Forecasting Fails: Unveiling Evasion Attacks in Weather Prediction Models

Huzaifa Arif[1], **Pin-Yu Chen**[2], **Alex Gittens**[1], **James Diffenderfer**[3], **Bhavya Kailkhura**[3]

[1]Rensselaer Polytechnic Institute, Troy, NY, United States
[2]IBM Research, Yorktown Heights, NY, United States
[3]Lawrence Livermore National Laboratory, Livermore, CA, United States
arifh@rpi.edu, pin-yu.chen@ibm.com, gittea@rpi.edu, diffenderfer2@llnl.gov, kailkhura1@llnl.gov

Abstract

With the increasing reliance on AI models for weather forecasting, it is imperative to evaluate their vulnerability to adversarial perturbations. This work introduces **Weather Adaptive Adversarial Perturbation Optimization (WAAPO)**, a novel framework for generating targeted adversarial perturbations that are both effective in manipulating forecasts and stealthy to avoid detection. **WAAPO** achieves this by incorporating constraints for channel sparsity, spatial localization, and smoothness, ensuring that perturbations remain physically realistic and imperceptible. Using the ERA5 dataset and FourCastNet (Pathak et al. 2022), we demonstrate **WAAPO**'s ability to generate adversarial trajectories that align closely with predefined targets, even under constrained conditions. Our experiments highlight critical vulnerabilities in AI-driven forecasting models, where small perturbations to initial conditions can result in significant deviations in predicted weather patterns. These findings underscore the need for robust safeguards to protect against adversarial exploitation in operational forecasting systems. The code for **WAAPO** is available at https://github.com/Huzaifa-Arif/WAPPO

Introduction

Recent research has focused extensively on the development of artificial intelligence models for weather prediction tasks, leading to the creation of advanced AI-based prediction models such as FourcastNet (Nvidia) (Pathak et al. 2022), GraphCast (Google) (Lam et al. 2022), ClimaX (Microsoft) (Huang et al. 2023), and PonguWeather (Huawei) (Bi et al. 2022). These models have demonstrated impressive accuracy and efficiency in weather forecasting, some of which can generate a 10-day forecast in just one minute (Rackow et al. 2024). Their accuracy often rivals that of traditional physics-based models, sparking significant interest from both private-sector companies and government agencies. The European Center for Medium-Range Weather Forecasts (ECMWF) and the National Oceanic and Atmospheric Administration (NOAA), which operates the Global Forecast System (GFS), have recognized the potential of these AI models. On 6 September 2023, ECMWF tweeted that three of its AI forecast models accurately predicted the slow westward movement of Hurricane Lee in the Atlantic Ocean, underscoring the practical application of these AI-driven forecasts (The Washington Post 2023).

Despite this success, limited research has been conducted to assess the vulnerability of these AI models to adversarial attacks. Traditional climate science institutes, such as ECMWF, employ rigorous protocols to ensure model reliability in prediction. Our study highlights a potential vulnerability by exploring whether these AI models could be susceptible to **adverse manipulation of initial weather fields**. We introduce a novel problem in the context of weather forecasting, specifically targeting the inference phase of weather forecasting models, demonstrating that they are highly vulnerable to input perturbations (see Figure 1).

Our study has significant implications, as creating false weather events or erasing or mistiming real ones could lead to serious consequences if these models are deployed without adversarial safeguards.

Adversarial Scenario in Weather Forecast Models

Climate forecasting relies on sensor data that are vulnerable to cyberphysical attacks (see Figure 1), enabling attackers to manipulate sensor data and inject inaccuracies into forecasting models. These models are often used by government-owned entities, such as NOAA. Even government-controlled sensors and models are not immune, as demonstrated by the following past incidents.

In September 2014, the National Oceanic and Atmospheric Administration (NOAA) experienced a significant cyberattack in which vulnerabilities in its IT security program were exploited. This attack compromised four NOAA websites and temporarily disrupted satellite imagery feeds, highlighting long-standing deficiencies in the NOAA cybersecurity infrastructure (Pagliery 2014). Similarly, in an unrelated application, in November 2024, China-linked hackers infiltrated multiple US telecommunications providers, extracting data on legal wiretaps and eavesdropping on government and political conversations. This breach, which affected roughly ten providers, including Verizon and AT&T, marked a substantial counterintelligence failure (Nakashima 2024).

These cases illustrate the tangible threat of cyberattacks on forecasting models, particularly as these models become more widely used and increasingly depend on data from in-

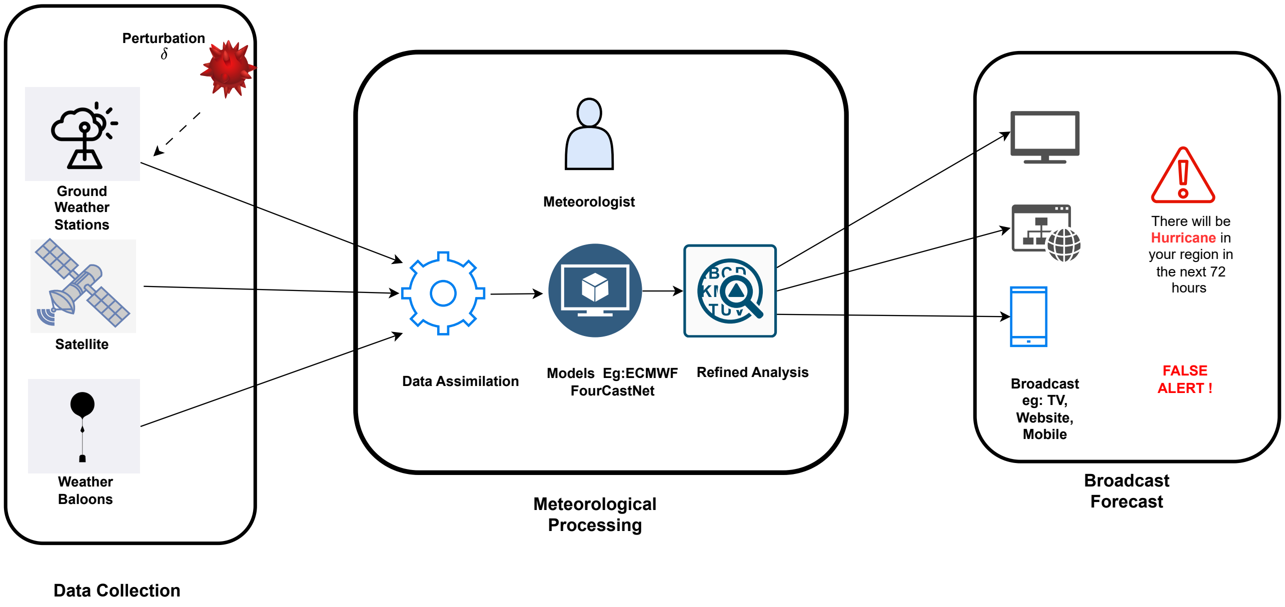


Figure 1: The weather forecasting process involves several key steps: **data collection, data assimilation, forecasting, analysis, and dissemination**. Data from sources like weather stations and satellites is processed through assimilation and forecasting models, refined through analysis, and shared with users via devices such as phones or TVs. Our study highlights vulnerabilities in this process, showing that adversaries can exploit the data collection phase to introduce perturbations and generate targeted false forecasts.

dependent sources, thereby expanding the attack surface. Attackers could exploit less secure third-party data providers to manipulate model inputs, potentially leading to inaccurate weather predictions. Such inaccuracies could result in economic losses, disrupted logistics, and public safety risks.

Therefore, vigilance and robust cybersecurity measures are critical to safeguarding forecasting systems against these potential threats.

Related Works

Typically, the literature focuses on adversarial attacks, where the target model is a classifier (Costa et al. 2024). In such settings, the goal of the adversary is defined as:

$$\arg\min_{\delta\mathbf{X}} \|\delta\mathbf{X}\| \quad \text{s.t.} \quad f(\mathbf{X}+\delta\mathbf{X}) = \mathbf{Y}^*, \tag{1}$$

$$\mathbf{X}^* = \mathbf{X} + \delta\mathbf{X}, \tag{2}$$

where f is the target classifier and $\mathbf{Y}^*$ is the target class defined by the adversary. For a clean example $(\mathbf{X}, \mathbf{Y})$, the adversary seeks a perturbation $\delta\mathbf{X}$ that results in the output $(\mathbf{X}+\delta\mathbf{X}, \mathbf{Y}^*)$. To maintain imperceptibility, the perturbation is constrained, typically using ℓ_∞ or ℓ_2-norms. This ensures that the perturbed input appears similar to the clean input, while causing the classifier to misclassify with high confidence. A visual depiction is provided in Figure 6, reproduced from (Akhtar and Mian 2018).

Numerous approaches to executing adversarial attacks in the classification setting have been proposed in the literature. Common methods include the box-constrained L-BFGS (Long et al. 2022), Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2014), and the Carlini and Wagner (C&W) attack (Carlini and Wagner 2017). Additional techniques are summarized in the surveys (Costa et al. 2024; Akhtar and Mian 2018).

With the rise of generative models, there has been a growing interest in exploring adversarial attacks and defenses in this domain (Kos, Fischer, and Song 2018; Rathore et al. 2020). These attacks are typically categorized as *untargeted* adversarial attacks (Belkhouja and Doppa 2022):

$$\{\mathbf{X}_{\text{adv}} \mid \|\mathbf{X}_{\text{adv}} - \mathbf{X}\|_p \leq \epsilon \text{ and } f_\theta(\mathbf{X}) \neq f_\theta(\mathbf{X}_{\text{adv}})\} \tag{3}$$

or *targeted* adversarial attacks (Liu et al. 2022), where t_{adv} is a predefined adversarial target:

$$\{\mathbf{X}_{\text{adv}} \mid \|\mathbf{X}_{\text{adv}} - \mathbf{X}\|_p \leq \epsilon \text{ and } f_\theta(\mathbf{X}_{\text{adv}}) = t_{\text{adv}}\}\,. \tag{4}$$

In this setting, an autoregressive model is repeatedly queried with some initial input over a fixed time horizon. While adversarial attacks have been extensively studied in tasks like time-series forecasting, their application to weather forecasting remains largely unexplored in the literature. This work aims to address this gap by introducing a study of *targeted* adversarial attacks specifically designed for weather forecasting models. Our analysis is conducted in the **white-box** setting, where the adversary has complete knowledge of the forecasting model and its parameters.

The objective of this novel adversarial attack strategy is to perturb the initial conditions of a weather forecasting model to cause significant deviations in the predicted trajectories for manipulating extreme weather events, such as hurricanes, toward some predefined target event. Importantly, these perturbations must satisfy constraints of physical realism and stealth, ensuring they remain relatively imperceptible, undetectable to unbounded perturbations and do not violate physical laws of weather fields.

Our experimental setup employs the state-of-the-art FourCastNet (Pathak et al. 2022; Nguyen et al. 2024) as the weather forecasting model, using the widely adopted ERA5 dataset for evaluation. The results demonstrate the effectiveness of this adversarial evasion attack in influencing weather forecasts by employing physically plausible and stealthy perturbations.

Problem Statement

An adversary can introduce subtle, realistic modifications to the **initial** weather conditions, mirroring plausible real-world scenarios, to manipulate weather forecasts toward desired outcomes. The practical implications of such perturbations are detailed in Appendix. For instance, if the adversary aims to create a **false temperature** event, they might adjust temperature profiles to trigger alerts for heat measures in a targeted region. By altering initial temperature fields, they could change the predicted temperature distribution across a region, affecting forecasts for heatwaves or cold spells. Each of these objectives relies on small, carefully controlled adjustments to particular atmospheric variables, allowing the adversary to steer forecasts toward their desired outcomes.

Table 1 provides an overview of the notations used in this paper. The autoregressive AI forecast model (e.g., FourCastNet (Pathak et al. 2022)) is denoted by ϕ, which generates a trajectory of forecasts $\mathbf{Z} \in \mathbb{R}^{T\times L\times M\times N}$ over a time horizon T. Here, N represents the prognostic variables (e.g., Temperature, Surface Wind Speed, etc.; see Table 4 for more details on the prognostic variables used by FourCastNet). L and M denote the number of latitude and longitude points, respectively (e.g., $L = 721$, $M = 1440$ for FourCastNet (Pathak et al. 2022)).

We use the subscript i to indicate the forecast at a specific time point. For instance, $\mathbf{Z}_3$ represents the model's third prediction for a given initial condition. The initial condition $\mathbf{Z}_0 \in \mathbb{R}^{L\times M\times N}$ is provided as input to $\phi(\cdot)$, which then generates the sequence of predictions $\mathbf{Z}_{1:T}$ which for simplicity is replaced by $\mathbf{Z}$ when discussing the whole forecast trajectory.

In this study, we address the challenge of designing a perturbation $\delta \in \mathbb{R}^{L\times M\times N}$ applied to the initial conditions $\mathbf{Z}_0$ to manipulate a weather forecast model $\phi(\cdot)$ (e.g., ForecastNet). Specifically, our objective is to minimize the squared ℓ_2-norm difference between the model's forecast at time T, denoted as $\mathbf{Z}_T = \phi_T(\mathbf{Z}_0 + \delta)$, and a desired adversarial future event t_{adv}. Here the subscript T of the model ϕ denotes the prediction of the model at the T^{th} timestep. Given a predetermined trajectory length T, the goal is to design the perturbation δ such that, when added to the initial condition, it causes the forecast to deviate significantly within the time frame T, making it resemble t_{adv}. This approach constitutes a **targeted** evasion attack for weather forecast task.

Table 1: Definitions of Variables

Symbol	**Description**
$\phi(\cdot)$	Weather forecast model (e.g., ForecastNet)
$\mathbf{Z}$	Set of T future forecast values, of shape $T\times L\times M\times N$
$\mathbf{Z}_0$	Initial condition, of shape $L \times M \times N$
δ	Perturbation applied to the initial conditions
t_{adv}	Desired adversarial event, of shape $L \times M \times N$
L	Longitude
M	Latitude
N	Number of prognostic variables (e.g., temperature, wind speed)

Are Weather Forecasts Secure to Unconstrained Attacks?

FourCastNet (Pathak et al. 2022) is evaluated on an ensemble of perturbed initial conditions. These initial conditions are generated by adding scaled Gaussian noise to the unperturbed state, where $\delta_{ijk} \sim \mathcal{N}(0, 1)$. Specifically, different ensembles are created as $\mathbf{Z}'_0 = \mathbf{Z}_0 + \sigma \cdot \delta$ with $\sigma = 0.3$. The resulting trajectories are produced as $\phi(\mathbf{Z}'_0)$. The paper shows that the ensemble mean predictions of FourCastNet closely follow the unperturbed control forecast, as assessed by the chosen evaluation metrics. This indicates that small Gaussian perturbations to the initial conditions do not cause significant deviations in the predictions, demonstrating the model's insensitivity to *i.i.d. random* minor variations in the initial weather fields.

However, to fully assess the robustness of FourCastNet to different inputs, it is essential to evaluate its performance under more targeted and potentially larger perturbations δ. We begin by constructing an attack aimed at solving optimization problem Equation (5), where no restrictions are placed on the initial perturbation:

$$\delta^* = \min_{\delta\in\mathbb{R}^{L\times M\times N}} \|\phi_T(\mathbf{Z}_0 + \delta) - t_{\text{adv}}\|_2^2 \tag{5}$$

The perturbation derived from solving Equation (5) influences all the prognostic variables listed in Table 4 (see Appendix), thereby altering the model's initial conditions across multiple atmospheric fields. Figure 2 exemplifies this effect on the temperature field. In this test case, the model is tasked with predicting conditions 24 hours into the future, while the adversarial target t_{adv} is set to reflect conditions at 120 hours. Such manipulations could yield serious real-world consequences. For instance, artificially elevated temperatures in the forecast might prompt authorities to prepare for a heatwave that will never materialize, potentially wasting resources and generating undue alarm. Conversely, underestimating temperatures could obscure an imminent heat risk, leaving communities unprepared and vulnerable to harm.

To visualize the distortion, Figure 2 compares the pointwise global temperature differences $\phi_T(\mathbf{Z}_0 + \delta^*) - t_{\text{adv}}$ and $\phi_T(\mathbf{Z}_0 + \delta^*) - \text{GT}$, where GT is the true temperature field at

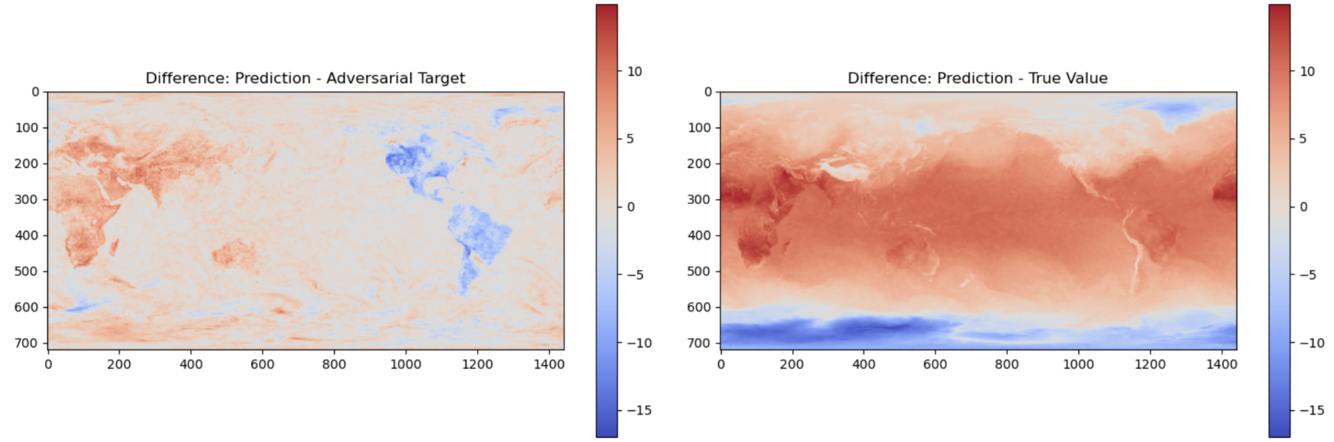
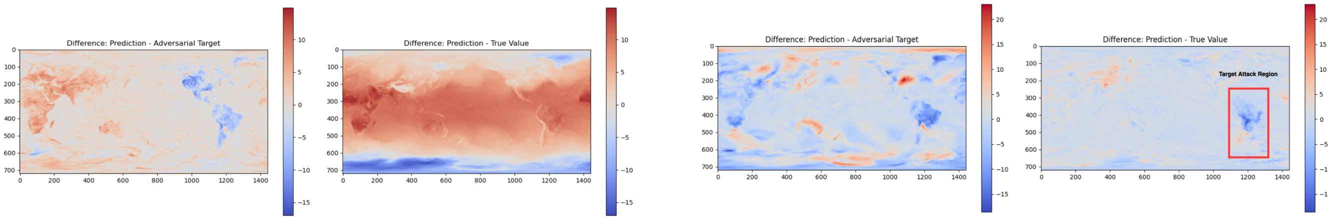

Difference: Prediction - Adversarial Target
Difference: Prediction - True Value
0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200
1400
10
5
0
−5
−10
−15

Figure 2: Pointwise temperature differences (in Kelvin) comparing the 24-hour perturbed forecast with both the 120-hour **adversarial target** (t_{adv}) and the 24-hour **ground truth (GT/True Value)**. Despite the adversarial target representing conditions 120 hours into the future, the unconstrained attack effectively manipulates the model's prediction to align more closely with the adversarial target than the actual 24-hour ground truth. This highlights the attack's capability to fabricate a false global temperature event, significantly overriding the model's original forecast.


Difference: Prediction - Adversarial Target
Difference: Prediction - True Value
Difference: Prediction - Adversarial Target
Difference: Prediction - True Value
Target Attack Region
20
15
10
5
0
−5
−10
−15

(a) WAAPO applied to only the **temperature (t2m)** channel.

(b) WAAPO with a **spatial mask** over South America.

Figure 3: These results illustrate the **Weather Adaptive Adversarial Perturbation Optimization (WAAPO)** framework from Algorithm 1. In **(a)**, WAAPO targets only the temperature ($t2m$) channel (Kelvin), showing that even single-channel perturbations can significantly alter predictions, closely mirroring the behavior seen in Figure 2. In **(b)**, a spatial mask M is applied over South America, demonstrating that localized, channel-specific perturbations can substantially reshape forecasts within the targeted region.

24 hours. Lighter regions correspond to smaller deviations, making it clear that the perturbed prediction adheres more closely to the adversarial target than to the authentic ground truth. To observe the individual temperature forecasts for the ground truth, unperturbed prediction, and perturbed prediction refer to Figure 5 (Appendix).

These results demonstrate that weather prediction models are inherently non-robust to arbitrary perturbations. However, such a *naive* attack is impractical in real-world scenarios, as a meteorologist could readily detect discrepancies in the initial data by analyzing the divergent trajectories of various fields. In the following section, we investigate whether incorporating a layer of imperceptibility into the attack can achieve comparable results while enhancing the realism and practical applicability of these perturbations in controlled settings.

Are Weather Forecast Models Secure to Localized Attacks on a Single Field?

Building on the previous section, we extend our approach to ensure *systemic stealthiness* in **targeted** adversarial attacks inspired by the approach as in Equation 4. This is crucial because trajectory observers (e.g., meteorologist) may detect data corruption if the perturbed trajectories diverge significantly from expected behavior or if validation against ground truth fails during a pilot run. The objective is to identify a *stealthy* perturbation, δ, that modifies the initial input $\mathbf{Z}_0$ to create an adversarial example while adhering to stealth constraints. We impose stealthiness through the following criteria: First, the perturbation should affect **only a minimal subset of channels**, representing correlated variables, to align with the adversary's goal of targeting specific events (e.g., temperature) while limiting observable discrepancies. Second, the attack **must be localized to a specific spatial region**, such as simulating a heatwave warning confined to a targeted area. This adds to obfuscating the observable discrepancy for a chosen targeted variable. Third,

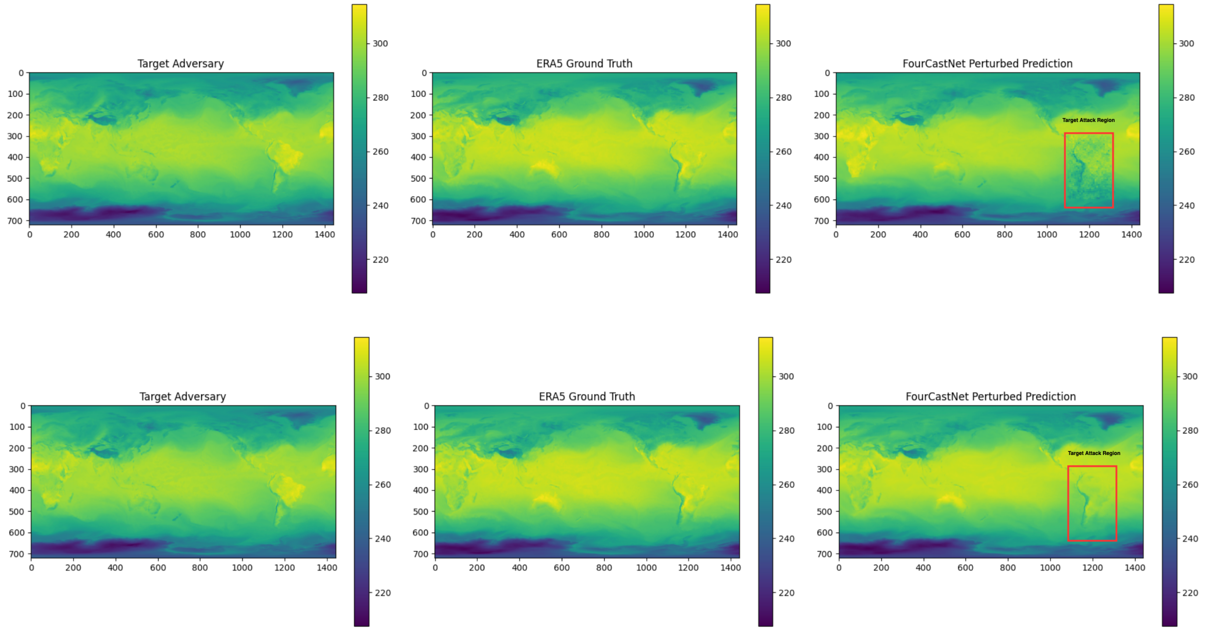


Figure 4: This figure demonstrates why a *smoothness constraint* is essential for creating imperceptible perturbations using **WAPPO**. The top row, without smoothness constraints, displays a clearly visible patch in the targeted area, whereas applying a smoothness constraint in the bottom row yields more diffused, subtle perturbations that blend naturally and are harder to detect.

the perturbation **must maintain physical realism** to avoid producing values that fall outside realistic bounds and could easily signal tampering. Finally, **transitions introduced by the perturbation must be smooth**, avoiding abrupt or spiky changes that could highlight corrupted data from sensors. These constraints collectively ensure that adversarial perturbations remain both effective and undetectable within real-world operational settings.

To impose the channel sparsity constraint, only a subset of channels $\mathbf{C}$, where $\mathbf{C} \subseteq \{1, 2, \ldots, N\}$, is perturbed. For the localization constraint, the perturbations are confined to a spatial mask $\mathbf{M}$.

To achieve these constraints, a projection operator $\mathcal{P}_{\mathbf{M},\mathbf{C}}$ is employed. The perturbed input $\mathbf{Z}_0^\delta$ is defined as $\mathbf{Z}_0 + \delta$, where $\delta = \mathcal{P}_{\mathbf{M},\mathbf{C}}(\delta)$. Each component of δ is constrained such that channels not in $\mathbf{C}$ are zeroed out, while those in $\mathbf{C}$ are further localized by the mask $\mathbf{M}$.

Additionally, the perturbations must generate forecast trajectories that are smooth and realistic, ensuring that the target fields remain within allowable limits. To impose these constraints, we utilize an adaptive loss objective that balances the primary objective with penalty terms for smoothness and realism. The primary objective, L_{primary}, minimizes the difference between the perturbed forecast and the adversarial target t_{adv}:

$$L_{\text{primary}} = \|\phi_T(\mathbf{Z}_0 + \delta) - t_{\text{adv}}\|_2^2$$

Two penalty terms encourage additional constraints: L_∞, which limits the maximum allowable values in the perturbed trajectories, and L_{TV}, which minimizes the total variation to promote spatial smoothness. Each channel $n \in \{1, \ldots, N\}$ is assigned a distinct penalty weight, $\lambda_{\infty,n}$ and $\lambda_{\text{TV},n}$, along with maximum allowable limits, ϵ_n, and smoothness parameters, τ_n. As previously defined, $\mathbf{Z}_t$ is the t^{th} prediction in a trajectory of length T which must obey these allowable and smoothness constraints:

$$L_\infty = \sum_{t=0}^{T-1} \sum_{n=1}^{N} \lambda_{\infty,n} \cdot \max\left(0, \|(\mathbf{Z}_t)_n\|_\infty - \epsilon_n\right)$$

$$L_{\text{TV}} = \sum_{t=0}^{T-1} \sum_{n=1}^{N} \lambda_{\text{TV},n} \cdot \max\left(0, \text{TV}\left((\mathbf{Z}_t)_n\right) - \tau_n\right)$$

The total loss function combines these terms to guide the optimization of δ, ensuring the perturbations are both realistic and stealthy:

$$L(\delta) = L_{\text{primary}} + L_\infty + L_{\text{TV}}.$$

To produce stealthy adversarial perturbations, we introduce **Weather Adaptive Adversarial Perturbation Optimization (WAAPO)**, summarized in Algorithm 1. WAAPO operates by iteratively updating the perturbation δ through gradient-based optimization. At each iteration, it computes the gradient of the total loss $L(\delta)$ with respect to δ, which includes the primary objective L_{primary} (ensuring the prediction aligns with the adversarial target t_{adv}) and two penalty terms, L_∞ and L_{TV}, previously discussed, that collectively enforce physical realism, boundedness, and smoothness.

The algorithm then performs a gradient descent step to refine δ. A subsequent projection step applies the channel and spatial masks $\mathbf{C}$ and $\mathbf{M}$, ensuring that only designated variables and regions are perturbed. By iterating this process until convergence or a fixed number of iterations is reached, WAAPO produces localized, channel-specific, and physically plausible adversarial examples.

Algorithm 1: Weather Adaptive Adversarial Perturbation Optimization for Weather Forecast (WAAPO)

```
Input  : Initial input $\mathbf{Z}_0$, target $t_{\text{adv}}$, learning rate $\alpha$, max iterations
         $K$, channels $\mathbf{C}$, spatial mask $\mathbf{M}$, penalties $\lambda_{\infty,n}$, $\lambda_{\text{TV},n}$,
         constraints $\epsilon_n$, $\tau_n$
Output: Optimized perturbation $\delta_{WAPPO}$
Initialize perturbation: $\delta^{(0)} \leftarrow \mathbf{0}$  for $k = 0$ to $K - 1$ do
    Compute Gradient:
     $\nabla_\delta L(\delta^{(k)}) \leftarrow \frac{\partial}{\partial \delta}\Big[\|\phi_T(\mathbf{Z}_0 + \delta^k) - t_{\text{adv}}\|_2^2 +$
     $\sum_{t=0}^{T-1}\sum_{n=1}^{N} \lambda_{\infty,n} \cdot \max\left(0, \|(\mathbf{Z}_t)_n\|_\infty - \epsilon_n\right) +$
     $\sum_{t=0}^{T-1}\sum_{n=1}^{N} \lambda_{\text{TV},n} \cdot \max\left(0, \text{TV}((\mathbf{Z}_t)_n) - \tau_n\right)\Big]$
     Gradient Descent Step: $\delta^{(k+1)} \leftarrow \delta^{(k)} - \alpha \nabla_\delta L(\delta^{(k)})$
     Projection Step:  for each channel $n = 1$ to $N$ do
         if $n \in C$ then
             $\delta_n^{(k+1)} \leftarrow \mathbf{M} \odot \delta_n^{(k+1)}$ ;    // Apply spatial mask
         else
             $\delta_n^{(k+1)} \leftarrow \mathbf{0}$ ;   // Zero out perturbation
         end
     end
end
return optimized perturbation $\delta_{WAPPO} = \delta^{(K)}$
```

Experimental Discussion

Hyperparameters:

We use samples from the ERA5 dataset for weather forecasting, starting from 2018, with data points spaced 6 hours apart. The initial condition $\mathbf{Z}_0$ represents the first sample, and t_{adv} corresponds to the forecast 120 hours (5 days) after $\mathbf{Z}_0$. Forecasting is performed using the FourCastNet model (Pathak et al. 2022). The number of prognostic variables N is set to 20 (Table 4). In experiments involving perturbation of specific channels, we perturb $\mathbf{C} = \{\textbf{t2m}\}$, representing the temperature at 2 meters above the surface. The spatial mask for perturbations is configured with a patch starting at $(L_0, M_0) = (1100, 300)$ and a patch size of $(L_p, M_p) = (200, 300)$, chosen to target the South American region. The penalty parameters $\lambda_{\infty,n}$ and $\lambda_{\text{TV},n}$ are set to 0.01, while the constraints ϵ_n are determined as the maximum value of each field over a 5-day forecast. The smoothness parameter τ_n is computed as the average smoothness value over the same period. Specific parameter values are detailed in the Appendix (Table 5). Optimization is performed using the Adam optimizer with a learning rate of $\alpha = 0.01$ and a total iteration count of $K = 1000$.

Performance of WAAPO

Utilizing the hyperparameters described, we solved for δ^* in the unconstrained optimization setting in Equation 5. However, solving for δ_{WAPPO}required additional hyperparameter selection. Channel-based masking initially led to gradient explosions, resulting in a non-smooth, spiky loss trajectory that hindered optimizer convergence. To address this issue, we implemented norm-based gradient clipping to stabilize the gradients and incorporated a learning rate scheduler for dynamic adjustment. These techniques significantly enhanced optimizer stability, leading to improved convergence and more effective perturbation discovery. Consequently, we successfully induced temperature deviations without affecting other channels. Figure 3a illustrates the results when the field **t2m** (temperature at 2m above the surface, as described in Table 4 in the Appendix) was only perturbed. It can be seen that the results are notably similar to 5. Notably, even though some variables are correlated with **t2m**, e.g. **U10**, **V10**, and **sp**, the perturbation was restricted solely to **t2m**, leaving other fields intact. **WAAPO** was thus able to successfully induce temperature deviations even while leaving correlated variables unchanged.

In a subsequent experiment, we applied localized temperature perturbations over South America to ensure that temperatures in other regions remained unaffected. This targeted perturbation involved applying a spatial mask to the perturbation, as shown in Figure 3b. The results indicate that the differences between the true forecast and the perturbed prediction are confined to the South American region, demonstrating the effectiveness of the spatial masking approach.

We also investigated the impact of the smoothness parameter on the perturbations. Figure 4 shows the results of localized perturbations with and without a smoothness penalty. As is evident from the figure, omitting the smoothness parameter produces coarse, visible patches, underscoring the importance of incorporating smoothness constraints. This comparison highlights the critical role of the smoothness parameter in ensuring realistic and physically consistent perturbations. Additionally, while the patch based attack is able to perturb the target region, the temperature profile of the affected region is still not similar to the target temperature profile. A possible reason could be the inherent spatial mixing that prevent effective execution of the targeted attack. In future work we explore how the adversary maybe able to utilize the potential inherent vulnerability of the process and improve **WAPPO** for the targeted localized patch based attacks.

While FourCastNet (Pathak et al. 2022) demonstrates robustness to random Gaussian noise added to its initial conditions, we observe that smaller, yet strategically crafted WAAPO perturbations can induce significant forecast errors. To quantify this effect, we introduce the Perturbation Magnitude Ratio relative to Gaussian (PMRG), which compares the Frobenius norm of δ_{WAAPO} to that of a scaled Gaussian perturbation, $\sigma \cdot \delta_{\text{Gaussian}}$, where $\delta_{\text{Gaussian}} \sim \mathcal{N}(0, 1)$ and $\sigma = 0.3$. The PMRG is defined as:

$$\text{PMRG} = \frac{\|\delta_{\text{WAAPO}}\|_F}{\|\sigma \cdot \delta_{\text{Gaussian}}\|_F}$$

indicates how the overall size of the WAAPO perturbation compares to a typical Gaussian disturbance of the same scale. Since

$$\|\sigma \cdot \delta_{\text{Gaussian}}\|_F \approx \sigma\sqrt{L \cdot M \cdot N},$$

this provides a meaningful benchmark for evaluating magnitude differences.

WAAPO Perturbation Variant	PMRG
S^{p}_{WAAPO} (patch-based)	0.105
S^{c}_{WAAPO} (channel-based)	0.565

Table 2: Comparison of WAAPO's perturbation magnitudes relative to Gaussian noise. Despite being smaller, both variants substantially impact the model's forecasts.

Table 2 shows that both the patch-based (S^{p}_{WAAPO}) and channel-focused (S^{c}_{WAAPO}) attacks yield perturbations smaller than the baseline Gaussian noise. Yet, these ostensibly imperceptible adjustments—especially when localized to a single region or variable—can lead to significant deviations in FourCastNet's predictions. This highlights a critical vulnerability: despite being smaller than random noise in magnitude, carefully targeted adversarial perturbations can exert a disproportionately large effect on AI-based weather forecasts.

Conclusion

In this work, we introduced *Weather Adaptive Adversarial Perturbation Optimization* **(WAAPO)**, a novel framework for generating adversarial perturbations in weather forecasting models. By enforcing channel sparsity, spatial localization, and smoothness constraints, **WAAPO** produces perturbations that are imperceptible, physically valid, and localized to specific regions. Our experiments, conducted using the ERA5 dataset and FourCastNet (Pathak et al. 2022), demonstrate that these carefully tailored attacks can align the model's forecasts closely with adversarial targets—revealing critical vulnerabilities in forecasting systems that heavily depend on accurate initial conditions. As weather models increasingly inform decisions in agriculture, disaster management, and transportation, adversarial attacks that generate false weather alerts or hide extreme events pose significant risks, highlighting the need for effective safeguards.

Notably, our patch-based experiments show that while **WAAPO** can successfully perturb a targeted region, the resulting temperature profile does not fully match the adversarial target. A possible explanation is the inherent spatial mixing within the forecasting model, which dilutes localized perturbations over time. Future work will address this limitation by refining WAAPO's methodology to exploit such mixing more effectively. In addition, the current study focuses mainly on the temperature field (**t2m**); expanding the attack space to other critical variables, such as surface pressure (**sp**), is essential for investigating impacts on hurricane forecasts and other severe weather scenarios. Beyond FourCastNet, evaluating adversarial robustness in alternative state-of-the-art models—including GraphCast (Lam et al. 2022), ClimaX (Huang et al. 2023), and PanguWeather (Bi et al. 2022)—represents a crucial research direction. Finally, building practical defenses (e.g., robustness-aware training or automated anomaly detection) will be vital to mitigating adversarial risks not only in weather forecasting but also in broader climate modeling and environmental simulation tasks.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 and LDRD Program Project No. 23-ERD-030 (LLNL-JRNL-871577).

References

Akhtar, N.; and Mian, A. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6: 14410–14430.

Belkhouja, T.; and Doppa, J. R. 2022. Adversarial framework with certified robustness for time-series domain via statistical features. *Journal of Artificial Intelligence Research*, 73: 1435–1471.

Bi, K.; Xie, L.; Liu, X.; and Xie, L. 2022. Pangu-Weather: A 3D High-Resolution Model for Fast and Accurate Global Weather Forecast. *arXiv preprint arXiv:2206.03408*.

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, 39–57. Ieee.

Costa, J. C.; Roxo, T.; Proença, H.; and Inácio, P. R. 2024. How deep learning sees the world: A survey on adversarial attacks & defenses. *IEEE Access*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Huang, B.; Song, H.; Song, Y.; and Wang, W. 2023. ClimaX: A foundation model for weather and climate. *arXiv preprint arXiv:2301.10343*.

Kos, J.; Fischer, I.; and Song, D. 2018. Adversarial examples for generative models. In *2018 ieee security and privacy workshops (spw)*, 36–42. IEEE.

Lam, R.; Sanchez-Gonzalez, A.; Willson, M.; Wirnsberger, P.; Fortunato, M.; Alet, F.; Ravuri, S.; Ewalds, T.; Eaton-Rosen, Z.; Hu, W.; et al. 2022. GraphCast: Learning skillful medium-range global weather forecasting. *arXiv preprint arXiv:2212.12794*.

Liu, L.; Park, Y.; Hoang, T. N.; Hasson, H.; and Huan, J. 2022. Robust multivariate time-series forecasting: Adversarial attacks and defense mechanisms. *arXiv preprint arXiv:2207.09572*.

Long, T.; Gao, Q.; Xu, L.; and Zhou, Z. 2022. A survey on adversarial attacks in computer vision: Taxonomy, visualization and future directions. *Computers & Security*, 121: 102847.

Nakashima, E. 2024. China-linked hackers stole wiretap data from telcos, FBI and CISA say. *The Washington Post*.

Nguyen, T.; Jewik, J.; Bansal, H.; Sharma, P.; and Grover, A. 2024. Climatelearn: Benchmarking machine learning for weather and climate modeling. *Advances in Neural Information Processing Systems*, 36.

Pagliery, J. 2014. U.S. weather system hacked, affecting satellites. *CNN Business*.

Pathak, J.; Subramanian, S.; Harrington, P.; Raja, S.; Chattopadhyay, A.; Mardani, M.; Kurth, T.; Hall, D.; Li, Z.; Azizzadenesheli, K.; et al. 2022. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*.

Rackow, T.; Koldunov, N.; Lessig, C.; Sandu, I.; Alexe, M.; Chantry, M.; Clare, M.; Dramsch, J.; Pappenberger, F.; Pedruzo-Bagazgoitia, X.; et al. 2024. Robustness of AI-based weather forecasts in a changing climate. *arXiv preprint arXiv:2409.18529*.

Rathore, P.; Basak, A.; Nistala, S. H.; and Runkana, V. 2020. Untargeted, targeted and universal adversarial attacks and defenses on time series. In *2020 international joint conference on neural networks (IJCNN)*, 1–8. IEEE.

The Washington Post. 2023. Hurricane Lee to scrape New England, bolt toward Canada: Storm updates.

Appendix

Real-World Adversarial Perturbations and Attack Strategies in Weather Forecasting Models

In the context of weather forecasting models like FourCastNet, feasible adversarial attacks would aim to subtly alter the input variables to achieve a significant change in the forecast output. Consider some examples of perturbations and attack strategies that could be employed:

These are specific instances of the attack and we can keep the setting to be very generic.

1. Changing Hurricane Location

Objective: Shift the predicted path of a hurricane to a different location.

Perturbation: Slightly adjust the wind speed, temperature, and pressure in the initial conditions around the hurricane's current location to steer it towards a different path.

2. Altering Storm Intensity

Objective: Modify the predicted intensity of a storm (e.g., making a hurricane appear stronger or weaker).

Perturbation: Small changes in sea surface temperature, pressure, and humidity around the storm's center to affect the storm's development and intensity.

3. Creating False Weather Events

Objective: Generate forecasts that predict non-existent weather events, such as a hurricane or severe storm where none would occur.

Perturbation: Introduce small but spatially consistent perturbations across temperature, wind, and pressure fields in regions where no significant weather is expected to create the illusion of a developing storm.

4. Suppressing True Weather Events

Objective: Prevent the model from predicting an actual severe weather event.

Perturbation: Apply small changes to key variables in the vicinity of the developing weather system to disrupt its formation in the model forecast.

5. (One Variable) Modifying Temperature Profiles

Objective: Change the predicted temperature distribution across a region.

Perturbation: Adjust the initial temperature field to create warmer or cooler forecast conditions in targeted areas, which could affect predictions of heatwaves or cold spells.

6. (One Variable) Wind Pattern Manipulation

Objective: Alter the predicted wind patterns, which could impact wind energy forecasts or general weather patterns.

Perturbation: Small, targeted changes to the initial wind field at various altitudes to influence the overall wind distribution.

7. Time Manipulation

Objective: Change the predicted speed and timing of a hurricane's movement, causing delays or acceleration, and spread out the uncertainty in its path.

Perturbation: Adjust the initial conditions to alter the hurricane's speed and trajectory timing. This includes modifying wind speeds, pressure gradients, and other relevant atmospheric variables to delay or accelerate the hurricane's movement and increase the uncertainty in its predicted path

Unconstrained Optimization

Table 3: Variables and Functions

Symbol	**Description**
$\mathbf{X}_0 \in \mathbb{R}^{N\times L\times M}$	Original input
$\delta \in \mathbb{R}^{N\times L\times M}$	Perturbation
$C \subseteq \{1, 2, \ldots, N\}$	Channels to perturb
$\mathbf{M} \in \mathbb{R}^{L\times M}$	Spatial mask
$\phi(\mathbf{Z}_t)_n$	Model output at time t for channel n
ϵ_n	Infinity norm constraint for channel n
τ_n	Total variation constraint for channel n
$\lambda_{\infty,n}, \lambda_{\text{TV},n}$	Penalty parameters for channel n
α	Learning rate
$\phi(\mathbf{Z}_{T-1} \mid \mathbf{X}_0^\delta)$	Model output at final time step with perturbed input
TV$(\cdot)$	Total variation operator
$\odot$	Element-wise multiplication

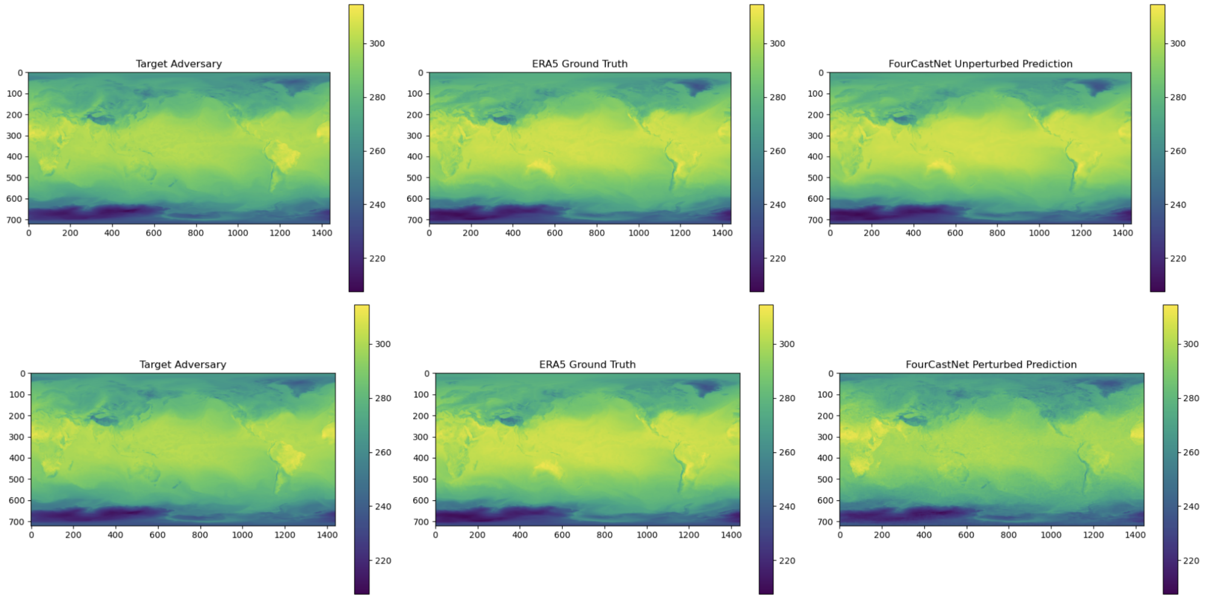
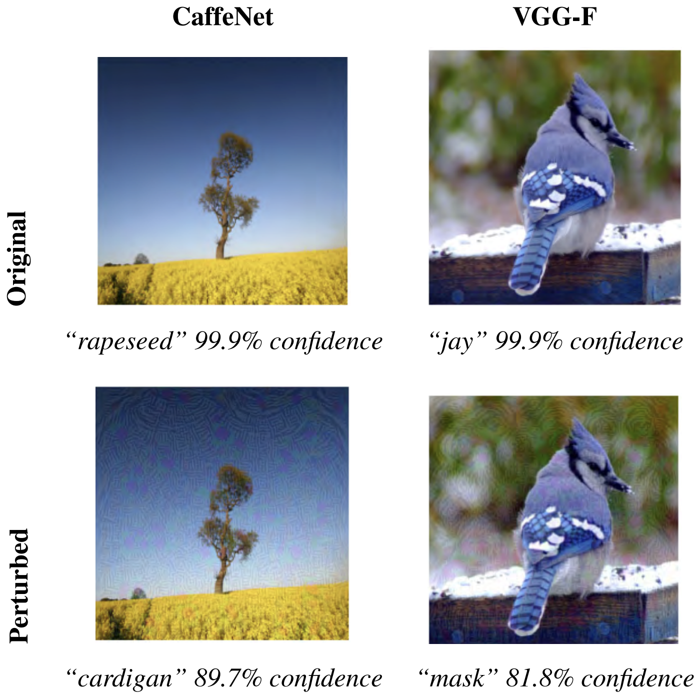


Figure 5: **Top:** Normal Prediction. **Bottom:** Perturbed Prediction. Illustration of the effect of the unconstrained adversarial attack on weather forecasts.

Table 4: 20 Prognostic Variables Modeled by FourcastNet(Pathak et al. 2022)

Variable	**Description**
Surface	
U10	Zonal wind velocity at 10 meters above the surface
V10	Meridional wind velocity at 10 meters above the surface
T2m	Temperature at 2 meters above the surface
sp	Surface pressure
mslp	Mean sea level pressure
1000 hPa	
U1000	Zonal wind velocity at 1000 hPa
V1000	Meridional wind velocity at 1000 hPa
Z1000	Geopotential height at 1000 hPa
850 hPa	
U850	Zonal wind velocity at 850 hPa
V850	Meridional wind velocity at 850 hPa
Z850	Geopotential height at 850 hPa
T850	Temperature at 500 hPa
R850	Relative humidity at 850 hPa
500 hPa	
U500	Zonal wind velocity at 500 hPa
V500	Meridional wind velocity at 500 hPa
Z500	Geopotential height at 500 hPa
T500	Temperature at 500 hPa
R500	Relative humidity at 500 hPa
50 hPa	
Z50	Geopotential height at 50 hPa
Integrated Variables	
TCWV	Total Column Water Vapor



“rapeseed” 99.9% confidence

“jay” 99.9% confidence

“cardigan” 89.7% confidence

“mask” 81.8% confidence

Figure 6: Comparison of Original and Perturbed Images for CaffeNet and VGG-F when the underlying target model is a classifier. An imperceptible perturbation leads to a different class with high confidence.

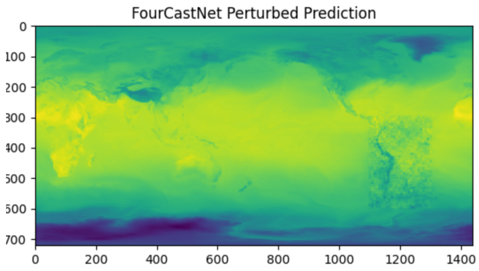


Figure 7: Coarse Patches.

Channel n	ϵ_n	τ_n
1	9.1473×10^{-8}	99165.6016
2	2.5559×10^{-6}	109694.2422
3	1.3177×10^{-6}	27920.4980
4	2.8082×10^{-7}	58921.9922
5	1.8507×10^{-6}	34445.3281
6	4.2667×10^{-7}	22821.9902
7	1.2589×10^{-7}	98763.7188
8	5.4461×10^{-7}	108050.5078
9	1.9669×10^{-6}	32448.5762
10	5.8189×10^{-7}	89158.7656
11	1.7013×10^{-6}	102990.0625
12	3.5134×10^{-6}	20403.5078
13	3.3896×10^{-7}	61842.3750
14	1.8124×10^{-7}	63398.3281
15	2.4146×10^{-6}	13680.4805
16	$1.8668 \times 10^{+0}$	19226.8984
17	3.6435×10^{-6}	9923.5332
18	4.8708×10^{-6}	114838.1641
19	1.6442×10^{-6}	142137.9844
20	6.4423×10^{-8}	49321.4219

Table 5: Values of ϵ and τ for different channels.