Unpaired Downscaling of Fluid Flows with Diffusion Bridges

TOBIAS BISCHOFF[a] AND KATHERINE DECK[a]

[a] *Climate Modeling Alliance, California Institute of Technology, Pasadena, California*

(Manuscript received 3 May 2023, in final form 4 December 2023, accepted 11 March 2024)

ABSTRACT: We present a method to downscale idealized geophysical fluid simulations using generative models based on diffusion maps. By analyzing the Fourier spectra of fields drawn from different data distributions, we show how a diffusion bridge can be used as a transformation between a low-resolution and a high-resolution dataset, allowing for new sample generation of high-resolution fields given specific low-resolution features. The ability to generate new samples allows for the computation of any statistic of interest, without any additional calibration or training. Our unsupervised setup is also designed to downscale fields without access to paired training data; this flexibility allows for the combination of multiple source and target domains without additional training. We demonstrate that the method enhances resolution and corrects context-dependent biases in geophysical fluid simulations, including in extreme events. We anticipate that the same method can be used to downscale the output of climate simulations, including temperature and precipitation fields, without needing to train a new model for each application and providing a significant computational cost savings.

SIGNIFICANCE STATEMENT: The purpose of this study is to apply recent advances in generative machine learning technologies to obtain higher-resolution geophysical fluid dynamics model output at lower cost compared with direct simulation while preserving important statistical properties of the high-resolution data. This is important because while high-resolution climate model output is required by many applications, it is also computationally expensive to obtain.

KEYWORDS: Downscaling; Neural networks; Statistical techniques

1. Introduction

Climate simulations are powerful tools for predicting and analyzing climate change scenarios, but they are often limited by computational resources and hence in their spatial and temporal resolution. As a result, simulations can both lack the high-resolution detail needed for many applications as well as carry an inherent bias due to the lack of representation of small-scale dynamical processes which feed back on larger scales. For example, horizontal resolutions of $O(10–100)$ km are still too coarse to accurately simulate important phenomena such as convective precipitation, tropical cyclone dynamics, and local effects from topography and land cover, and hence, these simulations can be of limited use for making predictions on regional and subregional scales. In particular, the low-resolution fields suffer from biases in extreme temperatures and precipitation rates, which in turn can reduce the accuracy of projections of climate hazards on smaller spatial scales, e.g., Abatzoglou and Brown (2012), Gutmann et al. (2014), and Hwang and Graham (2014).

Several approaches have been developed to address biases and increase resolution in fluid and climate simulations, a process referred to as "downscaling" of fluid flows (Fowler et al. 2007; Salathé et al. 2007; Maurer and Hidalgo 2008). Nudging techniques, which involve constraining the solution of a dynamical system to follow the large-scale information, are applied during simulation time and are a form of dynamical downscaling. On the other hand, statistical downscaling refers to methods which use a data-derived model to make a correction to a fluid simulation or to computed statistic after the simulation has been run, which can keep the computational budget comparatively lower. This is usually achieved by amortizing the training cost of these models over many evaluations during postprocessing, as opposed to solving the highly resolved fluid system whenever the output is needed (but not requiring any training time).

The predominant statistical method for bias correction and resolution enhancement of climate variables is the bias-correction spatial disaggregation (BCSD) algorithm (Panofsky et al. 1958; Wood et al. 2002, 2004; Thrasher et al. 2012). BCSD uses quantile mapping to correct biases and Fourier transforms to enhance resolution. However, the BCSD algorithm has limitations, including its inability to incorporate auxiliary datasets and its lack of multivariate capability. Quantile mapping can also adversely affect large-scale features such as the evolution of mean values (Hagemann et al. 2011; Pierce et al. 2013; Maurer and Pierce 2014; Ballard and Erinjippurath 2022). Constructed analogs and variants thereof are a multivariate method for downscaling which have been shown to outperform other methods, perhaps due to the fact that they take into account correlations between variables (Pierce et al. 2014; Abatzoglou and Brown 2012). However, these methods do not truly allow for sampling high-resolution data from a distribution.

At the same time, generative machine learning models like generative adversarial networks (GANs; e.g., Goodfellow et al. 2020),

Tobias Bischoff and Katherine Deck contributed equally to this work.

Corresponding author: Tobias Bischoff, tobias@caltech.edu

DOI: 10.1175/AIES-D-23-0039.1 e230039

variational autoencoders (VAEs; e.g., Kingma and Welling 2019), and normalizing flows (NFs; e.g., Papamakarios et al. 2021) have been demonstrated to be effective for superresolution and for domain translation (e.g., Wang et al. 2021; Zhu et al. 2017) and are now being applied to the downscaling task. In these use cases, the models are deep convolutional neural networks which map fields with multiple channels (e.g., fields of climate model variables at low resolution) into output fields with multiple channels (e.g., fields of climate model variables at high resolution). Statistical relationships between datasets are learned implicitly in supervised input–output or "generative" fashion. In supervised approaches, data points from the input and output spaces are aligned. When paired data points are unavailable or when the map between spaces is not injective, the domain translation task must become one of the modeling conditional data distributions and then generate samples from these distributions.

In this work, we propose to use generative models based on diffusion maps for generating downscaled fluid flows using unpaired training data. Diffusion models have shown great flexibility in generating realistic samples from a variety of learned high-dimensional probability distributions (e.g., fields, audio, and video; Dhariwal and Nichol 2021; Kong et al. 2021; Ho et al. 2022a,b). With respect to domain translation and downscaling, i.e., transforming a sample from a source distribution into a sample from a target distribution, generative diffusion models have distinct advantages over classical methods:

- Generative models allow for sampling from high-dimensional probability distributions. From these samples, any statistical quantity can be computed.
- Diffusion-based models can be trained with unpaired data and can therefore be used for multiple domain translation tasks without retraining for each source domain/target domain pair (Su et al. 2023).
- Pretrained diffusion-based models can be "repurposed" to sample from specific parts of the domain using guided sampling (e.g., Ho and Salimans 2022; Dhariwal and Nichol 2021).

These points put diffusion-based models into a class distinct from classical methods and in some cases distinct even from GAN-based methods, many of which require paired data or paired source/target domains. This suggests diffusion-based models as a promising candidate well suited for applications in fluid dynamics and climate science because

- retraining machine learning models frequently is undesirable due to the potentially high training cost involving high-dimensional data points (e.g., full climate fields);
- for downscaling tasks, paired datasets of high- and low-resolution climate *simulations* do not truly exist, due to deterministic chaos and the feedback of small-scale motion to large scales;
- extreme events with biased tail probabilities can be correlated across climate variables and spatial locations, and calibrating a downscaling method for all statistics of interest is challenging. As a result, the ability to generate samples can be highly desirable.

In this work, we provide a demonstration of how diffusion models can be used for domain translation between low- and high-resolution fluid simulations, without customization to the specific translation task under consideration. We focus on the generation of consistent high-resolution information given a low-resolution input and on the correction of important statistical biases, e.g., shifts in mean values, unresolved spatial scales, and underestimated tail events.

a. *Related work*

1) DOWNSCALING OF CLIMATE DATA

Machine learning–based methods for downscaling and bias correction have been applied to climate simulations successfully in prior works. Pan et al. (2021) use GANs to bias correct climate simulation data over the continental United States and focus on matching various statistical quantities of corresponding observational data. Using paired high-resolution radar measurements, ECMWF simulation data, and other contextual information, Harris et al. (2022) use GANs to downscale simulated low-resolution precipitation fields. They found that their model outperformed many conventional approaches, including on extreme rainfall events. Similarly, Price and Rasp (2022) show that conditional GANs can be used to directly bias correct and downscale low-resolution precipitation forecasts using high-resolution ground truth radar observations. Ballard and Erinjippurath (2022) use contrastic translation GANs and high-resolution observations to downscale CMIP low-resolution simulation data for daily maximum temperature and precipitation. This particular variant of the GAN model allows for training in an unsupervised, unpaired fashion (Park et al. 2020). Again, the authors find comparable or improved performance compared to existing methods. Similarly, Groenke et al. (2021) use unpaired datasets to learn a domain translation map from low-resolution simulation data to high-resolution, unbiased data by combining normalizing flows with a cycle consistency loss function similar to that of cycle GANs (CycleGANs) (Zhu et al. 2017). Methods based on CycleGANs generally require that a model is trained with access to both the source and the target data, so that a new model must be created for each translation task.

2) DIFFUSION MODELING

Diffusion is a dynamical process which erases initial conditions on long time scales. Using observed data (which are samples from an unknown data distribution) as initial conditions, we can integrate a trajectory forward in time under a diffusion model chosen so that as $t \to \infty$, the long-time steady state of the system corresponds to samples from a known distribution like a Gaussian (the prior distribution). Using samples from the prior distribution as initial conditions, solving the reverse-diffusion model will generate samples from the unconditional data distribution. Those diffusive processes can then be used as generative models, transforming samples from a known prior distribution into samples from an unknown data distribution via a diffusive process, which has been established by several authors, e.g., Sohl-Dickstein et al. (2015), Song and Ermon (2019), and Ho et al. (2020). Indeed,

more recently, Song et al. (2021b) showed that images created by generative diffusion models can be understood to be numerical solutions to "reverse diffusion" stochastic differential equations, with initial conditions equal to samples from the prior distribution. This relies on the fact that (forward) diffusion processes can be reversed if the score, related to the gradient of the data distribution, is known (Anderson 1982). Moreover, while the unconditional data distribution can be challenging to approximate directly, Hyvärinen and Dayan (2005) and Vincent (2011) have demonstrated how to approximate the score of the distribution using neural networks and gradient descent of a tractable loss function.

The mathematical results for unconditional distributions, described above, can be extended to conditional distributions, allowing for conditional sampling (Song et al. 2021b; Batzolis et al. 2021). Many conditional diffusion models require paired input data points, and many important conditional generation tasks provide this type of data [e.g., superresolution, inpainting, colorization, and other imputation tasks, including with temporal sequences—Tashiro et al. (2021), Saharia et al. (2023), Giannone et al. (2022), Ho et al. (2022a), and Saharia et al. (2022)]. Alternatively, Meng et al. (2022) show how to generate photo-realistic images from simple stroke paintings with little detail by choosing an appropriate starting point (initial condition) and starting time for a reverse-diffusion trajectory. As we will discuss, this can be interpreted as generating a high-resolution image conditional on the input stroke painting. Crucially, it is carried out using a model trained only on the high-resolution data (target domain data), without access to the stroke paintings (source domain data).

In contrast to CycleGANs (Zhu et al. 2017), Su et al. (2023) show how diffusion models can be used for domain translation such that a model is trained once per data domain and not once per translation of interest [StarGANS are another solution but at the expense of increased complexity Choi et al. (2018)]. This feature of diffusion models arises because diffusion models for two domains have easily relatable prior distributions and is advantageous because it allows for the same model to be used in many translation tasks. The translation works by completely diffusing an image from one domain, then turning that final state into a sample from the other domain's prior, and carrying out the reverse diffusion for the other domain using its model. The approach we will showcase in this work is based on combination of the ideas of Meng et al. (2022) and Su et al. (2023), where a chain of diffusion models acts as a bridge between data domains, although the general idea has been around for much longer (Chetrite et al. 2021).

b. *Our contribution*

Our long-term goal is to develop a flexible and performant method for unsupervised downscaling of fluid simulations which can be applied to climate simulations. As a first step toward this, here we develop and validate our method with a simpler dynamical system that exhibits sufficiently complex attributes: forced two-dimensional turbulence with non-Gaussian statistics and contextual information.

The method presented is based on chaining together diffusion-based generative models. It relies on the fact that coarsely resolved and highly resolved climate simulations differ on small and intermediate spatial scales but mostly agree on the largest scales. Because the diffusion processes we employ here erase information on the smallest scales first, we can start with samples from a source domain (low resolution), diffuse them until small-scale information is lost, and then reverse diffuse them using a pretrained diffusion model for the target domain (high resolution). How well the resulting fields match the source field on large scales while simultaneously containing fine-scale features which match statistics of the high-resolution data is governed by the time at which we stop the forward noising process and begin the reverse-diffusion process. Following Meng et al. (2022) and Su et al. (2023), we will refer to such a source-to-target diffusion model as a diffusion bridge. As described already, this approach has advantages over existing downscaling methods as it allows for sample generation, use with unpaired data, and the reusability of trained models, but it has not been tested yet for this application.

Additionally, we introduce architectural improvements for the neural network employed in the diffusion model. These improvements are secondary to our overall goal but improve performance metrics and decrease training time. Score-based diffusion models are known to suffer from a "color shift": generated fields may have the correct spatial features but are shifted to different average colors relative to the training data. The error grows for larger fields. One approach for improving this artifact is to use an exponential moving average (EMA) of the model parameters, typically using a very long memory implied by the exponential moving average (Song and Ermon 2020). As a consequence, a large number of training iterations are required to reach good performance. While alleviation of the color shift via other techniques is possible, e.g., Choi et al. (2022), we reduce the effect by introducing a bypass layer in the neural network architecture. This removes the need for the exponential moving average even for large field sizes (e.g., 512×512 pixels in spatial resolution).

2. Data and simulations

We use a two-dimensional advection–condensation model similar to the one proposed in O'Gorman and Schneider (2006) to create the fluid simulation data used in this work. The model is intended to provide an approximate representation of the dynamics of moisture on isentropes in the extratropical atmosphere on Earth-like planets (O'Gorman and Schneider 2006). As such, it is an idealized toy model. Nevertheless, the model allows for detailed investigations of spectral and distributional properties of its vorticity and moisture fields at low computation cost compared to a full climate model. Throughout this study, we focus on two quantities with nearly isotropic statistics, vorticity and an advected tracer that represents the supersaturation $q' = q - q_s$ in an Earth-like atmosphere, where q is the mass fraction of water relative to moist air (specific humidity) and q_s is the mass fraction when the air is saturated (saturation specific humidity).

The vorticity evolves according to the two-dimensional Euler equations with random forcing and linear dissipation. The supersaturation is advected by the flow field implied by the vorticity field. It is forced by a spatially homogeneous source e that can be interpreted as an evaporation field, adding moisture to the flow, as well as a spatially varying condensation which decreases moisture in situations of supersaturation $q > q_s$ ($q' > 0$). Condensation therefore represents the tail of the supersaturation tracer distribution, and extreme condensation events are correspondingly even further into the tail. For more mathematical details on the idealized advection–condensation model, see appendix A.

To mimic the meridional decay of the saturation specific humidity q_s along isentropes in Earth's atmosphere, we assume a linearly decaying profile that is modulated by a spatially periodic perturbation. The spatially periodic perturbation is useful because it can be used to impose spatial inhomogeneities in supersaturation tracer statistics at different length scales. Loosely speaking, these inhomogeneities can be interpreted as a very idealized version of orographic impact on the saturation specific humidity fields.

The time-independent form of q_s is given by the following expression:

$$q_s(x, y) = \gamma y + A \sin\left(\frac{2\pi k_x x}{L}\right)\sin\left(\frac{2\pi k_y y}{L}\right), \tag{1}$$

where γ denotes a background saturation specific humidity gradient, A is the modulation amplitude, L is the domain size in the x and y direction, and $k_x = k_y$ denotes wavenumbers that take values $k_{x,y} \in \{1, 2, 4, 8, 16\}$ that allow for different large-scale saturation specific humidity profiles. As such, we can generate a dataset of supersaturation tracer fields with different idealized orographic or supersaturation tracer forcings. Our goal is to understand how well the diffusion model can make use of *contextual information* when downscaling, as high frequency variations in topography, surface coverage, and other fields affect the atmospheric flow in a more realistic climate simulation. This site-specific information is often included in generative models, including the ones described in section 1. Training the model with context will in principle lead to better performance and to better generalization. It will be crucial in the climate simulation case as downscaling will be carried out in patches (smaller regions, as opposed to downscaling the entire globe at once), and because we may not have high-resolution data everywhere on the globe, and hence, we must rely on contextual information to generalize.

To generate data for training the denoising diffusion model, we performed a series of simulations at different resolutions and with different saturation specific humidity profiles, varying only the wavenumbers k_x, k_y of the saturation specific humidity modulation. We generated a set of high-resolution dataset (512 × 512 pixels) with varying background saturation specific humidity profiles and a low-resolution dataset (64 × 64 pixels) with a fixed and unmodulated background saturation specific humidity field, i.e., $A = 0$. The parameters used in the simulations are given in Table A1. Snapshots of these simulations once a steady state was reached were saved and used as training data for the diffusion model. We trained the diffusion model on the entire high-resolution dataset, including all context wavenumbers. Examples from the two-dimensional fluid dynamics models are shown in Fig. 1. The top row shows high-resolution snapshots, while the bottom row shows low-resolution snapshots. The left column shows the supersaturation tracer field, and the right column shows the vorticity field, respectively. Finally, we resized the low-resolution 64 × 64 fields using nearest-neighbor weighting to a resolution of 512 × 512 and removed high-frequency aliases by applying a low-pass filter. We ensured that the spectral information did not change between the true 64 × 64 fields and our resized ones.

3. Downscaling with diffusion bridges

a. *Diffusion models*

Our implementation of score-based generative models follows that of Song et al. (2021b). The forward diffusion ("noising") process involves adding independent samples of Gaussian noise to each pixel, where the added noise has a mean of zero and a variance that depends on time in a prescribed fashion. Concretely, given an initial condition $\mathbf{x}(t = 0) \sim p_{\text{data}}(\mathbf{x})$ drawn from the data distribution, the noising process is defined by the stochastic differential equation (SDE):

$$d\mathbf{x} = g(t)d\mathbf{W}, \tag{2}$$

where $g(t)$ is a nonnegative prescribed function of time and $d\mathbf{W}$ implies a Wiener process. At any time t, the solution to this SDE is the "noised" field $\mathbf{x}(t)$, which is drawn from a normal distribution

$$\mathbf{x}(t) \sim \mathcal{N}\big(\mathbf{x}(0), \sigma^2(t)\big) = p(\mathbf{x}(t)|\mathbf{x}(0)), \tag{3}$$

where $\sigma(t)^2$ is the variance defined by

$$\sigma^2(t) = \int_0^t g^2(t')dt'. \tag{4}$$

Here, we have chosen $g(t)$ such that at $t = 1$, the variance $\sigma^2(t = 1)$ is much larger in magnitude than the original pixel values, and hence, all memory of initial conditions is lost, i.e.,

$$p(\mathbf{x}(1)|\mathbf{x}(0)) \approx \mathcal{N}\big(0, \sigma^2(1)\big). \tag{5}$$

In this view, diffusion is a process which embeds a source field into a latent space, such that samples in the latent space $\mathbf{x}(1)$ are drawn approximately from a known distribution, which is independent of the source data.

To approximately sample from the data distribution, we reverse this process. First, we sample from the latent-space prior distribution to obtain $\mathbf{x}(1)$. This is the initial condition for the reverse-diffusion equation, which is solved from $t = 1$ to $t = 0$. The equation which reverses Eq. (2) is given by Anderson (1982) as

$$d\mathbf{x} = -g(t)^2\mathbf{s}(\mathbf{x}, t)dt + g(t)d\mathbf{W}, \tag{6}$$

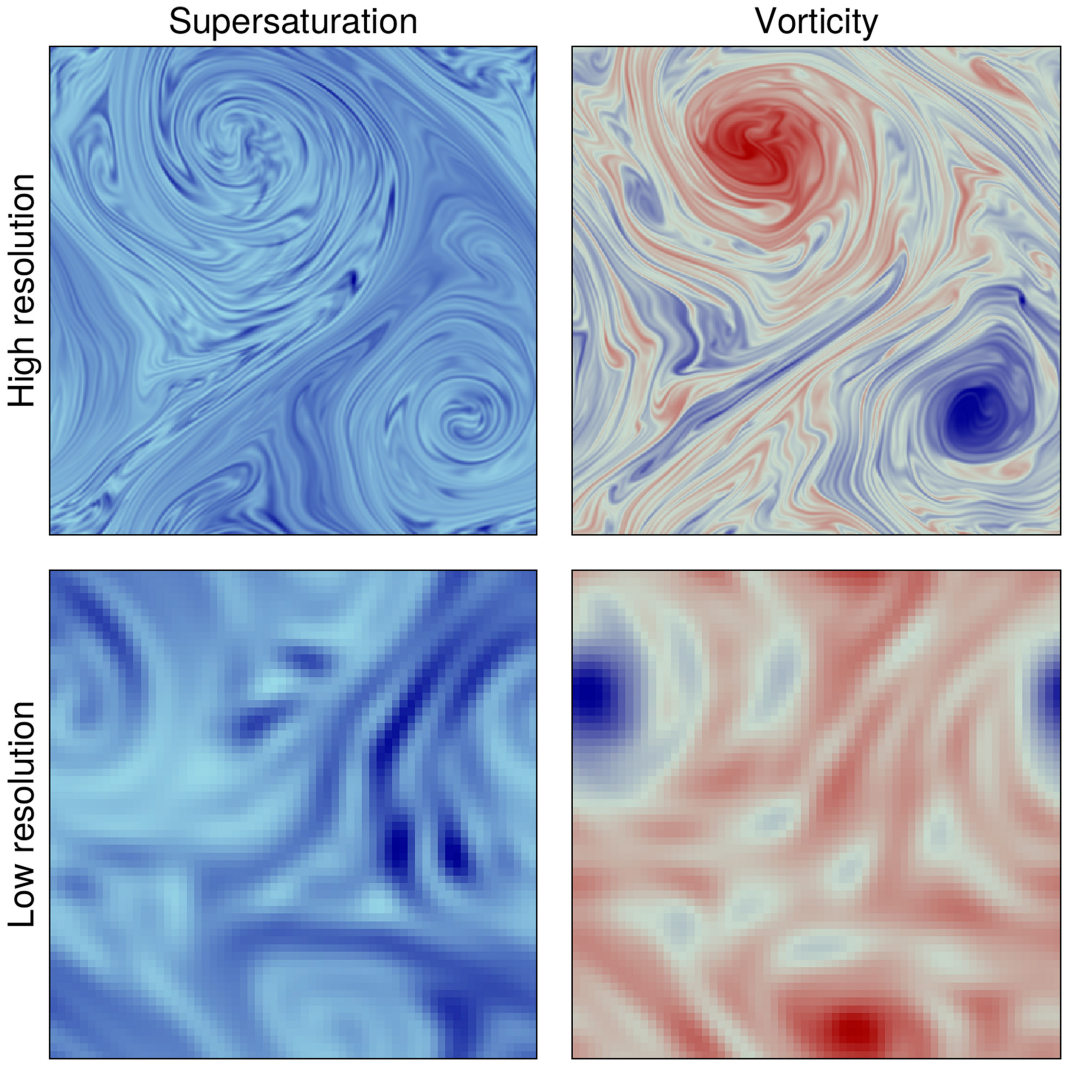

FIG. 1. Random snapshots from the two-dimensional fluid dynamics models. The top row shows high-resolution snapshots, while the bottom row shows low-resolution snapshots. The left column shows the supersaturation tracer field, and the right column shows the vorticity field. The values corresponding to the colors are irrelevant for the methods presented in this paper, but for the supersaturation field, positive values (blue colors) correspond to regions in which the condensation term in the simulation model is active (e.g., idealized rainfall events occur). The white regions in the supersaturation field are areas of saturation deficits (no idealized rainfall events). For the vorticity field, red colors correspond to positive values (cyclonic vorticity), while blue values correspond to negative values (anticyclonic vorticity).

where $\mathbf{s}(\mathbf{x}, t)$ is the score of the data distribution,

$$\mathbf{s}(\mathbf{x}, t) \equiv \nabla_x \log p_{\text{data}}(\mathbf{x}). \tag{7}$$

The goal of the training process used in diffusion modeling is to determine a parameterized representation of the score $\mathbf{s}_\theta(\mathbf{x}, t) \approx s(\mathbf{x}, t)$ through gradient descent on an appropriately chosen loss function. That is, we parameterize the time derivative appearing in the reverse SDE and learn it from the data. The final field $\mathbf{x}(t = 0)$ resulting from this reverse simulation, with a trained model for the score, is the new data sample.

b. *Diffusion bridges using spectral information*

Using a diffusion bridge for domain translation entails chaining together the forward model for a source dataset and the reverse model for a target dataset (e.g., Meng et al. 2022; Su et al. 2023)—note that the forward model [Eq. (2)] requires no training, while the reverse model [Eq. (6)] requires the score, which must be learned. Our downscaling procedure builds on this idea; the discussion below makes it explicit why the approach works by making the connection to differential noising of spatial scales.

According to Eq. (3), the noised field $\mathbf{x}(t)$ is the sum of $\mathbf{x}(0)$ and Gaussian noise, and so we can write its Fourier transform

as the sum of the Fourier transform of $\mathbf{x}(0)$ and Gaussian noise [as the Fourier transform of Gaussian noise is (complex valued) Gaussian noise]. As these are uncorrelated, we can approximate the power spectral density of $\mathrm{PSD}_{\mathbf{x}(t)}(k)$, where $k = \sqrt{k_x^2 + k_y^2}$ is the wavenumber, as

$$\mathrm{PSD}_{\mathbf{x}(t)}(k) \approx \mathrm{PSD}_{\mathbf{x}(0)}(k) + \mathrm{PSD}_{\boldsymbol{\eta}(t)}(k), \tag{8a}$$

$$\boldsymbol{\eta}(\mathbf{t}) \sim \mathcal{N}(\mathbf{0}, \sigma^2(t)). \tag{8b}$$

The power spectral density of white noise is independent of the wavenumber: $\mathrm{PSD}_{\boldsymbol{\eta}(t)}(k) = \sigma(t)^2/N^2$, where N is the field size. The power spectral density reduces the 2D Fourier transform into a 1D signal which is independent of direction. For arbitrary 2D fields, this procedure loses a significant amount of information; for isotropic and homogeneous 2D fields, the phase information lost is less important. More details are provided in appendix C. Snapshots of fluid flows exhibit a characteristic decay in power with increasing k. Hence, as the diffusion time increases from $t = 0$ to $t = 1$, and $\sigma(t)$ increases, the smallest scales (largest k) are noised first, cf. Fig. 2 (see also Choi et al. 2022; Rissanen et al. 2023).

We assume the existence of a spatial scale λ^* above which the low-resolution data are unbiased; a high-resolution simulation passed through a low-pass filter would agree with the low-resolution simulation for $\lambda > \lambda^*$. The existence of λ^* implies that the expected power spectral density $\mathrm{PSD}(k)$ of the low-resolution data and the high-resolution data agree for wavenumbers $k < k^*$, where $k^* = 2\pi/\lambda^*$. Given the value of k^*, one can therefore estimate the diffusion time t^* at which signals on all spatial scales smaller than λ^* have a signal-to-noise ratio of $\lesssim 1$,

$$t^* = \sigma^{-1}[\sqrt{N^2 \mathrm{PSD}(k^*)}], \tag{9}$$

since $\sigma(t)$ is a known analytic function.

We are interested in translating a field from a source domain $\mathbf{x}_{\mathscr{S}} \in \mathscr{S}$ into a field from a target domain $\mathbf{x}_{\mathscr{T}} \in \mathscr{T}$. More concretely, our samples from $\mathscr{T}$ are 512×512 fields generated by solving a fluid simulation at high resolution, and our samples from $\mathscr{S}$ are 512×512 fields generated by solving a fluid simulation at 64×64 resolution and then upsampling and low-pass filtering. All data $\mathbf{x}$ lie in $\mathbb{R}^{512\times512}$; by "target" and "source" domains $\mathscr{S}$ and $\mathscr{T}$, we refer to the lower-dimensional manifolds within $\mathbb{R}^{512\times512}$ that we assume the data lie on.[1] The downscaling (domain translation) algorithm defines a function mapping from $\mathscr{S} \in \mathbb{R}^{512\times512}$ to $\mathscr{T} \in \mathbb{R}^{512\times512}$. We use a sample $\mathbf{x}_{\mathscr{S}}$ as an initial condition and solve the forward noising model of the source domain to time t^*. We then use $\mathbf{x}(t^*)$ as an initial condition and solve the reverse denoising model of the target domain to $t = 0$. The resulting field $\mathbf{x}(0)$ is the generated field from $\mathscr{T}$. This transport map is probabilistic because different evaluations yield different samples from the target domain. That this process approximately samples from the conditional $p(\mathbf{x}_{\mathscr{T}}|\mathbf{x}_{\mathscr{S}})$ is not proven here, and may not be exact, but it is intuitive: the large-scale features between the two fields are kept fixed during this sampling process. The downscaling algorithm is defined more precisely in Algorithm 1.

[1] Those trajectories generated by fluid simulations are constrained to lower-dimensional manifolds seem plausible given the conserved quantities and partial differential equations governing the flow, and many large dimensional datasets are observed to lie on lower-dimensional manifolds, e.g., Brown et al. (2023).

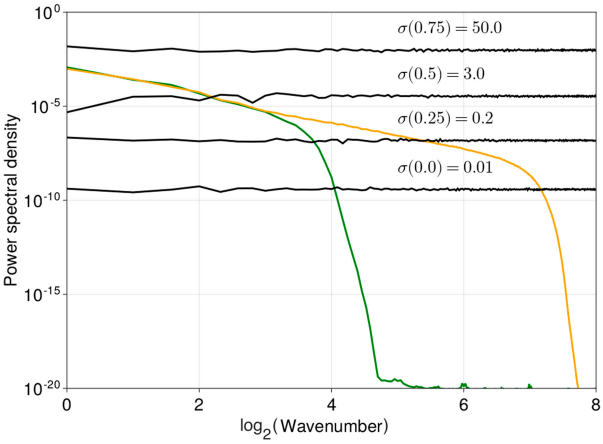

FIG. 2. The power spectral density for the vorticity field for both the low- and high-resolution datasets, along with the power spectral density of Gaussian white noise of different variance $\sigma(t)^2$. As larger amplitude Gaussian noise is added at larger diffusion times ($t \to 1$), the noising process will erase the information on the smallest scales first; see also Rissanen et al. (2023).

ALGORITHM 1 DOWNSCALING ALGORITHM. STEPS 1–3 ONLY ARE CARRIED OUT ONCE, WHILE STEPS 4–7 ARE CARRIED OUT FOR EACH DOWNSCALED DATA SAMPLE

1) Compute the expected power spectral densities for the source and target domains, $\mathrm{PSD}_{\mathscr{S}}(k)$ and $\mathrm{PSD}_{\mathscr{T}}(k)$
2) Solve for k^* such that $\mathrm{PSD}_{\mathscr{S}}(k^*) = \mathrm{PSD}_{\mathscr{T}}(k^*) \equiv \mathrm{PSD}^*$
3) Compute $t^*(\mathrm{PSD}^*)$ [Eq. (9)]
4) Sample $\mathbf{x}_{\mathscr{S}} \sim p_{\mathrm{data},\mathscr{S}}(\mathbf{x})$
5) Obtain $\mathbf{x}(t^*)$ by solving Eq. (2) from $t = 0$ to $t = t^*$, using $\mathbf{x}_{\mathscr{S}}$ as an initial condition
6) Obtain $\mathbf{x}(0)$ by solving Eq. (6), with $\mathbf{s}_{\mathscr{T}}(\mathbf{x}, t)$ as the score, from $t = t^*$ to $t = 0$, using $\mathbf{x}(t^*)$ as an initial condition
7) Return $\mathbf{x}(0) = \mathbf{x}_{\mathscr{T}} \sim p(\mathbf{x}_{\mathscr{T}}|\mathbf{x}_{\mathscr{S}})$

Figure 3 shows the generated fields resulting from the downscaling procedure for different values of t^*. For $t^* \lesssim 0.5$, the diffusion bridge has preserved the large-scale features of the low-resolution field, but only the finest-scale high-resolution features have been filled in. Intermediate scales are missing. For $t^* \gtrsim 0.5$, the forward noising process has erased some or all of the large-scale features we wish to preserve. In the limit of $t^* = 1$, we have sampled a high-resolution field from $p_{\mathrm{data},\mathscr{T}}$ without any information from the source field preserved. The optimal value, with respect to downscaling, is near $t^* = 0.5$, as also shown in Fig. 2.

The core idea of finding an optimal value of t^* was also explored in Meng et al. (2022) but without explaining the connection to spatial scales. In that work, they used a trade-off

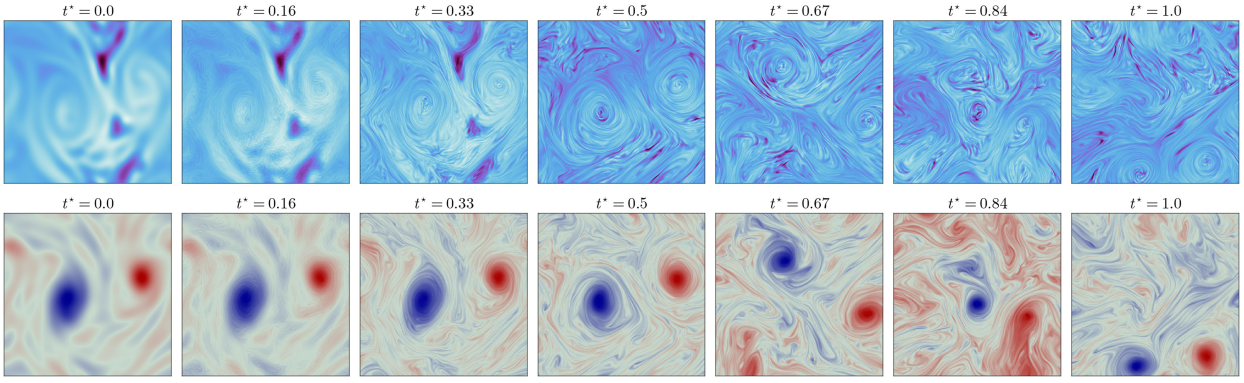

FIG. 3. The effect of t^* on the generated downscaled fields of the (top) supersaturation tracer and (bottom) vorticity. Too small a value does not result in realistic looking high-resolution fields, while too large a value leads to realistic, but randomly chosen, high-resolution fields. An optimal value of $t^* \lesssim 0.5$ yields a realistic downscaled version of the low-resolution source field ($t = 0$). For the supersaturation field, positive values (blue colors) correspond to regions in which the condensation term in the simulation model is active (e.g., idealized rainfall events occur). The white regions in the supersaturation field are areas of saturation deficits (no idealized rainfall events). For the vorticity field, red colors correspond to positive values (cyclonic vorticity), while blue values correspond to negative values (anticyclonic vorticity).

between faithfulness to the "guide" field (equivalent to our low-resolution field) and realism with respect to the target field and employed kernel inception scores and L2 norms. Choosing the optimal t^* will rely on the metrics of interest to a given problem, and many methods may work well.

It is important to note that this process requires that the power spectral density of the fields of interest decrease with wavenumber. Additionally, our current implementation relies on the fact that different channels of the field have similar spectral density shapes, so that a single value of t^* works for each channel. A more general algorithm would allow the different channels to have different schedules $\sigma(t)$.

c. *Contextual information*

In the case of real climate simulations and data, the transport map may also depend on contextual information, such as surface properties and topography. This is because the high-resolution flow is affected by these contextual fields on small scales, and so they can provide additional information with which to accurately downscale. Furthermore, when downscaling global climate data, it may be useful to first split the global data fields into smaller patches and perform downscaling on individual patches. The contextual information is crucial for encoding location-specific attributes to the flow. Ideally, the contextual information would have the same resolution as the high-resolution climate data. For example, global elevation data at 1-km resolution is available (Sandwell et al. 2014), and some land surface properties are available from satellite data. To use in training, these data would first need to be put on the same grid as the high-resolution climate fields and then appended as an additional "channel" to each snapshot of climate simulation data. To study how well the diffusion model can make use of contextual information, we employ a prescribed contextual field which our data-generating model, an advection–condensation model, depends on (the dependence is explained in section 2 and appendix A). This context has no physical connection to topography, but it is used in the same way that a topography field would be in downscaling realistic climate data.

The contextual information, which we denote generically as $\mathbf{x}_{\mathscr{C}}$, is aligned with the fluid state variable fields, denoted by $\mathbf{x}$. As such, we are able to sample pairs from the data distribution $(\mathbf{x}, \mathbf{x}_{\mathscr{C}}) \sim p(\mathbf{x}, \mathbf{x}_{\mathscr{C}})$. For the low-resolution runs, the context is taken to be flat. At high resolution, the context is available at the same resolution as the fluid state variable fields and is spatially varying. As described above, we treat the context as an additional channel as input into the convolutional neural network modeling the score function. We do not carry out any "diffusion" on these channels. More details on this can be found in appendix B.

Including contextual information does not change the explanation given above with respect to the diffusion bridge. We assume the existence of a spatial scale above which the low-resolution simulation is unbiased and above which contextual information has not affected the flow. A data point from the source (low resolution) domain is noised via Gaussian noise until the small-scale information is lost while the very largest scales remain approximately the same. Reverse diffusion is then applied to map the noised field toward the target (high resolution) domain. It is in the reverse-diffusion process where contextual information enters and plays a role. This means that depending on the contextual information, a different segment of the target domain is reached. This highlights how contextual information can be used to guide the generative process.

The process is illustrated in Fig. 4, where we use the same source field to generate downscaled fields with different contexts (we only show the supersaturation field; vorticity is unaffected by the context in our setup). For the $k_x = k_y = (8, 16)$ cases (top two rows), the same value of t^* can be used because the high-resolution and low-resolution power spectral densities only agree on spatial scales larger than the spatial scale of

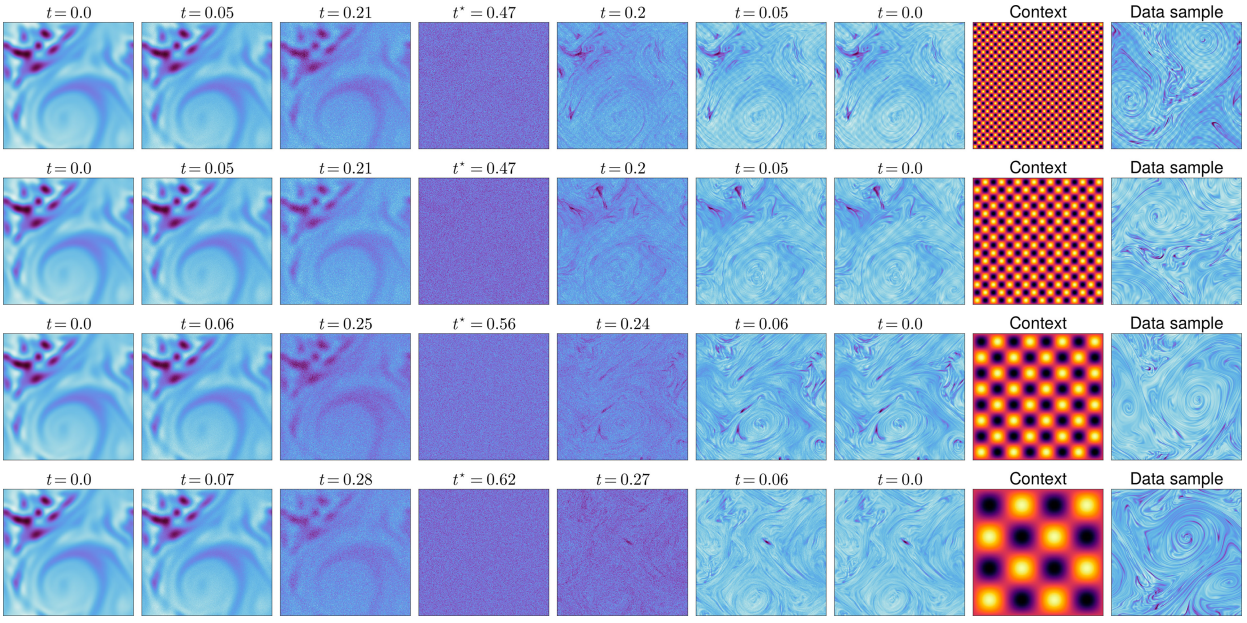

FIG. 4. Downscaling the same low-resolution supersaturation tracer field, using four different contexts (by row). On the left is the source field, at $t = 0$. Progressing to the right is equivalent to progressing through the downscaling procedure: the forward model of the source domain is used to noise the fields, and at $t = t^*$, we switch to the reverse model to integrate back to $t = 0$. The generated high-resolution samples (at $t = 0$) have different periodic signals in the fluid flow; these are due to their specific context (second to rightmost column). A randomly chosen data sample for each context is also shown (rightmost column). All noised fields have been scaled to have the same range, which is necessary as the variance of the added noise grows with time. For the supersaturation field, positive values (blue colors) correspond to regions in which the condensation term in the simulation model is active (e.g., idealized rainfall events occur). The white regions in the supersaturation field are areas of saturation deficits (no idealized rainfall events).

the contextual perturbation. On the other hand, for the $k_x = k_y = 2$ case (bottom row), we must use a larger value of t^* in order to recover the signal from the contextual field. At the same time, this value of t^* is in the regime where low-resolution information is being lost (as shown in Fig. 3). The wavenumber $k_x = k_y = 4$ case is in between. The large-scale features of the source field are present but distorted, and the modulation is less obvious than in the randomly drawn data sample with the same context. In section 4, we show the power spectral densities for the low- and high-resolution datasets, making this discussion more quantitative.

With respect to Fig. 4, note that 1) even if each downscaling simulation used the same contextual fields, the generated high-resolution fields would be different due to the probabilistic nature of the downscaling process, and 2) there is a single diffusion model used to generate the samples with different contexts. All data across all contexts were used to train this model.

d. A bypass for spatial mean bias reduction

As discussed in the introduction, diffusion models can struggle to produce fields with correct spatial means ["color shifts" in red–blue–green (RGB) fields] while producing realistic spatial variations (e.g., power spectra appear reasonable). The recommended solution to this is to employ an EMA of the parameters of the model with a long memory (Song and Ermon 2020). In some score network architectures, attention blocks are used (e.g., Ho et al. 2020) which may also improve the color shift, as self-attention allows for learning nonlocal features (Wang et al. 2018). Most of these approaches incur additional computational cost during training, which are not necessarily prohibitive but may nevertheless be avoided.

Errors in the spatial means of fields can only result from errors in the spatial mean of the score. Though the neural networks used in score modeling have the capability to predict this, they do not learn to do so efficiently. A practical solution is to split the network's task into two individual, and independent, tasks: predicting the spatial mean of the score and predicting the spatial variation about the mean of the score. We realize this by predicting the spatial mean of the score in a bypass layer of our network with independent parameters. This allows us to keep our neural network architecture simple, essentially consisting only of a basic U-Net (Ronneberger et al. 2015) and does not require the longer training process required by other methods during the training process. By avoiding more complex solutions, we are able to keep the number of trainable parameters smaller, thereby keeping our training and sampling procedures as computationally efficient as possible. As a result, our generated fields exhibit no discernible color shifts even when we generate samples with a simple Euler–Maruyama sampler.

e. Downscaling diffusion bridges

To carry out our diffusion-based downscaling method, we train a contextual diffusion model for our high-resolution

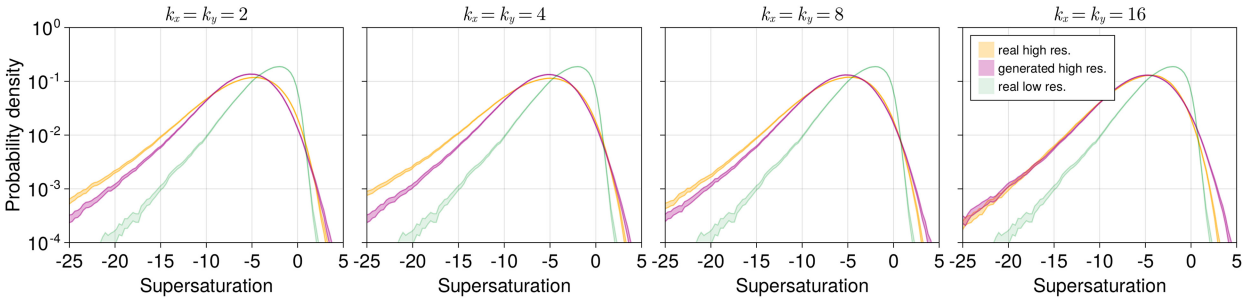

FIG. 5. Probability density function estimates for values of the supersaturation tracer field. Columns correspond to different high-resolution data subsets, where k indicates the saturation specific humidity modulation wavenumber. Orange distributions show distributions of real high-resolution samples, purple distributions show distributions of generated and downscaled high-resolution samples, and green distributions show distributions of real low-resolution samples. Estimates of probability densities are performed via kernel density estimation. Shaded areas are computed from 10 000 bootstrap samples at the 99% confidence interval.

dataset. Since the forward noising process is independent of the score function, we do not need to train a model for the low-resolution dataset (we would only need that if we wanted to generate low-resolution samples as well). In other words, the noising direction acts like a pretrained encoder does within a hierarchical autoencoder setup (Luo 2022).

Details on the construction of the diffusion models, the network architecture, the loss function, the training procedures, and the sampling method are provided in appendixes B and C. We note here that our network architecture does not preserve the doubly periodic nature of the flow fields. This is because this is a unique feature of this dataset which will not be present in most applications, for example, in downscaling patches of a larger fluid simulation.

4. Results

In this section, we assess the quality of the samples generated from our diffusion bridge according to several metrics. In terms of bias correction, we focus on biases in spatial mean values, intermediate scale biases, and biases in more extreme tail events (e.g., tails of distributions). We additionally quantify how well large-scale information is retained and how well small-scale information is added by our model and explore how the model generalizes to an unseen contextual field.

In what follows, we focus on comparing distributions and summary statistics between the low-resolution data, the generated downscaled data, and the real high-resolution data. We do this not because we do not have access to a ground truth (e.g., paired high-resolution and low-resolution data) but because the generative model samples from a distribution. This is a desirable feature since many high-resolution flow fields will be consistent with any given low-resolution simulation output. A demonstration of this is provided in appendix E.

a. *Distributions of supersaturation tracer vorticity values*

Figures 5 and 6 show probability density functions for the supersaturation tracer and vorticity fields for four different context fields. The value of $k_x = k_y$ indicates the modulation wavenumber used in the context field, according to Eq. (1). The green probability density functions show the distribution of values for the real low-resolution data, while the orange probability density functions show the values for the real high-resolution data. The purple probability density functions show the distribution of values from the downscaled (e.g., generated) high-resolution samples using the context-dependent diffusion bridge approach.

Figures 5 and 6 show that the context-dependent diffusion bridge approach shifts the distribution of the field values computed from the low-resolution dataset close to the distribution

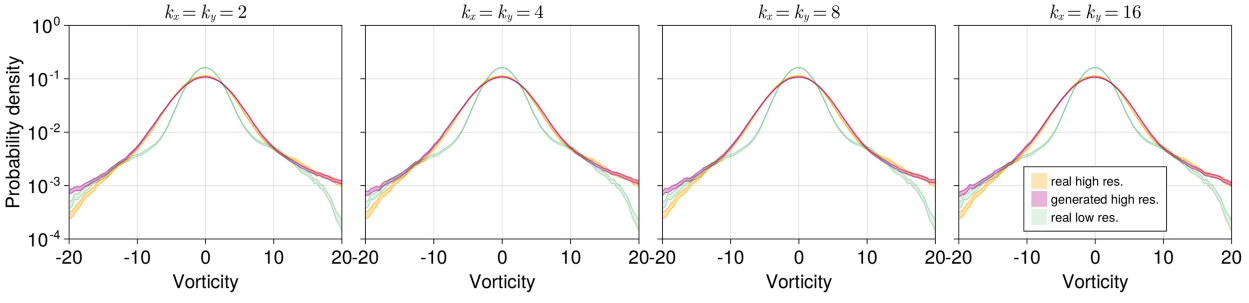

FIG. 6. As in Fig. 5, but we are now plotting all quantities for vorticity. Unlike for the supersaturation tracer field values, vorticity field values are distributed nearly symmetrical around zero because the vorticity forcing in the simulations also has this property, whereas this is not the case for the supersaturation tracer.

TABLE 1. Kolmogorov–Smirnov distances between the pixel distributions of downscaled high-resolution (*gh*) and real high-resolution (*rh*), and between real low-resolution (*rl*) and real high-resolution data. Note that the vorticity channel is unaffected by the context wavenumber. Please see the text for additional details.

Field	Wavenumber	$KS_{gh,rh}$	$KS_{lh,rh}$
Supersaturation tracer	2	0.078	0.360
Supersaturation tracer	4	0.040	0.391
Supersaturation tracer	8	0.038	0.369
Supersaturation tracer	16	0.089	0.324
Vorticity	2	0.009	0.087
Vorticity	4	0.011	0.087
Vorticity	8	0.011	0.087
Vorticity	16	0.011	0.087

of values computed from the high-resolution dataset. One can see that both the mean and variance of the distribution are adjusted by downscaling so that the generated and high-resolution distributions are much closer to each other than generated and low-resolution distributions. However, the left tails remain consistently underestimated, for the supersaturation tracer, and consistently overestimated for the vorticity, by an $\mathcal{O}(1)$ factor. We checked that random generated high-resolution fields (*not* downscaled low-resolution fields) demonstrated the same behavior (not shown), indicating that these errors originate in the model itself and not in the downscaling procedure. To improve the diffusion model, more training data could be used, or hyperparameter tuning could be carried out for the architecture or optimizer parameters.

To better quantify the similarity between these distribution functions, we computed the Kolmogorov–Smirnov distance between the real and generated high-resolution data and between the real high-resolution and real low-resolution data. To compute the Kolmogorov–Smirnov (KS) distance, we used 1.6 million random pixel samples from 800 independent fields in each dataset and computed the empirical cumulative distribution functions $F(x)$ for each. We then computed the distances as

$$\begin{aligned} KS_{gh,rh} &= \sup_x\{|F_{gh}(x) - F_{rh}|\} \\ KS_{rl,rh} &= \sup_x\{|F_{rl}(x) - F_{rh}|\}, \end{aligned} \tag{10}$$

where *r* and *g* stand for real and generated, respectively, and *l* and *h* stand for low and high resolution, respectively. The results are provided in Table 1.

b. *Spatial means of supersaturation tracer and vorticity*

Figure 7 shows the probability density function estimates for the spatial mean of the supersaturation tracer field. These demonstrate that the low-resolution simulations differ from the high-resolution simulations even in the spatial mean of the supersaturation tracer field. This indicates that the diffusion-bridge-based downscaling approach does more than just adding in the small spatial scale features; it corrects biases in large-scale features as well. Although we find that the spatial mean biases get corrected, the variance of spatial mean values is larger in the generated data samples than it is for the real high-resolution data. It is possible that a refinement of the mean-bypass layer could help alleviate this discrepancy.

Figure 8 shows the probability density function estimates for spatial means of the vorticity field. By design of the two-dimensional fluid dynamics model, the spatial mean of the vorticity fluctuates around zero and is nearly conserved. This is recovered in the generated data samples and is the result of our choice of including a mean-bypass layer in our neural network design, as described in section 3d and appendix C. Without the mean-bypass layer in our modified U-Net architecture, we find that it is more difficult to obtain data samples with minimal spread in spatial mean vorticity.

Again, we checked that random generated high-resolution fields (*not* downscaled low-resolution fields) demonstrated the same behavior (not shown), indicating that these errors originate in the model itself and not in the downscaling procedure.

c. *Power spectral densities and the role of contextual information*

Our fluid simulations depend on the contextual field via the supersaturation tracer equation, and the vorticity field is unaffected by this information. However, the role that the context plays in the high-resolution supersaturation field is not clear from the distribution of pixel values and spatial means (Figs. 5 and 7). Here, we explore the role of contextual information via the power spectral density of the flow. This metric also allows us to understand how well the downscaling method

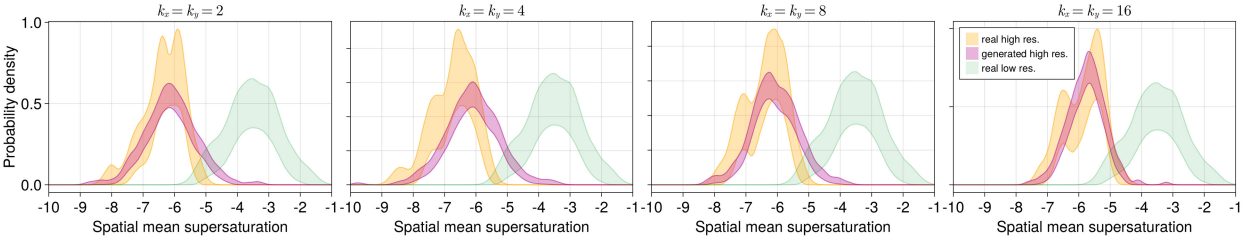

FIG. 7. Probability density function estimates for spatial means of the supersaturation tracer field. Columns correspond to different high-resolution data subsets, where *k* indicates the saturation specific humidity modulation wavenumber. Orange distributions show distributions of real high-resolution samples, purple distributions show distributions of generated and downscaled high-resolution samples, and green distributions show distributions of real low-resolution samples. Shaded areas are computed from 10 000 bootstrap samples at the 99% confidence interval.

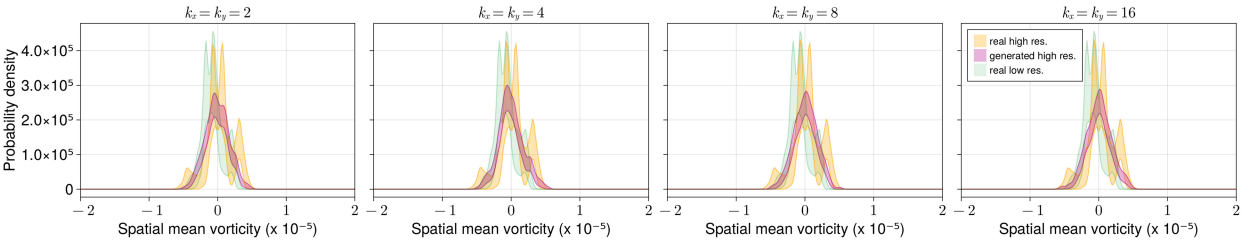

FIG. 8. As in Fig. 7, but we are now plotting all quantities for vorticity. Spatial mean vorticity is not conserved in our simulation but fluctuates around zero with small amplitude. The different simulations running with different contexts have slightly different mean vorticity values, leading to multiple peaks in the real data distributions. The generative model does not capture these small differences.

fills in information on small scales and corrects intermediate biases.

Figures 9 and 10 show how the context-dependent diffusion bridge downscaling algorithm performs in spectral space. Figure 9 shows the mean azimuthally averaged power spectral density for the supersaturation tracer field, and Fig. 10 shows the mean azimuthally averaged power spectral density for the vorticity field. The green spectra show the distribution of values for the real low-resolution data, while the orange spectra show the values for the real high-resolution data. The purple spectra show the distribution of values from the downscaled (i.e., generated) high-resolution samples using the context-dependent diffusion bridge approach. One can see that the real low-resolution spectra decay rapidly already at relatively low spatial wavenumbers. This is due to the increased damping of small scales in the fluid dynamical simulations. The context-dependent diffusion bridge approach not only "fills in" the missing part of the spectra when comparing low-resolution and high-resolution datasets but also corrects the intermediate scale bias stemming from the contextual information, i.e., the modulation of the background saturation specific humidity field in the high-resolution simulations.

For all wavenumbers, there can be an overall lack of power at all scales of the generated fields, compared with the real high-resolution fields, though the overall shape is correct. This implies that the correct spatial patterns are being learned, but that the overall contrast of the generated fields is slightly muted. This was observed during training; we speculate that a more refined neural network architecture for the score combined with more data would alleviate this disagreement. Again, this same behavior was seen in random generated fields and is therefore due to the model itself, and not to the downscaling procedure.

More importantly, the generated fields at wavenumbers $k_{x,y} = 4$ and $k_{x,y} = 2$ are lacking power at the wavenumber that the context imposes on the flow. This was also observed in Fig. 4 and discussed in section 3c. It is because our algorithm requires choosing a value of t^* which balances preserving the large-scale features of the flow with adding in the intermediate- and small-scale features. For the larger wavenumber contexts, this balance does not exist. That is, if the high-resolution and low-resolution fluid flows differ on essentially all scales, our method cannot work. Whether this type of translation would still be "downscaling" is unclear. We verified that this lack of power at the contextual wavenumbers in the generated fields was not due to the model itself; the purely generative high-resolution model (using $t^* = 1$) does correctly add in the contextual features in all cases; such a value of t^* would completely lose the low-resolution features we are trying to preserve in downscaling.

These figures also demonstrate the role that context plays in the downscaling procedure, as it clearly affects the power spectrum of the resulting fields. We took an initial step of testing our contextual diffusion model on an unseen context. To

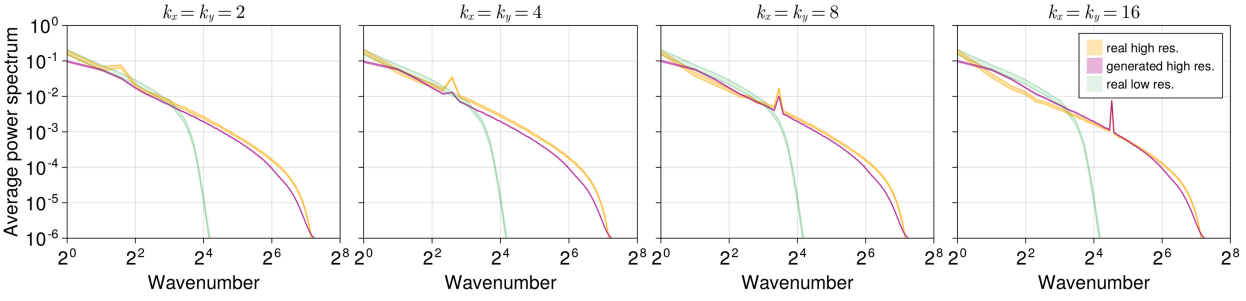

FIG. 9. Azimuthally averaged spectral density function estimates of supersaturation tracer field values. Columns correspond to different high-resolution data subsets, where k indicates the saturation specific humidity modulation wavenumber (the peak in the spectrum appears at the context modulation wavenumber). Orange spectra show spectra of real high-resolution samples, purple spectra show spectra of generated and downscaled high-resolution samples, and green spectra show spectra of real low-resolution samples. Shaded areas are computed from 10 000 bootstrap samples at the 99% confidence interval.

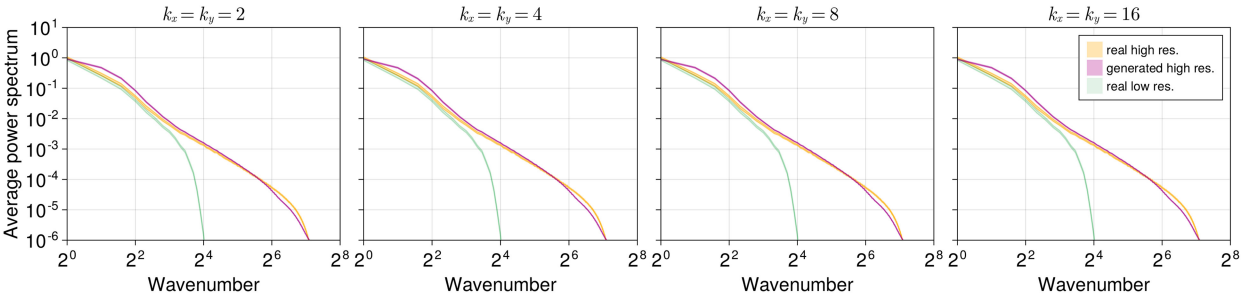

FIG. 10. As in Fig. 9, but we are now plotting all quantities for vorticity field values (e.g., the enstrophy spectrum).

assess this, we created a new context using two wavenumbers, one of which was not used during training, and carried out the downscaling algorithm. The generated field, context field, and power spectral density of the supersaturation tracer are shown in Fig. 11. The model correctly imposes the modulation, though it is hard to quantify its performance more quantitatively given that we do not have real fluid simulations with this context.

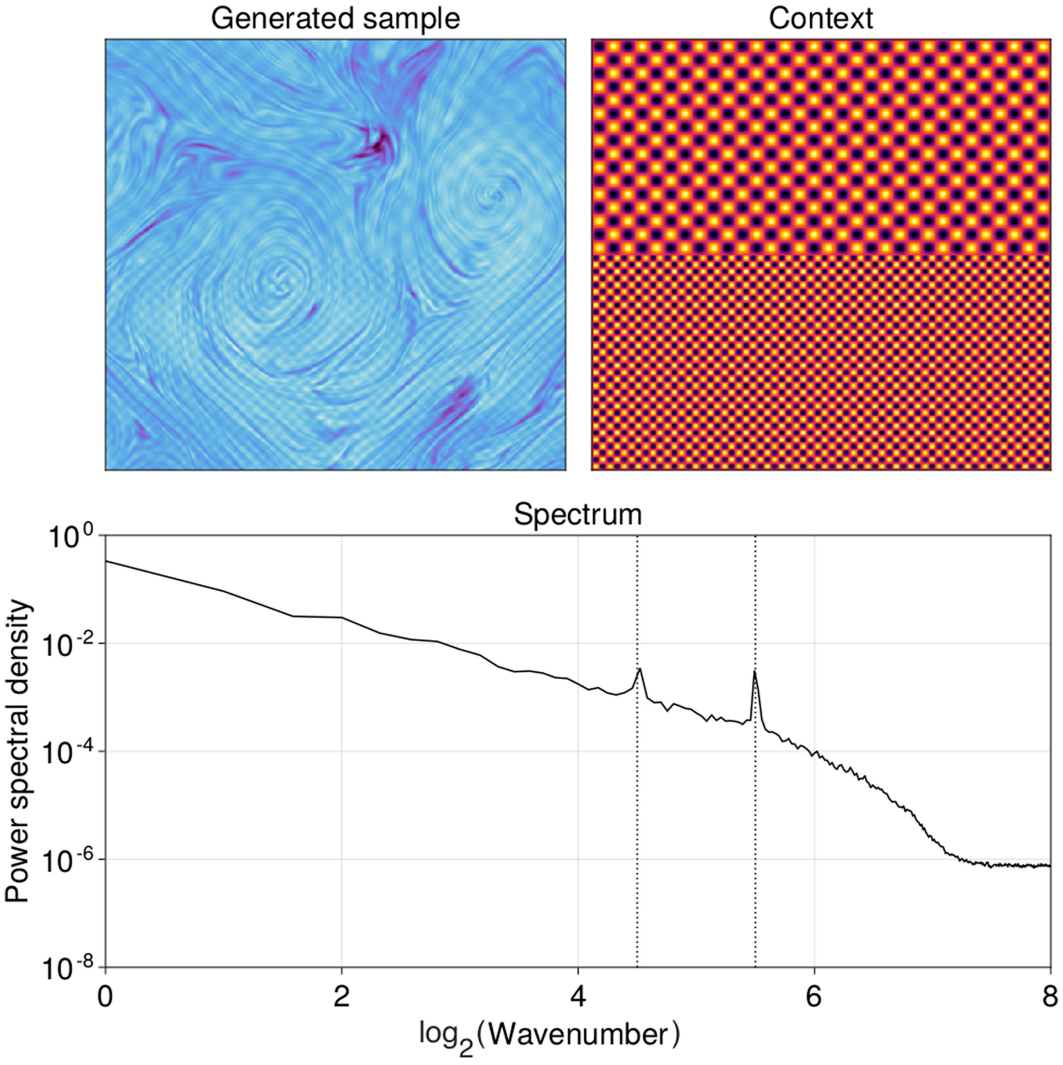

FIG. 11. Demonstration of the diffusion bridge algorithm using a contextual field that was not seen during training. (left) A downscaled supersaturation tracer field, (right) the context used during sampling, and (bottom) the power spectral density of the downscaled field. The dashed lines indicate the wavenumber of the two contextual spatial frequencies.

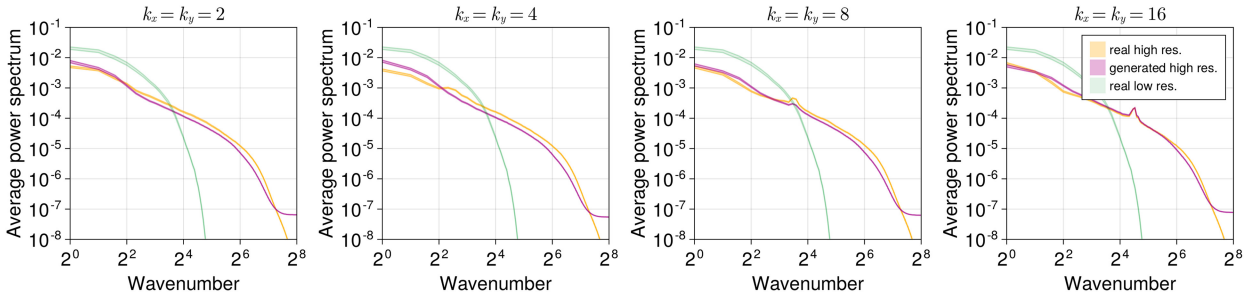

FIG. 12. Azimuthally averaged spectral density function estimates of the normalized composite field, defined in Eq. (11), created by multiplying the supersaturation tracer and vorticity field values at the pixel level. Columns correspond to different high-resolution data subsets, where k indicates the saturation specific humidity modulation wavenumber. Orange spectra show spectra of real high-resolution samples, purple spectra show spectra of generated and downscaled high-resolution samples, and green spectra show spectra of real low-resolution samples. Shaded areas are computed from 10 000 bootstrap samples at the 99% confidence interval.

d. *Spatial correlations between vorticity and supersaturation tracer*

The results presented in sections 4a–4c focused on the performance of the downscaling method by looking at the vorticity and supersaturation tracer fields independently. These fields are also correlated with each other, and it is important to also assess how well spatial correlations between these two fields are preserved as a function of scale. To do so, we computed the power spectral density of the composite field

$$\mathbf{y} = \frac{(\zeta - \overline{\zeta})(q' - \overline{q'})}{\sigma_{q'}\sigma_{\zeta}}, \tag{11}$$

where ζ is the vorticity, q' is the supersaturation tracer, overbars denote the computed spatial mean from the training data, and σ denotes the standard deviation computed using the training data. Figure 12 shows the computed power spectral density of $\mathbf{y}$ for each of the contextual wavenumbers. The results are similar in quality to those in section 4c: the downscaling procedure adds small-scale (large wavenumber) correlations between the two fluid fields with a magnitude that is comparable to that seen in the real high-resolution data.

e. *Condensation rate distributions*

To further assess the performance of our downscaling method, we compute the distribution of the condensation rate for low- and high-resolution datasets. To do this, we calculate a kernel density estimate of positive condensation rates over the data. The calculation of the condensation rate is given in appendix A and can be thought of as a rain formation rate in the idealized model. Figure 13 shows how the downscaling algorithm performs when evaluating the distributions of the condensation rate. The green distributions show the condensation rates for the low-resolution data, while the orange distributions show the condensation rates for the real high-resolution data. The purple distributions show the condensation rates from the downscaled (e.g., generated) high-resolution samples using the context-dependent diffusion bridge approach. One can see that the tails of the high-resolution data are underestimated by the low-resolution distribution by one or two orders of magnitude, especially for very rare events. This is because sharp peaks are smoothed out in low-resolution numerical simulations of fluid flows. The downscaling procedure "lifts" the tails up and alleviates the biases in condensation rate tail events.

However, we find that the generated samples overestimate the occurrence of very rare events (e.g., 1/1000 events). There

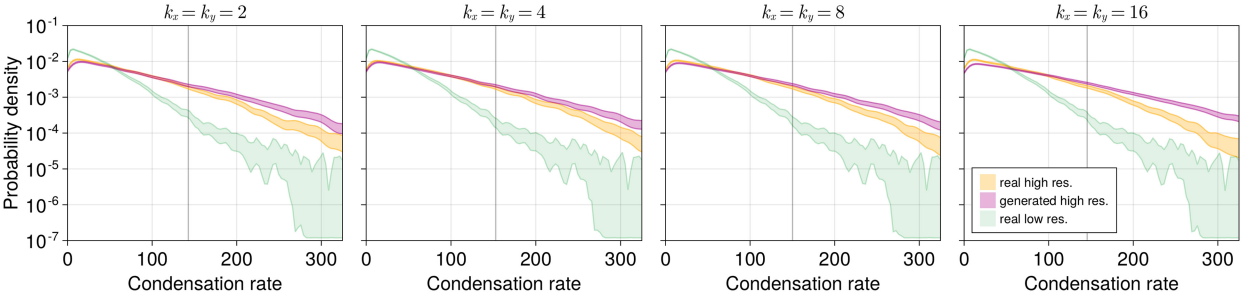

FIG. 13. Probability density function estimates for the instantaneous condensation rate. Columns correspond to different high-resolution data subsets, where k indicates the saturation specific humidity modulation wavenumber. Orange distributions show distributions of real high-resolution samples, purple distributions show distributions of generated and downscaled high-resolution samples, and green distributions show distributions of real low-resolution samples. Shaded areas are computed from 10 000 bootstrap samples at the 99% confidence interval. The vertical line approximately indicates the 90th percentile of true high-resolution condensation rates.

are many reasons why this could be the case. In general, machine learning models may perform poorly in the tails of distributions due to the lack of training data from this part of the data domain. However, we speculate that apart from this general difficulty, diffusion models can also leave a small amount of residual noise in the generated samples that is imperceptible to the human eye, but that manifests itself in tail statistics. This is due to the specific choice of noising schedule of many diffusion models in which the final noise added during sample generation is not equal to zero. To make further improvements to this issue, it may be necessary to find an improved noising schedule. Due to numerical instabilities that can appear when the final noise amplitude approaches zero, we leave this technical challenge for future work. We also note that errors in the tail of a distribution can arise from errors in the means. Our generated fields have larger variance in the means compared with the real fields (Fig. 7), which may also contribute to a shift in the tails.

f. *Conditional sampling assessment*

Using the notation from section 3c, we expect that a reasonable downscaling algorithm approximately generates samples from the conditional distribution $p(\mathbf{x}_{\mathscr{T}}|\mathbf{x}_{\mathscr{S}})$, where $\mathscr{T}$ is the high-resolution data domain and $\mathscr{S}$ is the low-resolution data domain. More concretely, we expect the large-scale spatial features to be preserved between $\mathbf{x}_{\mathscr{T}}$ and $\mathbf{x}_{\mathscr{S}}$. To test how well our algorithm meets this requirement, we compute the distance between downscaled fields and their low-resolution source fields using the pixelwise L^2 metric. We additionally compute the same statistic for two randomly chosen low-resolution fields and compare the distribution of these distances in each case to each other.

The resulting distribution of L^2-metric values is shown via boxplot in Fig. 14. This demonstrates that the downscaled fields are more similar to their low-resolution source fields than two randomly chosen low-resolution fields are to each other, indicating that broad spatial features are preserved by the diffusion bridge algorithm. Note that because biases exist between the high- and low-resolution datasets, as demonstrated in Figs. 7, 9, and 10, we first carried out the following transformation before computing the distance metric. We low-pass filtered the fields such that only spatial frequencies with $k < k^*$ are present. We then normalized the fields using the mean pixel value and standard deviation of the pixel values for the data domain in question. If we do not account for this, the L^2-metric value between the downscaled and real source fields can be very large but mostly due to the biases of the low-resolution data.

5. Conclusions

We have shown that a downscaling approach using context-dependent diffusion bridges can correct spatial mean biases and intermediate scale biases, as well as improve resolution in idealized low-resolution fluid dynamics simulations. In addition, we showed that this approach can help "lift the tails" of low-resolution condensation rates, leading to at least an order of magnitude correction in probability density values for the condensation rate tails. This suggests that diffusion-based generative models may be able to correct biases in extreme event rates even in more realistic settings and even without any explicit emphasis in the training loss function. We also demonstrated that the diffusion bridge method creates downscaled fields which match the statistics of the azimuthally averaged power spectrum and distributions of supersaturation tracer and vorticity values of the original high-resolution data and which match the statistics of the correlations between these variables as a function of scale. By introducing a bypass connection in the neural network used to model the score in the reverse-diffusion process within the diffusion bridge, the method alleviates the spatial mean bias (e.g., color shift) problem and preserves the value of the spatial mean vorticity. This implies that conservation laws based on global integrals may naturally be respected by diffusion models without further explicit emphasis in the loss function. While this may not be important for applications to realistic climate simulations, where smaller patches of the flow are downscaled one at a time, it may be useful in other contexts.

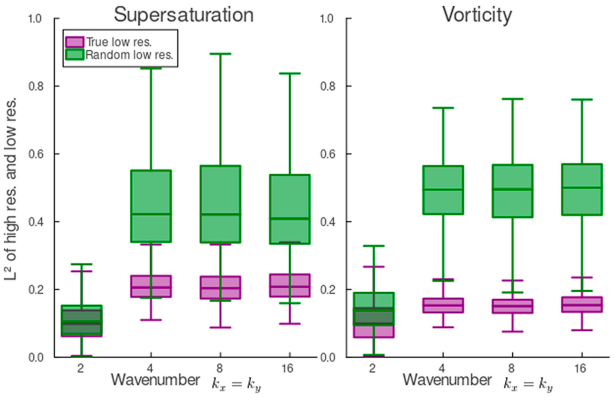

FIG. 14. Comparison of downscaled fields, by channel and by wavenumber, with their low-resolution source fields. The statistics of the pixelwise L^2-metric values between a filtered downscaled field and its filtered source (low resolution) field are shown in purple; the same statistics between two random low-resolution fields are shown in green.

As pointed out in the introduction, diffusion models can have advantages over classical and other generative machine learning methods for downscaling. We find that their usefulness can be summarized as follows:

- Diffusion models are flexible and reusable. The downscaling approach developed and applied in this work did not require any special tuning for the datasets at hand, and it did not require the low-resolution data during training at all. Domain translation tasks between data generated with other models or taken from observations only require training a diffusion model for each domain, thereby reducing the computational effort required during training.
- The loss functions used for training diffusion models in this work where generic and essentially unmodified and as such did not have any particular emphasis on extreme events. No quantile loss or spectral loss function was used in the training of our models.

- Diffusion bridges are able to approximately generate samples from high-resolution conditional distributions. This can be useful in application scenarios where complex statistical quantities need to be computed or where it is not known what kind of statistical quantities need to be computed later on after training.

While some of these advantages may not apply in every modeling scenario, we find that overall, the large flexibility of diffusion-based models makes them an appealing choice in generative modeling scenarios.

Alternatives and future directions

The data used in this work are comprised of a two-dimensional forced turbulent fluid and a supersaturation tracer. It included several features which are similar to a more complex climate simulation (non-Gaussian statistics of the supersaturation tracer and the influence of site-specific orography-like features). However, an obvious next step is to test the approach presented in this work with a realistic climate dataset and compare its performance more directly to other existing downscaling methods. In particular, it will be important to test with climate data the assumption that there is some scale above which the coarse-resolution simulations are unbiased (i.e., the existence of λ^* in section 3b), as it is possible that biases exist on the largest scales as well as intermediate scales. This of course will depend on what "large" and "intermediate" mean in terms of physical distances. An alternative is to split the debiasing and superresolution into two steps. As shown by Wan et al. (2023), this preserves many of the advantages discussed above, i.e., using unpaired data and using diffusion models, but this would not require our assumption. In addition, we also identify some possible research directions and outstanding questions:

- The work of Song et al. (2022) also suggests that one could use a projection step during sample generation to enforce constraints (e.g., matching small wavenumber features).
- Temporal coherence of samples may be achievable with diffusion models that are used in the context of video generation (Ho et al. 2022a). It would be interesting to test their performance on physical systems, but there may be drawbacks with respect to computational cost that need to be addressed.
- Guided sampling techniques for diffusion models, as introduced in Ho and Salimans (2022), may be useful in order to generate samples that have additional desirable characteristics, such as high values of certain climate indices.

However, there is still room to extend the scope of the current work. We have already identified minor discrepancies in the downscaled fields compared with the real high-resolution fields, as discussed in section 4, and determined that these were largely due to the model itself (rather than the downscaling procedure). A study refining network architectures and the training procedure may improve the model and results. Additionally, using more varied contexts in training, and truly demonstrating generalization to out-of-sample contexts, is an important next step.

Overall, it appears that diffusion-based models are promising candidates for future applications in the Earth sciences.

Acknowledgments. This research has been supported by Eric and Wendy Schmidt (by recommendations of the Schmidt Futures), by the Cisco Foundation, and by the National Science Foundation (Grant AGS-1835860). The authors thank Simone Silvestri, Ricardo Baptista, Nikola Kovachki, Hongkai Zheng, Andrew Stuart, and Tapio Schneider for stimulating discussion on this work. We thank Andre Souza, who assisted with an earlier version of the fluid dynamics model used in this study. Andre Souza and Tapio Schneider provided helpful comments on an early draft of this manuscript. We also acknowledge the anonymous reviewers who made valuable suggestions toward improving and clarifying the article. Our simulations are based on the fluid simulation code https://github.com/FourierFlows/FourierFlows.jl and https://github.com/FourierFlows/GeophysicalFlows.jl (Constantinou et al. 2021, 2023). All numerical calculations and model training used for this manuscript were performed with the help of Caltech's Resnick High Performance Computing Center.

Data availability statement. A GitHub repository with our training and analysis code for the diffusion model can be found at https://github.com/CliMA/diffusion-bridge-downscaling. A GitHub repository for fluid simulations can be found at https://github.com/CliMA/SimpleMoisture.jl.

APPENDIX A

Data-Generating Model

The data-generating model used in this work consists of a dynamical system that mimics the advection and condensation of moisture along isentropes Earth's extratropical atmosphere. It is idealized but contains enough complexity to test the performance of the machine learning algorithms outlined in this work. Specifically, it exhibits some desirable dynamical and statistical properties. For example, the supersaturation field is highly variable in space, and the associated idealized instantaneous condensation rate follows a distribution with approximately exponential tails (O'Gorman and Schneider 2006). These properties are useful when evaluating the skill of generative machine learning models in terms of spectral and statistical accuracy, especially with respect to extreme events.

The motivation for and behavior of this model are described extensively in O'Gorman and Schneider (2006), and as a result, we only recapitulate the main ingredients of the model here. At its heart, the model consists of the two-dimensional vorticity equation on a periodic domain forced randomly and damped via linear drag and hyperdiffusion (spectral filtering is an alternative). The governing equations of the vorticity field read

$$\partial_t \zeta + \partial_y \Psi \partial_x \zeta - \partial_x \Psi \partial_y \zeta = f - a\zeta - \kappa \Delta^8 \zeta, \tag{A1}$$

$$\Delta \Psi = \zeta, \tag{A2}$$

where ζ denotes the vorticity, Ψ is the streamfunction, f is a stochastic forcing with an isotropic wavenumber spectrum

TABLE A1. Parameter values for the data-generating model. The complete dataset generated for this work consists of six subsets, a low-resolution dataset without any saturation specific humidity modulation and five high-resolution subsets with varying modulation wavenumbers. All simulations were run for 200 000 time steps, and the first 100 000 time steps were discarded as spinup for the purposes of the work presented here. The subset size reported includes the spinup. All values are in nondimensional form.

Shared parameters

Domain size (L)	Time step (Δt)	Time steps (N_t)	Drag coefficient (a)	q_s gradient (γ)	Evaporation rate (e)
2π	1×10^{-3}	100 000	1×10^{-2}	1.0	1.0
Relaxation time (τ)	**Forcing wavenumber (k_f)**	**Bandwidth (Δk)**	**Energy input rate (ϵ)**		
1×10^{-2}	3	2	0.1		

Subset parameters

Name	Resolution ($L \times L$)	Amplitude (A)	Wavenumber ($k_{x,y}$)	Hyperdiffusivity (κ)	Subset size (N_d)
low-res	64×64	0	No modulation	1×10^{-8}	2000
high-res-1	512×512	1	1	1×10^{-16}	2000
high-res-2	512×512	1	2	1×10^{-16}	2000
high-res-4	512×512	1	4	1×10^{-16}	2000
high-res-8	512×512	1	8	1×10^{-16}	2000
high-res-16	512×512	1	16	1×10^{-16}	2000

and power contained in a narrow ring in wavenumber space centered on k_f with bandwidth Δk, a denotes a frictional time scale, and κ acts as a hyperdiffusivity parameter. The equation for the specific humidity field q is given by

$$\partial_t q + \partial_y \Psi \partial_x q - \partial_x \Psi \partial_y q = e - c - \kappa \Delta^8 q, \tag{A3}$$

$$c = \frac{1}{\tau}(q - q_s)\Theta(q - q_s), \tag{A4}$$

where Ψ is again the streamfunction, e is an evaporation rate, taken as fixed in space and time, and c denotes the instantaneous condensation rate. Here, κ is the same hyperdiffusivity as in Eq. (A1). The condensation rate c is proportional to the difference between the specific humidity q and the saturation specific humidity q_s, but condensation is only active when $q > q_s$, as it would be in Earth's atmosphere. In our simulations, we consider the case where the condensation time scale τ is small and finite. In other words, supersaturated ($q > q_s$) regions are relaxed back to the saturation specific humidity q_s over a time scale τ. The finiteness of τ mimics nonequilibrium thermodynamic processes but is not essential for the conclusions of this work. As described in the main text, we vary q_s as a function of space to mimic both the decay of q_s along isentropes in Earth's atmosphere and to impose spatial inhomogeneities at different length scales [Eq. (1)]. For large mean saturation deficits ($q < q_s$), condensation events are rare and the mean condensation rate tends to zero. For large evaporation rates, evaporation overpowers the ability of the turbulence to generate subsaturated fluid parcels through advection up the mean moisture gradient (cf. O'Gorman and Schneider 2006).

The complete dataset consists of six subsets, one low-resolution subset and five high-resolution subsets, with a total of 12 000 data points. A summary of the parameters used to generate the complete dataset is given in Table A1.

APPENDIX B

Score Modeling Details

a. Diffusion models

As described in the main text, our diffusion model's noising process adds Gaussian noise to the field at each time step. We have adopted the so-called variance-exploding schedule where

$$g(t) = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^t \sqrt{2\log\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)}, \tag{B1a}$$

$$\sigma^2(t) = \sigma_{\min}^2\left[\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{2t} - 1\right] \approx \sigma_{\min}^2\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{2t}, \tag{B1b}$$

where $\sigma_{\min}$ and $\sigma_{\max}$ are scalar parameters determining the shape of the variance with time. Other noising processes, including in the coefficient space after projecting onto a set of basis functions (Phillips et al. 2022) and via a blurring process (Rissanen et al. 2023; Hoogeboom and Salimans 2022), have also been used, but they do not change the core idea of the diffusion model.

Diffusion modeling parameterizes the score function $\mathbf{s}_\theta(\mathbf{x}, t) \approx s(\mathbf{x}, t)$ and optimizes the parameters through gradient descent on an appropriately chosen loss function. In practice, one usually represents the score function $\mathbf{s}_\theta$ with a neural network f_θ defined by

$$\mathbf{s}_\theta(\mathbf{x}, t) = \frac{f_\theta(\mathbf{x}, t)}{\sigma(t)} \approx \mathbf{s}(\mathbf{x}, t). \tag{B2}$$

The benefit of this is that the neural network output will always be of $O(1)$, which can lead to an easier training task for the neural network [as opposed to forcing it to learn the prescribed $\sigma(t)$ dependence as well]. The downside is that $\sigma(t = 0)$ should ideally be zero, and as a result, this introduces a singularity at $\sigma(t = 0)$. To avoid this, it is a standard practice (Song and Ermon 2020) to instead set $\sigma(t = 0) = \sigma_{\min}$, as given in the approximation of the expression for $\sigma^2(t)$ given by Eq. (B1).

The denoising score-matching loss function (Ho et al. 2020; Song et al. 2021b) is given by

$$\mathscr{L}(\boldsymbol{\theta}) = \mathbb{E}_{t,\mathbf{x}(0),\mathbf{x}(t)}\left[\lambda(t)^2\left\{\frac{\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, t)}{\sigma(t)} - \nabla_{\mathbf{x}} \log p[\mathbf{x}(t)|\mathbf{x}(0)]\right\}^2\right] \tag{B3a}$$

$$= \mathbb{E}_{t,\mathbf{x}(0),\mathbf{x}(t)}\left[\lambda(t)^2\left\{\frac{\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, t)}{\sigma(t)} - \frac{[\mathbf{x}(t) - \mathbf{x}(0)]}{\sigma^2(t)}\right\}^2\right] \tag{B3b}$$

$$= \mathbb{E}_{t,\mathbf{x}(0),\mathbf{x}(t)}\left[\frac{\lambda(t)^2}{\sigma(t)^2}[\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, t) - \boldsymbol{\epsilon}]^2\right], \tag{B3c}$$

where

$$\mathbb{E}_{t,\mathbf{x}(0),\mathbf{x}(t)} = \mathbb{E}_{t\sim U(0,1],\mathbf{x}(0)\sim p(\mathbf{x}(0)),\mathbf{x}(t)\sim p(\mathbf{x}(t)|\mathbf{x}(0))}, \tag{B4}$$

and $\mathbf{x}(t) = \mathbf{x}(0) + \sigma(t)\boldsymbol{\epsilon}$ is a noised field at time t, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$ is a Gaussian random vector, and $\lambda(t)$ is a weighting factor taken to be equal to $\sigma(t)$ (see Song et al. 2021a). From the last step in Eq. (B3), one can see that the score-matching loss is equivalent to making the neural net learn the added noise at time t. Note that although this involves an L_2 loss between the score function and the gradient of the logarithm of the conditional distribution, optimizing $\mathscr{L}(\theta)$ results in an approximation to the true score of the unconditional distribution (Vincent 2011).

We slightly modified the above loss function to monitor the specific loss values with respect to spatial means and variations about the mean. As $\boldsymbol{\epsilon}$ in Eq. (B3) is random Gaussian noise, the mean $\overline{\boldsymbol{\epsilon}}$ is independent of the variations about the mean, $\boldsymbol{\epsilon}' = \boldsymbol{\epsilon} - \overline{\boldsymbol{\epsilon}}$. Since $\mathbf{f}_\theta$ seeks to match $\boldsymbol{\epsilon}$, we anticipate that the same will be true for it once the network is well trained. In that case, we can rewrite the loss as

$$\mathscr{L}(\boldsymbol{\theta}) = \mathbb{E}_{t,\mathbf{x}(0),\mathbf{x}(t)}\left[\frac{\lambda(t)^2}{\sigma(t)^2}[\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, t) - \boldsymbol{\epsilon}]^2\right] \tag{B5}$$

$$\approx \mathbb{E}_{t,\mathbf{x}(0),\mathbf{x}(t)}\left[\frac{\lambda(t)^2}{\sigma(t)^2}\left\{[\mathbf{f}'_{\boldsymbol{\theta}}(\mathbf{x}, t) - \boldsymbol{\epsilon}']^2 + [\overline{\mathbf{f}}_{\boldsymbol{\theta}}(\mathbf{x}, t) - \overline{\boldsymbol{\epsilon}}]^2\right\}\right]. \tag{B6}$$

In practice, this should not affect the training procedure; we found it useful mainly to track errors in the means which result in color shifts.

b. Contextual diffusion models

When conditioning sampling on contextual information in climate modeling scenarios, such as topography, bathymetry, or land surface properties, we do have paired data points for high- and low-resolution fields. Denoting these contextual fields as $\mathbf{x}_{\mathscr{C}}$, we have access to samples from the joint distributions

$$\mathbf{z}_{\mathscr{S}} = (\mathbf{x}_{\mathscr{S}}, \mathbf{x}_{\mathscr{C}}) \sim p(\mathbf{z}_{\mathscr{S}}), \tag{B7a}$$

$$\mathbf{z}_{\mathscr{T}} = (\mathbf{x}_{\mathscr{T}}, \mathbf{x}_{\mathscr{C}}) \sim p(\mathbf{z}_{\mathscr{T}}), \tag{B7b}$$

where, as in the main text, $\mathscr{T}$ and $\mathscr{S}$ denote the target and source domains. Then, we can follow Song et al. (2021b) to allow for conditional sampling. By optimizing the loss function,

$$\mathscr{L}(\boldsymbol{\theta}) = \mathbb{E}_{t,\mathbf{z}(0),\mathbf{z}(t)}\left[\lambda(t)^2\left\{\frac{\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}, t)}{\sigma(t)} - \nabla_{\mathbf{x}} \log p[\mathbf{z}(t)|\mathbf{z}(0)]\right\}^2\right], \tag{B8}$$

where

$$\mathbb{E}_{t,\mathbf{z}(0),\mathbf{z}(t)} = \mathbb{E}_{t\sim U(0,1],\mathbf{z}(0)\sim p(\mathbf{z}(0)),\mathbf{z}(t)\sim p(\mathbf{z}(t)|\mathbf{z}(0))}, \tag{B9}$$

and $\mathbf{z}(t) = (\mathbf{x}(t), \mathbf{x}_{\mathscr{C}}(t))$ is the tuple containing the state of the fluid flow or climate model $\mathbf{x}(t)$ and the corresponding contextual information $\mathbf{x}_{\mathscr{C}}(t)$. We choose not to noise the context variables so that

$$p(\mathbf{z}(t)|\mathbf{z}(0)) = p(\mathbf{x}(t)|\mathbf{x}(0))\delta(\mathbf{x}_{\mathscr{C}}(t) - \mathbf{x}_{\mathscr{C}}(0)), \tag{B10}$$

and hence, the score functions can be related as

$$\nabla_{\mathbf{x}} \log p(\mathbf{z}(t)|\mathbf{z}(0)) = \nabla_{\mathbf{x}} \log p(\mathbf{x}(t)|\mathbf{x}(0)). \tag{B11}$$

The resulting score function in this contextual setup is then a known function, just like in the case of unconditional diffusion models. As shown in Batzolis et al. (2021), optimizing this loss function is equivalent to learning a function $\mathbf{f}_\theta(\mathbf{z}, t) = \nabla_{\mathbf{x}} \log p(\mathbf{x}(t)|\mathbf{x}_{\mathscr{C}})$, i.e., one that represents the conditional score. In implementation, we realize this by inputting $\mathbf{x}_{\mathscr{C}}$ as an additional channel of the diffusion model input. More discussion of the architecture is given in appendix C.

APPENDIX C

Network Architecture

a. Network architecture

The foundation of our score network is a U-Net (Ronneberger et al. 2015), which maps two inputs [$\mathbf{X}$, a tensor of size (N, N, C_{in}, B), and t, a tensor of size (B)], to a single output $\mathbf{Y}$, a tensor of size (N, N, C_{out}, B). That is, U-Net returns

$$\mathbf{Y} = \mathscr{U}(\mathbf{X}, t; \boldsymbol{\theta}), \tag{C1}$$

where $\mathscr{U}$ denotes the U-Net with parameters $\boldsymbol{\theta}$ described in more detail below.

The first input $\mathbf{X}$ holds a batch of fields, and the second t is a batch of times; B is the size of the batch. Any individual channel of the input or output is a field of size (N, N); there are C_{in} input channels and C_{out} output channels. For our dataset, our input fields have two noised channels: the fluid vorticity and supersaturation tracer concentration. Including the contextual information, we have $C_{\text{in}} = 3$ and $C_{\text{out}} = 2$.

In our default configuration, the U-Net has five distinct parts. The first is an initial lifting layer, which is a convolution that preserves the spatial dimensionality of $\mathbf{X}$, but increases the number of channels from C_{in} to 32. Three downsampling (convolutional) layers follow, which reduce the spatial dimensionality by a factor of 2 and which increase the number of

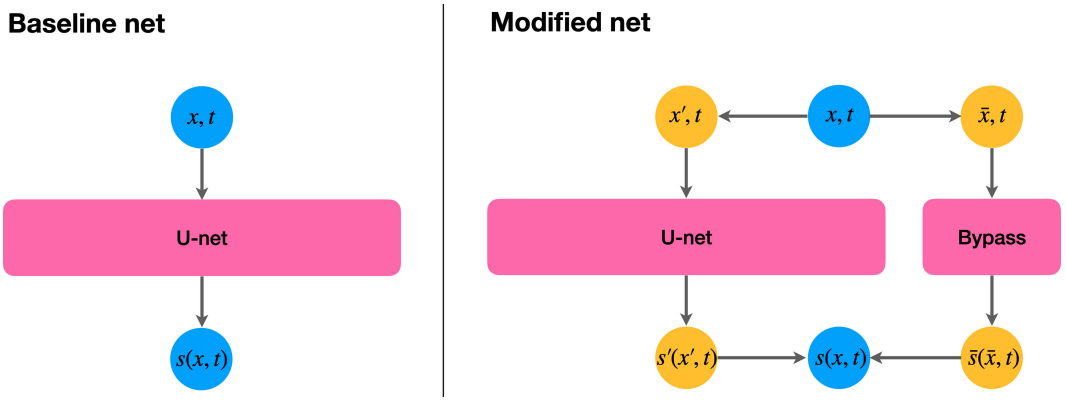

FIG. C1. Diagram of the two score networks discussed in this work. The left network is a baseline U-Net (Ronneberger et al. 2015), while the right network processes the spatial means of the input data separately via a nonlinear mean-bypass layer.

channels by a factor of 2. These transformed data are passed through eight residual blocks which preserve the dimensionality of the transformed data (He et al. 2016). Then, three upsampling layers, comprising nearest-neighbor upsampling, followed by convolutions, increase the spatial dimensionality while decreasing the number of channels, mirror the downsampling layers. Finally, a projection layer decreases the number of channels to C_{out}. We use 3×3 convolutional kernels, group normalization (Wu and He 2018), and the Swish function as a nonlinearity (Ramachandran et al. 2017).

The time variable is first embedded using a random Fourier projection (Tancik et al. 2020). This embedded time is then transformed by a dense network at each up- and downsampling layer, after which it is added to the up- or downsampled field. The sum is then group normalized and operated on by the Swish function, following Song et al. (2021b) and Ho et al. (2020), before being passed to the next layer.

b. Modifications to U-Net: Mean-bypass network

We modified the neural network architecture introduced in the previous section of this appendix. to include a bypass connection. The incoming batch is split into a component that has spatial variations (in fact, the original field after subtracting the channel-and-batchwise spatial means) and the spatial average of each channel and batch member. The spatial average is then fed through the bypass network, while the spatially varying component is fed through the U-Net. At the final layer, the output from the U-Net and the bypass is added together, after removing the spatial average of the output of the U-Net. In this way, we have a completely separate network handling the spatial means and the spatial variations about the mean. Along with our choice of loss function, this has the added advantage of making the mean prediction and spatial variation prediction entirely independent tasks. A diagram of this modified network architecture compared with the baseline architecture is shown in Fig. C1.

Concretely, we compute the spatial mean, by channel and batch member, of the input $\mathbf{X}$, denoted by $\overline{\mathbf{X}}$. The spatial variation about the mean is denoted as $\mathbf{X}' = \mathbf{X} - \overline{\mathbf{X}}$ and is processed by the U-Net as discussed in the previous section of this appendix, to produce an output $\mathscr{U}(\mathbf{X}', t; \boldsymbol{\theta})$. We then subtract the spatial mean (by channel and batch member) from this output, i.e., we produce a tensor $\mathbf{Y}' = \mathscr{U}(\mathbf{X}', t; \boldsymbol{\theta}) - \overline{\mathscr{U}}(\mathbf{X}', t\theta)$ that has zero spatial mean. A separate function $\mathscr{M}(\overline{\mathbf{X}}, t; \phi)$, with trainable parameters ϕ operates on $\overline{\mathbf{X}}$ and t, and returns a tensor of the same size as $\overline{\mathbf{X}}$, denoted by $\overline{\mathbf{Y}}$. In the last layer, we combine the outputs of these individual components to produce the final output $\mathbf{Y}$ as

$$\mathbf{Y} = \mathscr{U}(\mathbf{X}', t; \boldsymbol{\theta}) - \overline{\mathscr{U}}(\mathbf{X}', t; \theta) + \mathscr{M}(\overline{\mathbf{X}}, t; \phi) = \mathbf{Y}' + \overline{\mathbf{Y}}. \tag{C2}$$

The network $\mathscr{M}$ consists of a three-layer dense feed-forward network, consisting of two linear transformations followed by a normalization and nonlinear activation function, and a single final linear transformation, without an activation or normalization. The embedded time is handled in the exact same way as for $\mathscr{U}$; it is passed through a linear transformation before being added to the transformed input, prior to normalization and activation.

Note that because of this, our implemented solution does not take advantage of correlations between the spatial variations about the mean and the mean. If spatial variations of the input are useful for predicting the spatial mean of the score, or vice versa, our prediction will not make use of that information. Through limited testing, we found that letting $\mathscr{U}$ have access to the entire input $\mathbf{X}$ yielded slightly worse performance after training for the same number of epochs. More investigation is required in order to take into account these correlations.

c. Model training

We follow the recommendations of Song et al. (2021b) and Ho et al. (2020) in setting up the optimizer for score-matching denoising diffusion models. We use an Adam optimizer with a learning rate of $\lambda_0 = 2 \times 10^{-4}$, $\epsilon = 1 \times 10^{-8}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. We employ gradient norm clipping to a value of 1.0. We additionally employ a linear warmup schedule in the learning rate, from 0 to λ_0, over 5000 gradient

updates. A batch size of 4 was used for all runs. We generally train for 125 epochs. In our tests, we found that a type of overfitting would occur if we ran for longer, and we used dropout in the residual layers, with a probability of 0.5, to help alleviate this.

With respect to preprocessing the raw fields, we proceed as follows. We first split each data sample into a constant mean field and a field of deviations from the mean. Over these two components of the data, we carry out an independent minimum–maximum scaling, such that the minimum pixel value (over all of the preprocessed data) is -1 and the maximum (over all of the preprocessed data) pixel value is 1. We then add the two back together. The resulting dataset no longer has a minimum and maximum pixel value of exactly ± 1, but because the maximum and minimum values of the mean are not necessarily correlated with the fields that have the maximum and minimum spatial deviations from the mean, the distribution of pixels is still mostly contained within the $[-1, 1]$ range (and at worst, in the $[-2, 2]$ range).

Our main motivation for this preprocessing step is because the total vorticity is conserved, and so the distribution of the total vorticity is a delta function. Floating point error turns this into a Gaussian with a very small variance. Our preprocessing step then turns this into a much wider distribution, which will be easier to learn. However, we expect that this is beneficial in general given that means are handled by an independent neural network; this is akin to preprocessing the input of that network as is standard practice.

d. Sample generation

To generate all of the results shown here, we use the Euler–Maruyama (EM) method with a fixed time step for solving the stochastic differential equations. For the SDE,

$$dx = f(x, t)dt + g(x, t)dW, \tag{C18}$$

the update rule is as follows:

$$x(t + \Delta t) = x(t) + f(x, t)\Delta t + g(x, t)\eta\sqrt{\Delta t}, \tag{C4}$$

where $\eta \sim \mathcal{N}(0, 1)$. For all simulations, we use a fixed time step of 0.002, which corresponds to 500 steps from $t = \epsilon = 1 \times 10^{-5}$ to $t = 1$. Field generation was not the dominant computational cost for this project, so we did not explore varying the time-stepping algorithm or time step. Testing alternate time-stepping schemes is an activate area of research in the field.

APPENDIX D

Azimuthally Averaged Power Spectral Density

For each channel in the input data, we have a two-dimensional field of dimensions $N \times N$. We compute the discrete Fourier transform of the field,

$$\tilde{I}(k_x, k_y) = \sum_{x=-N/2}^{N/2-1} \sum_{y=-N/2}^{N/2-1} I(x, y)\exp[-i2\pi/N(k_x x + k_y y)]. \tag{D1}$$

The power spectrum for wavenumbers (k_x, k_y) is given by

$$\mathrm{PS}(k_x, k_y) = \frac{1}{N^4}\tilde{I}(k_x, k_y)\tilde{I}^*(k_x, k_y). \tag{D2}$$

This can be converted into a power spectral density $\mathrm{PSD}(k_x, k_y)$ by dividing by an area in wavenumber space (Youngworth et al. 2005). We may convert to polar coordinates (k, ϕ), where $k = \sqrt{k_x^2 + k_y^2}$. For isotropic flows, the expectation of $|\tilde{I}(k, \phi)|$ over different regions of the flow is independent of ϕ. This means that carrying out an integral of the azimuthal angle ϕ leads to no loss of information (in expectation). We can write the azimuthally averaged power spectral density as

$$\mathrm{PSD}(k) = \frac{\dfrac{1}{N^4}\displaystyle\int_0^{2\pi}\int_k^{k+1} \tilde{I}^*\tilde{I}k'\, dk'\, d\phi}{\displaystyle\int_0^{2\pi}\int_k^{k+1} k'\, dk'\, d\phi} \tag{D3}$$

$$\approx \frac{1}{N^4}\frac{\displaystyle\sum_{k_x}\sum_{k_y} \tilde{I}^*\tilde{I}\Theta[k^2 \le k_x^2 + k_y^2 < (k+1)^2]}{\displaystyle\sum_{k_x}\sum_{k_y} \Theta[k^2 \le k_x^2 + k_y^2 < (k+1)^2]}, \tag{D4}$$

where Θ(condition) is a function which returns 1 when the condition is true and 0 otherwise. This metric becomes less informative for fields of flows with preferred directions or inhomogeneities, in which case, the 2D Fourier transformed field itself may be more useful.

Our Algorithm 1 requires knowing the power spectrum for white noise. One can show that if $I(x, y) \sim \mathcal{N}(0, \sigma^2)$, $|\tilde{I}(k_x, k_y)^2| \sim \mathrm{Exp}[1/(\sigma^2 N^2)]$ when k_x or k_y is greater than zero. This has an expected value of $\sigma^2 N^2$. Plugging this into Eq. (D3), we see that the PSD(k) of Gaussian white noise is independent of wavenumber k and has an expected value of σ^2/N^2 for $k > 0$.[2]

APPENDIX E

Downscaling as Sampling from a Distribution

Our goal in downscaling is to sample from a distribution of high-resolution images conditional on a biased and low-resolution input. Sampling is a desirable feature since many high-resolution fields are consistent with any given low-resolution field and since statistics are often of most interest, compared with any given instantaneous realization of the dynamical system. To illustrate this, we created a synthetic low-resolution "pair" to a single real high-resolution data sample by low-pass filtering the fields and carried out the downscaling procedure 100 times. For three pixels across the fields, we compared the true high and low-resolution pixel values with the distribution of values obtained from the 100 downscaled ensemble members. The results for the

[2] When $k_x = k_y = 0$, $|\tilde{I}(0,0)^2|$ is not drawn from an exponential distribution. Instead, $|\tilde{I}(0,0)^2|/(\sigma^2 N^2) \sim \chi_1^2$, the chi-squared distribution. We subtract the means prior to computing the PSD, so $|\tilde{I}(0, 0)^2| \approx 0$.

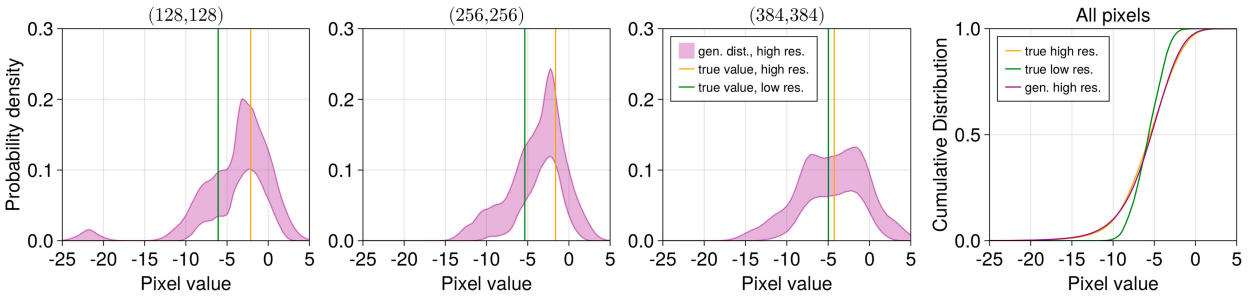

FIG. E1. Probability density function estimates for specific pixels of the supersaturation tracer field in the wavenumber = 16 data subset. The first three columns from the left correspond to different pixels, showing the distribution of values obtained from downscaling (purple), compared with the "true" value for that pixel in the original high-resolution sample (orange) and the "true" value for the low-resolution sample created by low-pass filtering the high-resolution truth (green). Shaded areas are computed from 10 000 bootstrap samples at the 99% confidence interval. In the rightmost column, we show the cumulative distribution functions for all 512^2 pixels in the real high- and low-resolution samples, as well as that computed using 512^2 randomly chosen pixels from an ensemble of downscaled fields.

supersaturation tracer variable are shown in Fig. E1. This shows that even in the case where there is a "correct answer," obtained with paired data, the algorithm is still sampling from a distribution. Because of this, the metrics that make the most sense are those that compare distributions or summary statistics. For example, Fig. E1 also shows the cumulative distribution for all of the pixel values in the real high-resolution data sample and its low-resolution pair and a randomly chosen subset of pixels from the generated high-resolution images.

REFERENCES

Abatzoglou, J. T., and T. J. Brown, 2012: A comparison of statistical downscaling methods suited for wildfire applications. *Int. J. Climatol.*, **32**, 772–780, https://doi.org/10.1002/joc.2312.

Anderson, B. D. O., 1982: Reverse-time diffusion equation models. *Stochastic Process. Appl.*, **12**, 313–326, https://doi.org/10.1016/0304-4149(82)90051-5.

Ballard, T., and G. Erinjippurath, 2022: Contrastive learning for climate model bias correction and super-resolution. arXiv, 2211.07555v1, https://doi.org/10.48550/arXiv.2211.07555.

Batzolis, G., J. Stanczuk, C.-B. Schönlieb, and C. Etmann, 2021: Conditional image generation with score-based diffusion models. arXiv, 2111.13606v1, https://doi.org/10.48550/arXiv.2111.13606.

Brown, B. C. A., A. L. Caterini, B. L. Ross, J. C. Cresswell, and G. Loaiza-Ganem, 2023: Verifying the union of manifolds hypothesis for image data. arXiv, 2207.02862v3, https://doi.org/10.48550/arXiv.2207.02862.

Chetrite, R., P. Muratore-Ginanneschi, and K. Schwieger, 2021: E. Schrödinger's 1931 paper "On the Reversal of the Laws of Nature" ["Über die Umkehrung der Naturgesetze", *Sitzungsberichte der preussischen Akademie der Wissenschaften, physikalisch-mathematische Klasse, 8* N9 144–153]. *Eur. Phys. J. H*, **46**, 28, https://doi.org/10.1140/epjh/s13129-021-00032-7.

Choi, J., J. Lee, C. Shin, S. Kim, H. Kim, and S. Yoon, 2022: Perception prioritized training of diffusion models. *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, IEEE Computer Society, 11 462–11 471.

Choi, Y., M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, 2018: StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, Institute of Electrical and Electronics Engineers, 8789–8797.

Constantinou, N. C., G. L. Wagner, L. Siegelman, B. C. Pearson, and A. Palóczy, 2021: GeophysicalFlows.jl: Solvers for geophysical fluid dynamics problems in periodic domains on CPUs & GPUs. *J. Open Source Software*, **6**, 3053, https://doi.org/10.21105/joss.03053.

——, and Coauthors, 2023: Fourierflows/fourierflows.jl: v0.10.3. Zenodo, https://doi.org/10.5281/zenodo.7631539.

Dhariwal, P., and A. Nichol, 2021: Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, Curran Associates, Inc., 8780–8794.

Fowler, H. J., S. Blenkinsop, and C. Tebaldi, 2007: Linking climate change modelling to impacts studies: Recent advances in downscaling techniques for hydrological modelling. *Int. J. Climatol.*, **27**, 1547–1578, https://doi.org/10.1002/joc.1556.

Giannone, G., D. Nielsen, and O. Winther, 2022: Few-shot diffusion models. arXiv, 2205.15463v1, https://doi.org/10.48550/arXiv.2205.15463.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, 2020: Generative adversarial networks. *Commun. ACM*, **63**, 139–144, https://doi.org/10.1145/3422622.

Groenke, B., L. Madaus, and C. Monteleoni, 2021: ClimAlign: Unsupervised statistical downscaling of climate variables via normalizing flows. *CI2020: Proc. 10th Int. Conf. on Climate Informatics*, New York, NY, Association for Computing Machinery, 60–66.

Gutmann, E., T. Pruitt, M. P. Clark, L. Brekke, J. R. Arnold, D. A. Raff, and R. M. Rasmussen, 2014: An intercomparison of statistical downscaling methods used for water resource assessments in the United States. *Water Resour.*, **50**, 7167–7186, https://doi.org/10.1002/2014WR015559.

Hagemann, S., C. Chen, J. O. Haerter, J. Heinke, D. Gerten, and C. Piani, 2011: Impact of a statistical bias correction on the projected hydrological changes obtained from three GCMs and two hydrology models. *J. Hydrometeor.*, **12**, 556–578, https://doi.org/10.1175/2011JHM1336.1.

Harris, L., A. T. McRae, M. Chantry, P. D. Dueben, and T. N. Palmer, 2022: A generative deep learning approach to

stochastic downscaling of precipitation forecasts. *J. Adv. Model. Earth Syst.*, **14**, e2022MS003120, https://doi.org/10.1029/2022MS003120.

He, K., X. Zhang, S. Ren, and J. Sun, 2016: Deep residual learning for image recognition. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, Institute of Electrical and Electronics Engineers, 770–778.

Ho, J., and T. Salimans, 2022: Classifier-free diffusion guidance. arXiv, 2207.12598v1, https://doi.org/10.48550/arXiv.2207.12598.

——, A. Jain, and P. Abbeel, 2020: Denoising diffusion probabilistic models. *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems*, Curran Associates Inc., 6840–6851.

——, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, 2022a: Video diffusion models. arXiv, 2204.03458v2, https://doi.org/10.48550/arXiv.2204.03458.

——, and Coauthors, 2022b: Imagen video: High definition video generation with diffusion models. arXiv, 2210.02303v1, https://doi.org/10.48550/arXiv.2210.02303.

Hoogeboom, E., and T. Salimans, 2022: Blurring diffusion models. arXiv, 2209.05557v2, https://doi.org/10.48550/arXiv.2209.05557.

Hwang, S., and W. D. Graham, 2014: Assessment of alternative methods for statistically downscaling daily GCM precipitation outputs to simulate regional streamflow. *J. Amer. Water Resour. Assoc.*, **50**, 1010–1032, https://doi.org/10.1111/jawr.12154.

Hyvärinen, A., and P. Dayan, 2005: Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.*, **6**, 695–709.

Kingma, D. P., and M. Welling, 2019: An introduction to variational autoencoders. *Found. Trends Mach. Learn.*, **12**, 307–392, https://doi.org/10.1561/2200000056.

Kong, Z., W. Ping, J. Huang, K. Zhao, and B. Catanzaro, 2021: DiffWave: A versatile diffusion model for audio synthesis. arXiv, 2009.09761v3, https://doi.org/10.48550/arXiv.2009.09761.

Luo, C., 2022: Understanding diffusion models: A unified perspective. arXiv, 2208.11970v1, https://doi.org/10.48550/arXiv.2208.11970.

Maurer, E. P., and H. G. Hidalgo, 2008: Utility of daily vs. monthly large-scale climate data: An intercomparison of two statistical downscaling methods. *Hydrol. Earth Syst. Sci.*, **12**, 551–563, https://doi.org/10.5194/hess-12-551-2008.

——, and D. W. Pierce, 2014: Bias correction can modify climate model simulated precipitation changes without adverse effect on the ensemble mean. *Hydrol. Earth Syst. Sci.*, **18**, 915–925, https://doi.org/10.5194/hess-18-915-2014.

Meng, C., Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, 2022: SDEdit: Guided image synthesis and editing with stochastic differential equations. arXiv, 2108.01073v2, https://doi.org/10.48550/arXiv.2108.01073.

O'Gorman, P. A., and T. Schneider, 2006: Stochastic models for the kinematics of moisture transport and condensation in homogeneous turbulent flows. *J. Atmos. Sci.*, **63**, 2992–3005, https://doi.org/10.1175/JAS3794.1.

Pan, B., G. J. Anderson, A. Goncalves, D. D. Lucas, C. J. W. Bonfils, J. Lee, Y. Tian, and H.-Y. Ma, 2021: Learning to correct climate projection biases. *J. Adv. Model. Earth Syst.*, **13**, e2021MS002509, https://doi.org/10.1029/2021MS002509.

Panofsky, H. A., G. W. Brier, and W. H. Best, 1958: *Some Application of Statistics to Meteorology*. Pennsylvania State University, 224 pp.

Papamakarios, G., E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, 2021: Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, **22**, 1–64.

Park, T., A. A. Efros, R. Zhang, and J.-Y. Zhu, 2020: Contrastive learning for unpaired image-to-image translation. *European Conference on Computer Vision: ECCV 2020*, Springer, 319–345.

Phillips, A., T. Seror, M. Hutchinson, V. D. Bortoli, A. Doucet, and E. Mathieu, 2022: Spectral diffusion processes. arXiv, 2209.14125v2, https://doi.org/10.48550/arXiv.2209.14125.

Pierce, D. W., and Coauthors, 2013: Probabilistic estimates of future changes in California temperature and precipitation using statistical and dynamical downscaling. *Climate Dyn.*, **40**, 839–856, https://doi.org/10.1007/s00382-012-1337-9.

——, D. R. Cayan, and B. L. Thrasher, 2014: Statistical downscaling using localized constructed analogs (LOCA). *J. Hydrometeor.*, **15**, 2558–2585, https://doi.org/10.1175/JHM-D-14-0082.1.

Price, I., and S. Rasp, 2022: Increasing the accuracy and resolution of precipitation forecasts using deep generative models. *Proc. 25th Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, Valencia, Spain, PMLR, 10 555–10 571.

Ramachandran, P., B. Zoph, and Q. V. Le, 2017: Searching for activation functions. arXiv, 1710.05941v2, https://doi.org/10.48550/arXiv.1710.05941.

Rissanen, S., M. Heinonen, and A. Solin, 2023: Generative modelling with inverse heat dissipation. arXiv, 2206.13397v7, https://doi.org/10.48550/arXiv.2206.13397.

Ronneberger, O., P. Fischer, and T. Brox, 2015: U-Net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*, Springer International Publishing, 234–241.

Saharia, C., W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi, 2022: Palette: Image-to-image diffusion models. *SIGGRAPH'22: ACM SIGGRAPH 2022 Conf. Proc.*, Vancouver, BC, Canada, Association for Computing Machinery, 1–10.

——, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, 2023: Image super-resolution via iterative refinement. *IEEE Trans. Pattern Anal. Mach. Intell.*, **45**, 4713–4726, https://doi.org/10.1109/TPAMI.2022.3204461.

Salathé, E. P., Jr., P. W. Mote, and M. W. Wiley, 2007: Review of scenario selection and downscaling methods for the assessment of climate change impacts on hydrology in the United States Pacific Northwest. *Int. J. Climatol.*, **27**, 1611–1621, https://doi.org/10.1002/joc.1540.

Sandwell, D., W. Smith, and J. Becker, 2014: SRTM30+ global 1-km Digital Elevation Model (DEM): Version 11: Land surface. Pacific Islands Ocean Observing System (PacIOOS), accessed 16 November 2023, http://pacioos.org/metadata/srtm30plus_v11_land.html.

Sohl-Dickstein, J., E. A. Weiss, N. Maheswaranathan, and S. Ganguli, 2015: Deep unsupervised learning using nonequilibrium thermodynamics. *Proc. 32nd Int. Conf. on Machine Learning*, Lille, France, JMLR, 2256–2265.

Song, Y., and S. Ermon, 2019: Generative modeling by estimating gradients of the data distribution. *NIPS'19: Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Curran Associates, Inc., 11 918–11 930.

——, and ——, 2020: Improved techniques for training score-based generative models. *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems*, Curran Associates. Inc., 12 438–12 448.

——, C. Durkan, I. Murray, and S. Ermon, 2021a: Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems (NeurIPS 2021)*, Curran Associates. Inc., 1415–1428.

——, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, 2021b: Score-based generative modeling through stochastic differential equations. arXiv, 2011.13456v2, https://doi.org/10.48550/arXiv.2011.13456.

——, L. Shen, L. Xing, and S. Ermon, 2022: Solving inverse problems in medical imaging with score-based generative models. arXiv, 2111.08005v2, https://doi.org/10.48550/arXiv.2111.08005.

Su, X., J. Song, C. Meng, and S. Ermon, 2023: Dual diffusion implicit bridges for image-to-image translation. arXiv, 2203.08382v4, https://doi.org/10.48550/arXiv.2203.08382.

Tancik, M., and Coauthors, 2020: Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems (NeurIPS 2020)*, Curran Associates. Inc., 7537–7547.

Tashiro, Y., J. Song, Y. Song, and S. Ermon, 2021: CSDI: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems (NeurIPS 2021)*, Curran Associates. Inc., 24 804–24 816.

Thrasher, B., E. P. Maurer, C. McKellar, and P. B. Duffy, 2012: Technical note: Bias correcting climate model simulated daily temperature extremes with quantile mapping. *Hydrol. Earth Syst. Sci.*, **16**, 3309–3314, https://doi.org/10.5194/hess-16-3309-2012.

Vincent, P., 2011: A connection between score matching and denoising autoencoders. *Neural Comput.*, **23**, 1661–1674, https://doi.org/10.1162/NECO_a_00142.

Wan, Z. Y., R. Baptista, Y.-f. Chen, J. Anderson, A. Boral, F. Sha, and L. Zepeda-Núñez, 2023: Debias coarsely, sample conditionally: Statistical downscaling through optimal transport and probabilistic diffusion models. arXiv, 2305.15618v2, https://doi.org/10.48550/arXiv.2305.15618.

Wang, X., R. Girshick, A. Gupta, and K. He, 2018: Non-local neural networks. *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, IEEE, 7794–7803.

Wang, Z., J. Chen, and S. C. H. Hoi, 2021: Deep learning for image super-resolution: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, **43**, 3365–3387, https://doi.org/10.1109/TPAMI.2020.2982166.

Wood, A. W., E. P. Maurer, A. Kumar, and D. P. Lettenmaier, 2002: Long-range experimental hydrologic forecasting for the eastern United States. *J. Geophys. Res.*, **107**, 4429, https://doi.org/10.1029/2001JD000659.

——, L. R. Leung, V. Sridhar, and D. P. Lettenmaier, 2004: Hydrologic implications of dynamical and statistical approaches to downscaling climate model outputs. *Climatic Change*, **62**, 189–216, https://doi.org/10.1023/B:CLIM.0000013685.99609.9e.

Wu, Y., and K. He, 2018: Group normalization. *European Conference on Computer Vision (ECCV)*, Springer, 3–19.

Youngworth, R. N., B. B. Gallagher, and B. L. Stamper, 2005: An overview of power spectral density (PSD) calculations. *Proc. SPIE*, **5869**, 206–216, https://doi.org/10.1117/12.618478.

Zhu, J.-Y., T. Park, P. Isola, and A. A. Efros, 2017: Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, Venice, Italy, Institute of Electrical and Electronics Engineers, 2242–2251.