SeasonCast: A Masked Latent Diffusion Model for Skillful Subseasonal-to-Seasonal Prediction

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Accurate weather prediction on the subseasonal-to-seasonal (S2S) scale is critical for anticipating and mitigating the impacts of climate change. However, existing data-driven methods struggle beyond the medium-range timescale due to error accumulation in their autoregressive approach. In this work, we propose SeasonCast, a scalable and skillful probabilistic model for S2S prediction. SeasonCast consists of two components, a VAE model that encodes raw weather data into a continuous, lower-dimensional latent space, and a diffusion-based transformer model that generates a sequence of future latent tokens given the initial conditioning tokens. During training, we mask random future tokens and train the transformer to estimate their distribution given conditioning and visible tokens using a per-token diffusion head. During inference, the transformer generates the full sequence of future tokens by iteratively unmasking random subsets of tokens. This joint sampling across space and time mitigates compounding errors from autoregressive approaches. The low-dimensional latent space enables modeling long sequences of future latent states, allowing the transformer to learn weather dynamics beyond initial conditions. SeasonCast performs competitively with leading probabilistic methods at the medium-range timescale while being 10× to 20× faster, and achieves state-of-the-art performance at the subseasonal-to-seasonal scale across accuracy, physics-based, and probabilistic metrics.

1 Introduction

Subseasonal-to-seasonal (S2S) prediction is crucial for disaster preparedness, resource management, and long-term planning [40, 31, 41, 7]. Yet, this timescale remains particularly difficult due to the dual importance of initial conditions, which drive short- and medium-range forecasts, and boundary conditions, which dominate seasonal and climate predictions [22, 23]. Numerical weather prediction (NWP) models have long been the backbone of S2S forecasting [31, 37, 38], but solving high-dimensional systems of differential equations incurs significant computational costs and restricts ensemble size. Recent deep learning approaches trained on reanalyses such as ERA5 [12, 13, 33, 34, 1, 18, 28] have surpassed operational IFS [39] in medium-range forecasting, but their performance in S2S remains limited [26] due to rapid error accumulation of their autoregressive designs.

We propose SeasonCast, a latent diffusion model for probabilistic S2S prediction. Our method first compresses raw weather fields into a continuous latent space with a VAE, then trains a masked transformer to model future token distributions [2, 44] with a per-token diffusion head. Unlike autoregressive methods, SeasonCast generates full spatiotemporal sequences by progressively unmasking tokens, thereby reducing error accumulation and jointly addressing initial- and boundary-condition problems. Our experiments on ChaosBench [26] show that SeasonCast achieves state-of-the-art performance across accuracy, physical consistency, and probabilistic metrics in comparison with leading numerical and AI methods.

Submitted to the Machine Learning and the Physical Sciences workshop (NeurIPS 2025).

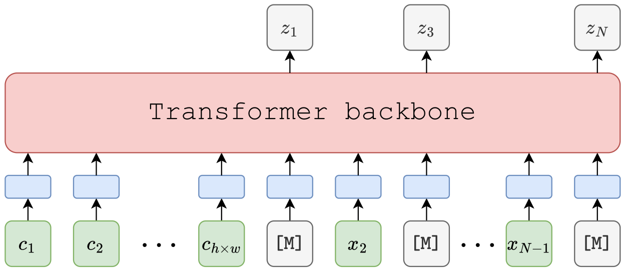


Figure 1: SeasonCast processes the latent tokens through a transformer backbone that outputs a vector z_i for each position i in the sequence.

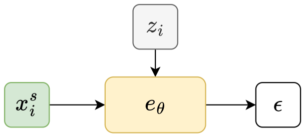


Figure 2: The denoising network e_θ predicts the noise ϵ from z_i and x_i^s.

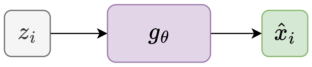


Figure 3: The deterministic network predicts directly x_i from z_i.

2 Methodology

Unlike the dominant autoregressive paradigm, which iteratively forecasts future states over short time intervals, SeasonCast is a generative model that learns the distribution of the entire sequence of future weather states $X_{1:T}$ based on initial conditions X_0. Our framework is inspired by masked generative modeling, a powerful approach for video generation. It consists of two main components: a VAE model that compresses raw weather data into a lower-dimensional latent space, and a masked generative transformer model that operates on these latent representations.

2.1 VAE for Weather Data Embedding

The first component of SeasonCast is a VAE that embeds a weather state $X \in \mathbb{R}^{V\times H\times W}$ into a latent map of $h \times w$ tokens, where $h < H$ and $w < W$. A common practice in masked generative modeling is to use vector-quantized VAEs, which discretize the latent space into a fixed-size vocabulary. However, this approach is problematic for weather data with hundreds of input channels, leading to an extreme compression requirement. For example, compressing 100-variable weather data with 32-bit values by $4\times$ spatially, using a discrete latent space with a vocabulary size of 8192 (13 bits per token), results in a compression ratio of approximately 3938. Such aggressive compression introduces substantial reconstruction errors, which degrade the performance of the subsequent generative model.

To overcome this, we adopt a continuous VAE model for SeasonCast, where each token in the $h \times w$ latent map is a continuous vector of dimension D. Using $D = 16$, for instance, reduces the compression ratio to 100, which results in significantly better reconstruction quality. While it is possible to compress the weather data in both spatial and temporal dimensions, our preliminary experiments showed no clear benefits from temporal compression, so we adopt a per-frame approach.

2.2 Masked Generative Modeling for S2S Prediction

The continuous VAE embeds the initial weather state X_0 into a sequence of continuous tokens $\mathbf{c} = (c_1, \ldots, c_{h\times w})$, and the sequence of future weather states $X_{1:T}$ into a sequence of future tokens $\mathbf{x} = (x_1, \ldots, x_N)$, where $N = T \times h \times w$. Each token x_i is a continuous vector of dimension D. The objective of our generative model is to learn the conditional distribution $p(\mathbf{x} \mid \mathbf{c})$. We achieve this using a masked generative framework, as illustrated in Figure 1. During training, we randomly sample a binary mask $\mathbf{m} = [m_i]_{i=1}^N$ and replace the tokens at masked positions ($m_i = 1$) with a learnable `[MASK]` token. We then train a bi-directional transformer to recover the original tokens at the masked positions, conditioned on both the unmasked future tokens and the initial condition tokens $\mathbf{c}$. The transformer processes the concatenation of conditioning tokens and corrupted future tokens, along with positional encodings, to produce a vector z_i for each masked position.

To model the continuous distribution of each masked token x_i, we employ a diffusion model where the transformer's output vector z_i serves as the conditioning information. This is implemented by a small MLP on top of the transformer that acts as the denoising network (Figure 2). We train the transformer backbone and the denoising network jointly using the diffusion loss: $\mathcal{L}_{\text{gen}}(\theta) = \mathop{\mathbb{E}}_{\mathbf{m}\sim p_{\mathcal{U}}} \left[\sum_{i \text{ s.t. } m_i=1} \mathcal{L}_{\text{diff}}(\theta)\right]$, where $\mathcal{L}_{\text{diff}}(\theta) = \mathbb{E}_{\epsilon,x}\left[\|\epsilon_\theta(x_s, s, z) - \epsilon\|^2\right]$. This joint objective encourages the model to produce representations z_i that facilitate effective denoising and thus accurate generation of the masked tokens.

Auxiliary Deterministic Objective. To further improve the accuracy of near-term forecasts, we introduce an auxiliary mean-squared error (MSE) loss applied to the first 10 days of future frames. A separate MLP head, trained jointly with the transformer, produces deterministic predictions $\hat{x}_i$ from the representations z_i. Because weather dynamics are more predictable in the short term, this loss is applied only to the first 10 frames and is weighted with an exponentially decreasing scheme to emphasize the importance of earlier frames. The complete training objective is the sum of the generative and auxiliary deterministic losses: $\mathcal{L}(\theta) = \mathcal{L}_{\text{gen}}(\theta) + \mathcal{L}_{\text{deter}}(\theta)$, where $\mathcal{L}_{\text{deter}}(\theta) = \mathop{\mathbb{E}}_{\mathbf{m}\sim p_{\mathcal{U}}} \left[\sum_{m_i=1} w(i)||x_i - \hat{x}_i||_2^2\right]$.

Sampling from SeasonCast. At inference time, we generate samples from the learned distribution $p(\mathbf{x} \mid \mathbf{c})$ through an iterative decoding process. This process begins with a fully masked sequence of future tokens. In each iteration, the transformer first processes the conditioning and corrupted future tokens to produce vectors z_i for each masked position. Next, a subset of masked positions is randomly selected for unmasking according to a predefined schedule. Finally, for each selected position, the diffusion model generates a continuous token x_i by conditioning on z_i and performing a fixed number of reverse diffusion steps. This iterative process continues until all future tokens are revealed. The generated tokens are then decoded back into the weather domain using the VAE decoder. We can generate an ensemble of forecasts by simply replicating the initial tokens and performing independent sampling for each copy, with hyperparameters such as the number of unmasking iterations, unmasking order, diffusion steps, and diffusion temperature controlling the generation process.

3 Experiments

We compare SeasonCast with state-of-the-art deep learning and numerical methods on medium-range weather forecasting and S2S prediction, using WeatherBench2 [34] (WB2) and ChaosBench [26] as benchmarks, respectively. We also conduct extensive ablation studies to assess the contribution of each component in SeasonCast, and evaluate its scalability under varying inference compute budgets.

Across both tasks, we train and evaluate SeasonCast on 69 variables from the ERA5 reanalysis dataset [13], including four surface-level variables – 2-meter temperature (T2m), 10-meter U and V wind components (U10, V10), and mean sea-level pressure (MSLP), as well as five atmospheric variables – geopotential (Z), temperature (T), U and V wind components, and specific humidity (Q), each at 13 pressure levels {50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 850, 925, 1000} hPa. For medium-range forecasting, we use native $0.25°$ resolution (721×1440 grids) and follow WB2 to train on years 1979–2018, validate on 2019, and test on 2020 using initial conditions at 00UTC and 12UTC. For S2S prediction, we downsample the data to $1.40625°$ (128×256 grids) and follow ChaosBench to train on 1979–2020, validate on 2021, and test on 2022 using 00UTC initializations.

3.1 SeasonCast for S2S prediction

Training and inference details We train a VAE that embeds each weather state of shape $69 \times 128 \times 256$ into a latent map of shape $1024 \times 8 \times 16$, reducing spatial dimensions by a factor of 16. The architectural details and training process of the VAE are described in Appendix B.1. We train SeasonCast to forecast a sequence of $T = 44$ future weather states at 24hr intervals, covering lead times from 1 to 44 days. Each training example consists of $45 \times 8 \times 16 = 5760$ latent tokens, including the initial condition. During inference, we generate the complete future sequence in 44 iterations (1 iteration per frame) using a diffusion temperature of $\tau = 1.3$. We produce an ensemble of 50 forecast sequences for each initial condition.

Baselines We compare SeasonCast with PanguWeather (PW) [1] and GraphCast (GC) [18], two leading open-sourced deep learning methods, and ensemble systems of four numerical models from different national agencies: UKMO-ENS (UK) [42], NCEP-ENS (US) [35], CMA-ENS (China) [43], and ECMWF-ENS (Europe) [9]. We refer to ChaosBench for details about these baselines. Following ChaosBench, we report results on T850, Z500, and Q700 at lead times from 1 to 44 days. We additionally compare SeasonCast with ClimaX [27] and Stormer [28] in Appendix C.3. We do not compare against Fuxi-S2S [4] as Fuxi-S2S forecasts daily average values from past daily averages, making it incomparable with SeasonCast and the rest of the methods, which perform point-in-time weather forecasting based on an initial condition. We are also not able to run Gencast [32] and NeuralGCM [17] for S2S due to their significant computational demands.

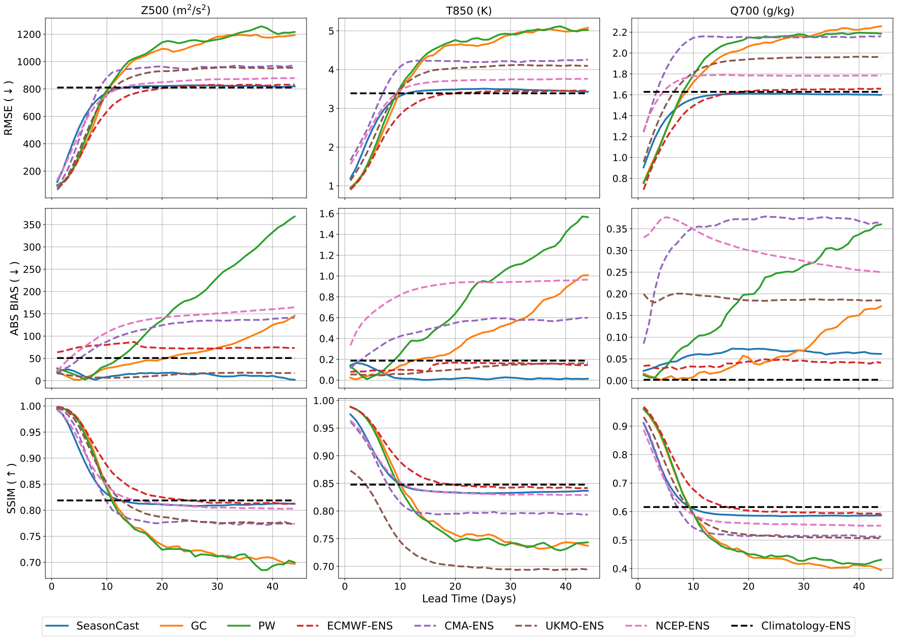


Figure 4: Deterministic performance of different methods at 1-44 days across three key variables. Solid denotes deep learning methods and dashed denotes numerical methods.

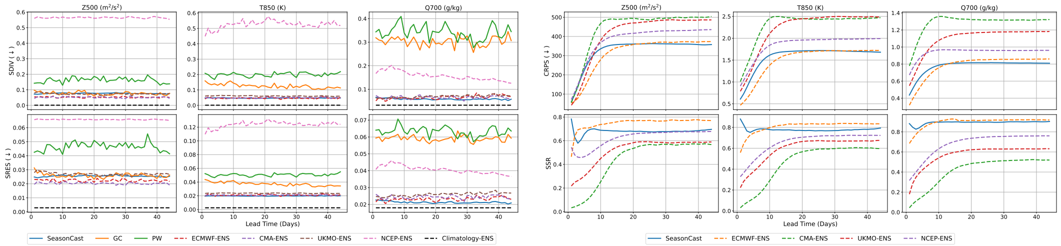


Figure 5: Physics (left) and probabilistic (right) metrics of different methods at 1-44 days across three key variables. Solid denotes deep learning methods and dashed denotes numerical methods.

Results Figure 4 compares different methods on three deterministic metrics: Root Mean-Squared Error (RMSE), Absolute Bias (ABS BIAS), and Multi-scale Structural Similarity (SSIM). At shorter lead times, SeasonCast shows slightly worse performance on RMSE and SSIM than other baselines, which is expected since we train SeasonCast to model a full sequence of future weather states rather than optimizing for short- and medium-range predictions. However, SeasonCast's relative performance improves with increasing lead time, ultimately matching ECMWF-ENS as one of the top two performing methods beyond day 10. Notably, SeasonCast demonstrates the lowest bias among all baselines, maintaining near-zero bias across all three target variables.

Physical consistency also plays a crucial role in S2S prediction, particularly for ensemble systems. We evaluate this aspect using two physics-based metrics: Spectral Divergence (SDIV) and Spectral Residual (SRES), which measure how closely the power spectra of predictions match those of ground-truths. As shown in Figure 5, SeasonCast achieves substantially better physical consistency than other deep learning methods, and often outperforms all baselines on these metrics. These results demonstrate how SeasonCast effectively preserves signals across the frequency spectrum.

Finally, we compare SeasonCast with the four numerical ensemble systems on two probabilistic metrics: Continuous Ranked Probability Score (CRPS) and Spread/Skill Ratio (SSR) (closer to 1 is better). Figure 5 shows that SeasonCast and ECMWF-ENS are the two leading methods across variables and lead times. Similar to deterministic results, SeasonCast performs worse than ECMWF-ENS at shorter lead times but outperforms this baseline beyond day 15.

References

[1] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 619(7970):533–538, 2023.

[2] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022.

[3] Kang Chen, Tao Han, Junchao Gong, Lei Bai, Fenghua Ling, Jing-Jia Luo, Xi Chen, Leiming Ma, Tianning Zhang, Rui Su, et al. Fengwu: Pushing the skillful global medium-range weather forecast beyond 10 days lead. *arXiv preprint arXiv:2304.02948*, 2023.

[4] Lei Chen, Xiaohui Zhong, Jie Wu, Deliang Chen, Shangping Xie, Qingchen Chao, Chensen Lin, Zixin Hu, Bo Lu, Hao Li, et al. Fuxi-s2s: An accurate machine learning model for global subseasonal forecasts. *arXiv preprint arXiv:2312.09926*, 2023.

[5] Lei Chen, Xiaohui Zhong, Feng Zhang, Yuan Cheng, Yinghui Xu, Yuan Qi, and Hao Li. FuXi: A cascade machine learning forecasting system for 15-day global weather forecast. *arXiv preprint arXiv:2306.12873*, 2023.

[6] Guillaume Couairon, Christian Lessig, Anastase Charantonis, and Claire Monteleoni. Archesweather: An efficient ai weather forecasting model at 1.5 {\deg} resolution. *arXiv preprint arXiv:2405.14527*, 2024.

[7] Daniela IV Domeisen, Christopher J White, Hilla Afargan-Gerstman, Ángel G Muñoz, Matthew A Janiga, Frédéric Vitart, C Ole Wulff, Salomé Antoine, Constantin Ardilouze, Lauriane Batté, et al. Advances in the subseasonal prediction of extreme events: Relevant case studies across the globe. *Bulletin of the American Meteorological Society*, 103(6):E1473–E1501, 2022.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] ECMWF. *IFS Documentation CY41R1 - Part V: The Ensemble Prediction System*. Number 5. ECMWF, 2015 2015. doi: 10.21957/eow1lonc. URL `https://www.ecmwf.int/node/9212`. <p> Operational implementation 12 May 2015</p>.

[10] Jayesh K Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized pde modeling. *arXiv preprint arXiv:2209.15616*, 2022.

[11] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[12] Hans Hersbach, Bill Bell, Paul Berrisford, Gionata Biavati, András Horányi, Joaquín Muñoz Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Iryna Rozum, Dinand Schepers, Adrian Simmons, Cornel Soci, Dick Dee, and Jean-Noël Thépaut. ERA5 hourly data on single levels from 1979 to present. *Copernicus Climate Change Service (C3S) Climate Data Dtore (CDS)*, 10(10.24381), 2018.

[13] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, , Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

[14] Jessica Hwang, Paulo Orenstein, Judah Cohen, Karl Pfeiffer, and Lester Mackey. Improving subseasonal forecasting in the western us with machine learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2325–2335, 2019.

[15] Ryan Keisler. Forecasting global weather with graph neural networks. *arXiv preprint arXiv:2202.07575*, 2022.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Dmitrii Kochkov, Janni Yuval, Ian Langmore, Peter Norgaard, Jamie Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter Düben, et al. Neural general circulation models for weather and climate. *Nature*, 632(8027):1060–1066, 2024.

[18] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 0(0):eadi2336, 2023. doi: 10.1126/science.adi2336. URL `https://www.science.org/doi/abs/10.1126/science.adi2336`.

[19] Simon Lang, Mark Rodwell, and Dinand Schepers. Ifs upgrade brings many improvements and unifies medium-range resolutions. *ECMWF Newsletter*, 176:21–28, 2023.

[20] Simon Lang, Mihai Alexe, Matthew Chantry, Jesper Dramsch, Florian Pinault, Baudouin Raoult, Mariana CA Clare, Christian Lessig, Michael Maier-Gerber, Linus Magnusson, et al. Aifs-ecmwf's data-driven forecasting system. *arXiv preprint arXiv:2406.01465*, 2024.

[21] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024.

[22] Edward N Lorenz. Forced and free variations of weather and climate. *Journal of Atmospheric Sciences*, 36(8):1367–1376, 1979.

[23] Annarita Mariotti, Paolo M Ruti, and Michel Rixen. Progress in subseasonal to seasonal prediction through a joint weather and climate community effort. *Npj Climate and Atmospheric Science*, 1(1):4, 2018.

[24] Soukayna Mouatadid, Paulo Orenstein, Genevieve Flaspohler, Miruna Oprescu, Judah Cohen, Franklyn Wang, Sean Knight, Maria Geogdzhayeva, Sam Levang, Ernest Fraenkel, et al. Subseasonalclimateusa: a dataset for subseasonal forecasting and benchmarking. *Advances in Neural Information Processing Systems*, 36, 2024.

[25] Congyi Nai, Xi Chen, Shangshang Yang, Yuan Liang, Ziniu Xiao, and Baoxiang Pan. Boosting weather forecast via generative superensemble. *arXiv preprint arXiv:2412.08377*, 2024.

[26] Juan Nathaniel, Yongquan Qu, Tung Nguyen, Sungduk Yu, Julius Busecke, Aditya Grover, and Pierre Gentine. Chaosbench: A multi-channel, physics-based benchmark for subseasonal-to-seasonal climate prediction. *arXiv preprint arXiv:2402.00712*, 2024.

[27] Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. ClimaX: A foundation model for weather and climate. *arXiv preprint arXiv:2301.10343*, 2023.

[28] Tung Nguyen, Rohan Shah, Hritik Bansal, Troy Arcomano, Romit Maulik, Veerabhadra Kotamarthi, Ian Foster, Sandeep Madireddy, and Aditya Grover. Scaling transformer neural networks for skillful and reliable medium-range weather forecasting. *arXiv preprint arXiv:2312.03876*, 2023.

[29] Joel Oskarsson, Tomas Landelius, Marc Peter Deisenroth, and Fredrik Lindsten. Probabilistic weather forecasting with hierarchical graph neural networks. *arXiv preprint arXiv:2406.04759*, 2024.

[30] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, Pedram Hassanzadeh, Karthik Kashinath, and Animashree Anandkumar. FourCastNet: A global data-driven high-resolution weather model using adaptive Fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

[31] Kathy Pegion, Ben P Kirtman, Emily Becker, Dan C Collins, Emerson LaJoie, Robert Burgman, Ray Bell, Timothy DelSole, Dughong Min, Yuejian Zhu, et al. The subseasonal experiment (subx): A multimodel subseasonal prediction experiment. *Bulletin of the American Meteorological Society*, 100(10):2043–2060, 2019.

[32] Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Timo Ewalds, Andrew El-Kadi, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. Gencast: Diffusion-based ensemble forecasting for medium-range weather. *arXiv preprint arXiv:2312.15796*, 2023.

[33] Stephan Rasp, Peter D Dueben, Sebastian Scher, Jonathan A Weyn, Soukayna Mouatadid, and Nils Thuerey. WeatherBench: a benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203, 2020.

[34] Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russel, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, Matthew Chantry, Zied Ben Bouallegue, Peter Dueben, Carla Bromberg, Jared Sisk, Luke Barrington, Aaron Bell, and Fei Sha. WeatherBench 2: A benchmark for the next generation of data-driven global weather models. *arXiv preprint arXiv:2308.15560*, 2023.

[35] Suranjana Saha, Shrinivas Moorthi, Xingren Wu, Jiande Wang, Sudhir Nadiga, Patrick Tripp, David Behringer, Yu-Tai Hou, Hui-ya Chuang, Mark Iredell, et al. The ncep climate forecast system version 2. *Journal of climate*, 27(6):2185–2208, 2014.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[37] Frédéric Vitart. Evolution of ecmwf sub-seasonal forecast skill scores. *Quarterly Journal of the Royal Meteorological Society*, 140(683):1889–1899, 2014.

[38] Frederic Vitart, Constantin Ardilouze, Axel Bonet, Anca Brookshaw, M Chen, C Codorean, M Déqué, L Ferranti, E Fucile, M Fuentes, et al. The subseasonal to seasonal (s2s) prediction project database. *Bulletin of the American Meteorological Society*, 98(1):163–173, 2017.

[39] NP Wedi, P Bauer, W Denoninck, M Diamantakis, M Hamrud, C Kuhnlein, S Malardel, K Mogensen, G Mozdzynski, and PK Smolarkiewicz. *The modelling infrastructure of the Integrated Forecasting System: Recent advances and future challenges*. European Centre for Medium-Range Weather Forecasts, 2015.

[40] Christopher J White, Henrik Carlsen, Andrew W Robertson, Richard JT Klein, Jeffrey K Lazo, Arun Kumar, Frederic Vitart, Erin Coughlan de Perez, Andrea J Ray, Virginia Murray, et al. Potential applications of subseasonal-to-seasonal (s2s) predictions. *Meteorological applications*, 24(3):315–325, 2017.

[41] Christopher J White, Daniela IV Domeisen, Nachiketa Acharya, Elijah A Adefisan, Michael L Anderson, Stella Aura, Ahmed A Balogun, Douglas Bertram, Sonia Bluhm, David J Brayshaw, et al. Advances in the application and utility of subseasonal-to-seasonal predictions. *Bulletin of the American Meteorological Society*, 103(6):E1448–E1472, 2022.

[42] KD Williams, CM Harris, A Bodas-Salcedo, J Camp, RE Comer, D Copsey, D Fereday, T Graham, R Hill, T Hinton, et al. The met office global coupled model 2.0 (gc2) configuration. *Geoscientific Model Development*, 88(55):1509–1524, 2015.

[43] Tongwen Wu, Yixiong Lu, Yongjie Fang, Xiaoge Xin, Laurent Li, Weiping Li, Weihua Jie, Jie Zhang, Yiming Liu, Li Zhang, et al. The beijing climate center climate system model (bcc-csm): The main progress from cmip5 to cmip6. *Geoscientific Model Development*, 12(4):1573–1600, 2019.

[44] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023.

A Related Work

Data-driven weather forecasting Deep learning has become a promising approach in the field of weather forecasting. Recent advancements with powerful architectures have achieved significant successes, providing faster inference and superior forecasting accuracy compared to IFS, the gold-standard numerical weather prediction system. Notable methods include FourCastNet [30], which utilizes an adaptive neural operator architecture; Keisler [15]'s, GraphCast [18], and AIFS [20], which leverage graph neural networks; and a series of transformer-based models such as PanguWeather [1], Stormer [28], and others [27, 5, 3, 6]. Beyond deterministic predictions, the field has increasingly focused on probabilistic forecasting to account for forecast uncertainty. Common approaches involve integrating existing architectures with generative frameworks, including diffusion models [32, 25], normalizing flows [6], and latent variable models [29]. Others explore ensemble predictions through initial condition perturbations, exemplified by methods like AIFS-CRPS [20] and NeuralGCM [17].

Data-driven S2S prediction Recent benchmarks have emerged to evaluate data-driven methods at S2S timescales. While many focus on regional forecasts such as the US [14, 24], ChaosBench [26] offers a comprehensive framework for global S2S prediction, providing extensive numerical baselines and physics-based metrics. A key finding from ChaosBench shows that state-of-the-art deep learning methods struggle to extend to S2S timescales. These methods predominantly rely on autoregressive approaches that generate predictions iteratively at short time intervals, leading to error accumulation with increasing lead times. While multi-step finetuning helps mitigate this issue for medium-range forecasts, it becomes computationally prohibitive for S2S predictions due to the extensive number of required forward passes. Moreover, training models with short time intervals fails to capture boundary conditions essential for long-term weather patterns. While Fuxi-S2S [4] was proposed for S2S prediction, it focuses on forecasting daily averaged statistics, which fundamentally alters the underlying weather dynamics and makes it inapplicable to forecasting at instantaneous time steps.

B Implementation details

Architectural details For the transformer backbone, we adopt the encoder-decoder architecture from Masked Autoencoder (MAE) [11]. The model processes an input sequence in two stages: first, the encoder processes the conditioning and visible tokens; second, the encoded sequence is augmented with learnable `[MASK]` tokens at appropriate positions and passed through the decoder to produce z_i for each position i. Both the encoder and decoder are bidirectional, employing full attention. Before feeding to either the encoder or decoder, we add the input sequences with positional embeddings that combine two components: temporal embeddings to distinguish different frames, and spatial embeddings to differentiate tokens within each frame. The encoder and decoder follow the Transformer [36] implementation in ViT [8], each having 16 layers with 16 attention heads, a hidden dimension of 1024, and a dropout rate of 0.1.

Mask sampling During training, we sample a masking ratio $\gamma \sim \mathcal{U}[0.5, 1.0]$ and generate a corresponding binary mask $\mathbf{m}$, where $\gamma = 0.75$ indicates that 75% of entries in $\mathbf{m}$ are 1. For inference, we start with full masking ($\gamma = 1.0$) and gradually reduce it to 0.0 with a cosine schedule [2]. We set the number of unmasking iterations to match the number of future weather states T. We employ random masking orders across both spatial and temporal dimensions for training and inference.

Diffusion loss details We use a linear noise schedule with 1000 steps at training time that are resampled to 100 steps at inference. The denoising network ϵ_θ is implemented as a small MLP following Li et al. [21]. Specifically, the network consists of six residual blocks, each comprising a LayerNorm (LN), a linear layer, a SiLU activation, and another linear layer, with a residual connection around the block. Each block maintains a width of 2048 channels. The network takes the vector z_i from the transformer as conditioning information, which is combined with the time embedding of the diffusion step s through adaptive layer normalization (AdaLN) in each block's LN layers.

B.1 VAE details

Our VAE model follows the UNet implementation from PDEArena [10]. We use the following hyperparameters for UNet in our experiments.

Table 1: Default hyperparameters of UNet

Hyperparameter	Meaning	Value
Padding size	Padding size of each convolution layer	1
Kernel size	Kernel size of each convolution layer	3
Stride	Stride of each convolution layer	1
Input channels	The number of channels of the input	69
Input channels	The number of channels of the output	69
Base channels	The base hidden dimension of the UNet	256
Channel multiplications	Determine the number of output channels for Down and Up blocks	$[1, 2, 4, 4, 8]$
Dimension of z	The dimension of the latent space	1024
Blocks	Number of blocks	2
Use attention	If use attention in Down and Up blocks	False
Dropout	Dropout rate	0.0

The VAE encoder embeds each weather state of shape $69 \times 128 \times 256$ to a latent map of shape $1024 \times 8 \times 16$, reducing the spatial dimensions by 16. We use a KL weight of $5e-5$ and optimize the VAE model with Adam [16] for 200 epochs with a batch size of 32, a base learning rate of $2e-4$, parameters $(\beta_1 = 0.9, \beta_2 = 0.95)$, and weight decay of $1e-5$. The learning rate follows a linear warmup for the first 20 epochs, followed by a cosine decay schedule for the remaining 180 epochs.

B.2 Weighted deterministic objective

In SeasonCast, we employ a weighted MSE objective to encourage accurate deterministic predictions for near-term frames. The objective is formulated as:

$$\mathcal{L}_{\text{deter}}(\theta) = \mathop{\mathbb{E}}_{\mathbf{m}\sim p_{\mathcal{U}}} \left[\sum_{m_i=1} w(i) ||x_i - \hat{x}_i||_2^2 \right], \tag{1}$$

where $w(i)$ is an exponentially decreasing weighting function. We compute this weight in three steps. First, for each token i, we determine its corresponding frame index $k = \lfloor \frac{i}{h \times w} \rfloor$, where $h \times w$ represents the spatial dimensions of each frame's latent map. Second, we assign weights to tokens based on their frame index: $w(i) = e^{-k} = e^{-\lfloor \frac{i}{h \times w} \rfloor}$, ensuring all tokens from the same frame receive equal weight. Third, we set $w(i) = 0$ for tokens beyond frame 10 and normalize the remaining weights to sum to one.

B.3 Optimization details

We optimize SeasonCast with AdamW [16] for 100 epochs with a batch size of 32, a base learning rate of $2e-4$, parameters $(\beta_1 = 0.9, \beta_2 = 0.95)$, and weight decay of $1e-5$. The learning rate follows a linear warmup for the first 10 epochs, followed by a cosine decay schedule for the remaining 90 epochs.

C Additional experiments

C.1 SeasonCast for medium-range forecasting

In addition to its strong performance on the S2S task, we demonstrate that SeasonCast also performs competitively at the medium-range timescale. We train a VAE model with a spatial downsampling ratio of 16, compressing each weather state of shape $69 \times 721 \times 1440$ into a latent representation of size $256 \times 45 \times 90$. We then train SeasonCast to predict two steps ahead at 12-hour intervals, following the setup of Gencast [32]. During inference, we use autoregressive sampling, recursively feeding the most recent predicted frame as the new initial condition until the target lead time is reached. We generate forecasts using a single sampling iteration per frame with a diffusion temperature $\tau = 1.0$, and produce an ensemble of 50 members.

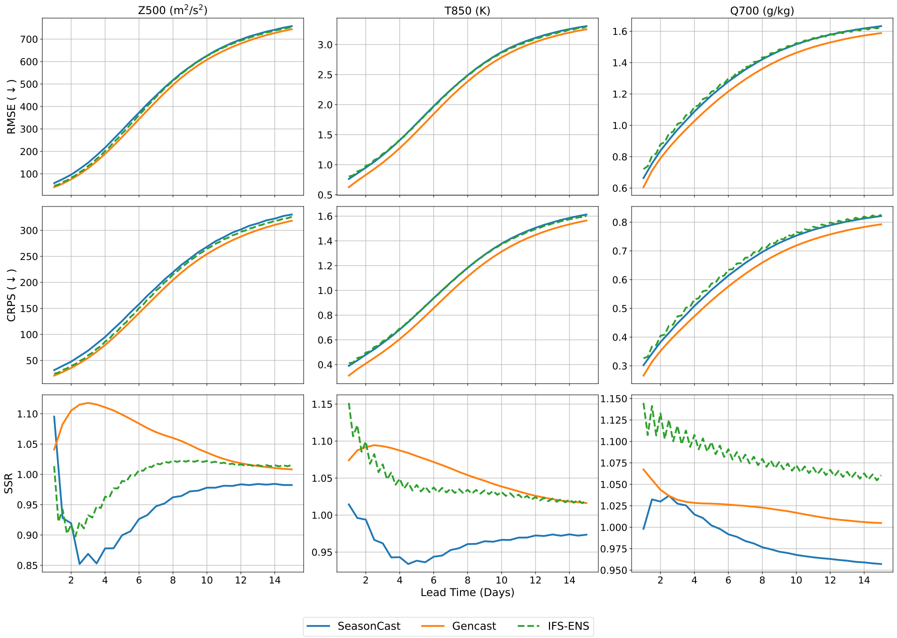


Figure 6: Probabilistic performance of different methods in medium-range forecasting. Solid curves are deep learning methods and dashed curves are numerical methods.

We compare SeasonCast against Gencast [32], a leading deep learning method for probabilistic forecasting, and IFS-ENS [19], the gold-standard numerical ensemble system. Following WeatherBench2, we use ensemble RMSE, CRPS, and spread-skill ratio (SSR) as evaluation metrics. Figure 6 shows that SeasonCast performs comparably with IFS-ENS across all variables and metrics, and is only slightly behind Gencast. Moreover, our analysis in Appendix C.2 further shows that SeasonCast is $10\times$ to $20\times$ faster than all baselines. These results indicate strong performance across both medium-range and S2S timescales of SeasonCast.

C.2 Efficiency of SeasonCast

Beyond its empirical performance, SeasonCast offers substantial efficiency gains over existing methods. We train SeasonCast for 4 days using 32 NVIDIA A100 GPUs. In comparison, Gencast requires 5 days of training on 32 TPUv5e devices – hardware significantly more powerful than A100s, and NeuralGCM [17] requires 10 days on 128 TPUv5e devices. Additionally, Gencast employs a two-stage training pipeline, first pretraining on $1.0°$ resolution and then finetuning on $0.25°$, while SeasonCast is trained in a single stage.

At inference time, SeasonCast is orders of magnitude faster than Gencast, NeuralGCM, and IFS-ENS. Figure 7 compares the runtime (in seconds) required to generate a 15-day forecast across different resolutions. At $0.25°$ resolution, Gencast requires 480 seconds on TPUv5, whereas SeasonCast achieves the same forecast in just 29 seconds on an A100. At $1.0°$, SeasonCast completes inference in only 11 seconds, compared to 224 seconds for Gencast on the same hardware. These results highlight the scalability and practicality of SeasonCast for operational forecasting.

The efficiency of SeasonCast stems from two key architectural innovations. First, SeasonCast operates in a much lower-dimensional latent space (45×90 latent grid vs 721×1440 original grid), significantly reducing the computational cost of training and inference. Second, SeasonCast employs a highly efficient sampling mechanism. Unlike Gencast, which performs 50 full forward passes through the entire network for 50 diffusion steps, SeasonCast requires only a single forward pass through the transformer backbone. The subsequent diffusion steps involve only lightweight forward passes through a compact MLP diffusion head, resulting in orders-of-magnitude lower inference time. Together, these design choices enable SeasonCast to deliver fast and scalable forecasts.

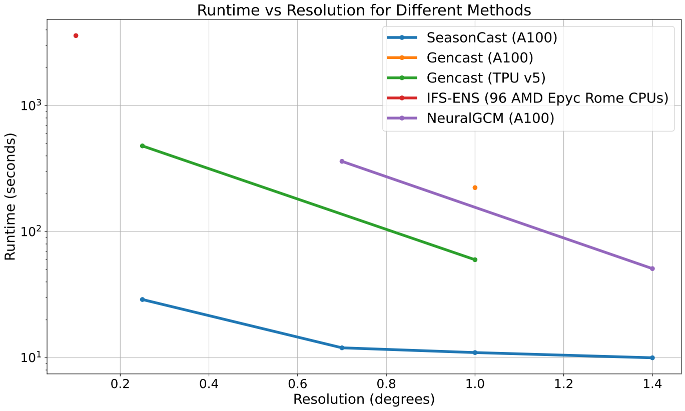


Figure 7: Runtime vs resolution of different methods to produce one forecast at 15-day lead time.

C.3 Comparison with more deep learning baselines

In addition to PanguWeather and GraphCast, we compare SeasonCast with two advanced transformer-based methods: ClimaX [27] and Stormer [28]. Figure 8 shows that Stormer achieves superior accuracy in short-to-medium timescales, consistent with its reported results. However, as an autoregressive method, its performance degrades more rapidly than SeasonCast, eventually falling below Climatology, albeit at a slower rate than PanguWeather and GraphCast. ClimaX takes a different approach as a direct forecasting method, where a model trained on large-scale climate data is finetuned specifically for individual lead times. This approach avoids error accumulation and achieves comparable performance with SeasonCast at S2S scales. However, ClimaX requires fine-tuning separate models for each target lead time, while a single SeasonCast model can simultaneously generate the complete sequence of future weather states.

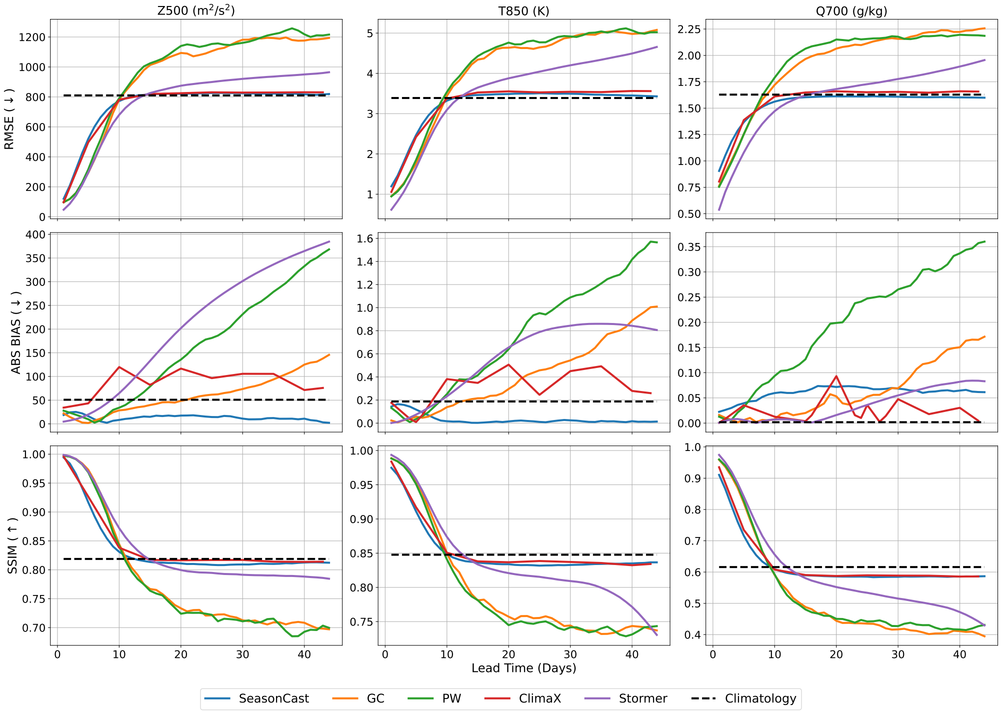


Figure 8: Comparison of deterministic performance of SeasonCast with more deep learning methods.

C.4 Impact of IC perturbations

Initial condition (IC) perturbations—adding random noise to initial conditions X_0 – are a standard technique in numerical methods for generating ensemble forecasts. This approach complements our generative framework. Figure 9 evaluates SeasonCast's performance across different noise levels, varying the standard deviation of the Gaussian distribution used for generating perturbations. The results demonstrate SeasonCast's robustness to input noise, maintaining consistent RMSE and CRPS scores across noise levels from 0.0 to 0.2, with only minor variations in SSR scores at short lead times.

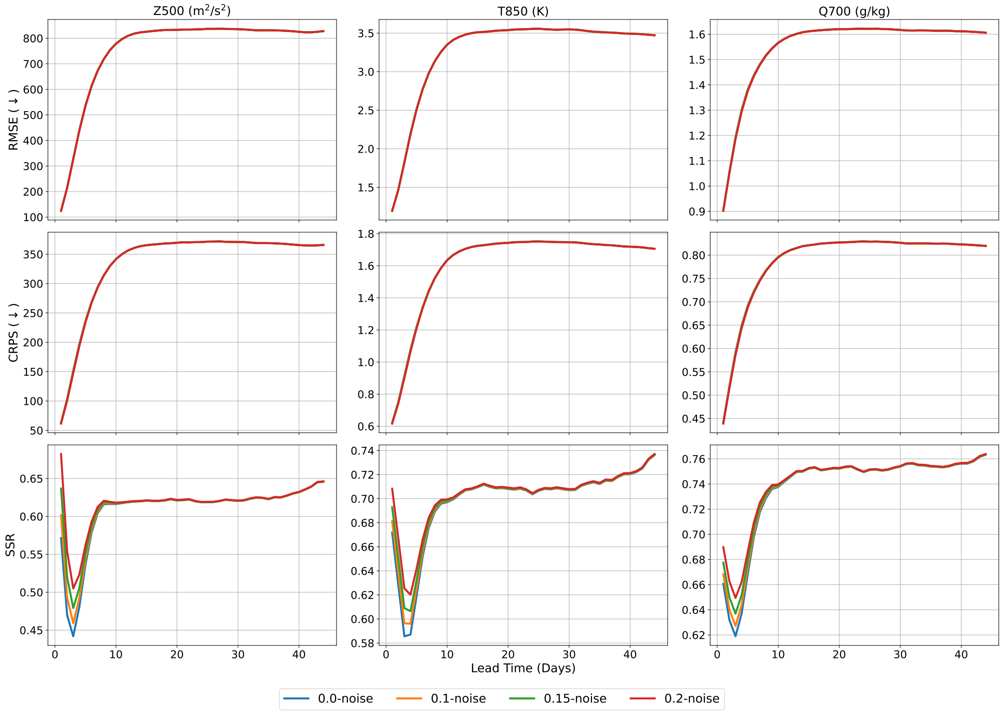


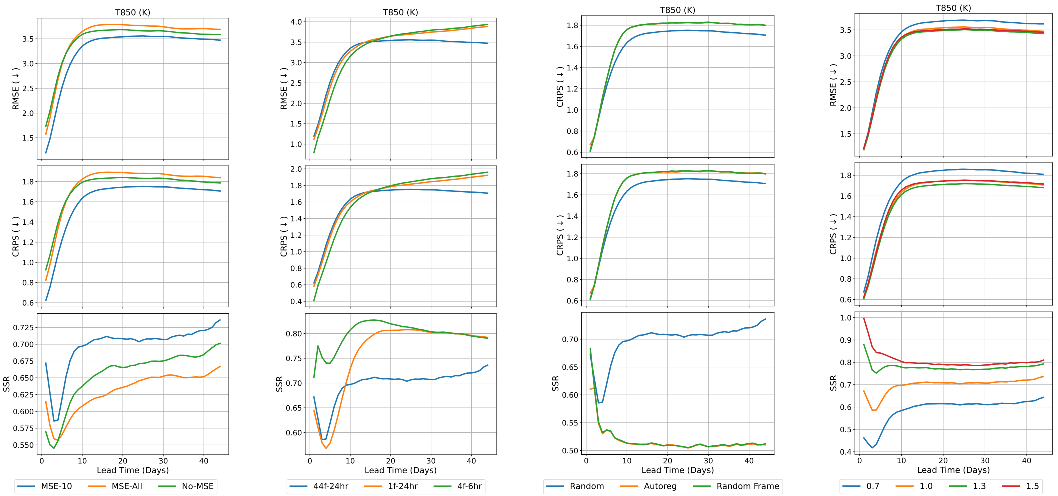
Figure 9: Performance of SeasonCast with different levels of IC noise.

C.5 Ablation studies

We analyze four key factors that influence SeasonCast's performance: the auxiliary deterministic objective, training sequence length T, unmasking order during sampling, and diffusion sampling temperature τ. We present results for T850 on RMSE, CRPS, and SSR. We additionally study the impact of IC perturbations in Appendix C.4.

Impact of the deterministic objective Figure 10a demonstrates the important role of the deterministic loss in SeasonCast's performance. Removing the MSE objective (No-MSE) degrades both RMSE and CRPS scores, with particularly noticeable impact at short lead times. However, naively applying MSE to all future frames (MSE-All-Frames) also proves counterproductive, as it forces deterministic predictions even for S2S timescales where weather systems become inherently chaotic. Our approach of applying MSE only to the first 10 frames achieves the best RMSE and CRPS scores across medium-range and S2S timescales.

Impact of training sequence length In our main experiments, we train SeasonCast to generate 44 future weather states at 24 hour intervals. One could alternatively train the model on shorter sequences and/or smaller intervals, then apply multiple roll-outs during inference to reach longer horizons, similar to autoregressive approaches. Figure 10b shows that models trained on shorter sequences or smaller intervals excel at short- and medium-range forecasting but underperform at S2S timescales. This trade-off emerges because shorter sequences allow models to specialize in near-term predictions, leading to better performance at shorter lead times. However, these models suffer from



(a) Impact of the deterministic objective.

(b) Impact of the training sequence length.

(c) Comparison of different unmasking strategies.

(d) Impact of diffusion sampling temperature τ.

Figure 10: Ablation studies showing the impact of different components in SeasonCast.

error accumulation at longer horizons, ultimately performing worse than the model trained on full sequences.

Impact of unmasking orders While our approach randomly masks tokens across both space and time during training, one may try more structured masking strategies at inference. We evaluate two such alternatives: an autoregressive strategy that unmasks entire frames sequentially, and a random framewise approach that unmasks complete frames in random order. Figure 10c shows that our fully randomized strategy achieves the best SSR scores, while both alternatives produce under-dispersive ensemble predictions. The superior performance of the fully randomized approach stems from its introduction of additional randomness through the fully random unmasking order, generating more diverse ensemble forecasts. This greater diversity consequently leads to better performance across other metrics.

Impact of diffusion sampling temperature The temperature τ controls the generation diversity, with higher values producing more diverse forecasts. Figure 10d demonstrates this relationship empirically. Setting $\tau < 1$ produces under-dispersive ensembles, degrading performance across other metrics. Increasing τ boosts sample diversity, improving SSR scores and overall better performance. However, pushing τ too high (e.g., $\tau = 1.5$) causes samples to deviate from the mean prediction, compromising RMSE and CRPS performance. We identify $\tau = 1.3$ as the optimal value, providing the best balance between ensemble diversity and forecast quality, which we adopt for our main experiments.

C.6 Scaling inference compute

Finally, we examine how increasing inference compute affects SeasonCast's performance through two hyperparameters: the number of ensemble forecasts and the average number of unmasking iterations per frame, i.e., 1-iter means a total of 44 iterations for 44 frames. Figure 11 shows that generating more ensemble forecasts improves both system diversity (higher SSR) and mean prediction accuracy (lower RMSE). Interestingly, while increasing the number of unmasking iterations shows minimal impact on RMSE, it yields slight improvements in SSR. This improvement likely stems from the increased randomness in unmasking order with more iterations, leading to greater ensemble diversity.

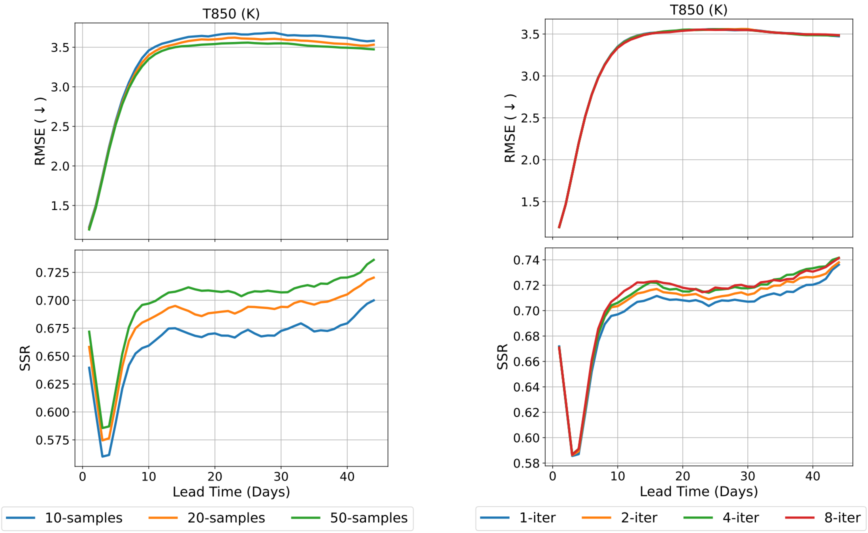


Figure 11: Performance of SeasonCast as we vary the number of ensemble forecasts (left) and the number of unmasking iterations.