forecasting. Section VI presents relevant results, Section VII discusses these results, and conclusions are made in Section VIII.

II. Related Work

The current methods and the new model must be well understood in order to apply the new model properly. They are presented first. Notable open-source methods currently used for weather field reconstruction with aircraft surveillance data include the Meteo-Particle (MP) (Subsection II-A) and with a Physically Inspired Neural Network (PINN) (Subsection II-B). Furthermore, the state-of-the-art in generative DL methods and their current applications will be discussed (Subsection II-C).

A. Meteo-Particle model

Sun et al.'s Meteo-Particle (MP) is a weather field reconstruction model, based on Lagrangian transportation, using aircraft surveillance data [7]. Particles carry information on the state of wind and temperature through space. They are generated at locations where measurements have been made and propagate through space with a Gaussian random walk model, biased by lateral wind, while decaying over time. One can estimate information at any location within the spatial bounds by combining the weighted states of nearby particles.

The model assumes the true states of wind and temperature are geographically and temporally stable at a scale of tens of kilometers and minutes, respectively. Furthermore, the burst error rate of single-aircraft observations is assumed to be low. This is hard to avoid, although the effects are reduced with the addition of probabilistic measurement rejection.

Various factors influence the confidence level of an estimation. These include the number of particles in the vicinity of the target location, the mean distances between these particles and the target location, the homogeneity of states carried by the particles, and lastly, the strength of the particles as determined by an aging function. The model output could be combined with Global Forecast System (GFS) estimates in low-confidence areas. Zhu et al. improved precision and accuracy by introducing a mixed evaluation index and optimizing constant parameters and control factors [13].

The MP model can be used for short-term forecasting. A prior is created with a short history of estimated states at the desired location. This is a statistical model as a function of time. A Gaussian process regressor then uses this as input. Using various kernels (combined in summation or multiplication) that represent the properties of the system, a covariance function can be constructed to make a forecast. Sun et al. use a constant kernel, a squared exponential kernel, and a white noise kernel to describe the system.

B. Physically Inspired Neural Network

Malfliet approached the same problem with a Physically Inspired Neural Network (PINN), used for inpainting [14]. The U-net architecture (encoder/dense-blocks/decoder blocks with skip connections [15]) was used. This research investigated the added value of incorporating physical processes into the cost function. Besides the observational loss (difference between predicted and truth fields), continuity and vorticity were considered.

This research's scope excluded specific weather phenomena such as convection, jet streams, and thunderstorms. It assumed 2D flow on isobarics, and similarly to the MP model, it assumed stable flow geographically and temporally on the scale of tens of kilometers and minutes, respectively.

Atmospheric flow occurs at multiple scales, which makes the U-net architecture suitable because of its performance in identifying features at different scales [15]. Notably, given the problem being treated as an inpainting task, partial convolutions are used to cope with the unknown values.

Similarly to the MP model, GFS data was included in the background in addition to the aircraft measurements as input, to assist the model. This was especially useful to guide the model during low flight activity (i.e., nighttime) and in areas with few measurements.

ECMWF ERA5 reanalysis data [6] and GFS [16] data were used to train the model. These datasets contain U-component (West-East), V-component (South-North), and geopotential height information of distinct pressure levels [14]. Vorticity and continuity losses were included in the loss function to introduce awareness of physical processes, in addition to the standard observational loss.

Malfliet's model was validated using a portion of the simulated input as a validation sample and real flight data. In the former, complete ERA5 reanalysis wind fields were masked to simulate aircraft paths, with the unmasked regions used to validate the output. In the latter, a set of aircraft-derived data was split, with some flight tracks being used as input to the model and others to validate the output of the model.

C. State-of-the-art Deep Learning for Content Generation

Generative Artificial Intelligence (AI) is currently one of the most popular research topics within computer vision. Its impact is widespread, from computer graphics to art, design, and medical imaging [12]. In the realm of image and video generation, diffusion models have prevailed to yield the best results, surpassing other methods such as General Adversarial Networks (GANS) [17, 18, 19] and auto-regressive Transformers [20, 21, 22]. While commonly used to generate content based on text input (text-to-image), these methods can also be used for image inpainting. The application of this paper is inpainting; however, it is now done with wind fields instead of everyday images. The strength of diffusion models lies in their ability to generate semantically correct and diverse samples.

Diffusion models are a class of latent variable probabilistic generative models. Inspired by nonequilibrium thermodynamics, the algorithm involves incrementally transforming a complex distribution (such as an image) into unstructured noise, and learning to reverse this diffusion process [23]. The former is referred to as the *forward process* or *diffusion process* and the latter as the *reverse process*. Additionally, there is a sampling procedure to generate new data.

There are three primary methods, namely, Score-based Generative Models (SGMs), Stochastic Differential Equationss (SDEs), and Denoising Diffusion Probabilistic Models (DDPMs). The forward process is shared between them and is defined by a time-indexed (diffusion time) Markov chain that incrementally adds Gaussian noise to the data following a variance schedule [23].

SGM use a neural network trained with score matching [24] to learn the score [25] of the logarithmic data density. The logarithmic data density is defined as the gradient of the log-density function at the input data point [26]. This is often visualized as a vector field directing where the log data density grows most [26]. With score matching, a model can be trained to estimate the score of a probability density without the probability density itself. This is crucial because the probability density function is intractable in complex real-world situations. Samples are produced using annealed Langevin dynamics. This allows the production of samples from a probability density using the score function. To summarize this method, data is perturbed with varying noise, and a neural network is trained to approximate the score at each noise level. Samples are then generated by chaining the learned score functions together, simulating the true reverse process [12].

The Score SDE method generalizes the concept of perturbing data at multiple noise levels to an infinite number of noise scales [12]. The diffusion process is modeled as a solution to an SDE. Samples are then generated through a reverse-time SDE [27].

Lastly, DDPM, the chosen method, involves two Markov chains for the forward and reverse process. The reverse process learns transition kernels to reverse the forward process with a neural network [12]. The two most common network architectures are the U-Net [15] and Transformer [28]. Initially proposed for image segmentation, the U-Net has a U-shaped encoder-decoder architecture. The downsampling layers of the encoder combined with skip connections to the upsampling layers of the decoder allow the model to learn data features at various levels. The Transformer also has an encoder-decoder structure and includes self-attention functions. These measure significance between all inputs regardless of spatial location [28]. A U-Net structure is often added to Transformers to get the benefit of both [29]. This is beyond the scope of this paper, however. Learning is done by optimizing for the variational lower bound. A key property of both network architectures is that the data dimensionality is conserved between input and output. This is necessary to preserve the dimensions of the latents during the reverse process. Sampling is done by starting with a random vector from the prior distribution and sampling ancestrally through the learned transition kernels of the reverse Markov chain, to produce an output.

III. Data Sources

This chapter explains the data sources used in this work. First, aircraft-derived meteorological data are discussed (Subsection III-A), followed by a description of the model training data (Subsection III-B). These are crucial as they determine the model's restrictions and reveal opportunities.

A. Aircraft-Derived Meteorological Data

The source of the real-time data used to initialize the nowcasting of wind fields is aircraft-derived. Automatic Dependent Surveillance-Broadcast (ADS-B) combined with Mode-S Comm-B responses from aircraft interrogated by Secondary Surveillance Radar (SSR) will be used to derive the meteorological data.

The ADS-B technology requires no interrogation. The information the messages contain, also in Mode-S format, is automatically broadcast twice per second and can be received by anyone with a (low-cost off-the-shelf) receiver [30]. Flight states such as location, altitude, and ground speed are included. Airspeed is not transmitted, so SSR interrogation responses are needed. Note that here ground speed means relative speed between the aircraft and the ground, whereas airspeed means the relative speed between the aircraft and the surrounding air.

SSR replies are not designed to be read by entities other than the SSR itself. Unlike ADS-B, the message contains no information on its contents. This is because the interrogating radar knows the target aircraft and the expected information in the message. Using open data like this, in a way it was not initially designed for, poses challenges in identifying the International Civil Aviation Organization (ICAO) address of the broadcasting aircraft [7].

A reliable way to identify the ICAO address of an aircraft is by cross-referencing with ADS-B data, in addition to a reverse parity check. To obtain the information in the transmission, many checks assuming different message types must be made. This is further elaborated upon by Sun et al. in their paper on weather field reconstruction [7]. Measurement, transmission, and truncation errors should not be forgotten. The truncation of values can also lower the numerical resolution of measurements [31].

While not used in this research, the open-source library, pyModeS, can be used to handle the ADS-B and Mode-S Comm-B transmissions [32]. The preprocessing done by this library reduces the uncertainty caused by the potential errors mentioned and incorrect Mode-S Comm-B message type identification.

The ADS-B and Mode-S Comm-B supply the following aircraft states:

- Barometric altitude
- Ground speed (GS)

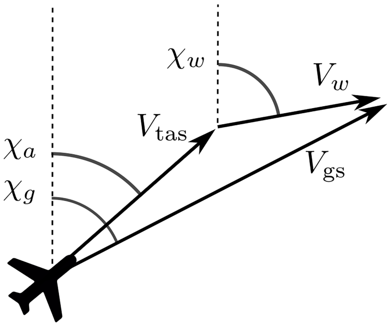


Fig. 1. Inference of wind vector from GS and TAS vectors. χ_a, χ_g, and χ_w refer to the airspeed, ground speed, and wind headings, respectively. Image taken from Sun et al. [7], https://doi.org/10.1371/journal.pone.0205029.g002

- True airspeed (TAS)
- Indicated airspeed (IAS)
- Mach number

The wind vector can be derived with knowledge of the GS and TAS in combination with heading and track angle (Fig. 1) [7]:

$$\overrightarrow{V}_{wind} = \overrightarrow{V}_{GS} - \overrightarrow{V}_{TAS} \tag{1}$$

Since the TAS update frequency is lower than desired, TAS is also calculated with IAS or Mach number when combined with knowledge of temperature. The temperature can be derived from TAS and pressure. Further details regarding inference of meteorological conditions in this way are described by de Leege et al. [33] and Sun et al. [7].

Barometric altitude determines if aircraft are on the same pressure level (isobaric). Calculated with measured pressure and International Standard Atmosphere (ISA) standard sea level pressure [33], it is used to determine the flight level at which an aircraft flies. Importantly, often barometric altitude does not coincide with geometric altitude. This is because the pressure at a geometric altitude can vary depending on where on the globe an aircraft is. Safety is ensured using a common reference standard (ISA) based on pressure. Aircraft flight levels are based on this standard and will be used in this work.

This study will use data from the European Meteorological Aircraft Derived Data Center (EMADDC) [34]. This wind data is derived in the same way as described earlier.

B. Model Training Data

To train the model, windfields from the ECMWFs Copernicus European Regional ReAnalysis (CERRA) [35] dataset have been used. This assimilation system has been optimized for the European area (Fig. 2) and was chosen for its high vertical and horizontal resolution. The horizontal grid spacing is 5.5km, and 106 vertical (model) levels exist between the surface and the 1 hPa pressure level. The temporal density is 3 hours, with analyses at UTC 00, 03, 06, etc. This work concentrates on cruise altitudes; hence, data from model levels 45 to 54 were applicable for training [36]. This corresponds to approximately FL300 to FL400 in steps of around 10 flight levels. The dataset contains information on U and V wind components, temperature, and specific humidity. Given computational constraints, specific humidity was left out of the model to keep the model size as small as possible.

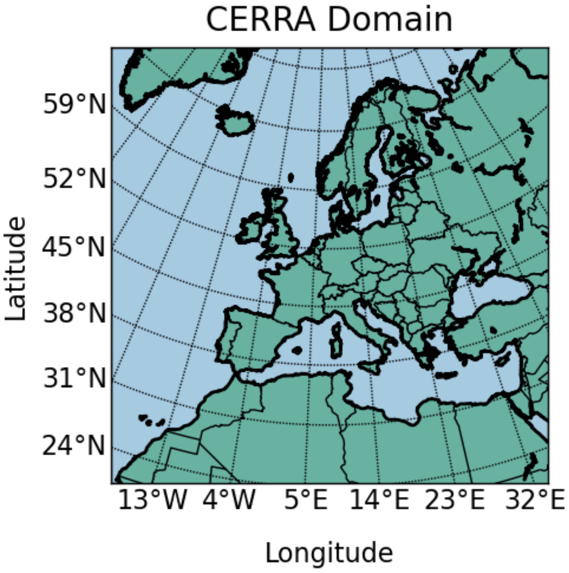


Fig. 2. The CERRA model domain.

IV. Denoising Diffusion Probabilistic Model

The model aims to learn the data distribution of wind fields and use this to generate new, semantically accurate wind fields. The model description is divided into three parts.

The forward process is first discussed (Subsection IV-A). During this process, the training data is perturbed with Gaussian noise until it is pure Gaussian noise. This is a prerequisite for the reverse process, the part of the model trained to approximate the original data distribution.

Next, the reverse process, including the neural network architecture that acts as the backbone, is explained (Subsection IV-B). The network configuration slightly differs depending on whether the goal is nowcasting or forecasting. During this process, the network is trained to undo the perturbations from the forward process.

Lastly, the sample inference procedure is discussed (Subsection IV-C). This includes how the available part of the wind field is used to guide the generation of the complete wind field from the learned data distribution in the nowcasting scenario. Additionally, it includes how forecasting is done, guided by a complete nowcast.

A. Forward Process

The forward process starts with a sample from a real data distribution $\mathbf{x}_0 \sim q(\mathbf{x})$. The process ends with a sample equivalent to isotropic Gaussian noise. While similar to the encoder of a Variational Autoencoder (VAE), the forward process only adds noise for perturbation and does not have any trainable parameters [23]. First, the general noise addition process will be discussed. The rationale behind variance scheduling during this noise addition process follows, with a useful reparameterization discussed last.

1) Sample Noising: The initial sample, $\mathbf{x}_0$, is a complete wind/temperature field. Gaussian noise is added to the sample in T steps, resulting in a sequence of noisy samples $\mathbf{x}_1, \ldots, \mathbf{x}_T$. Eq. (2) describes a single noise addition step, where β_t represents the variance of the noise.

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}\right) \tag{2}$$

The forward process is a Markov process, meaning that the state at each step depends only on the state at the previous step. The state at any step T is thus described by Eq. (3).

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{3}$$

The sample $\mathbf{x}_0$ gradually becomes indishtinguable as t becomes larger. As $T \to \infty$, the sample ($\mathbf{x}_T$) becomes equivalent to an isotropic Gaussian distribution (Fig. 3).

2) Variance Scheduling: The variance schedule, or amount of noise added at each diffusing step, is defined by $\{\beta_t \in (0,1)\}_{t=1}^{T}$. The goal of variance scheduling is threefold. The final sample, $\mathbf{x}_T$, should be approximately Gaussian, too rapid or slow signal deconstruction should be avoided, and stable and efficient learning of the reverse process should be facilitated.

The signal should ideally be destroyed minimally near $t = 0$ and $t = T$, with a linear dropoff in the middle. This balances the requirement to achieve as close to pure noise as possible by the end of the forward process, without destroying the signal too quickly or adding too much noise when nearing $t = T$. This is closely associated with the requirement for stable and efficient learning. Let $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \Pi_{i=0}^{t}\alpha_i$. α_t represents how much of the signal is retained during a step in the chain, and $\bar{\alpha}_t$ represents how much of the signal remains at any diffusion step t. With this in mind, the chosen $\bar{\alpha}_t$ schedule is cosine-based. The cosine schedule used (Eq. 4) was designed by Nichol and Dhariwal [37]. To prevent β_t from being too small near $t = 0$, a small offset, s, is introduced. The value of s has been set to 0.008, as was done by Nichol and Dhariwal. It has a linear drop-off of $\bar{\alpha}$ in the middle of the diffusion process, while being more subtle near the start and end of the diffusion process (Fig. 4). In practice, β_t ($\beta_t = \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}$) is clipped to be no bigger than 0.999 to prevent singularities near $t = T$. This balances the three aforementioned goals.

$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, \quad f(t) = \cos\left(\frac{t/T + s}{1+s} \cdot \frac{\pi}{2}\right)^2 \tag{4}$$

3) Reparamatization: A useful property of the forward process is that the sample $\mathbf{x}_t$ at any arbitrary step t can be obtained in a closed form using the so-called reparameterization trick [38]. This writes a random variable as a deterministic function of a noise variable. Recall $\alpha_t := 1 - \beta_t$, $\bar{\alpha}_t := \Pi_{i=0}^{t}\alpha_i$, and let the noise $\epsilon_{t-1}, \epsilon_{t-2}, \cdots \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

$$\begin{aligned}\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\ &= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\boldsymbol{\epsilon}}_{t-2} \\ &= \ldots \\ &= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}\end{aligned} \tag{5}$$

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right) \tag{6}$$

Two Gaussians are merged in line two of Eq. (5). Note that merging two Gaussians of different variance, such as $\mathcal{N}(\mathbf{0}, \sigma_1^2\mathbf{I})$ and $\mathcal{N}(\mathbf{0}, \sigma_2^2\mathbf{I})$ results in a new distribution described by $\mathcal{N}(\mathbf{0}, (\sigma_1^2 + \sigma_2^2)\mathbf{I})$. In this case the merged standard deviation is $\sqrt{(1-\alpha_t) + \alpha_t(1-\alpha_{t-1})} = \sqrt{1-\alpha_t\alpha_{t-1}}$.

B. Reverse Process

The goal of the reverse process is to reverse the forward process. Firstly, the theory on how this process is learned will be elaborated upon, followed by a description of the architecture of the neural network trained to learn according to the theory.

1) Modeling the posterior distribution: In an ideal world, one would sample from $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, recreating the original sample from a Gaussian noise input. Given that β_t is small enough, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ will also be Gaussian. Unfortunately, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ needs to use the entire dataset and is intractable. Thus, a model, p_θ, of the conditional probabilities must be learned to approximate the ideal reverse process (Eq. (7)). The process starts at $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \boldsymbol{I})$.

$$\begin{aligned}p_\theta(\mathbf{x}_{0:T}) &:= p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \\ p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) &:= \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))\end{aligned} \tag{7}$$

Together, q and p form a VAE. The Variational Lower Bound (VLB) can be maximized as a proxy for learning how to perfectly model the true latent posterior distribution [39]. The definitions below can describe it [37]:

$$L_{\text{vlb}} := L_0 + L_1 + \ldots + L_{T-1} + L_T \tag{8}$$

$$L_0 := -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \tag{9}$$

$$L_{t-1} := D_{\text{KL}}\left(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \,\|\, p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\right) \tag{10}$$

$$L_T := D_{\text{KL}}\left(q(\mathbf{x}_T|\mathbf{x}_0) \,\|\, p(\mathbf{x}_T)\right) \tag{11}$$

L_0 (Eq. (9)) represents the reconstruction likelihood of the decoder from the variational distribution [39]. In other words, this measures whether the chosen parametrization of the latents is suitable. L_T (Eq. (11)), the prior matching term, represents how close the output of the forward process, $p(\mathbf{x}_T)$, is to a standard Gaussian. It is not trainable and is zero under our assumptions [39]. All other terms in Eq. (8) are Kullback–Leibler (KL) divergences between two Gaussian distributions, namely $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is compared directly to forward process posteriors, which are tractable when conditioned on $\mathbf{x}_0$. These can be evaluated in closed form. By reparameterizing and using Bayes' theorem, the posterior $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$

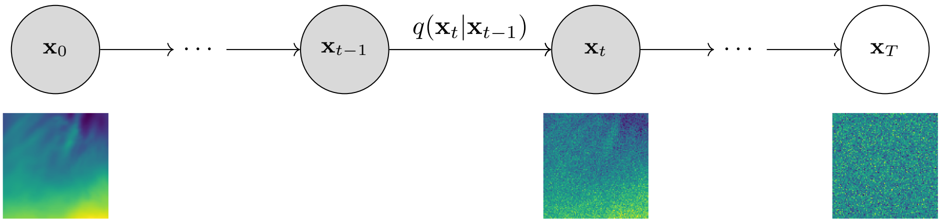


Fig. 3. Depiction of the forward process of a DDPM. $\mathbf{x}_0$ is a sample from a real data distribution, in this case, a wind/temperature field. Gaussian noise is added stepwise. As $T \to \infty$, $\mathbf{x}_T$, the sample becomes equivalent to an isotropic Gaussian distribution. Inspired by an image from Ho et al. [23].

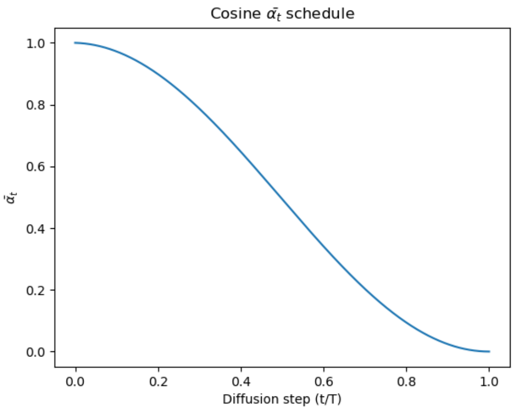


Fig. 4. Schedule of $\bar{\alpha}_t$, representing how much of the signal remains at any diffusion step t, throughout the entire diffusion process. Inspired by Nichol and Dhariwal [37].

can be calculated in terms of $\tilde{\beta}_t$ and $\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0)$. They are defined as follows [23]:

$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{12}$$

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \tag{13}$$

$$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I}) \tag{14}$$

Given this, the objective of Eq. (8) is a sum of terms, L_{t-1}. Reparametrization (Eq. (6)) allows simple sampling from any timestep of the forward process to estimate L_{t-1} using Eq. (7) and Eq. (14). Randomly sampling t and taking the expected value of L_{t-1} given t, $\mathbf{x}_0$ and ϵ, can be done to estimate L_{vlb}.

Various options exist to parameterize $\mu_\theta(\mathbf{x}_t, t)$ in the prior. It could be directly predicted with a neural network. $\mathbf{x}_0$ could also be predicted, using Eq. (13) to obtain $\mu_\theta(\mathbf{x}_t, t)$. Furthermore, ϵ (the noise) could be predicted. Eq. (5) and Eq. (13) can then be used to derive Eq. (15), where ϵ_θ is a function approximator for predicting ϵ from $\mathbf{x}_t$.

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right) \tag{15}$$

Ho et al. found that this approach, to predict ϵ, works best, particularly when combined with a reweighted loss function:

$$L_{\text{simple}} = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2\right] \tag{16}$$

This is the Mean Squared Error (MSE) between the true and predicted noise. The improved results are explained by the connection with generative score matching [40].

Currently, there is no learning signal for $\mathbf{\Sigma}_\theta(\mathbf{x}_t, t))$. The chosen parameterization is an interpolation between $\sigma_t^2 = \beta_t$ and $\sigma_t^2 = \tilde{\beta}_t$ in the log domain, representing the upper and lower bounds of the variance. The former is the variance given by $q(\mathbf{x}_0)$ being Gaussian noise (unknown, random final generated sample). The latter is the variance given a delta function (deterministic final generated sample, recall Eq. (12)). The model outputs a vector, v, with one component per dimension. This can be turned into a variance as follows [37]:

$$\mathbf{\Sigma}_\theta(\mathbf{x}_t, t) = e^{\left(v \log \beta_t + (1-v)\log \tilde{\beta}_t\right)} \tag{17}$$

v is unconstrained, meaning the model could predict variances outside the previously mentioned range. Although in practice, this is not observed [37].

We now define a final, hybrid, objective:

$$L_{\text{hybrid}} = L_{\text{simple}} + \lambda L_{\text{vlb}} \tag{18}$$

λ is set to 0.001 to prevent L_{vlb} from overwhelming L_{simple}.

2) Neural Network Architecture (U-Net): The choice of neural network is heavily determined by the requirement to have equal input and output dimensions. The U-Net is the most common for DDPMs [41], and will also be utilized here. The U-Net [15] has an encoder-decoder structure (contracting path and expansive path) of blocks. The contracting path diminishes spatial dimensions and enlarges the number of feature channels with downsampling. Compressing the input, each block contains a combination of convolutional layers, group/batch normalizations, attention layers, and max-pooling layers. After the contracting path, there is the bottleneck. There, the sample is in its most compressed and abstracted form. Next is the expansive path, where the spatial information is upsampled. This path mirrors the contracting path, resulting in the output having the same dimensions as the input. Skip connections link encoder feature maps to the corresponding decoder feature maps. This is crucial to the success of this architecture. They assist in the creation of better features and prevent gradient degradation during backpropagation [41]. The network configuration used for nowcasting was based on Nichol and Dhariwal's codebase (OpenAI), described in Appendix A. The network configuration used for forecasting was based on an adaptation by Rombach et al. [42], described in Appendix B.

C. Inference

Currently, the model is trained as an unconditional decoder. Starting with Gaussian noise, it is possible to sample semantically correct wind fields based on the data distribution on which it has been trained. However, the goal is to guide the final sample with an input. In the case of nowcasting, this input is a partially complete wind field, and in the case of forecasting, this is a complete nowcasted field at the 0th time frame. The following sampling procedure, the so-called Repaint approach of Lugmayr et al. [43], will ensure this happens by adjusting the unconditional reverse process.

After each reverse process step, a masking procedure is applied to condition the next reverse process step on the known region. Let m denote a mask, which is the same size as the input; it contains a one at every pixel location that is known, and a zero at every unknown pixel location. The known region, $\mathbf{x}_{t-1}^{\text{known}}$, at any reverse process step is then defined by Eq. (19), which is deterministic. The unknown region, $\mathbf{x}_{t-1}^{\text{unknown}}$, is sampled from the model and is defined by Eq. (20). The output of the reverse step is a combination $\mathbf{x}_{t-1}^{\text{known}}$ and $\mathbf{x}_{t-1}^{\text{unknown}}$, as given by the mask, m, in Eq. (21). This procedure is illustrated in Fig. 5.

$$\mathbf{x}_{t-1}^{\text{known}} \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\,\mathbf{I}\right) \tag{19}$$

$$\mathbf{x}_{t-1}^{\text{unknown}} \sim \mathcal{N}\left(\mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)\right) \tag{20}$$

$$\mathbf{x}_{t-1} = m \odot \mathbf{x}_{t-1}^{\text{known}} + (1-m) \odot \mathbf{x}_{t-1}^{\text{unknown}} \tag{21}$$

Applying this procedure solely guides the output of the DDPM based on the given context from the known region. However, the surrounding unknown regions are not semantically coherent with the mask. The unknown regions are, thus, not harmonized well enough with the known regions [43]. To create harmony in the final output, a characteristic of a DDPM - the aim to create structures consistent with the training data distribution - is leveraged. The concept of resampling allows this characteristic to be utilized optimally. Resampling involves diffusing the output of a reverse step, $\mathbf{x}_{t-1}$, back in the forward direction, to x_t, by sampling from Eq. (2). Despite this adding noise back to the sample, some information in $\mathbf{x}_{t-1}^{\text{unknown}}$, which was conditioned by $\mathbf{x}_t^{\text{known}}$, is preserved in $\mathbf{x}_t^{\text{unknown}}$. This results in a new $\mathbf{x}_t^{\text{unknown}}$ that is, firstly, more harmonized with $\mathbf{x}_t^{\text{known}}$ and, secondly, contains conditional information from it. Despite harmonizing a single step, it will not add to the semantic meaning of the entire reverse process. To achieve the latter, a time horizon for resampling is defined. This is the jump length, j. In the discussed example, $j = 1$. The number of times this resampling procedure is done, per jump length, is defined as r. In the final model, T is set to 250 timesteps, $r = 10$, and $j = 10$. The 250 timesteps indicate that, during the inference process, given that a total of 1000 diffusion steps were taken during training, four reverse diffusion steps are executed per inference timestep. This is to speed up the inference process. Furthermore, $j = 10$ indicates that ten inference timesteps are taken before resampling occurs, and $r = 10$ indicates that the resampling process occurs ten times. The result is that, starting at inference step 250, the sample will be denoised to inference step 240. Then, it is renoised, with the conditioning information gained, ten times back and forth between inference steps 240 and 250, before being denoised to inference step 230. Here it will be denoised and noised ten times between inference steps 230 and 240, before taking reverse steps to inference step 220, and so on. A pseudocode for the entire inference procedure is provided in Algorithm 1 for a situation where $j = 1$ and $U = r$. More details on the implementation can be found in the original paper from Lugmayr et al. [43]. This procedure is unique and useful for this application because the mask can be anything. The mask is not a part of the training process.

Algorithm 1 Inpainting using the RePaint approach [43].

```
$x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ do
    for $u = 1, \ldots, U$ do
        $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\epsilon = \mathbf{0}$
        $x_{t-1}^{\text{known}} = \sqrt{\bar{\alpha}_t}x_0 + (1-\bar{\alpha}_t)\epsilon$
        $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $z = \mathbf{0}$
        $x_{t-1}^{\text{unknown}} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t z$
        $x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1-m) \odot x_{t-1}^{\text{unknown}}$
        if $u < U$ and $t > 1$ then
            $x_t \sim \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I})$
        end if
    end for
end for
return $x_0$
```

V. Nowcasting and Forecasting

The essence of the nowcasting problem is to learn a model of the data distribution of a wind field at cruise altitude. Once learned, samples can be generated from the model. The sample generation can then be conditioned with the available part of the wind fields to generate a semantically correct, harmonized, complete wind field. The forecasting problem is similar, with time as an extra dimension, and the mask covering (only) the entire nowcasted frame. The domain of interest in this research is a 528km by 528km box centered on Schiphol. This chapter outlines the training routine for nowcasting and forecasting (Subsection V-A). Next is the validation setup, which includes the error metrics and the nowcasting and forecasting setup (Subsection V-B).

A. Training Setup

Both the forecasting and nowcasting models have been trained in two spatial dimensions due to computational constraints. Given the three channels, U-component wind, V-component wind, and temperature, when nowcasting in two spatial dimensions, it is already 3D for the computer. Adding a third spatial dimension is equivalent to 4D computations. In the case of forecasting, with the added time dimension, it is equivalent to 5D. With the available computing power, it was not possible to train the models in more than two spatial dimensions.

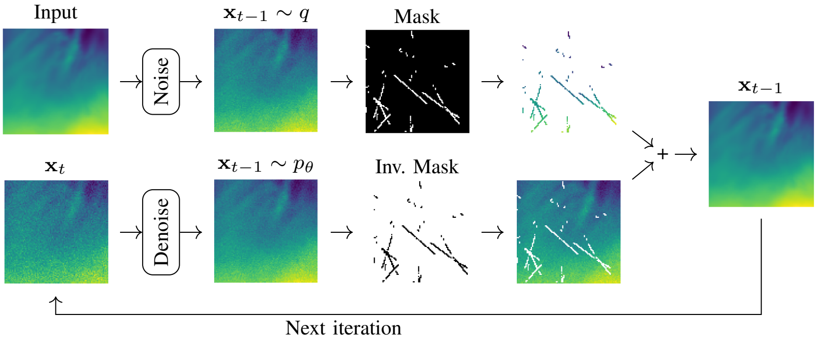


Fig. 5. Illustration of Repaint approach to conditional inference of a DDPM. At each step, the known region is sampled from the input (top), and the unknown region is sampled from the output of the previous step (bottom). In the first reverse step, the unknown region is sampled from the Gaussian noise prior. This figure is inspired by a figure from Lugmayr et al. [43].

1) Nowcasting Model Training: The nowcasting model was trained on samples from the CERRA dataset. The subset used contained 192 samples on every even day of the month in 2020, at model level 45, corresponding to FL400. The wind is assumed to behave the same way between FL300 and FL400. This is a necessary assumption due to computational constraints. Training a model per flight level is not possible within the given timeframe. See Appendix A for the sample transformations, model parameters, and diffusion parameters. The author found good results after 500,000 training steps. Further training provided little benefit.

2) Forecasting Model Training: The forecasting model was trained on a subset of the CERRA dataset containing 186 samples, each containing four frames from 12:00, 15:00, 18:00, and 21:00 on every even day of the month in 2018, again at model level 45 (see Paragraph V-A1). See Appendix B for the sample transformations, model parameters, and diffusion parameters. The author trained the model for 800,000 steps, and no notable difference in results was observed by increasing the number of training steps.

B. Validation Setup

First, the error metrics and the general behavior of the wind above The Netherlands will be described, followed by the validation setups for nowcasting and forecasting.

1) Error Metrics: Error metrics allow comparison between this work and others.

a) Absolute Magnitude Error: The absolute magnitude error (ΔV_{mag}) is computed by taking the absolute difference (Eq. (22)).

$$\Delta V_{\text{mag}} = |\overrightarrow{V}_{\text{pred}} - \overrightarrow{V}_{\text{obs}}| \tag{22}$$

b) Absolute Direction Error: The direction error ($\Delta\theta$) can be calculated with a rearranged version of the dot product formula (Eq. (23)).

$$\Delta\theta = \arccos \frac{\overrightarrow{V}_{\text{pred}} \cdot \overrightarrow{V}_{\text{obs}}}{|\overrightarrow{V}_{\text{pred}}| \cdot |\overrightarrow{V}_{\text{obs}}|} \tag{23}$$

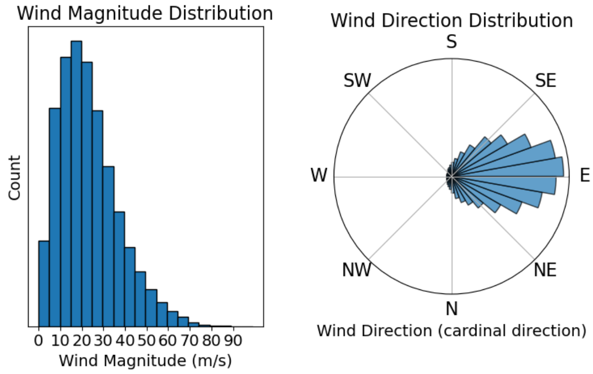


Fig. 6. Distributions of wind magnitude and direction of the CERRA training dataset in the area of interest (528km by 528km box centered on Schiphol). The magnitude of the wind in generally in the order of 10s of m/s and the direction is almost exclusively Westerly (from West to East).

c) Root-Mean Square Error (RMSE): The Root-Mean Square Error (RMSE) can be determined by Eq. (24). It combines the wind field's magnitude and direction error in one metric.

$$\text{RMSE} = \sqrt{\frac{\Sigma[(u_{\text{pred}} - u_{\text{obs}})^2 + (v_{\text{pred}} - v_{\text{obs}})^2]}{n_{\text{obs}}}} \tag{24}$$

2) Behavior of Wind Above The Netherlands: The absolute magnitude error and absolute direction error metrics can be put into context by the distributions of the magnitude and the direction of representative wind conditions in the domain of interest (528km by 528km box centered on Schiphol). This is done in Fig. 6 with the entire CERRA training dataset. Regarding wind magnitude, it is generally in the order of 10s of m/s. With occasional locations where the wind is less than 5 m/s, the most common conditions are between 5 m/s and 40 m/s. Sometimes, the magnitude increases above 50 m/s, but this is uncommon. Regarding the wind direction, it is almost exclusively Westerly (from West to East), nearly always remaining between a North-Westerly and South-Westerly wind.

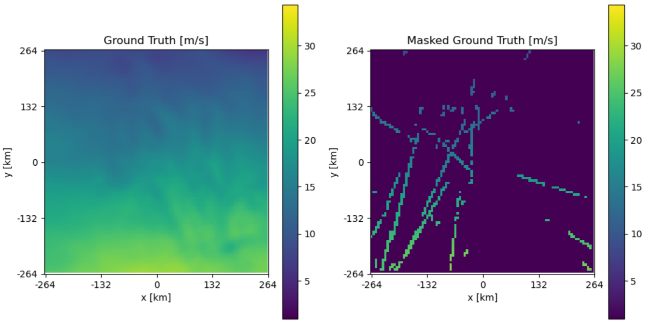


Fig. 7. An example of the U-component of a complete wind field from the CERRA dataset (left) and the input to the model (right). The input to the model is the Ground Truth wind field masked with real aircraft trajectories.

3) *Nowcasting Validation:* The nowcasting model was validated in two ways. First, a mask based on real aircraft paths was used over a complete reanalysis wind field. The model output was then compared to the entire unmasked wind field. Second, data derived from aircraft measurements was split into model input and validation sets. A nowcast was done with the input set, and the model output was compared to the validation set at locations where the validation set was present. Both methods utilized the sample aircraft trajectory dataset (EMADDC). The first method only used the flight paths, whereas the second used the wind/temperature data derived from the aircraft measurements. The aircraft trajectory set came from between approximately FL362 and FL372, from 15:00 to 15:15, in January or February 2024. The goal was to use data from all aircraft climbing to or at FL370. This is generally the flight level where most aircraft observations are available [14]. The time of day was chosen arbitrarily to represent daytime flight activity. The period was selected based on the most recent available EMADDC data.

a) Using Reanalysis Data: Using 190 wind fields from the 2019 CERRA dataset, yet unseen by the model, in combination with real aircraft flight paths to create a mask, the model was validated. The masks were based on a random element of the set of real aircraft trajectories mentioned above. Fig. 7 visualizes an example of this. The remaining masks are in Appendix C. The model's output is a complete windfield and was compared to the entire ground truth field using both RMSE and absolute magnitude/direction error.

b) Using Aircraft Measurements: The model was also validated with 190 samples at 15:00 from July to December 2024, and splitting this into model input and validation subsets (Fig. 8). The input subset consisted of 80% of the flights in a trajectory set, where the remaining 20% was used for validation. The input subset was fed into the model to generate a complete nowcast, and the output was compared at locations where the validation subset had a value with both RMSE and absolute magnitude/direction error.

4) *Forecasting Validation:* The forecasting model was validated using an input of the same description as the training data, but unseen to the model. The validation data was taken from the period January and February 2024. In this forecasting case, the mask is the first frame of the series. The forecasted frames of the output were compared to the ground truth CERRA frames using both RMSE and absolute magnitude/direction error.

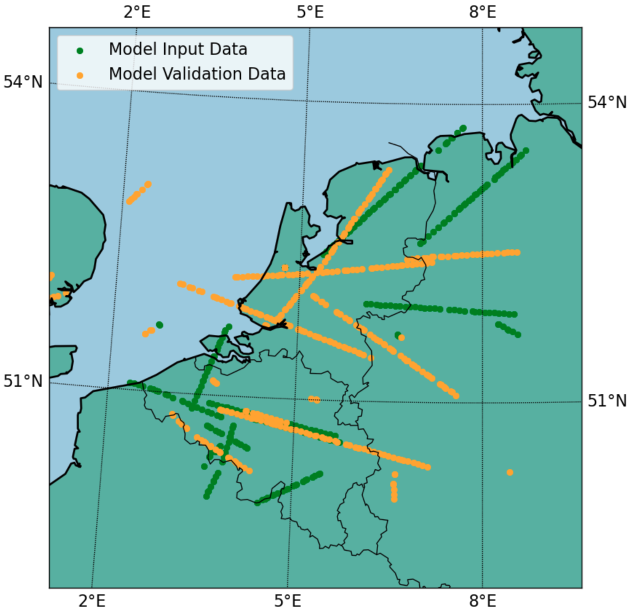


Fig. 8. An example of the model input (green) and validation (orange) subsets from a complete EMADDC sample window.

VI. Results

This section will present the results. First, nowcasting results will be presented with reanalysis data and aircraft measurements (Subsection VI-A). Second, forecasting results will be presented with reanalysis data (Subsection VI-B).

A. *Nowcasting Results*

The nowcasting model was sampled 190 times with each method. The error metrics discussed only measure results for unmasked areas. The average magnitude error was 2.91 m/s for the reanalysis method and 2.03 m/s for the aircraft measurement split method. Fig. 9 shows boxplots of these distributions. The average directional error was 11.3° for the reanalysis method and 4.3° for the aircraft measurement split method. Fig. 10 shows boxplots of these distributions.

To put the outliers into perspective, within the area of interest, the training dataset showed magnitudes generally between 5 m/s and 40 m/s and directions within a 90° range. This indicates that occasionally the model predicts a magnitude and direction potentially on the opposite extreme of the distribution of common wind behavior.

Furthermore, the average RMSE was 4.94 m/s and 3.99 m/s, respectively. The reanalysis data method displaying a higher RMSE is attributed to the fact that the validation pixels are on average much further from the mask pixels than with the aircraft data split method. This is because generally aircraft

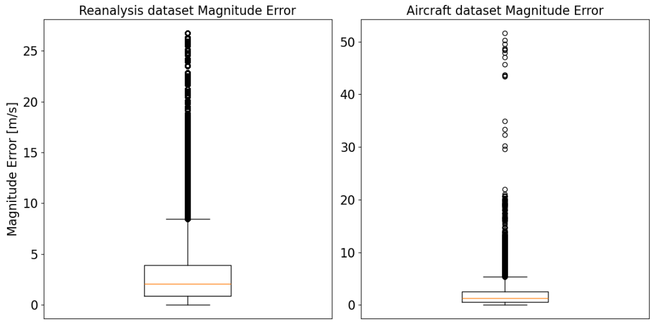


Fig. 9. Boxplot showing nowcasting magnitude error results. The results were based on 190 samples using the reanalysis data method (left) and 190 samples using the aircraft measurement split method (right).

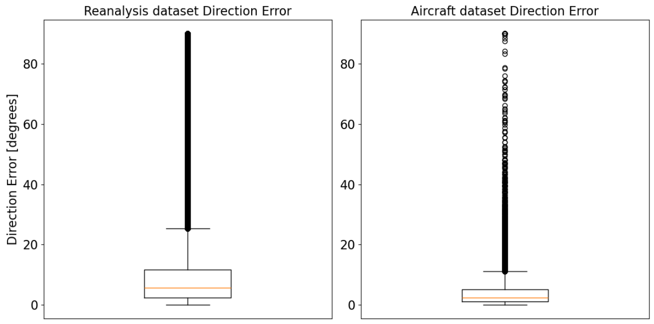


Fig. 10. Boxplot showing nowcasting directional error results. The results were based on 190 samples using the reanalysis data method (left) and 190 samples using the aircraft measurement split method (right). Outliers above 90° have been removed from the plot for readability.

paths cluster in specific areas. Some of these paths are masks, and others are validation pixels. Previous work, namely the MP model [7] and the PINN [14], show an average magnitude error of 4.78 m/s and 2.85 m/s, respectively, and an average directional error of 17.3° and 11.2°, respectively. The DDPM improves on the best of these (PINN) by 29% and 62% for magnitude and directional error, respectively, when using the results from the aircraft measurement data method. The results from the aircraft measurement data validation method are used because it is the same validation method as used by Malfliet with the PINN.

Fig. 11 shows an example of a single realization of the model with CERRA validation data. This figure illustrates how errors are distributed within the domain, depending on the available flight measurements. The top right of the domain has few measurements, and this is also where the magnitude error is the highest. Furthermore, in areas of rapid changes of magnitude, as seen in the center of the top half of the ground truth, the model struggles.

Experiments with 5x ensemble nowcasting were also performed; these produced better results than single realizations. Despite this, the author concluded that using an ensemble, firstly, removed too much detail from the output, and secondly, could not be fairly evaluated as an ensemble average well-suited to the available error metrics. This topic is elaborated upon in Appendix D.

B. *Forecasting Results*

The forecasting model was sampled 50 times using the reanalysis data method. The error metrics discussed only measure results for unmasked areas. The average magnitude and directional errors at the various forecasting horizons are presented in Table I. Boxplots are presented, showing the distributions of the magnitude (Fig. 12) and directional (Fig. 13) errors at the various forecasting horizons.

Despite being able to produce coherent and semantically correct wind fields, the forecasting model was unsuccessful in making useful predictions. The average magnitude error, directional error, and RMSE are high and are the same across all forecasting horizons. Given the distribution of the wind magnitude and direction (Subsubsection V-B2), these averages suggest that the model output is no better than a guess within the range of the most common conditions, around the mean of the input. This is supported by the fact that the error does not increase with forecasting horizon, which is expected.

The development of errors within the domain of the forecasting model is less clear than for nowcasting. Illustrated in Fig. 17 of Appendix E, the model struggles to be conditioned by the input (nowcast). This behavior is suggestive that the mean of the input conditions the output well, but the details are not utilized.

Table I
Forecasting errors using the reanalysis data method at various time horizons.

Forecasting horizon (hours)	3	6	9
Magnitude error (m/s)	8.10	8.11	8.10
Directional error (°)	35.93	35.98	36.04
RMSE (m/s)	11.27	10.76	10.29

VII. Discussion

Paramount to safety and efficiency in ATM, this work aimed to employ aircraft surveillance data in combination with a Denoising Diffusion Probabilistic Model (DDPM) for the nowcasting (inpainting) and forecasting (image-to-video) of wind fields. With the available data sources (Section III), a DDPM (Section IV) was applied to both the nowcasting and forecasting problems (Section V). The model was tested and showed successful results when nowcasting, but was unsuccessful in the forecasting task.

The assumptions made in this work are discussed in Subsection VII-A. Subsequently, the nowcasting results will be examined (Subsection VII-B), followed by the forecasting results (Subsection VII-C). Lastly, recommendations for further research are provided (Subsection VII-D).

A. *Assumptions Underlying the Model's Application*

The necessary assumptions will be presented to clarify the context in which the results of this research apply. This list does not include model architecture-level assumptions as they are not relevant to the application.

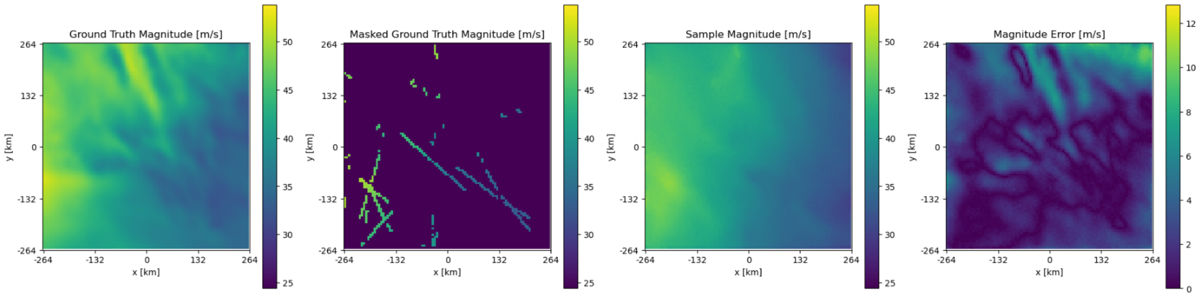


Fig. 11. Wind magnitude plots of a single realization of the nowcasting model. From left to right, the ground truth, the masked ground truth (the model input), the model output (sample), and the magnitude error.

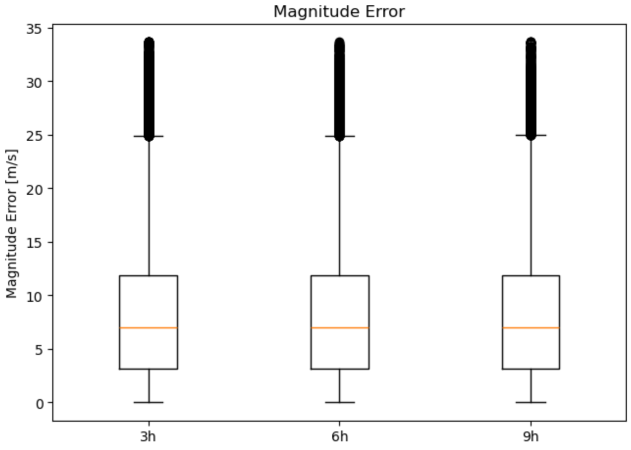


Fig. 12. Boxplots showing the forecasting magnitude error results, sampled 50 times. From left to right, the distributions of the error at forecasting horizons of 3, 6, and 9 hours.

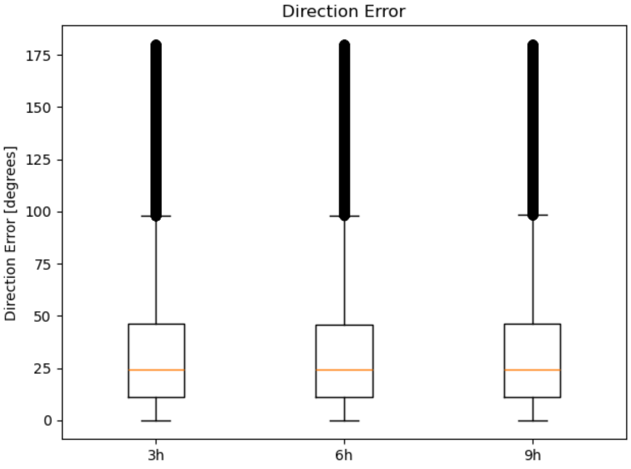


Fig. 13. Boxplots showing the forecasting directional error results, sampled 50 times. From left to right, the distributions of the error at forecasting horizons of 3, 6, and 9 hours.

1) 2-Dimensional flow on constant pressure levels (isobaric): The model utilizes input data from a vertical range of 1000ft, and outputs the predicted wind field in this vertical range. No data from above or below is used to assist in the prediction. Hence, no interaction is assumed, despite there being interaction in practice. This assumption was necessary due to computational constraints. Extending the model to three spatial dimensions with 3D convolutions is straightforward to implement, but requires ~50x more compute time.

2) Stable conditions at the scale of minutes and kilometers: The model assumes stable conditions within a 15-minute time window at constant locations on the assimilation grid. To give the model sufficient input data to generate reasonable results, data derived from aircraft over these 15 minutes was needed. The displacement of these conditions in space was limited to within the model's 5.5km square grid elements. Therefore, severe weather conditions remain an issue, as is with most other current models. In nominal flow conditions, with gradual changes in wind magnitude and direction, this assumption has little impact.

3) Same wind behavior between FL300 and FL400: With the initial goal of nowcasting and forecasting in three spatial dimensions not being achieved due to computational constraints, the approach for 3D nowcasting and forecasting has been to stack 2D models on top of each other. Again, due to computational limitations, it was not realistic to train a 2D model on each model level in the dataset. Therefore, the assumption was made that the wind's behavior is the same at all cruise altitudes. This allowed the use of one 2D model for all of the isobarics.

B. Nowcasting Discussion

The nowcasting implementation of the DDPM demonstrated notable improvements over existing wind field nowcasting methods. All results are presented in Subsection VI-A. This subsection offers possible explanations for these results.

The model's accuracy is highly dependent on the mask coverage. Results showed that in regions with dense aircraft measurements, performance was strong and the model effectively captured the complexity of the wind field. Conversely, in areas with sparse mask coverage, errors increased rapidly as the distance from known wind conditions grew. Small clusters of known pixels, typically caused by aircraft climbing through the FL of interest, do not provide sufficient conditioning. Instead, a wider spatial distribution of known pixels, such as those from an aircraft cruising at the relevant FL, is necessary to guide the model toward an accurate reconstruction of the complete wind field. This is illustrated in Fig. 11.

In conditions involving convergence or divergence, where abrupt changes in wind magnitude and direction occur, the availability of sufficient input data becomes critical. These complex scenarios are underrepresented in the training data, as they are rare compared to the prevailing Westerly winds over Europe. This highlights a key limitation of using deep learning: functioning as a black-box model. Modifying it to better capture rare phenomena requires altering the training dataset itself. This limitation also applies to the PINN and is a broader characteristic of this class of predictive models.

Moreover, for any given input, multiple semantically valid wind field completions may exist. A DDPM generates different realizations with each sampling, based on its learned representation of the data distribution. This behavior is analogous to traditional image inpainting, where a partially known image may correspond to many plausible completions. Consequently, the model cannot express a level of confidence in its predictions.

The RMSE of the DDPM is significantly lower than that of both the MP model and the PINN. The author considers this the primary contribution of this work. The lower RMSE indicates a reduced spread in the prediction error. While the inability of DL-based models to quantify uncertainty remains a drawback, the consistently lower variance in the error implies a low maximum error, which can serve as a basis for assessing the model's suitability for operational use.

C. Forecasting Discussion

The forecasting implementation of the DDPM failed to produce sensible results. The predicted wind fields were semantically correct, but incoherent with the truth fields. This indicates that the model was unable to learn the temporal evolution of the wind field distribution. Although the exact reason for this failure is unknown, several potential explanations are considered.

Incorporating the time dimension into the model architecture involved switching to 3D convolutions and appropriately handling time during the upsampling and downsampling stages. Aside from increasing computational costs, adding an extra dimension may have introduced sufficient complexity to the data distribution, making it too difficult for the model to learn effectively.

Forecasting horizons of 3, 6, and 9 hours were selected based on the availability of CERRA data. However, these time intervals may be too long for the model to handle accurately. The further into the future a forecast attempts to reach, the more possible future states can arise from a given nowcast input. Without guidance from physics and with only a limited training distribution, the model may struggle to converge on a meaningful distribution of the wind. This is supported by the results: both the magnitude and variance errors are high but relatively constant, suggesting that the input exerts minimal influence on the output.

One possible improvement is to evaluate the forecasting model at shorter time horizons, similar to the aircraft-based validation used for nowcasting. For example, using partial future wind fields from aircraft measurements to compute forecast errors could provide insight into model performance. However, obtaining suitable training data for this approach is challenging. The only feasible option may be to use the nowcasting model to generate synthetic training data for the forecasting model. While this is technically viable, it introduces the risk of model collapse, as training a model on its own generated outputs can quickly degrade performance.

D. Recommendations for Future Work

This research has started investigating the application opportunities of a DDPM to wind nowcasting and forecasting. There are several promising opportunities the author recommends be explored in future research.

A significant assumption is the exclusion of vertical interaction between model layers. This is not representative of the atmosphere, and much could be gained in exploring the effects of these interactions. Given the immense computational requirements of the necessary 3D convolutions, the author recommends incorporating information from the directly vertically adjacent model levels when generating a wind field. Modeling all desired levels at once is ideal, but it is not realistic.

Similar model architectures could be explored, such as using a Transformer [28] as the neural network backbone. Furthermore, an Latent Diffusion Model (LDM) [44] could be used to reduce computational requirements. These models incorporate an encoder and decoder around the DDPM, such that the noising and denoising processes are performed in latent space rather than directly in pixel space, as is the case with conventional DDPMs. A limitation of this approach is that it further reduces the interpretability of the model's intermediate representations, which are already inherently difficult to understand.

Using aircraft-derived data as input to a model has drawbacks at times of low aviation activity. Although the current application is aviation, meaning that the model is most needed at times of high aviation activity, weather modeling has a much broader scope. Incorporating existing NWP data could significantly improve results for other deployments. The author recommends exploring the convex combination (weighted average) with a NWP-derived background field during the inference process to guide the model output.

The model is trained to generate samples based on the data distribution of its training set. Currently, the model struggles with severe weather conditions. These are inherently less common than nominal conditions and, therefore, less common in the training data distribution. Training the model on only severe conditions could be beneficial when it is clear from the aircraft-derived data that the conditions are not nominal.

The scope of this study is cruise altitudes. This is a good starting point, as this is where most of the aircraft-derived data resides. Nevertheless, the applications of DDPMs in the aviation and ATM sectors are numerous. Combining aircraft-derived measurements with stationary ground sensors provides prospects for wind estimation in and around airports. This has applications for determining active runway use and safety considerations on the apron related to weather.

VIII. Conclusion

The purpose of this study was to explore the effectiveness of using a Denoising Diffusion Probabilistic Model (DDPM) for the nowcasting (inpainting) and forecasting (image-to-video) of wind fields when using aircraft-derived meteorological data. The experiments demonstrated that a DDPM with a U-Net neural network backbone is useful for nowcasting wind fields, given a reasonable set of aircraft-derived meteorological data as a starting point. Forecasting remains challenging, and the author could not produce sensible results. Nowcasting results of previous studies', including the Lagrangian transportation-based Meteo-Particle (MP) model [7] and the Physically Inspired Neural Network (PINN) [14], have been improved upon - 29% in magnitude error and 62% in directional error when compared to the previous best (PINN), with the same validation method. The nowcasting magnitude error is 2.03 m/s, and the directional error is 4.2° when using a test set of 190 samples at 15:00 from July to December 2024. The primary contribution of the DDPM lies in its ability to produce results with significantly reduced error variance, demonstrating markedly tighter error distributions compared to prior approaches. This was established by improving the RMSE of the PINN by 29% to 3.99 m/s.

The ECMWFs CERRA reanalysis dataset was used for training both the nowcasting and forecasting models. Simulation of real-world nowcasting performance was done by masking (previously unseen) complete CERRA windfields with real aircraft tracks, and comparing the model output to the true wind field. Additionally, validation was done by splitting real aircraft-derived meteorological data into a model input set and an output validation set. The DDPM can successfully reconstruct wind fields, given a reasonable amount of input data equivalent to typical daytime aircraft activity in a 528km by 528km square centered around Schiphol.

The results of this research contribute to the rapidly expanding field of generative Deep Learning (DL) model applications. It has demonstrated that DDPMs can also be applied in fields seemingly unrelated to the conventional use of daily image and video generation. However, limitations of this work exist. First, the immense computational power required to train the model restricts testing potential. Second, the issue of handling converging and diverging weather conditions and quantifying confidence in the accuracy of the generated wind fields remains unsolved. Lastly, generating reasonable forecasts is yet to be successfully implemented.

Regarding future research, the author has several recommendations. The extension into the vertical spatial dimension could provide performance improvements. Exploring other variants of the DDPM could reveal architectures better suited to this specific application. Moreover, the inclusion of NWP assimilations to guide the inference process in low-observed areas could broaden the scope of its applications. Similarly, training the model on severe weather conditions is expected to improve its performance in this domain. The author sees many opportunities for this model class in the ATM sector beyond wind field generation at cruise altitudes. Investigating the usefulness of applications at lower altitudes and closer to airports could provide efficiency and safety benefits for runway and apron operations.

Acknowledgements

The author would like to thank Junzi Sun and Evert Westerveld for their superb guidance on this project. Furthermore, the author is grateful for the reanalysis data provided by the ECMWFs Climate Data Store and EMADDC data supplied by the The Royal Netherlands Meteorological Institute (KNMI). Finally, sincere thanks are extended to the many employees at LVNL whose contributions—whether administrative or through insightful discussions during informal exchanges—have supported and enriched this work.

The code used in this research can be found at: https://github.com/matthijsslobbe/wind-ddpm.

References

[1] M. Steiner, R. Bateman, D. Megenhardt, Y. Liu, M. Xu, M. Pocernich, and J. Krozel, "Translation of ensemble weather forecasts into probabilistic air traffic capacity impact," *Air Traffic Control Quarterly*, vol. 18, no. 3, pp. 229–254, 2010. [Online]. Available: https://doi.org/10.2514/atcq.18.3.229

[2] M. Marinescu, A. Olivares, E. Staffetti, and J. Sun, "On the estimation of vector wind profiles using aircraft-derived data and gaussian process regression," *Aerospace*, vol. 9, no. 7, 2022. [Online]. Available: https://www.mdpi.com/2226-4310/9/7/377

[3] M. Leutbecher and T. Palmer, "Ensemble forecasting," p. 31, Feb. 2007. [Online]. Available: https://www.ecmwf.int/node/10729

[4] J. Jeppesen, "Fact sheet: Ensemble weather forecasting," Mar. 2017. [Online]. Available: https://www.ecmwf.int/en/about/media-centre/focus/2017/fact-sheet-ensemble-weather-forecasting

[5] P. Bauer, A. Thorpe, and G. Brunet, "The quiet revolution of numerical weather prediction," *Nature*, vol. 525, pp. 47–55, Sep. 2015.

[6] H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horányi, J. Muñoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, A. Simmons, C. Soci, S. Abdalla, X. Abellan, G. Balsamo, P. Bechtold, G. Biavati, J. Bidlot, M. Bonavita, G. De Chiara, P. Dahlgren, D. Dee, M. Diamantakis, R. Dragani, J. Flemming, R. Forbes, M. Fuentes, A. Geer, L. Haimberger, S. Healy, R. J. Hogan, E. Hólm, M. Janisková, S. Keeley, P. Laloyaux, P. Lopez, C. Lupu, G. Radnoti, P. De Rosnay, I. Rozum, F. Vamborg, S. Villaume, and J. Thépaut, "The ERA5 global reanalysis," *Quarterly Journal of the Royal Meteorological Society*, vol. 146, no. 730, pp. 1999–2049, Jul. 2020, publisher: Wiley. [Online]. Available: https://rmets.onlinelibrary.wiley.com/doi/10.1002/qj.3803

[7] J. Sun, H. Vû, J. Ellerbroek, and J. M. Hoekstra, "Weather field reconstruction using aircraft surveillance data and a novel meteo-particle model," *PLOS ONE*, vol. 13, no. 10, pp. 1–33,

Oct. 2018. [Online]. Available: https://doi.org/10.1371/journal.pone.0205029

[8] S. de Haan and A. Stoffelen, "Assimilation of high-resolution mode-s wind and temperature observations in a regional nwp model for nowcasting applications," *Weather and Forecasting*, vol. 27, no. 4, pp. 918 – 937, 2012. [Online]. Available: https://journals.ametsoc.org/view/journals/wefo/27/4/waf-d-11-00088_1.xml

[9] P. De Jong, J. Van Der Laan, A. In 't Veld, M. Van Paassen, and M. Mulder, "Wind-profile estimation using airborne sensors," *Journal of Aircraft*, vol. 51, no. 6, pp. 1852–1863, 2014.

[10] R. Keisler, "Forecasting global weather with graph neural networks," 2022. [Online]. Available: https://arxiv.org/abs/2202.07575

[11] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds, Z. Eaton-Rosen, W. Hu, A. Merose, S. Hoyer, G. Holland, O. Vinyals, J. Stott, A. Pritzel, S. Mohamed, and P. Battaglia, "Graphcast: Learning skillful medium-range global weather forecasting," 2023. [Online]. Available: https://arxiv.org/abs/2212.12794

[12] Z. Xing, Q. Feng, H. Chen, Q. Dai, H. Hu, H. Xu, Z. Wu, and Y.-G. Jiang, "A survey on video diffusion models," 2024. [Online]. Available: https://arxiv.org/abs/2310.10647

[13] J. Zhu, H. Wang, J. Li, and Z. Xu, "Research and optimization of meteo-particle model for wind retrieval," *Atmosphere*, vol. 12, no. 9, 2021. [Online]. Available: https://www.mdpi.com/2073-4433/12/9/1114

[14] J. M. L. Malfliet, "Estimating Wind Fields Using Physically Inspired Neural Networks With Aircraft Surveillance Data," Master's thesis, Delft University of Technology, Delft, The Netherlands, Jan. 2023, available at https://repository.tudelft.nl/record/uuid:7b93584f-0043-4fbd-a16a-610b50429cf3.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[16] National Centers for Environmental Prediction, National Weather Service, NOAA, U.S. Department of Commerce, "Ncep gfs 0.25 degree global forecast grids historical archive," Boulder CO, 2015. [Online]. Available: https://rda.ucar.edu/datasets/d084001/

[17] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014. [Online]. Available: https://arxiv.org/abs/1406.2661

[18] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," 2021. [Online]. Available: https://arxiv.org/abs/2106.12423

[19] A. Sauer, K. Schwarz, and A. Geiger, "Stylegan-xl: Scaling stylegan to large diverse datasets," 2022. [Online]. Available: https://arxiv.org/abs/2202.00273

[20] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," 2021. [Online]. Available: https://arxiv.org/abs/2012.09841

[21] M. Ding, W. Zheng, W. Hong, and J. Tang, "Cogview2: Faster and better text-to-image generation via hierarchical transformers," 2022. [Online]. Available: https://arxiv.org/abs/2204.14217

[22] J. Yu, Y. Xu, J. Y. Koh, T. Luong, G. Baid, Z. Wang, V. Vasudevan, A. Ku, Y. Yang, B. K. Ayan, B. Hutchinson, W. Han, Z. Parekh, X. Li, H. Zhang, J. Baldridge, and Y. Wu, "Scaling autoregressive models for content-rich text-to-image generation," 2022. [Online]. Available: https://arxiv.org/abs/2206.10789

[23] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020. [Online]. Available: https://arxiv.org/abs/2006.11239

[24] A. Hyvärinen, "Estimation of non-normalized statistical models by score matching," *Journal of Machine Learning Research*, vol. 6, no. 24, pp. 695–709, 2005. [Online]. Available: http://jmlr.org/papers/v6/hyvarinen05a.html

[25] Q. Liu, J. D. Lee, and M. I. Jordan, "A kernelized stein discrepancy for goodness-of-fit tests and model evaluation," 2016. [Online]. Available: https://arxiv.org/abs/1602.03253

[26] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," 2020. [Online]. Available: https://arxiv.org/abs/1907.05600

[27] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," 2021. [Online]. Available: https://arxiv.org/abs/2011.13456

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[29] X. Jing, Y. Chang, Z. Yang, J. Xie, A. Triantafyllopoulos, and B. W. Schuller, "U-dit tts: U-diffusion vision transformer for text-to-speech," 2023. [Online]. Available: https://arxiv.org/abs/2305.13195

[30] J. Sun, *The 1090 Megahertz Riddle: A Guide to Decoding Mode S and ADS-B Signals*, 2nd ed. TU Delft OPEN Publishing, 2021.

[31] S. de Haan, "High-resolution wind and temperature observations from aircraft tracked by mode-s air traffic control radar," *Journal of Geophysical Research: Atmospheres*, vol. 116, no. D10, 2011. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2010JD015264

[32] J. Sun, H. Vû, J. Ellerbroek, and J. Hoekstra, "pymodes: Decoding mode-s surveillance data for open air transportation research," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 2777–2786, 2019.

[33] A. de Leege, M. Mulder, and M. M. Van Paassen, "Automatic dependent surveillance-broadcast for meteorological monitoring," *Journal of Aircraft*, vol. 50, pp. 249–261, Jan. 2013.

[34] P. de Jong, S. Haan, J. Sondij, M. Koutek, A. Hoekstra, and J. Bokhorst, "Operational use of aircraft derived data for meteorology and other applications," Oct. 2018.

[35] M. Ridal, E. Bazile, P. Le Moigne, R. Randriamampianina, S. Schimanke, U. Andrae, L. Berggren, P. Brousseau, P. Dahlgren, L. Edvinsson, A. El-Said, M. Glinton, S. Hagelin, S. Hopsch, L. Isaksson, P. Medeiros, E. Olsson, P. Unden, and Z. Q. Wang, "Cerra, the copernicus european regional reanalysis system," *Quarterly Journal of the Royal Meteorological Society*, vol. 150, no. 763, pp. 3385–3411, 2024. [Online]. Available: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.4764

[36] S. Schimanke, L. Isaksson, and L. Edvinsson. Copernicus European Regional ReAnalysis (CERRA): product user guide. [Online]. Available: https://climate.copernicus.eu/copernicus-regional-reanalysis-europe-cerra

[37] A. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," 2021. [Online]. Available: https://arxiv.org/abs/2102.09672

[38] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022. [Online]. Available: https://arxiv.org/abs/1312.6114

[39] C. Luo, "Understanding diffusion models: A unified perspective," 2022. [Online]. Available: https://arxiv.org/abs/2208.11970

[40] Y. Song and S. Ermon, "Improved techniques for training score-based generative models," 2020. [Online]. Available: https://arxiv.org/abs/2006.09011

[41] D. Gallon, A. Jentzen, and P. von Wurstemberger, "An overview of diffusion models for generative artificial intelligence," 2024. [Online]. Available: https://arxiv.org/abs/2412.01371

[42] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer,

"High-resolution image synthesis with latent diffusion models," 2021.

[43] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. V. Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," 2022. [Online]. Available: https://arxiv.org/abs/2201.09865

[44] A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. W. Kim, S. Fidler, and K. Kreis, "Align your latents: High-resolution video synthesis with latent diffusion models," 2023. [Online]. Available: https://arxiv.org/abs/2304.08818

Appendix

A. Nowcasting Training, Model, and Diffusion Parameters

Table II
An overview of all of the nowcasting parameters and settings used to get the presented results.

Variable	Value	Description
Training Parameters		
Random horizontal flip	p=0.5	-
Random vertical flip	p=0.5	-
Random rotation	between 0°and 30°	-
Center crop	to 96x96 pixels[2]	-
Learning rate (σ)	2e-5	-
Batch size	1	-
Exponential moving average (EMA) weight	0.9999	Defines the rate at which past weights of the network are maintained. A value of nearly 1 means that past weights are gradually forgotten.
Weight decay	0.1	L2 regularization strength, preventing overfitting by discouraging large weights in the network.
Diffusion parameters		
Number of diffusion steps	1000	This is the number of times that noise is added/removed during the forward/backward process.
Number of timesteps	250	This is the number of steps taken in the training/inference process. With 1000 diffusion steps, a quarter of the diffusion steps were evaluated. This speeds up training and inference time.
Model parameters		
Image size	3x96x96	96x96 image with 3 channels (U-component, V-component and Temperature).
Number of channels	256	This determines the base number of feature channels in the U-Net.
Number of heads	4	This specifies the number of attention heads in the self-attention layers (both down- and upsampling sides) of the U-Net.
Number of residual blocks	2	Indicates the number of residual blocks at each level of the U-Net.
Number of channels per head	64	This is determined by dividing the number of channels by the number of heads.
Attention resolutions	12,6	Specifies at which spatial resolutions the model applies attention blocks.
Channel multiplier	1,2,3,4	Specifies the feature channel multiplier at each level of the U-net. Given we have 256 channels, the levels would have 256, 256, 512, etc. channels.
Dropout	0.1	This controls the dropout rate (regularization). 0.1 indicates that during each forward pass during training, 10% of the neurons will be set to 0.
Gradient checkpointing	true	This is a GPU memory-saving technique. It slows training time because not all intermediate activations needed for backpropagation are saved. These activations are recalculated during backpropagation. This was necessary given the available hardware.
ResBlock UpDown	true	Determines whether upsampling and downsampling are integrated into the residual blocks (true) or handled by separate layers (false).
Scale Shift Normalization	true	Determines whether to apply learned scaling and shifting of the normalized output (after normalizing the activations within a layer).
Inference Parameters		
Jump length (j)	10	The resampling time horizon. This indicates the number of inference time steps across which resampling occurs (i.e., the total amount of renoising inference time steps).
Number of resamples per resampling procedure (r)	10	The number of re-noising steps within the resampling jump length. It must be smaller than the jump length, and the jump length must be divisible by it.

B. Forecasting Training, Model, and Diffusion Parameters

Table III
An overview of all of the forecasting parameters and settings used to get the presented results. Only additions and changes to values as compared to the nowcasting table (Tab. A.II).

Variable	Value	Description
Model parameters		
Image size	3x4x96x96	There are four tensors in the time dimension. The current winf field (which is the input) and three future realizations.
Seq. factor	-	Addition of a variable in the up- and downsampling blocks of the U-Net to indicate to the model that there is a difference between each of the four frames, and that the sequence is important.

[2] Achieving the desired 528x528 km grid, given the CERRA data's 5.5km grid spacing.

C. Nowcasting Masks for Sampling with Reanalysis Data

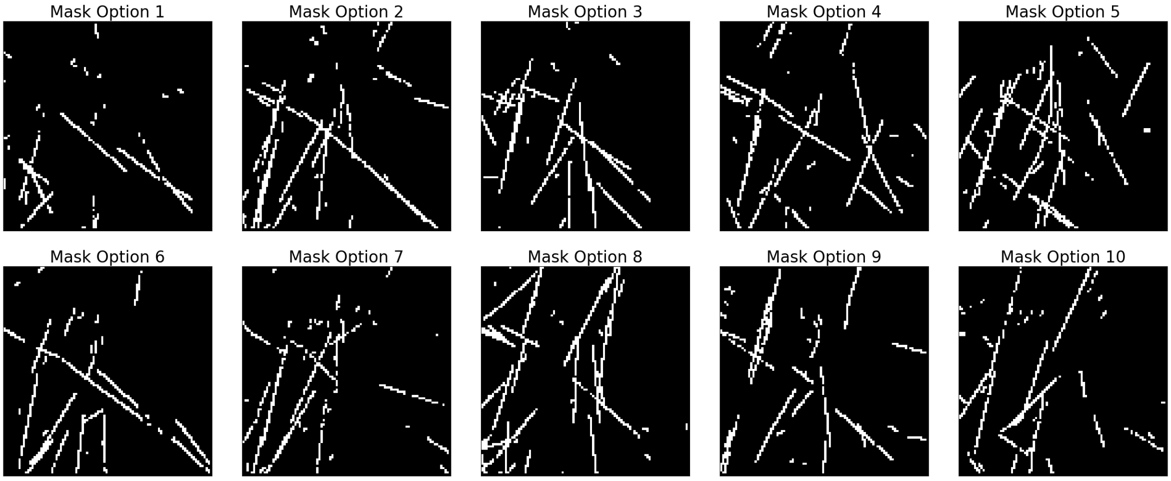


Fig. 14. The set of masks used when sampling the nowcasting model with the reanalysis dataset. For each sample, a random mask from this set is used. They are intended to be representative of nominal daytime flight activity.

D. Ensemble Nowcasting Results

The nowcasting model has also been sampled as an ensemble with the CERRA dataset. With this approach, the model was sampled five times with the same input (i.e., the same ground truth and mask), and the average of each pixel value was taken. In the context of conventional NWP, ensembles are made by perturbing the inputs of a model to various degrees and taking the average of the results. In the context of a DDPM, this initial perturbation is not done because the model's output is different every time. This is due to the nature of the model architecture.

Fig. 15 and Fig. 16 show the wind magnitude output of a single realization of the model and a five-member ensemble realization. The ensemble wind fields are significantly smoothed as compared to the single realizations. Tab. IV and Tab. V display the ensemble errors in comparison to the single realization errors for the reanalysis data and aircraft measurement validation methods, respectively. The ensemble method shows a notable difference, especially with the reanalysis data method. It may be the case that the chosen error metrics are well-suited to the inherent side effects of averaging. The author has decided not to use this ensemble method because the goal of implementing a model as complex as a DDPM is to capture the details of the wind's behavior. By taking the average of multiple model realizations, these fine details are lost. This is also noticeable when comparing Fig. 15 and Fig. 16. Furthermore, no relevant suitable alternative error metrics have been found to compare the methods more fairly.

Table IV
Average nowcasting errors when sampling the model once as compared to a five-member ensemble with the CERRA reanalysis data method.

	Av. Magnitude error (m/s)	Av. Directional error (°)	RMSE
1x Sample	2.91	11.3	4.94
5x Ensemble	2.24	8.5	3.72

Table V
Average nowcasting errors when sampling the model once as compared to a five-member ensemble with the aircraft measurement data method.

	Av. Magnitude error (m/s)	Av. Directional error (°)	RMSE
1x Sample	2.03	4.4	3.99
5x Ensemble	1.92	4.3	3.84

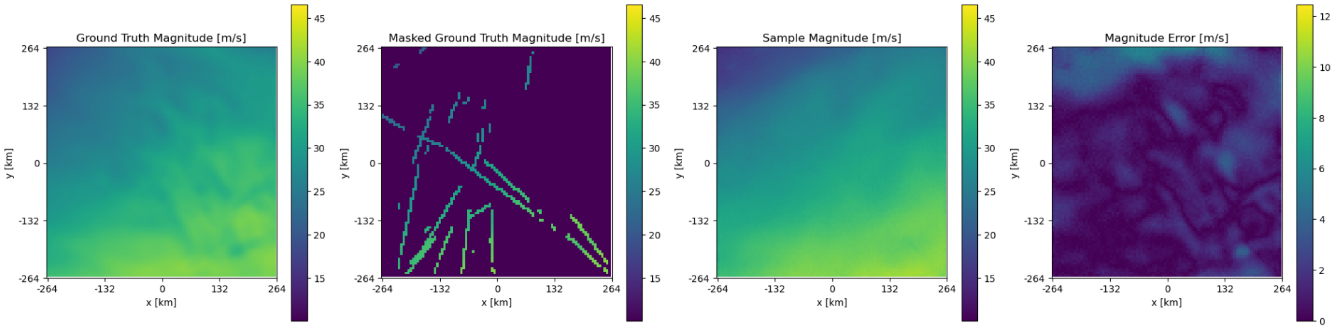


Fig. 15. Wind magnitude output of a single model realization.

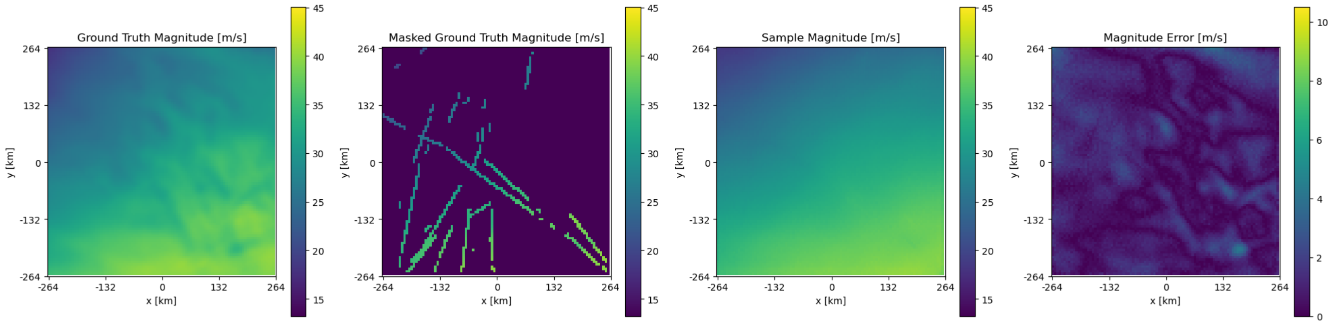


Fig. 16. Wind magnitude output of an ensemble of 5 model realizations.

E. Forecasting Magnitude Error Example

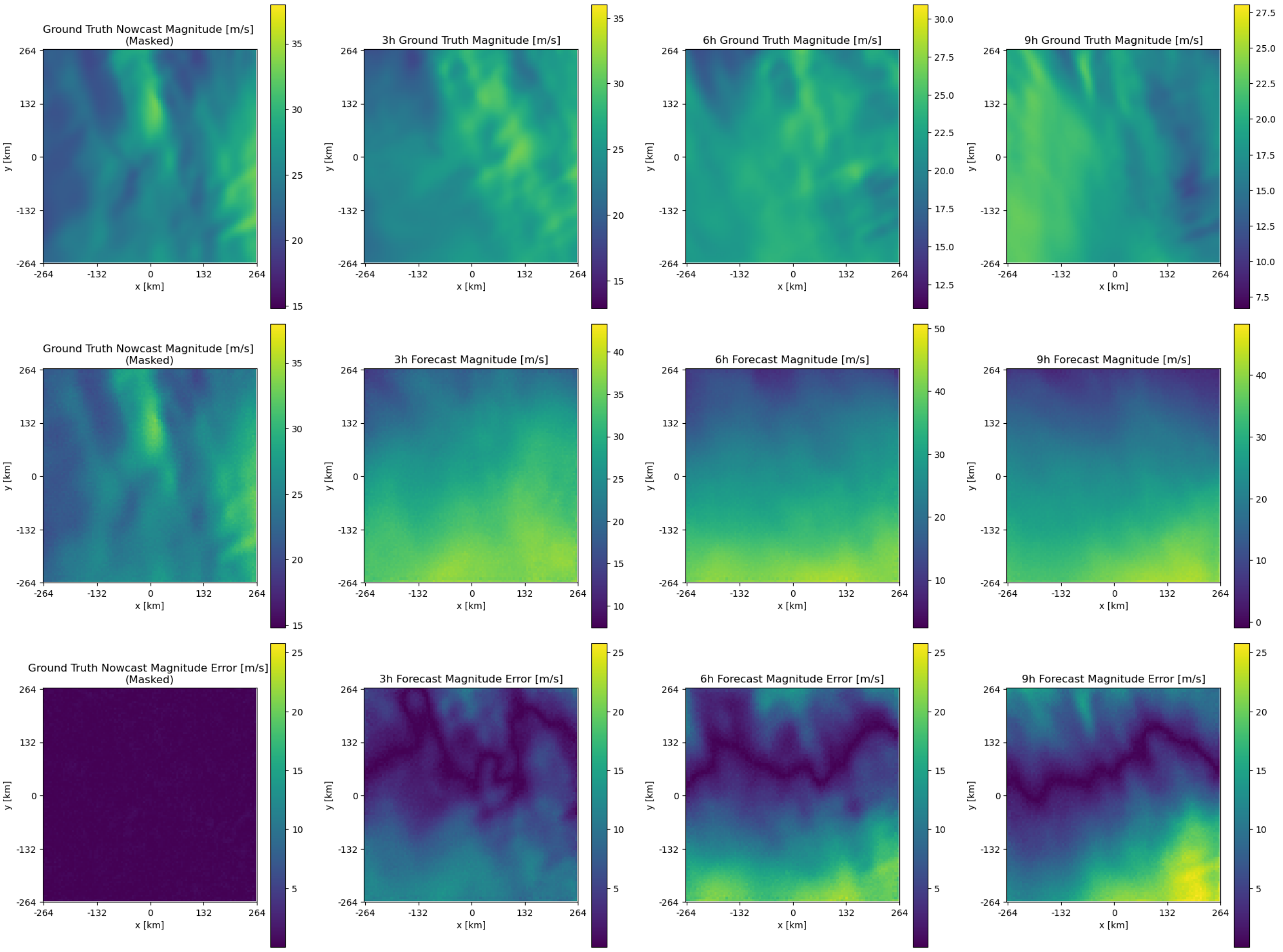


Fig. 17. Wind magnitude plots of a single realization of the forecasting model. From left to right, the ground truth nowcast (model input), followed by the 3-hour, 6-hour, and 9-hour forecasts. The top row is the ground truth for each of the forecasting horizons, the middle row is the model realization, and the bottom row is the magnitude error. The nowcast magnitude error is zero everywhere because it is the input to the model and assumed to be correct.

Part II

Research Proposal and Literature Review

MSc Thesis Research Proposal

Matthijs Slobbe

October 2024

1 Introduction

The weather is unique because it affects everyone - all the time. Good forecasting can save lives and poor forecasting can cost a lot of money (World Meteorological Organization, 2023). The wind plays a large role in weather forecasting, especially in the aviation industry. It is a dominant source of uncertainty in the current and future paradigm of Air Traffic Management (ATM) (Marinescu et al., 2022), influencing almost all aspects of operations. This is not only the case for aviation. Constant development in this sector is essential for facilitating a sustainable future in all domains.

Wind fields are important in determining optimal trajectories and weather risk, as well as active runway use at airports, in addition to contributing to Air Traffic Controller (ATCo) decision-making to keep airspaces manageable, and hence safe (Steiner et al., 2010). Within the commercial aviation industry, there is a high demand for *accurate wind now- and forecasting*, allowing air traffic service providers (ANSPs) and airlines alike to *safely and efficiently* conduct day-to-day operations.

Accurate wind field generation is particularly challenging due to the non-linear dynamic nature of the atmosphere, in addition to uncertainties, and variety in measurement methods as well as numerical solvers (Martin Leutbecher, 2007). Advancements have been, and are constantly being made in an effort to increase the accuracy of forecasts (Bauer et al., 2015). Often, however, these forecasts lack local real-time accuracy. This is exactly what is needed by airlines and ANSPs to make well-informed decisions. This follows from conversations with Air Traffic Control the Netherlands (LVNL).

To address this problem, two-dimensional (2D) wind field reconstruction models have been developed. There is a Monte Carlo method, the so-called Meteo-Particle (MP) model, which propagates particles through space with a Gaussian random walk model (Sun et al., 2018). The wind can then be estimated by interpolating surrounding particles. Furthermore, a physically inspired cost function U-net neural network has also been applied for inpainting partially available wind fields (Malfliet, 2023). These methods both struggled in converging or diverging wind conditions and in less observed areas.

Machine learning (ML) has also been applied with purely data-driven approaches (Keisler, 2022; Lam et al., 2023). Success has been observed in predicting global weather patterns, struggling with extreme weather conditions. A potential solution to this in the future is to deviate from the common minimization of Root Mean Squared Error (RMSE) scoring method (Maskell, 2023). When compared to state-of-the-art Numerical Weather Prediction (NWP) models, ML models lack in uncertainty handling (Lam et al., 2023). Despite this, there is great potential for combined ML and NWP approaches.

Additional research in this domain is needed to investigate real-time spatially dense machine learning techniques for the reconstruction of wind fields. This project will target this with a diffusion probabilistic model using aircraft surveillance data, in three dimensions (3D), applied to the aviation industry. If successful, this work has the potential to be applied further in a wide range of disciplines, such as general weather forecasting.

2 Literature Study

To gain more knowledge on the problem, as well as explore current advancements and challenges in the domain of wind and weather prediction, a literature study has been done on multiple relevant topics. The starting point was previous work with similar research objectives. This was followed by a study of the past and current state of numerical weather prediction. Lastly, machine learning methods in weather prediction were analyzed and the link to other computer vision machine learning methods was made, posing the author with an opportunity for the current research.

2.1 Previous Work on Wind Field Reconstruction

This section will concentrate on the previous work done in an effort to improve wind field reconstruction capabilities at high altitudes. Firstly, the data source and how the data is obtained will be explained, followed by an explanation of the Meteo-Particle model (Sun et al., 2018) and a physically inspired neural network approach (Malfliet, 2023).

2.1.1 Data from aircraft for Partial Wind Fields

Obtaining the data

In the past, multiple methods for extracting wind data from aircraft have been devised. One such method was observing the trajectories of aircraft from the ground and estimating the wind velocity by combining Bayesian filtering and ground speed. This method requires assumptions of quasi-constant wind velocity and constant aircraft airspeed during maneuvers. This idea was first developed by Hollister et al., 1989 and was later extended (Delahaye et al., 2003, D. Delahaye and Puechmorel, 2009).

Another method, applied by The Royal Netherlands Meteorological Institute (KNMI), uses Mode-S to construct wind vectors and temperature observations (De Haan, 2011). This showed good quality wind observations, comparable to the poorly available Aircraft Meteorological Data Relay (AMDAR) observations, after correction and calibration. However, the temperature observations, even after correction were insufficient. Further research managed to further improve the results (De Haan & Stoffelen, 2012).

Since then, data gathered by interrogating aircraft has become the norm for estimating local wind (and other weather) conditions through Automatic Dependent Surveillance-Broadcast (ADS-B) (de Leege et al., 2013, De Jong et al., 2014). Important to note is that ADS-B data was not originally intended for this use. This will be discussed further later in this section.

Currently, as is also used by Sun et al., 2018, the best option is to combine ADS-B and Mode-S data sources to infer the wind at locations where aircraft are flying. ADS-B enabled aircraft broadcast information including GPS coordinates, barometric altitude, and ground speed at a near-constant rate (de Leege et al., 2013). Mode-S is gathered in the form of Comm-B messages when a Secondary Surveillance Radar (SSR) interrogates an aircraft. Included in this down-link includes parameters such as aircraft position, velocity, operational parameters, and meteorological data. It is however challenging to interpret these Mode-S Comm-B replies. This is because the ICAO address of the originating aircraft is not directly known with only the Comm-B message. Additionally challenges in acquiring the content of the message and the quality of the content (certainty of values) exist. Sun et al., 2018 solve these issues as follows.

- The ICAO address can be obtained by performing a reverse parity check of the Comm-B messages and cross-referencing the resulting addresses with ADS-B, which is always correct if the message is not corrupt.
- The content of the messages, or BDS code, contains status bits that indicate whether or not specific aircraft parameters are available in the message. Multiple checks assuming various message times need to be performed since messages can match multiple BDS codes.

- Lastly, the quality of the messages is checked, which may be altered due to faulty aircraft measurements or transmission errors. Messages can also have a lower resolution due to the truncation of values (De Haan, 2011). This process occurs after signal processing and can be done with the PyModeS library, which also handles ADS-B decoding (Sun et al., 2020).

Deriving wind speed and temperature

With the now decoded ADS-B and Mode-S Comm-B reply data, the aircraft state can be used to derive wind speed and temperature. Despite not being intended for this purpose, the computed values are accurate. The following method is used by Sun et al., 2018 to do these computations. Note that only the computations for Mach numbers above 0.3 are shown, as these are the most relevant conditions for this research.

ADS-B provides:

- Barometric altitude
- Ground speed

And Mode-S Comm-B messages provide:

- True airspeed (TAS) (BDS 5,0)
- Mach number (BDS 7,0)

The final goal of the calculations is to get the TAS and the temperature. Although the TAS can be directly read from Mode-S, the update rate of BDS 5,0 is low, hence it is also calculated with the temperature and Mach number. Mach number (BDS 7,0) has a higher update rate. This is not an issue since the temperature was desired anyway, and is assumed to be relatively constant within the update intervals. Given ISA conditions Equation 1 is used to calculate the temperature and Equation 2 is then used to calculate the TAS at an increased update rate. a_0, the speed of sound at sea level, is $340.29ms^{-1}$ and T_0, the temperature at sea level, is $288.15K$.

$$T = \frac{V_{tas,50}^2 \cdot T_0}{M^2 \cdot a_0^2} \tag{1}$$

$$V_{tas} = M \cdot a_0 \sqrt{\frac{T}{T_0}} \tag{2}$$

Combining the resulting TAS with the heading, ground speed, and tack angle the wind vector can be obtained with Equation 3. This is based on Figure 1 (Sun et al., 2018).

$$\overrightarrow{V_w} = \overrightarrow{V_{gs}} - \overrightarrow{V_{tas}} \tag{3}$$

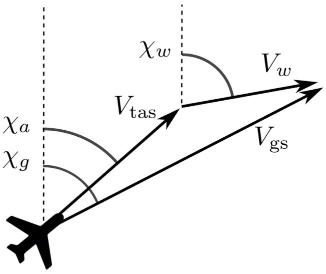


Figure 1: The relation between TAS, GS, and the wind vector. χ_g, χ_a and χ_w are track angle, aircraft heading, and wind direction, with respect to the true north, respectively (Sun et al., 2018).

2.1.2 Meteo-Particle Model

This subsection will describe the model proposed by Sun et al., 2018, which was designed with the same objective in mind as that of this project, although having a different approach. Furthermore, the additions to the model by Zhu et al., 2021 are noted.

The model itself
In the Meteo-Partical (MP) model, particles carry information on the state of wind and temperature through space. They are generated at locations where measurements have been made and propagate through space while decaying over time. One can infer the wind at any location within the spatial bounds by combining the weighted states of nearby particles.

The model requires several assumptions, namely that the true states of wind and temperature are geographically and temporally stable at the scale of tens of kilometers and minutes, respectively. Additionally, it is assumed that the burst error rate of observations from a single aircraft is low despite the implementation of probabilistic measurement rejection within the model. This probabilistic rejection is based on a tolerance parameter and the state of the current wind field.

The particles carrying the desired states propagate through space with a Gaussian random walk model, with a bias in lateral wind direction (but not in vertical wind direction). With each step, the age of each particle is increased and resampling occurs to remove particles outside the spatial boundaries and ensure there are more newer particles than older particles in the system.

To compute the wind at a specific location, the weighted values from neighboring particles are used. The weights are dependent on the relative location of the particle and the distance the particle is from its origin. When the temperature is calculated the origin altitude of the particle is additionally taken into account. Importantly, there is no fixed grid, the domain is continuous.

When the wind and temperature fields have been reconstructed, the confidence level of estimates is determined by a number of factors. These include the number of particles in the vicinity of the target location, the mean distances between these particles and the target location, the homogeneity of states carried by the particles, and lastly the strength of the particles as determined by an aging function.

Short-term forecasting
By making a statistical model as a function of time, in the form of a Gaussian process predictor, a prior over functions is fitted over the observations. With different kernels (covariance functions), the forecast can be derived from the prior. Sun et al., 2018 use a summation of 3 different kernels.

Further optimization of the model
Zhu et al., 2021 used the MP model as a base and improved the precision and accuracy by introducing a mixed evaluation index, documenting periods during the day and altitudes at which the number of aircraft is scarce, and lastly by optimizing constant parameters and control factors of the model. They were able to slightly improve the mode and concluded that the MP model is suitable for situations with large observation datasets.

2.1.3 Physically Inspired Neural Networks

A paper by Malfliet, 2023 introduces another method for reaching the same goals, by using a physically inspired neural network. This paper used the data gathering and preparation as proposed by Sun et al., 2018 for the MP model. The implementation included the use of an encoder/dense-blocks/decoder composition with a U-net structure and skip connections, where the cost function was used to incorporate physical processes such as continuity and vorticity in addition to the observational loss (difference between predicted and truth fields).

To train and validate this model, ERA5 reanalysis data (Bell et al., 2021) and GFS data (National Weather Service, 2015) were used. Noteworthy is that for training the data was split on entire aircraft trajectories, not randomly. An exception to this was with forecasting, where data was split with time. To assist the model where there are little or no aircraft measurements, the input was combined with GFS data.

The resulting model showed a better representation of the wind fields than the MP model (40% magnitude error decrease, 27% directional error decrease) or the GFS alone, although the contribution of physical losses remains unclear. The effect of adding vorticity to the cost function is inconclusive, and the effect of adding divergence is minimal. This is because these loss functions both smooth the flow fields, which can coincidentally improve or deteriorate the results.

To produce the model, Malfliet, 2023 was required to make assumptions such as that the flow was stable and non-turbulent and the scale of minutes and at an altitude of 10km. This means that the model is not valid during extreme weather events, in addition to affecting the model outputs in general converging/diverging wind conditions. Furthermore, since the model was in cartesian coordinates the curvature of the earth was not taken into account. These assumptions are necessary to make, but sketch the applicability of the model well.

2.2 Numerical Weather Prediction

The goal of this project is to improve upon the current system of predicting wind fields. To this end, an understanding of the current state-of-the-art in wind field prediction, and hence general weather prediction, algorithms is necessary. This section will summarize the traditional numerical weather prediction (NWP) approach, and describe data assimilation as well as the impact of ensemble forecasts and reanalysis datasets.

NWP methods use the Navier-Stokes equations, mass continuity (including earth rotation) with the first law of thermodynamics and the ideal gas law to predict, in space and time, the state of wind, pressure, density, and temperature in the atmosphere. Since the conception of the idea at the beginning of the 20th century, the forecast skill has continually increased significantly on all scales (Bauer et al., 2015) up to even 10 days in advance.

2.2.1 Data Assimilation and Ensemble Forecasts

Data assimilation, meaning the production of weather predictions for the current time based on earlier observations (Jeppesen, 2017), is an important topic to investigate. The strength of this technique lies in the use of data in a time frame as opposed to at a specific point in time (Guzzi, 2016). The source of data can be anything from weather stations, radiosondes, satellites, ground-based radar, or even aircraft and ships (Jeppesen, 2017). Aircraft data for assimilation will be the main topic in this subsection.

The biggest benefit of using aircraft data is that the observations are high-resolution (in space and time), making them ideal for nowcasting and short-range forecasting up to 2-3 hours (De Haan & Stoffelen, 2012). Ensemble weighted averages have been shown to produce good results (Kikuchi et al., 2018). Polynomial chaos expansion-based Gaussian process regression has also promising results (Marinescu et al., 2023). These methods perform very well near the flight paths of aircraft, which is where the research discussed in Section 2.1 and the current research comes in intending to fill in the areas where there are few flights.

An important technique used in NWP data assimilation is ensemble forecasting. With a particular set of observations, the uncertainty of a forecast can be determined by slightly perturbing the initial conditions for separate simulation runs, creating an ensemble of many forecasts. The set of this ensemble of forecasts represents the uncertainty in the initial conditions as well as the approximations in the model. With that, a forecast that indicates the chance of, for example, the temperature rising above a specific threshold, can be made as opposed to a single deterministic forecast (Jeppesen, 2020). As will be explored later in the literature review, this is a huge advantage of NWP over machine learning methods of forecasting weather conditions. The drawback, however, which machine learning techniques do solve, is the large amount of computational effort and time required to make the forecasts.

2.2.2 Reanalysis datasets derived from NWP

A byproduct of many years of gathering data for NWP and the improvement of NWP methods is the creation of large reanalysis datasets. A reanalysis in this context refers to the complete picture of past weather and climate (Jeppesen, 2023). It is the combination of observations with past short-range weather forecasts rerun with modern forecasting models and is made through data assimilation (Jeppesen, 2023). This is particularly useful for training data-driven machine learning models as will become clear in the next section.

Currently, the two most complete and high-resolution reanalysis datasets publically available are the European Centre for Medium-Range Forecasts's (ECMWF) ERA5 dataset (Hersbach et al., 2020) and the U.S. National Science Foundation's (NSF) Global Forecast System (GFS) (National Weather Service, 2015).

2.3 Machine Learning in Weather Forecasting

In recent years applying machine learning algorithms to the topic of weather forecasting has increased in popularity and effectiveness rapidly. Global weather forecasting has been a main topic of interest, with local high-resolution forecasting being less common but showing potential for improvements over traditional methods.

Graph neural networks (Lam et al., 2023, Keisler, 2022) and well as deep convolutional neural networks (Weyn et al., 2019) are common examples of successfully implemented machine learning architectures. These models are often much faster and have a lower computational cost (once trained), but show mixed results. With some models claiming to perform better than current operational NWP models in test cases (Bi et al., 2022, Lam et al., 2023) and others having potential but requiring further development (Weyn et al., 2019, Keisler, 2022).

These fully machine learning-based methods for forecasting still rely heavily on Integrated Forecasting Systems (IFS) for training and validation, as well as for initial conditions to base the forecast on (Maskell, 2023). This is not an issue, however, it should be made clear that ML models add to the current forecasting methods rather than replace them. Potential ways to further improve data-driven machine learning methods is to reduce the smoothing of predictions, allowing the forecasts to predict extremes (when the models are at their most valuable). This could be achieved by stepping away from training models to minimize RMSE, towards a generative approach (Maskell, 2023). Additionally, the way that ML approaches handle uncertainty is worse than NWP methods and is still in development (Lam et al., 2023).

As identified by researchers attempting fully data-driven as well as physics-aware neural network approaches, it has become clear that weather predictions can be made in a similar fashion to computer vision inpainting techniques (Schweri et al., 2021, Bi et al., 2022). This is especially the case for high-resolution local predictions such as in the scope of this project. Therefore, taking inspiration from other domains within machine learning, such as applying partial convolutions to image inpainting (Liu et al., 2018) could produce good results. Diffusion probabilistic models are known for generating image samples extremely well (Ho et al., 2020) or generating a video from an image (Wang et al., 2024), however, they have also shown promise for image inpainting (Lugmayr et al., 2022, Saharia et al., 2022, Corneanu et al., 2024). This machine-learning architecture will be the basis of the coming research, applied to wind fields instead of images of daily objects and scenes.

3 Research Question(s)

Research Objective

As a consequence of the background research the research objective is decided to be:

To contribute to the development of accurate wind field reconstruction by investigating the applications of a diffusion probabilistic model to partially available wind fields.

This objective can be divided into three secondary objectives. The first is to gain a comprehensive understanding of the diffusion probabilistic model and tools that can be used to achieve the desired goals by applying them to simplified versions of the problem. The second is to extensively evaluate the model by applying it first in 2D, then in 3D, and exploring short-term forecasting capabilities by extending the model to 4D. The last objective is to quantify the applicability of the model in the aviation industry by testing the output on Air Traffic Flow Management (ATFM) decision support tooling.

Scope

To accomplish the objectives of this project, a scope has been defined as a foundation for the research to be conducted. The focus will lie on using aircraft surveillance data as a source, and on constructing wind fields at altitudes where there are aircraft. This means at cruise altitudes for en route locations and lower altitudes near airports.

Research Questions

Following the research objectives and the scope, the research questions can be set. These questions intend to fulfill the objective within the given scope. The goal is to set a structure for the coming research. These research questions are subject to change during the project, if necessary.

Research Question 1

How can a diffusion probabilistic model be utilized to create accurate and useful wind field estimations based on aircraft surveillance data?

Sub-questions

1. How does the output of the model in 2D compare to current methods for wind field reconstruction?
2. What are the implications of extending the model to 3D and how do these affect the accuracy of the results?
3. Can the model be applied to short-term forecasting? If so, how does it compare to current models?

Research Question 2

How does the accuracy of ATFM decision support tooling change when used with the output of the model?

4 Methods, tools and expected results

In the process of answering the aforementioned research questions, a brief literature review will be completed. Subsequently, a number of methods will be used, which are introduced in this section.

Preparation for model training
When making a model designed to construct a wind field when only a part of the wind field is known, it is clear that there is no way to validate the accuracy of the predictions without the complete wind field. Hence, the partial wind field needs to be simulated on a dataset that is complete. The chosen data is the ECMWF global reanalysis data. This is a suitable option because it has data for every hour since 1940 and is easily accessible online in `.grib` format.

To this end, firstly a target region for which nowcasting is desired needs to be selected and the most flown aircraft routes, commonly known as "highways in the sky", within this region distinguished. Following this, complete weather reanalysis data needs to be obtained and split. For each input, there will be a masked version, only containing wind field data at locations where aircraft fly, and a complete wind field to compare the output against. The entire dataset will be split, with approximately 80% for training, 10% for validation, and the last 10% for testing.

Development of diffusion model
Based on an investigation into the diffusion model architecture, the model needs to be developed and adapted to the current application in 2D and then in 3D. The model will be written in Python, as is common practice with ML research. The PyTorch library will utilized for its superior flexibility for academic research. No special hardware is strictly necessary to train and run the model, given the modest size of the problem.

The model will be validated and tested, with a cyclic process of hyperparameter tuning applied for the best results that the model architecture can provide. It can then be directly compared to other wind field reconstruction models currently found in the literature. This will be done on the same datasets used to evaluate the current models (Sun et al., 2018, Malfliet, 2023). It is expected that the diffusion model will show a lower error magnitude and variance in wind speeds and behave better in converging and diverging wind conditions. This can be displayed with boxplots, for example.

Extension to short-term forecasting
The model will be extended into 4D, with the fourth dimension being time. Multiple methods will be explored, as it is not possible to know beforehand which will perform best. Firstly, the model will be trained in the same way as previously but trained to a reanalysis dataset in the future compared to the masked input. Furthermore, as can be done with images that are converted to videos (Wang et al., 2024), the nowcasted output may be able to be extended to forecasts.

There are currently no short-term forecasting methods that can easily be directly compared to the implementation of the diffusion model. As such, the model will be compared to itself in 2D. It is expected that the model will perform better in high vertical wind conditions, but suffer in most other conditions due to the significantly increased complexity of the model.

Testing on operational infrastructure
With the trained, tested, and validated Python diffusion model, the outputs can be used to perform testing with ATFM decision support tooling. The LVNL Decision Support Tool (DST) will use the outputs of the model as an input and the Estimated Time of Arrival (ETOA) of aircraft can be measured. The accuracy and variance of the output of the DST can be compared to output when using the current operational input, provided by The Royal Netherlands Meteorological Institute (KNMI).

5 Planning

In this section, the project's planning will be elaborated upon. The planning will be detailed from the time of submitting this proposal up until the mid-term (27=-01-2025). The planning after the mid-term will only be briefly discussed and will be made at a later stage when well-informed planning can be made. First, a description of all of the tasks that need to be completed will be given and their duration will be estimated. Next a diagrammatic representation of the workflow will be provided.

To ensure that the goals of the project are met within the set amount of time, the approach will be to make the planning in a bottom-up fashion; starting with the outputs and working backward as required.

The end goal is to have a model in three forms. Namely, in 2D, 3D, and 4D. They all build off each other and hence will also be made in that order. Secondly, the model needs to be tested on operational ATFM decision support tooling. Additionally, the results need to be gathered, discussed, and presented in a scientific paper-like format. Before any of this can happen, the author also needs a deep understanding of both the tools to complete these tasks as well as the theory of the model.

With the above in mind, the following steps need to be completed before the mid-term:

- Learn how to utilize the Python PyTorch library correctly (1 week).
- Learn the details of the diffusion probabilistic model network architecture (1 week).
- Gather input data for the model (instant, however, the chance of issues is high, so hence process must be started immediately).
- Understand how to design the output of the model to be suitable for ATFM decision support tooling by talking to KDC stakeholders (continuous process, 1/2 week spread over a longer period).
- Construct, train, and optimize hyperparameters of the model in 2D (3 weeks).
- Prepare a mid-term report and presentation (1 1/2 weeks).

With the goal to achieve the above before the mid-term, the following remains to be completed before the end of the research:

- Further 2D model hyperparameter tuning post-mid-term evaluation (1 week).
- Extension of model to 3D + hyperparameter tuning (4 weeks).
- Extension of model to 4D + hyperparameter tuning (4 weeks).
- Test model with ATFM decision support tooling (1 week).
- Iterate the model to yield better results with operational tooling (1 week).
- Write the scientific report describing all methods, results, and conclusions (2 weeks).
- Prepare final presentation (2 weeks).

These are the steps foreseen by the author in this project and have been visualized in the form of a Gantt chart in Appendix A. Lastly, some additional remarks on the planning have been taken into consideration when constructing the Gannt chart. All processes before the initial construction of the model are independent of each other and can in principle be done in parallel. Construction of the model, training, and hyperparameter tuning have been combined into blocks as the process is iterative, and they are all dependent on each other. Lastly, holidays total 4 planned weeks at this stage. Space for unforeseen circumstances and delays has been accounted for between the mid-term and the green-light meeting (currently blank in the chart).

6 Conclusions

The aim of this research proposal was to introduce the project's area of interest, define the advancements, challenges, and opportunities, provide an overview of the relevant literature, and develop clear research questions. Additionally, a plan for the research activities has been provided. This report lays the foundations for the research to be completed in the coming academic year.

A literature study has been performed to gain valuable insights into the current state of the research field. Firstly, previous work with similar objectives from Sun et al., 2018 and Malfliet, 2023 was discussed. This was followed by an analysis of the past and current state of numerical weather prediction methods. To conclude the literature study, machine learning methods in weather prediction were analyzed and the link to other computer vision machine learning methods was made.

Given the high demand for accurate wind now- and forecasting by airlines and ANSPs to be able to safely and efficiently conduct day-to-day operations and the rise of various machine learning network architectures, the research proposed is as follows: develop a diffusion probabilistic model for accurate wind fiend reconstruction in 2D, 3D and 4D with aircraft derived wind data as input. The goal is to compare this new application of machine learning models to previous work and to observe the effects on current operational ATFM decision support tooling.

The aforementioned goals are intended to be achieved by first thoroughly understanding the tools that can be used to make the model as well as understanding the model architecture itself. This will be followed by the implementation of the model in Python and the extension from 2D to 3D. The methods for expanding the model to 4D will require various approaches to find which is the most suitable for this new application.

Lastly, detailed planning of the first research phase of the project (until 27/01/2024) was presented and rough planning of the research activities of the second research phase was outlined. The current planning suggests that the 2D model be complete by the end of the first research phase.

References

Bauer, P., Thorpe, A., & Brunet, G. (2015). The quiet revolution of numerical weather prediction. *Nature*, *525*, 47–55. https://doi.org/10.1038/nature14956

Bell, B., Hersbach, H., Simmons, A., Berrisford, P., Dahlgren, P., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Radu, R., Schepers, D., Soci, C., Villaume, S., Bidlot, J.-R., Haimberger, L., Woollen, J., Buontempo, C., & Thépaut, J.-N. (2021). The ERA5 global reanalysis: Preliminary extension to 1950 [Publisher: Wiley]. *Quarterly Journal of the Royal Meteorological Society*, *147*(741), 4186–4227. https://doi.org/10.1002/qj.4174

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., & Tian, Q. (2022, November). Pangu-Weather: A 3D High-Resolution Model for Fast and Accurate Global Weather Forecast [arXiv:2211.02556 [physics]]. https://doi.org/10.48550/arXiv.2211.02556

Corneanu, C., Gadde, R., & Martinez, A. M. (2024). LatentPaint: Image Inpainting in Latent Space with Diffusion Models. *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 4322–4331. https://doi.org/10.1109/WACV57701.2024.00428

De Haan, S. (2011). High-resolution wind and temperature observations from aircraft tracked by Mode-S air traffic control radar. *Journal of Geophysical Research Atmospheres*, *116*(10). https://doi.org/10.1029/2010JD015264

De Haan, S., & Stoffelen, A. (2012). Assimilation of high-resolution mode-S wind and temperature observations in a regional NWP model for nowcasting applications. *Weather and Forecasting*, *27*(4), 918–937. https://doi.org/10.1175/WAF-D-11-00088.1

De Jong, P., Van Der Laan, J., In 't Veld, A., Van Paassen, M., & Mulder, M. (2014). Wind-profile estimation using airborne sensors. *Journal of Aircraft*, *51*(6), 1852–1863. https://doi.org/10.2514/1.C032550

Delahaye, D., & Puechmorel, S. (2009). TAS and wind estimation from radar data. *2009 IEEE/AIAA 28th Digital Avionics Systems Conference*, 2.B.5–1–2.B.5–16. https://doi.org/10.1109/DASC.2009.5347547

Delahaye, Puechmorel, & Vacher. (2003). Windfield estimation by radar track Kalman filtering and vector spline extrapolation. *22nd Digital Avionics Systems Conference Proceedings (Cat No 03CH37449) DASC-03*, 5.E.2–51. https://doi.org/10.1109/DASC.2003.1245869

de Leege, A., Mulder, M., & Van Paassen, M. M. (2013). Automatic Dependent Surveillance-Broadcast for Meteorological Monitoring. *Journal of Aircraft*, *50*, 249–261.

Guzzi, R. (2016). *Data Assimilation: Mathematical Concepts and Instructive Examples*. Springer International Publishing. https://doi.org/10.1007/978-3-319-22410-7

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., . . . Thépaut, J.-N. (2020). The ERA5 global reanalysis [Publisher: Wiley]. *Quarterly Journal of the Royal Meteorological Society*, *146*(730), 1999–2049. https://doi.org/10.1002/qj.3803

Ho, J., Jain, A., & Abbeel, P. (2020, December). Denoising Diffusion Probabilistic Models [arXiv:2006.11239 [cs, stat]]. https://doi.org/10.48550/arXiv.2006.11239

Hollister, W. M., Bradford, E. R., & Welch, J. D. (1989). Using aircraft radar tracks to estimate winds aloft.

Jeppesen, J. (2017, March). Fact sheet: Ensemble weather forecasting. Retrieved September 17, 2024, from https://www.ecmwf.int/en/about/media-centre/focus/2017/fact-sheet-ensemble-weather-forecasting

Jeppesen, J. (2020, March). Fact sheet: Earth system data assimilation. Retrieved September 17, 2024, from https://www.ecmwf.int/en/about/media-centre/focus/2020/fact-sheet-earth-system-data-assimilation

Jeppesen, J. (2023, August). Fact sheet: Reanalysis. Retrieved September 17, 2024, from https://www.ecmwf.int/en/about/media-centre/focus/2023/fact-sheet-reanalysis

Keisler, R. (2022, February). Forecasting Global Weather with Graph Neural Networks [arXiv:2202.07575 [physics]]. https://doi.org/10.48550/arXiv.2202.07575

Kikuchi, R., Misaka, T., Obayashi, S., Inokuchi, H., Oikawa, H., & Misumi, A. (2018). Nowcasting algorithm for wind fields using ensemble forecasting and aircraft flight data [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/met.1704]. *Meteorological Applications*, *25*(3), 365–375. https://doi.org/10.1002/met.1704

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., Ravuri, S., Ewalds, T., Eaton-Rosen, Z., Hu, W., Merose, A., Hoyer, S., Holland, G., Vinyals, O., Stott, J., Pritzel, A., Mohamed, S., & Battaglia, P. (2023). Learning skillful medium-range global weather forecasting [Publisher: American Association for the Advancement of Science]. *Science*. https://doi.org/10.1126/science.adi2336

Liu, G., Reda, F. A., Shih, K. J., Wang, T.-C., Tao, A., & Catanzaro, B. (2018, December). Image Inpainting for Irregular Holes Using Partial Convolutions [arXiv:1804.07723 [cs]]. https://doi.org/10.48550/arXiv.1804.07723

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., & Gool, L. V. (2022, August). RePaint: Inpainting using Denoising Diffusion Probabilistic Models [arXiv:2201.09865]. https://doi.org/10.48550/arXiv.2201.09865

Malfliet, J. M. L. (2023, January). *Estimating Wind Fields Using Physically Inspired Neural Networks With Aircraft Surveillance Data* [Doctoral dissertation, Delft University of Technology].

Marinescu, M., Olivares, A., Staffetti, E., & Sun, J. (2023). Polynomial Chaos Expansion-Based Enhanced Gaussian Process Regression for Wind Velocity Field Estimation from Aircraft-Derived Data. *Mathematics*, *11*(4). https://doi.org/10.3390/math11041018

Marinescu, M., Olivares, A., Staffetti, E., & Sun, J. (2022). On the Estimation of Vector Wind Profiles Using Aircraft-Derived Data and Gaussian Process Regression. *Aerospace*, *9*(7), 377. https://doi.org/10.3390/aerospace9070377

Martin Leutbecher, T. N. P. (2007). Ensemble forecasting. Retrieved September 17, 2024, from https://www.ecmwf.int/en/elibrary/75394-ensemble-forecasting

Maskell, K. (2023, June). The rise of machine learning in weather forecasting. Retrieved September 19, 2024, from https://www.ecmwf.int/en/about/media-centre/science-blog/2023/rise-machine-learning-weather-forecasting

National Weather Service. (2015, January). NCEP GFS 0.25 Degree Global Forecast Grids Historical Archive. https://doi.org/10.5065/D65D8PWK

Saharia, C., Chan, W., Chang, H., Lee, C. A., Ho, J., Salimans, T., Fleet, D. J., & Norouzi, M. (2022, May). Palette: Image-to-Image Diffusion Models [arXiv:2111.05826 [cs]]. https://doi.org/10.48550/arXiv.2111.05826

Schweri, L., Foucher, S., Tang, J., Azevedo, V. C., Günther, T., & Solenthaler, B. (2021). A Physics-Aware Neural Network Approach for Flow Data Reconstruction From Satellite Observations [Publisher: Frontiers]. *Frontiers in Climate*, *3*. https://doi.org/10.3389/fclim.2021.656505

Steiner, M., Bateman, R., Megenhardt, D., Liu, Y., Xu, M., Pocernich, M., & Krozel, J. (2010). Translation of Ensemble Weather Forecasts into Probabilistic Air Traffic Capacity Impact. *Air Traffic Control Quarterly*, *18*, 229–254. https://doi.org/10.2514/atcq.18.3.229

Sun, J., Vu, H., Ellerbroek, J., & Hoekstra, J. M. (2020). pyModeS: Decoding Mode-S Surveillance Data for Open Air Transportation Research. *IEEE Transactions on Intelligent Transportation Systems*, *21*(7), 2777–2786. https://doi.org/10.1109/TITS.2019.2914770

Sun, J., Vû, H., Ellerbroek, J., & Hoekstra, J. M. (2018). Weather field reconstruction using aircraft surveillance data and a novel meteo-particle model [Publisher: Public Library of Science]. *PLOS ONE*, *13*(10), e0205029. https://doi.org/10.1371/journal.pone.0205029

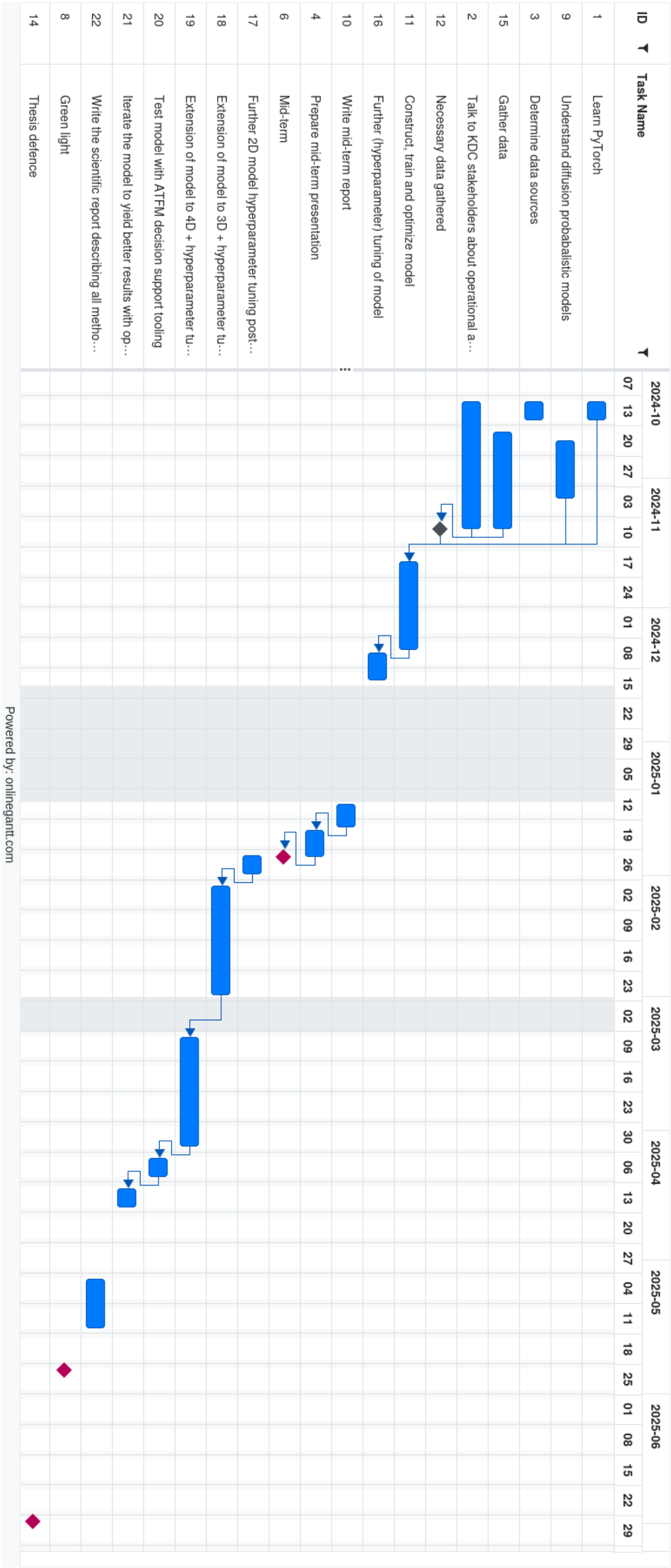
Wang, C., Gu, J., Hu, P., Xu, S., Xu, H., & Liang, X. (2024, September). DreamVideo: High-Fidelity Image-to-Video Generation with Image Retention and Text Guidance [arXiv:2312.03018]. https://doi.org/10.48550/arXiv.2312.03018

Weyn, J. A., Durran, D. R., & Caruana, R. (2019). Can Machines Learn to Predict Weather? Using Deep Learning to Predict Gridded 500-hPa Geopotential Height From Historical Weather Data [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2019MS001705]. *Journal of Advances in Modeling Earth Systems*, *11*(8), 2680–2693. https://doi.org/10.1029/2019MS001705

World Meteorological Organization. (2023, May). Economic costs of weather-related disasters soars but early warnings save lives. Retrieved September 27, 2024, from https://wmo.int/news/media-centre/economic-costs-of-weather-related-disasters-soars-early-warnings-save-lives

Zhu, J., Wang, H., Li, J., & Xu, Z. (2021). Research and optimization of meteo-particle model for wind retrieval. *Atmosphere*, *12*(9). https://doi.org/10.3390/atmos12091114

A Gantt Chart



ID	Task Name
1	Learn PyTorch
9	Understand diffusion probabalistic models
3	Determine data sources
15	Gather data
2	Talk to KDC stakeholders about operational a...
12	Necessary data gathered
11	Construct, train and optimize model
16	Further (hyperparameter) tuning of model
10	Write mid-term report
4	Prepare mid-term presentation
6	Mid-term
17	Further 2D model hyperparameter tuning post...
18	Extension of model to 3D + hyperparameter tu...
19	Extension of model to 4D + hyperparameter tu...
20	Test model with ATFM decision support tooling
21	Iterate the model to yield better results with op...
22	Write the scientific report describing all metho...
8	Green light
14	Thesis defence