BIFOLD

ADVERSARIAL OBSERVATIONS IN WEATHER FORECASTING

Erik Imgrund, Thorsten Eisenhofer, Konrad Rieck

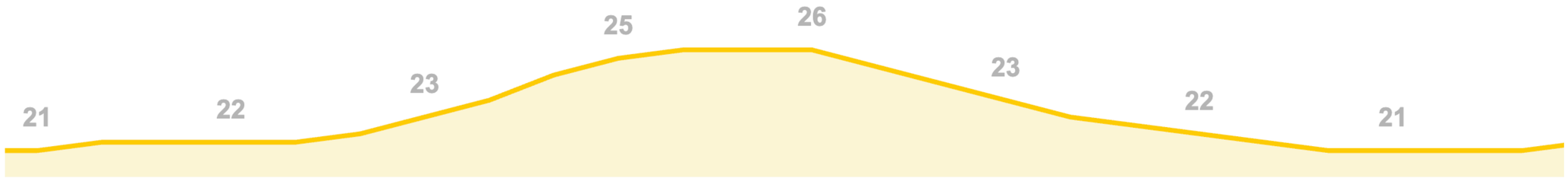
22 °C | °F

Precipitation: 90%
Humidity: 70%
Wind: 17 km/h

Weather

Thursday
Light rain

Temperature | Precipitation | Wind



03:00	06:00	09:00	12:00	15:00	18:00	21:00	00:00

Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
31° 26°	31° 26°	27° 24°	22° 22°	26° 23°	27° 23°	26° 23°	26° 23°

28 °C

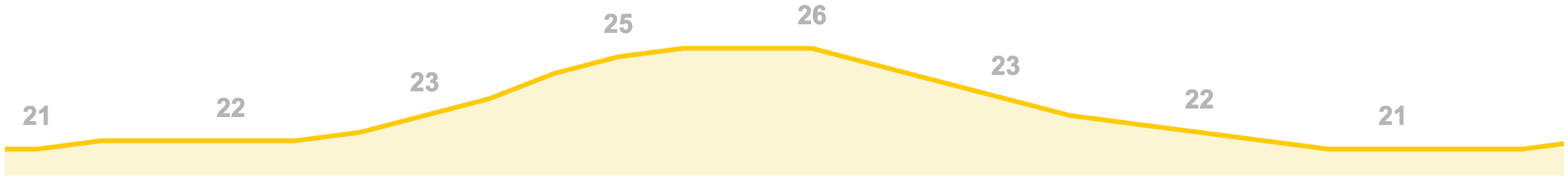
22 °C | °F

Precipitation: 90%
Humidity: 70%
Wind: 17 km/h

Weather

Thursday
Light rain

Temperature | Precipitation | Wind



03:00	06:00	09:00	12:00	15:00	18:00	21:00	00:00

Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
31° 26°	31° 26°	27° 24°	22° 22°	26° 23°	27° 23°	26° 23°	26° 23°

Can we manipulate the weather?

Can we manipulate the weather?

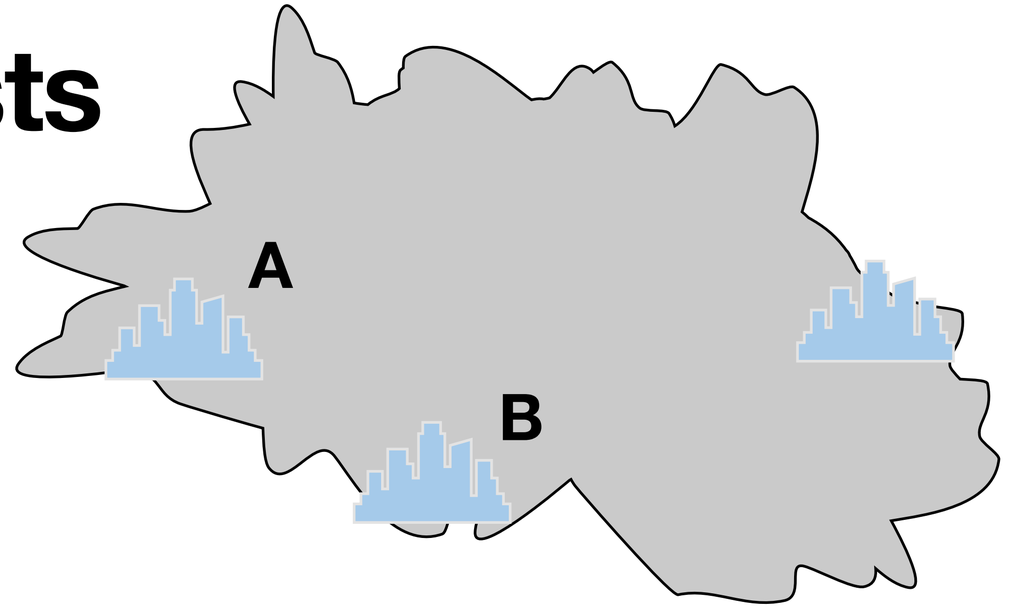
no

forecast

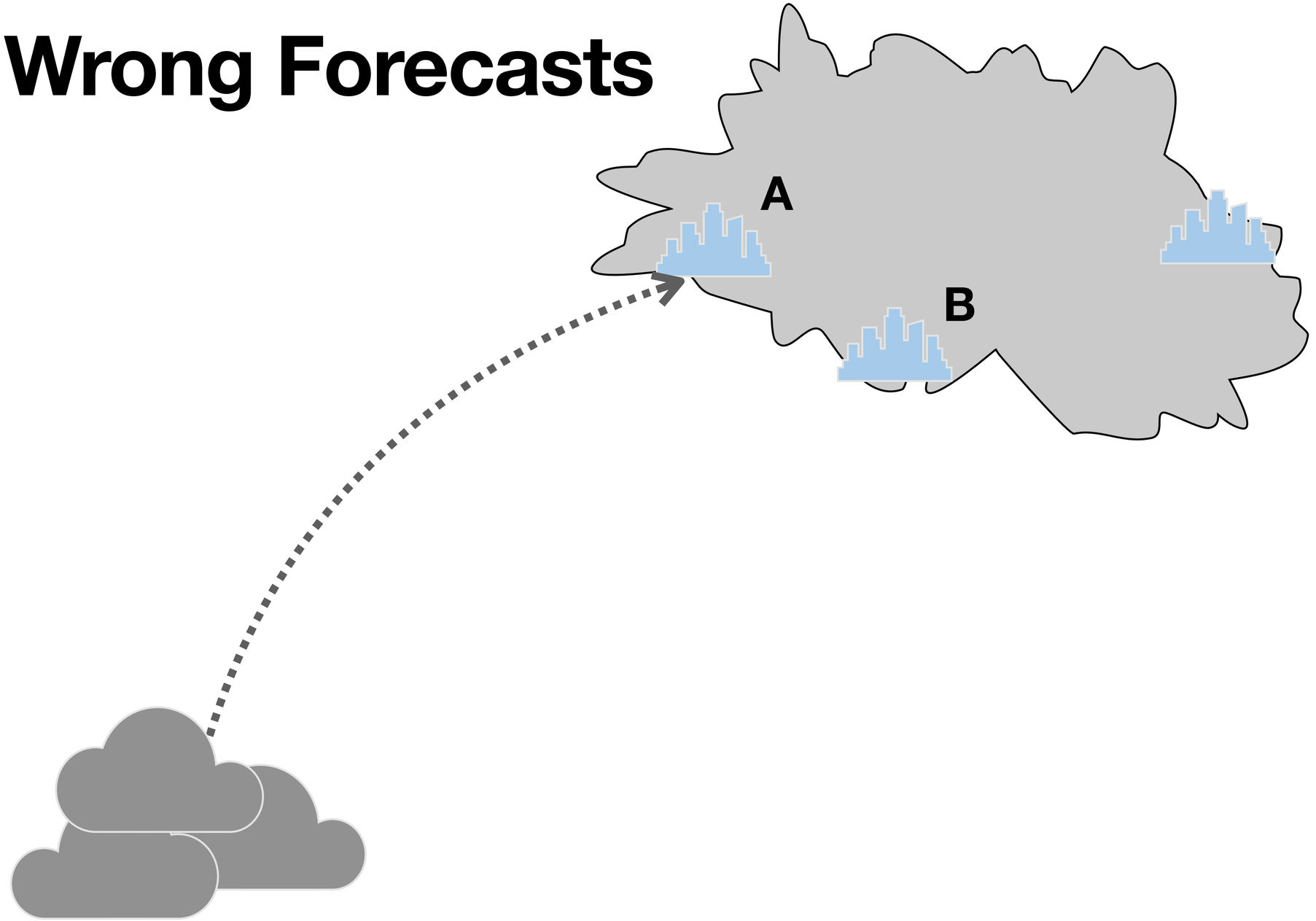
Can we manipulate the weather?

Risk of Wrong Forecasts

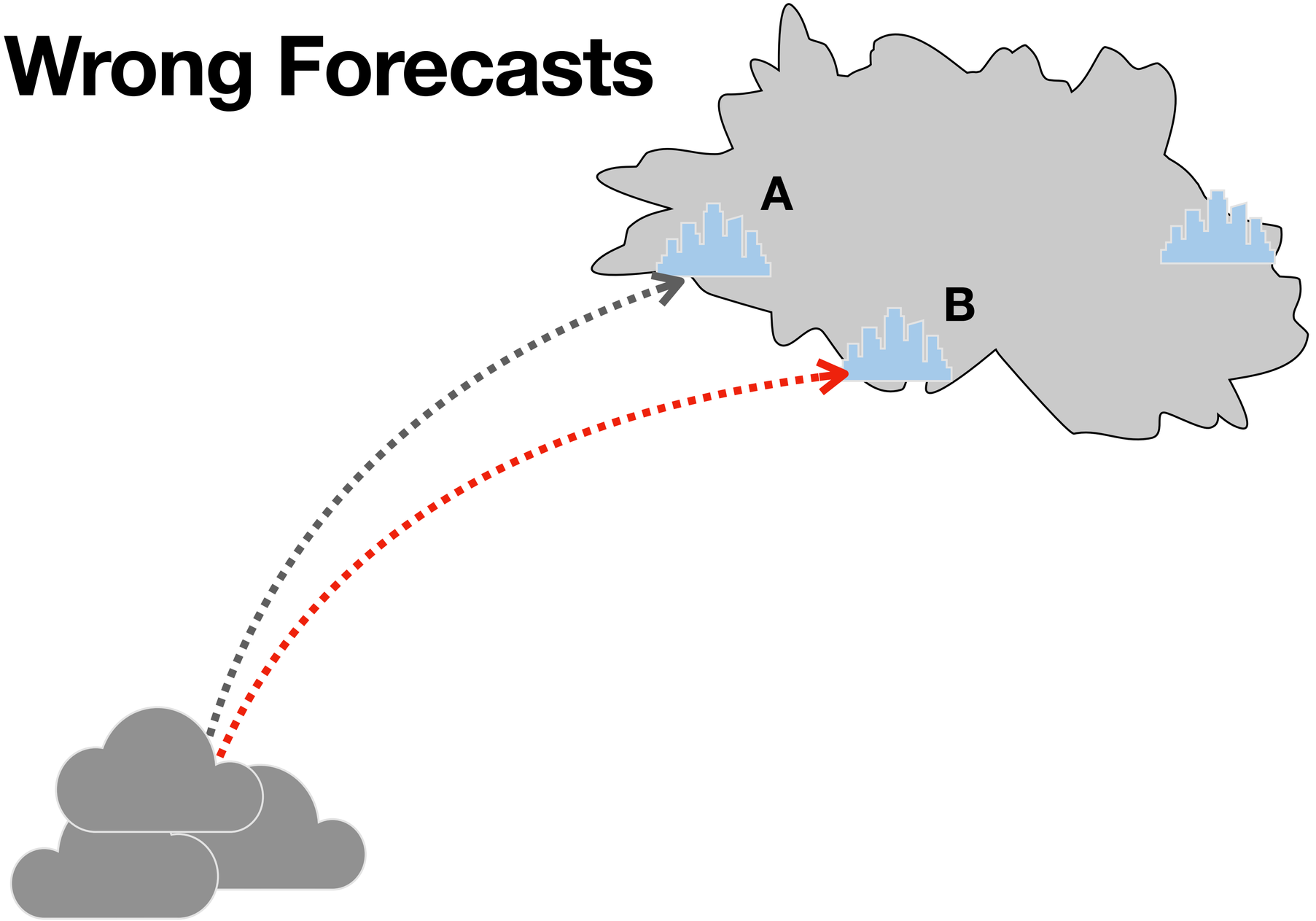
Risk of Wrong Forecasts



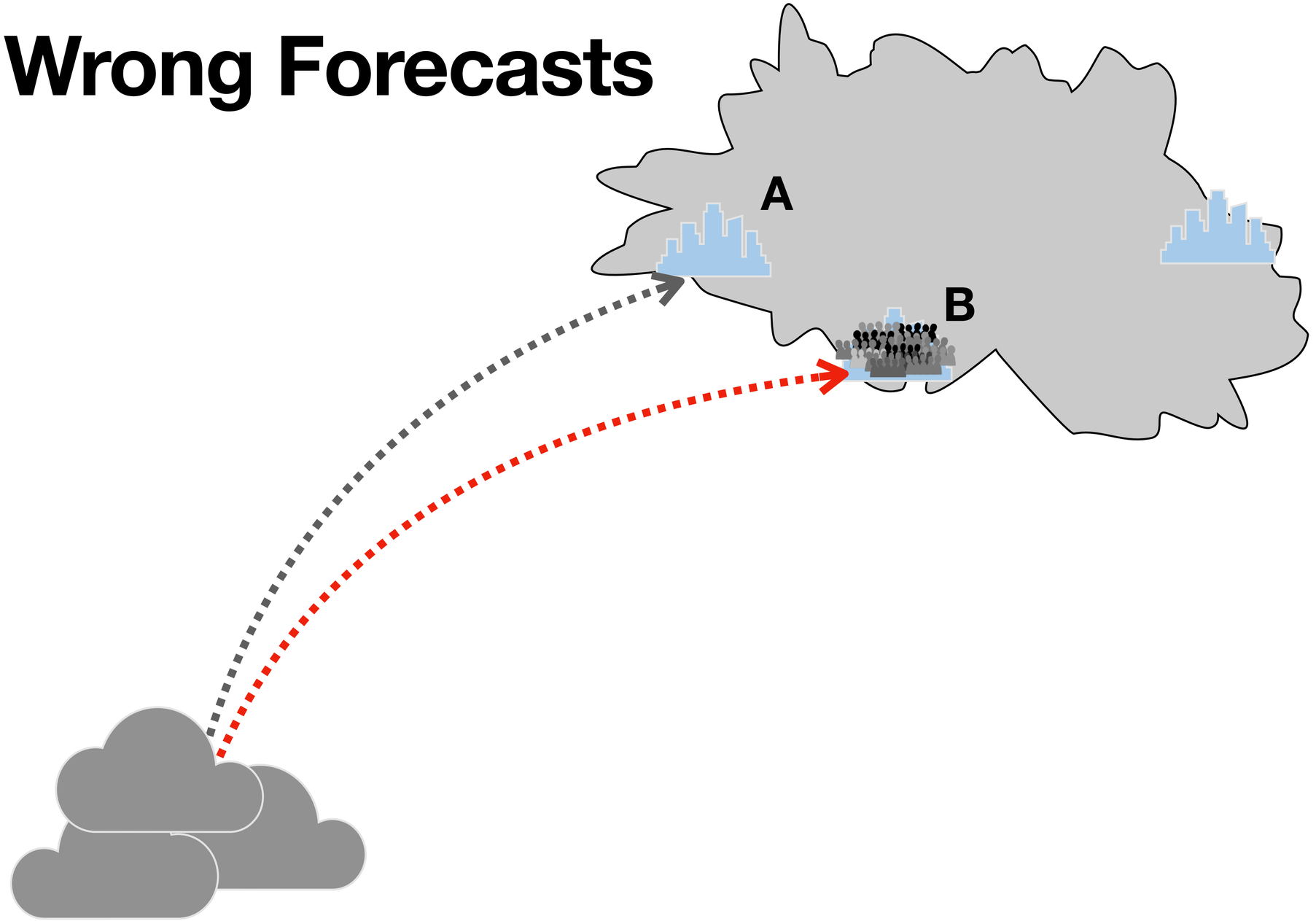
Risk of Wrong Forecasts



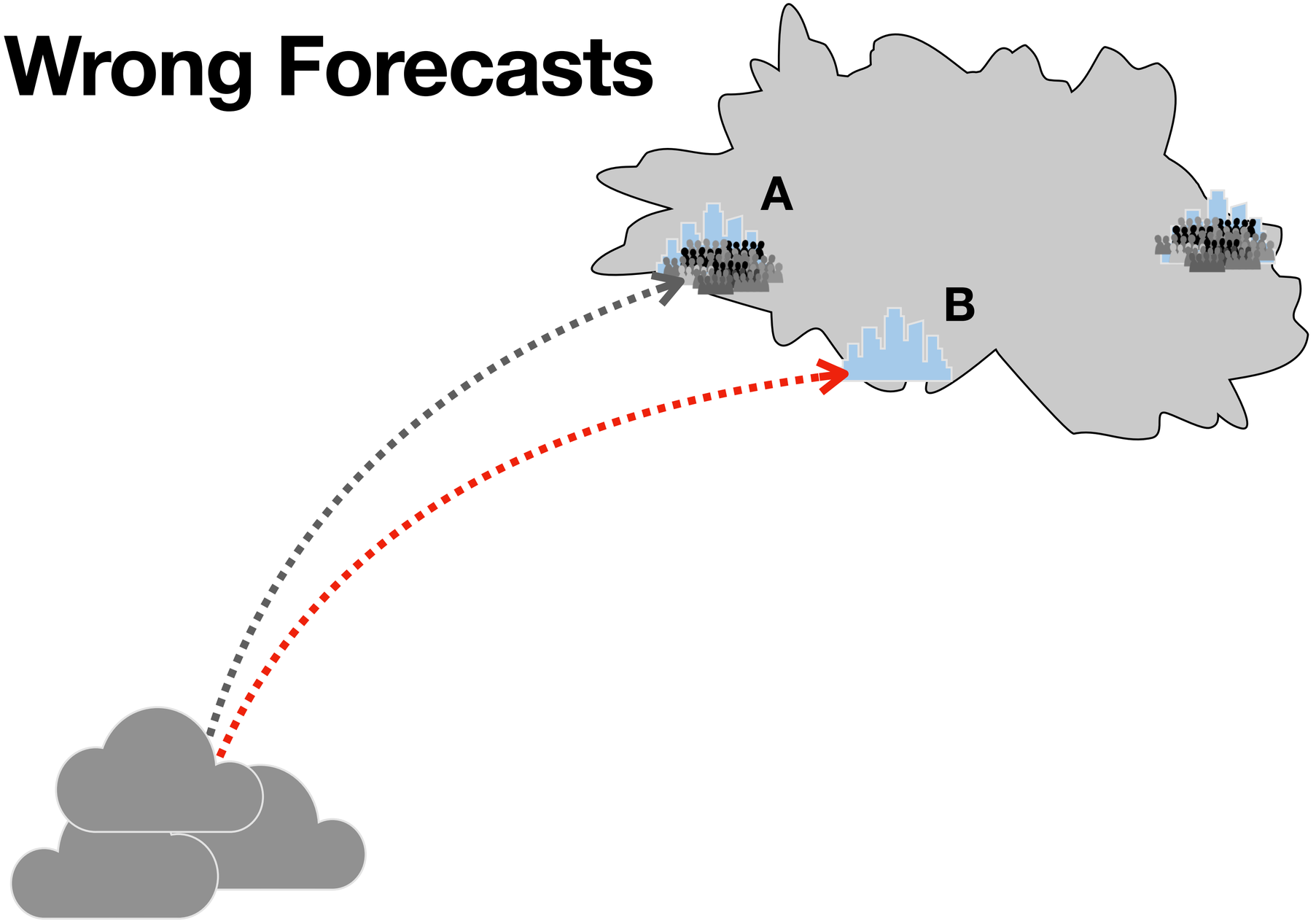
Risk of Wrong Forecasts



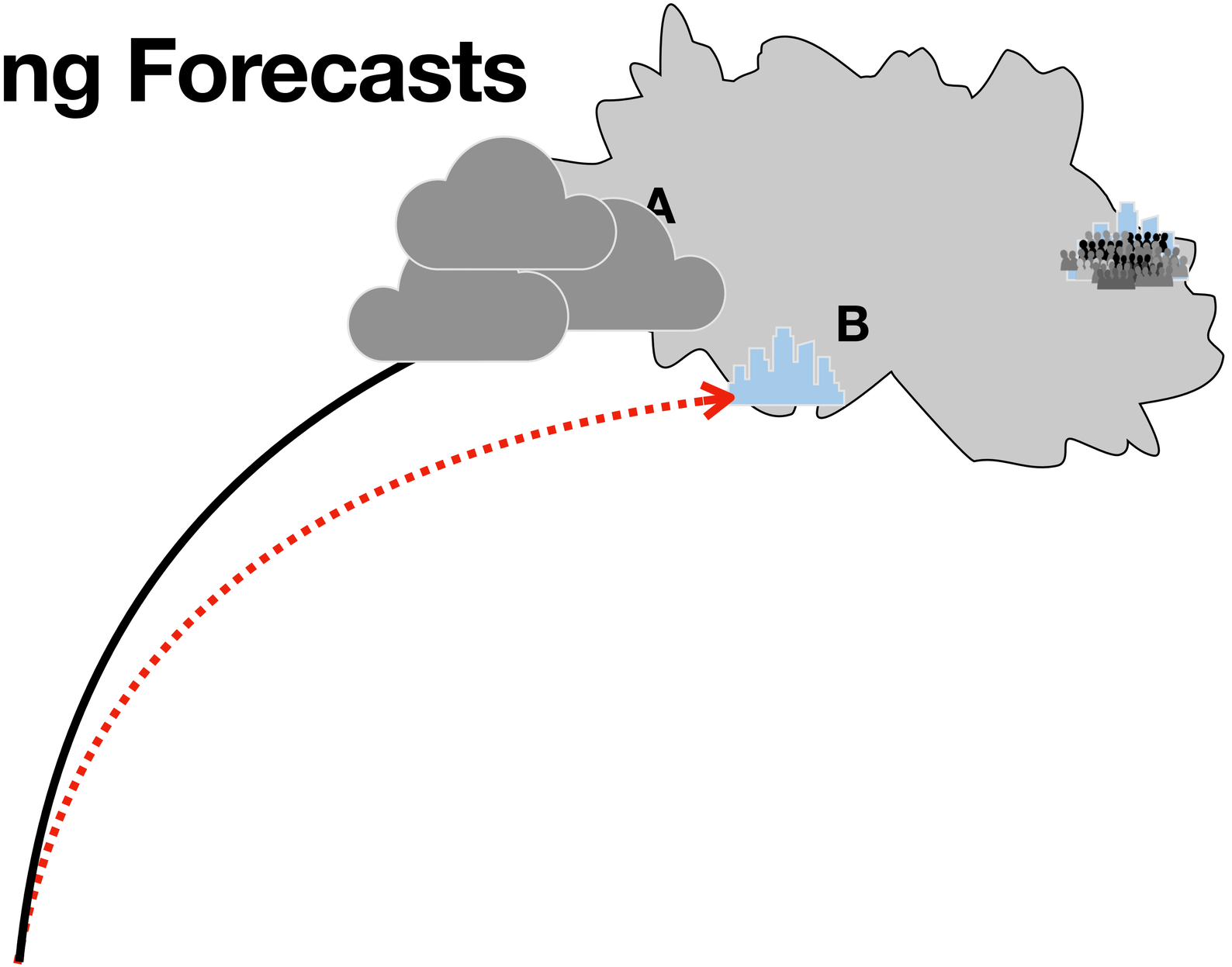
Risk of Wrong Forecasts



Risk of Wrong Forecasts

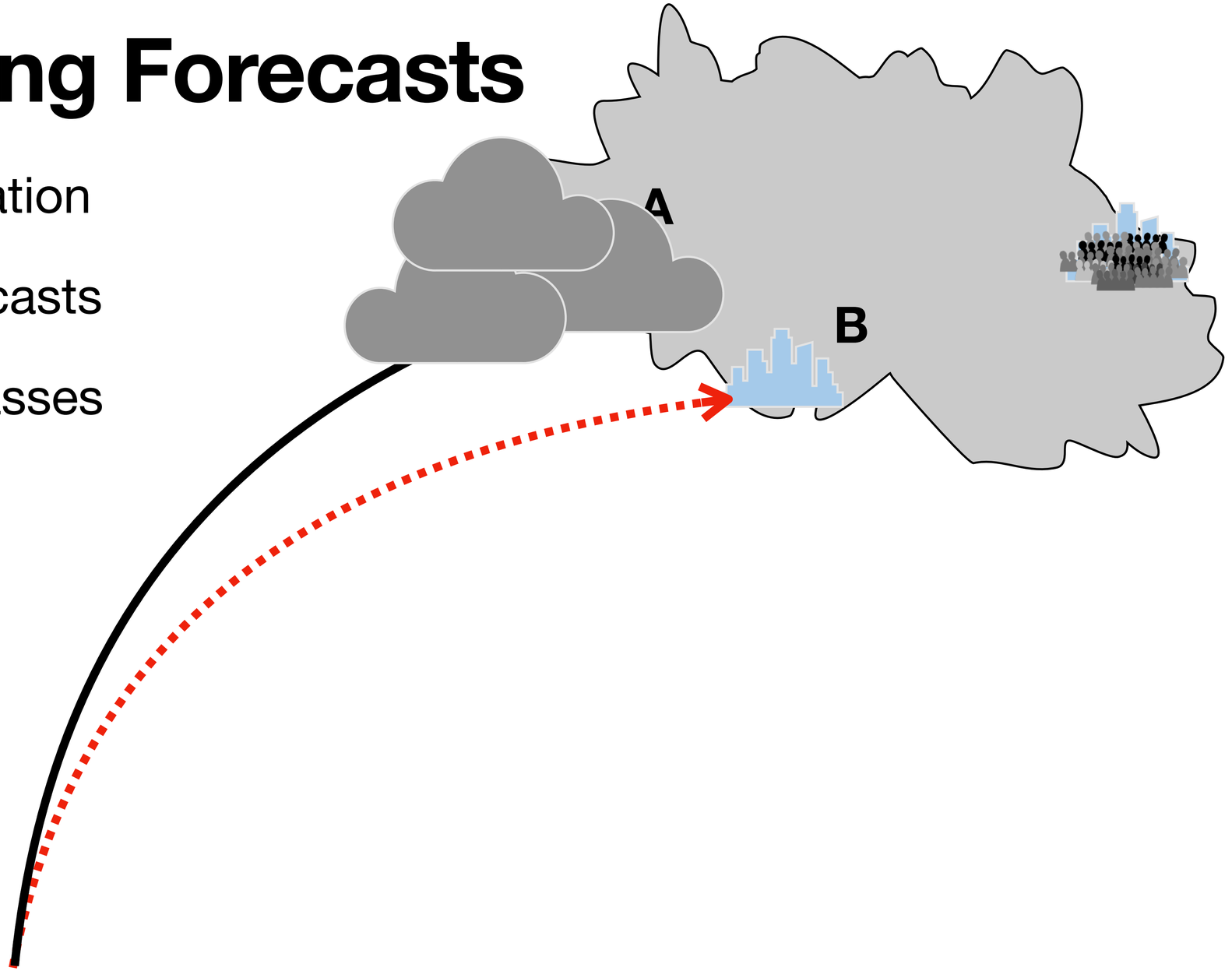


Risk of Wrong Forecasts



Risk of Wrong Forecasts

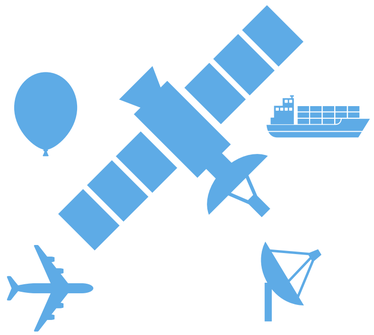
- Can hinder preparation
- Lower trust in forecasts
- Induce panic in masses



Medium-range Weather Forecasting

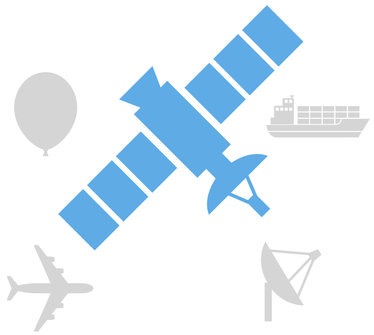
Medium-range Weather Forecasting

- Based on diverse sensor data



Medium-range Weather Forecasting

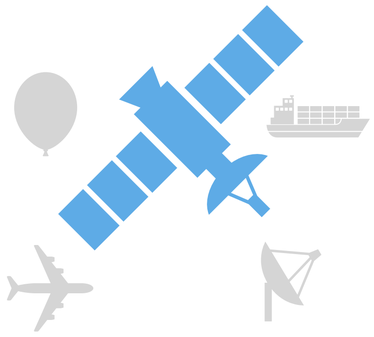
- Based on diverse sensor data
  - Satellites have by far the largest impact[1]



[1] ECMWF “Atmospheric Model Data Sources”, Forecast User Guide

Medium-range Weather Forecasting

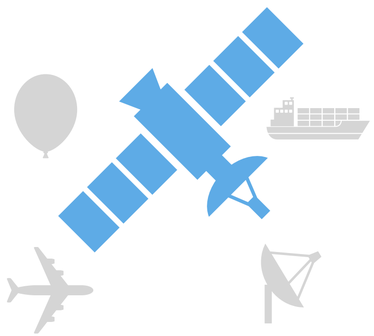
- Based on diverse sensor data
  - Satellites have by far the largest impact[1]
- Recently: AI-based forecasting outperforms traditional forecasting



[1] ECMWF “Atmospheric Model Data Sources”, Forecast User Guide

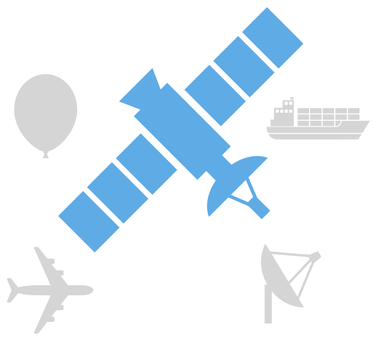
Medium-range Weather Forecasting

- Based on diverse sensor data
  - Satellites have by far the largest impact[1]
- Recently: AI-based forecasting outperforms traditional forecasting
- SotA: Autoregressive diffusion model GenCast[2]



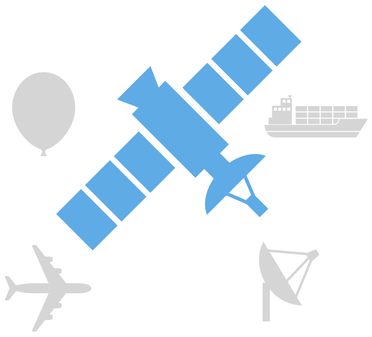
[1] ECMWF “Atmospheric model data sources”, Forecast User Guide
[2] Price et al. “Probabilistic weather forecasting with machine learning”, Nature

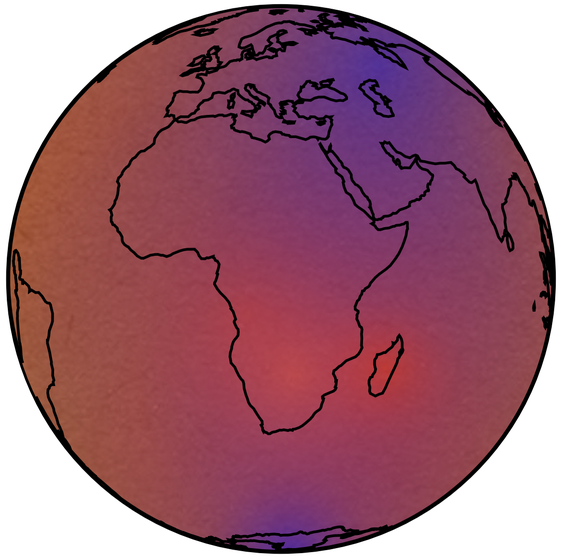
Medium-range Weather Forecasting



Medium-range Weather Forecasting

1. Assimilation of sensor data into global state

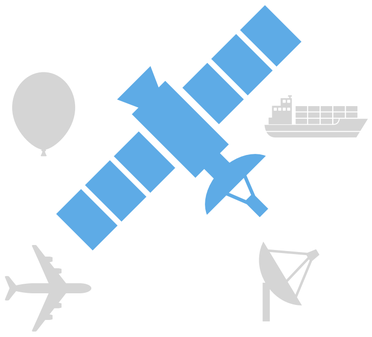


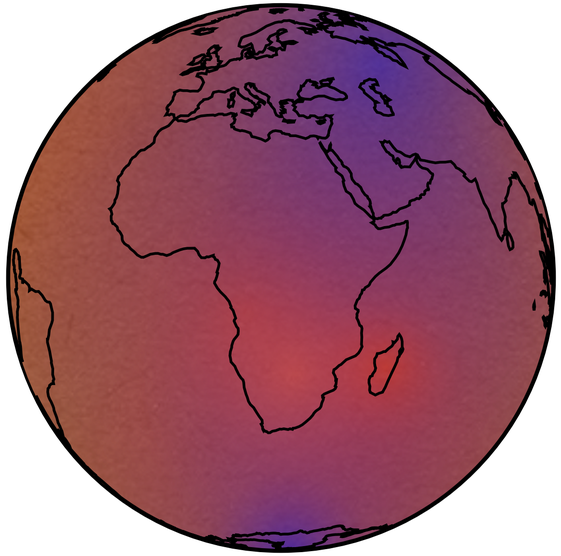


$\tau = 0\text{h}$

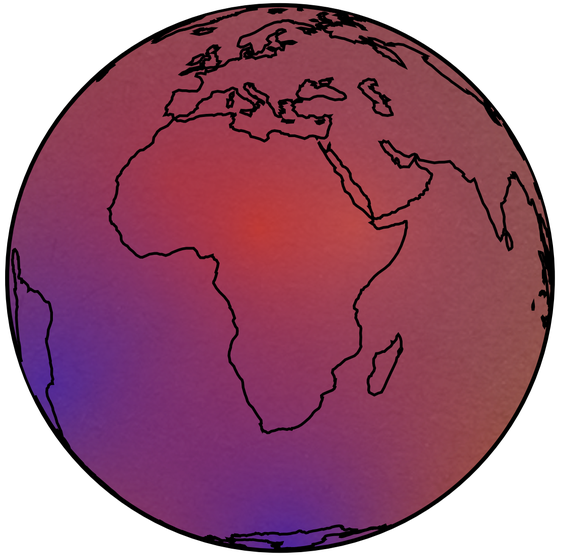
Medium-range Weather Forecasting

1. Assimilation of sensor data into global state
2. Diffusion model predicts state 12h later





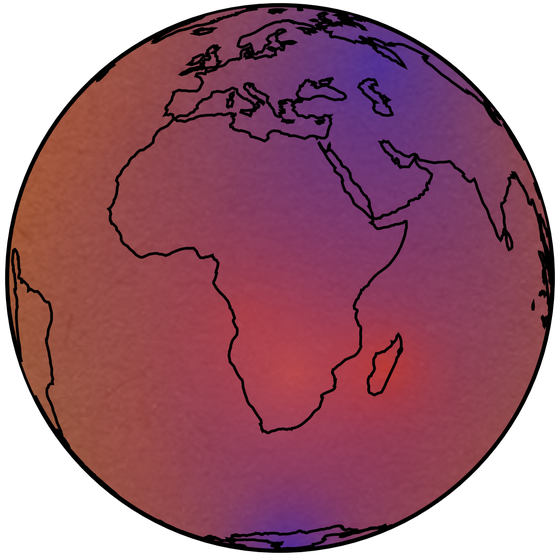
τ = 0h



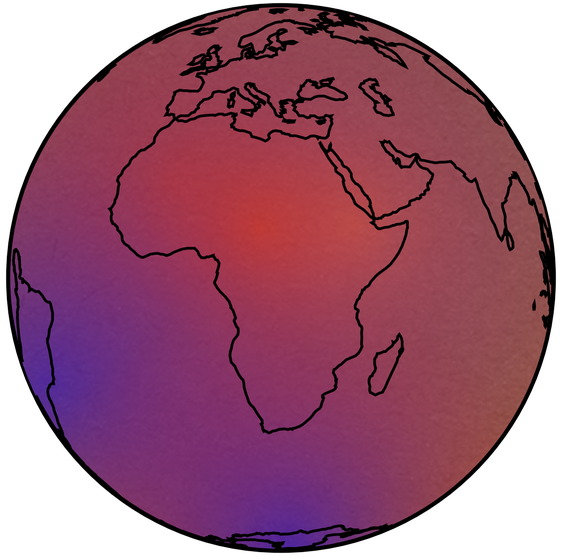
τ = 12h

Medium-range Weather Forecasting

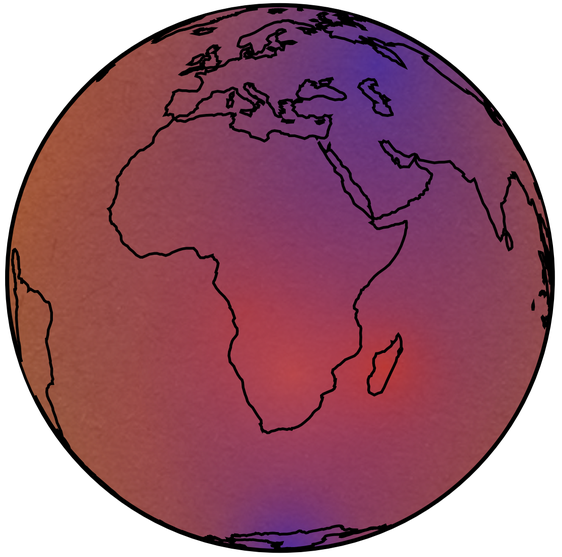
1. Assimilation of sensor data into global state
2. Diffusion model predicts state 12h later
3. Repeat 2. with predicted state



τ = 0h



τ = 12h



τ = 24h

Threat Model

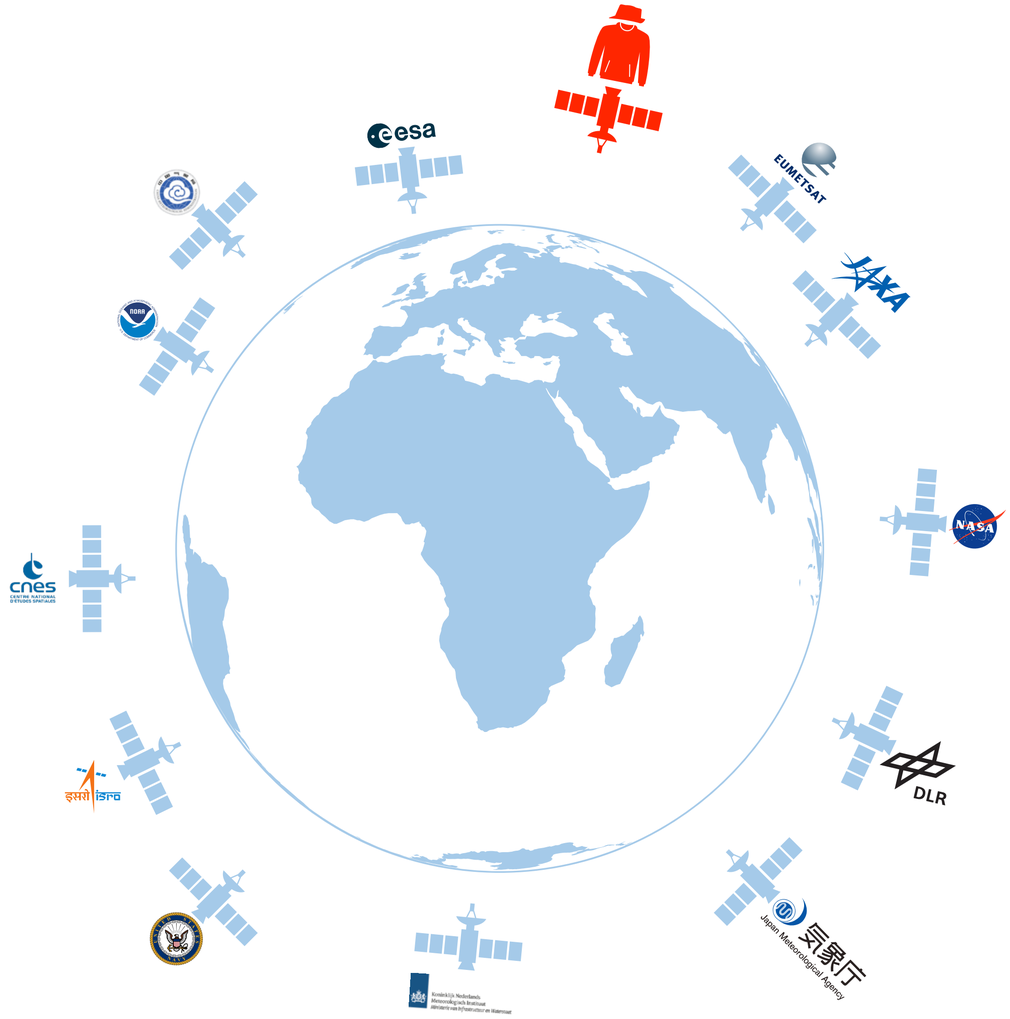
Threat Model

≈100 meteorological satellites contribute data

Threat Model

≈100 meteorological satellites contribute data

- Attacker controls one satellite[1]



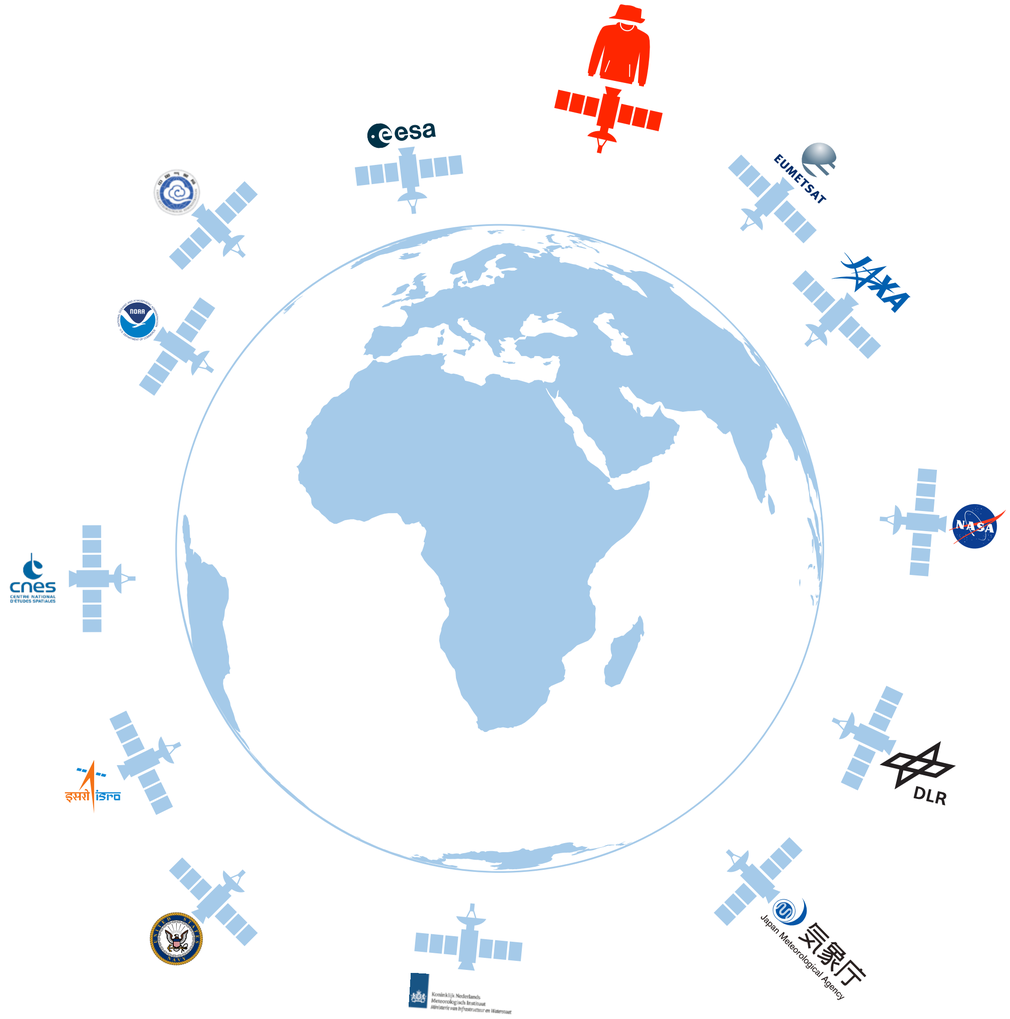
[1] Willbold et al. „Space odyssey: An experimental software security analysis of satellites“, S&P’23

Threat Model

≈100 meteorological satellites contribute data

- Attacker controls one satellite[1]

$$\overline{X} = X + \underbrace{\mathcal{N}\left(0, \sigma^2\right)}_{\text{Natural Noise}} + \underbrace{\mathcal{N}\left(0, \epsilon^2\right)}_{\text{Attacker}}$$



[1] Willbold et al. „Space odyssey: An experimental software security analysis of satellites“, S&P'23

Threat Model

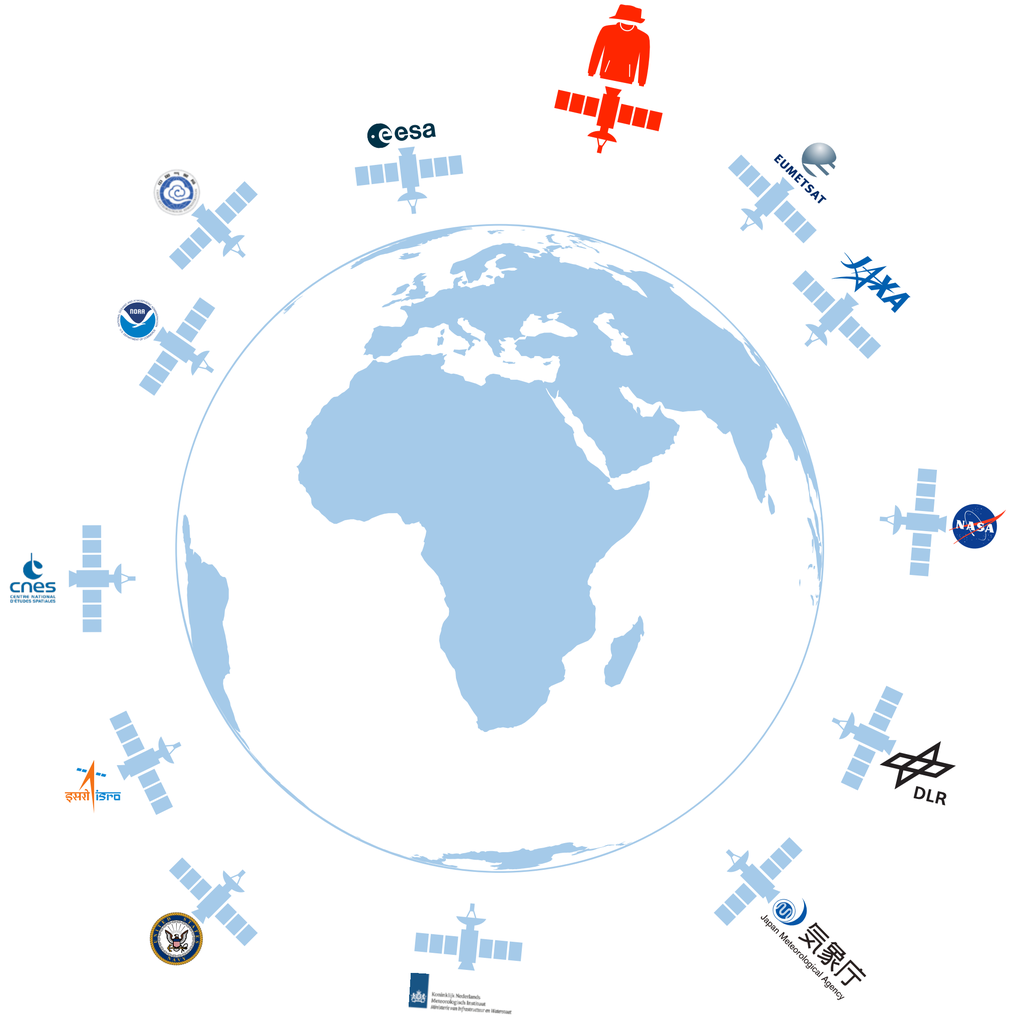
≈100 meteorological satellites contribute data

- Attacker controls one satellite[1]

$$\overline{X} = X + \underbrace{\mathcal{N}\left(0, \sigma^2\right)}_{\text{Natural Noise}} + \underbrace{\mathcal{N}\left(0, \epsilon^2\right)}_{\text{Attacker}}$$

- Adds noise with small variance

$$\sigma^2 \gg \epsilon^2$$



[1] Willbold et al. „Space odyssey: An experimental software security analysis of satellites“, S&P'23

Attack Algorithm

Input: attack budget ϵ, number of attack steps N, lead time steps j, inputs $\mathbf{X}^t, \mathbf{X}^{t-1}$

Output: adversarial perturbation $\boldsymbol{\delta}^t, \boldsymbol{\delta}^{t-1}$

$\mathbf{m}_0 \leftarrow \mathbf{0}$

$\boldsymbol{\delta}_0 = (\boldsymbol{\delta}_0^t, \boldsymbol{\delta}_0^{t-1}) \leftarrow \mathbf{0}$

for $i \leftarrow 1$ **to** N **do**

$\tilde{\mathbf{X}}^{t+j} = f(\mathbf{X}^t + \boldsymbol{\delta}_{i-1}^t, \mathbf{X}^{t-1} + \boldsymbol{\delta}_{i-1}^{t-1}, j, n)$

$\mathbf{g}_i \leftarrow \nabla_{\boldsymbol{\delta}_{i-1}} \mathcal{A}(\tilde{\mathbf{X}}^{t+j})$

$\mathbf{m}_i \leftarrow \beta \cdot \mathbf{m}_{i-1} + (1-\beta) \cdot \Pi_1(\mathbf{g}_i)$

$\alpha_i' \leftarrow \frac{\epsilon}{N} + \frac{1}{2}\left(2\epsilon - \frac{\epsilon}{N} \cdot 1 + \cos\ \frac{(i-1)\cdot\pi}{N}\right)$

$\alpha_i \leftarrow \frac{\alpha_i'}{(1-\beta)^i}$

$\boldsymbol{\delta}_i \leftarrow \Pi_\epsilon(\boldsymbol{\delta}_{i-1} - \alpha_i \mathbf{m}_i)$

return $\boldsymbol{\delta}_N^t, \boldsymbol{\delta}_N^{t-1}$

Attack Algorithm

Gradient of diffusion output

Input: attack budget ϵ, number of attack steps N, learning rate steps j, inputs $\mathrm{X}^t, \mathrm{X}^{t-1}$

Output: adversarial perturbation δ^t, δ^{t-1}

$\mathrm{m}_0 \leftarrow 0$

$\delta_0 = (\delta_0^t, \delta_0^{t-1}) \leftarrow 0$

for $i \leftarrow 1$ **to** N **do**

$\tilde{\mathrm{X}}^{t+j} = f(\mathrm{X}^t + \delta_{i-1}^t, \mathrm{X}^{t-1} + \delta_{i-1}^{t-1}, j, n)$

$\mathrm{g}_i \leftrightarrow \nabla_{\delta_{i-1}} \mathcal{A}(\tilde{\mathrm{X}}^{t+j})$

$\mathrm{m}_i \leftarrow \beta \cdot \mathrm{m}_{i-1} + (1-\beta) \cdot \Pi_1(\mathrm{g}_i)$

$\alpha_i' \leftarrow \frac{\epsilon}{N} + \frac{1}{2}\left(2\epsilon - \frac{\epsilon}{N} \cdot 1 + \cos\frac{(i-1)\cdot\pi}{N}\right)$

$\alpha_i \leftarrow \frac{\alpha_i'}{(1-\beta)^i}$

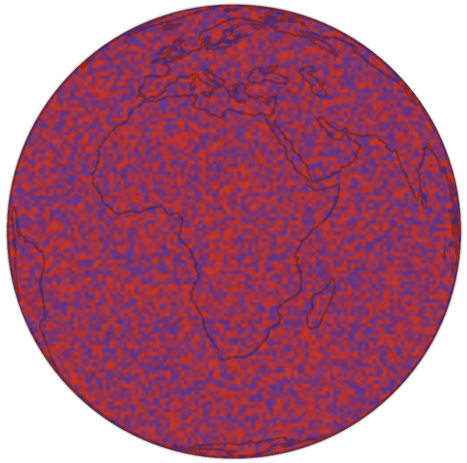
$\delta_i \leftarrow \Pi_\epsilon(\delta_{i-1} - \alpha_i \mathrm{m}_i)$

return $\delta_N^t, \delta_N^{t-1}$

Diffusion Inference

Diffusion Inference

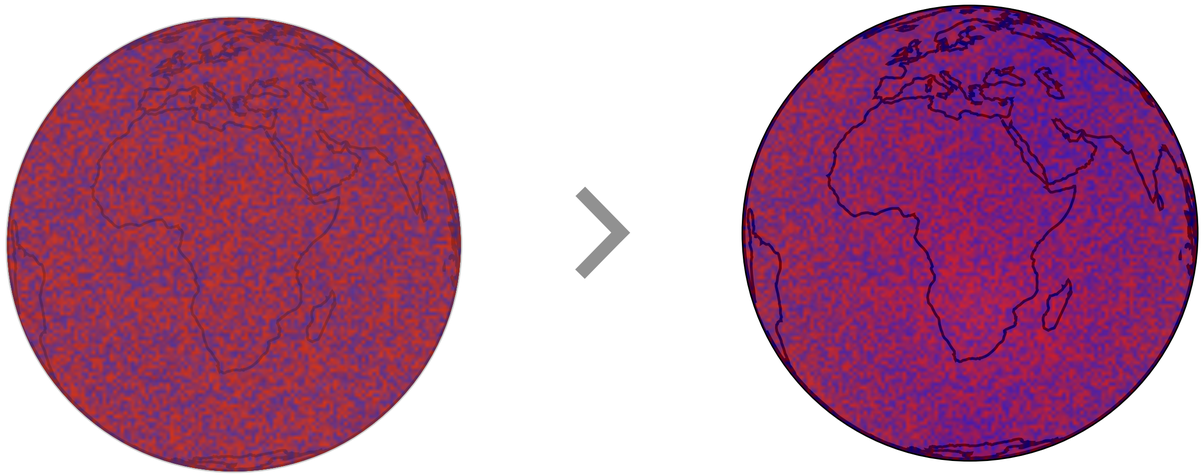
- Starts with random noise as input



$i = 0$

Diffusion Inference

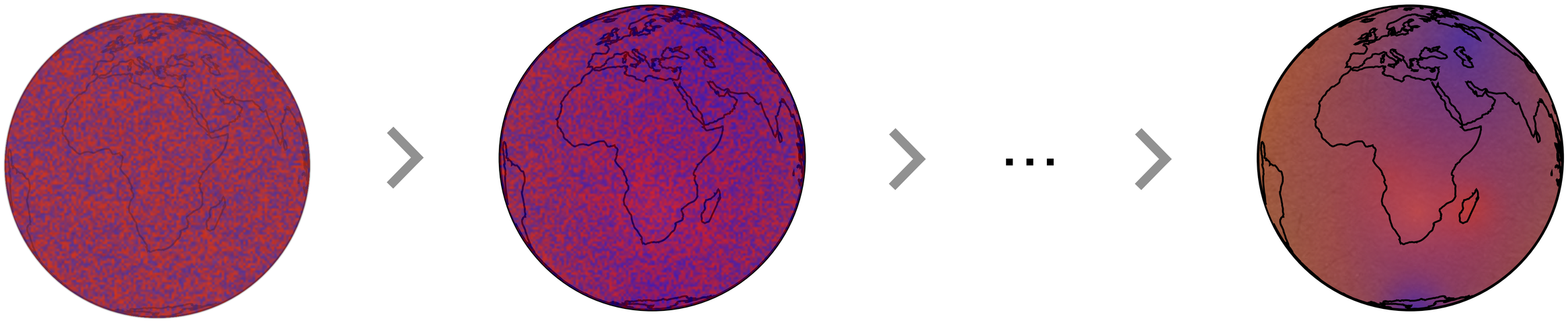
- Starts with random noise as input
- N denoising steps



$i = 0$ $i = 1$

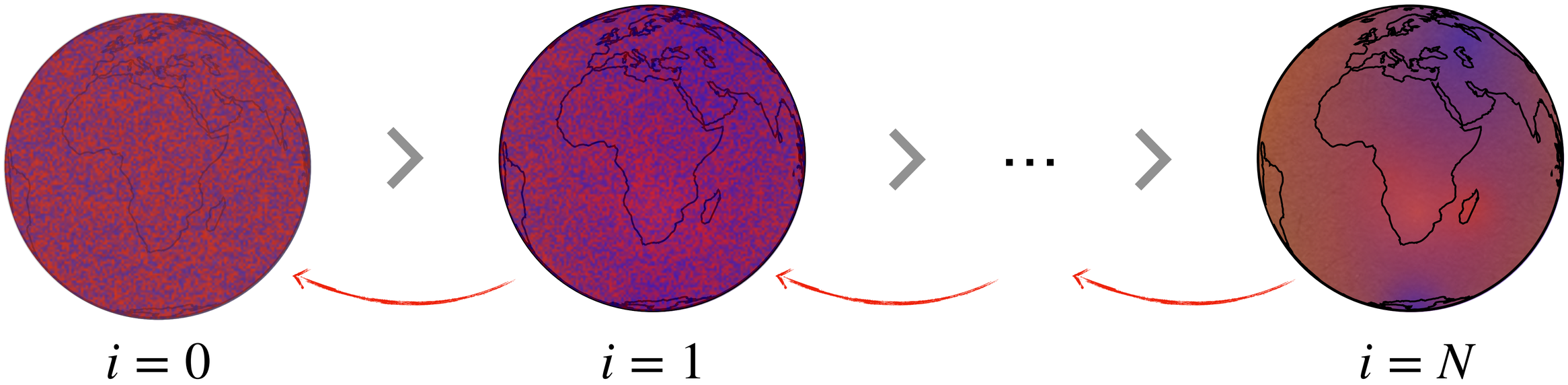
Diffusion Inference

- Starts with random noise as input
- N denoising steps



Diffusion Inference

- Starts with random noise as input
- N denoising steps
- Gradient back propagation computationally infeasible



Attack Algorithm

Gradient of diffusion output

Input: attack budget ϵ, number of attack steps N, learning rate α, number of diffusion steps j, inputs $\mathrm{X}^t, \mathrm{X}^{t-1}$

Output: adversarial perturbation δ^t, δ^{t-1}

$\mathrm{m}_0 \leftarrow 0$

$\delta_0 = (\delta_0^t, \delta_0^{t-1}) \leftarrow 0$

for $i \leftarrow 1$ **to** N **do**

$\tilde{\mathrm{X}}^{t+j} = f(\mathrm{X}^t + \delta_{i-1}^t, \mathrm{X}^{t-1} + \delta_{i-1}^{t-1}, j, n)$

$\mathrm{g}_i \leftarrow \nabla_{\delta_{i-1}} \mathcal{A}(\tilde{\mathrm{X}}^{t+j})$

$\mathrm{m}_i \leftarrow \beta \cdot \mathrm{m}_{i-1} + (1-\beta) \cdot \Pi_1(\mathrm{g}_i)$

$\alpha_i' \leftarrow \frac{\epsilon}{N} + \frac{1}{2}\left(2\epsilon - \frac{\epsilon}{N} \cdot 1 + \cos\frac{(i-1)\cdot\pi}{N}\right)$

$\alpha_i \leftarrow \frac{\alpha_i'}{(1-\beta)^i}$

$\delta_i \leftarrow \Pi_\epsilon(\delta_{i-1} - \alpha_i \mathrm{m}_i)$

return $\delta_N^t, \delta_N^{t-1}$

Must be approximated

Approximated Diffusion Inference

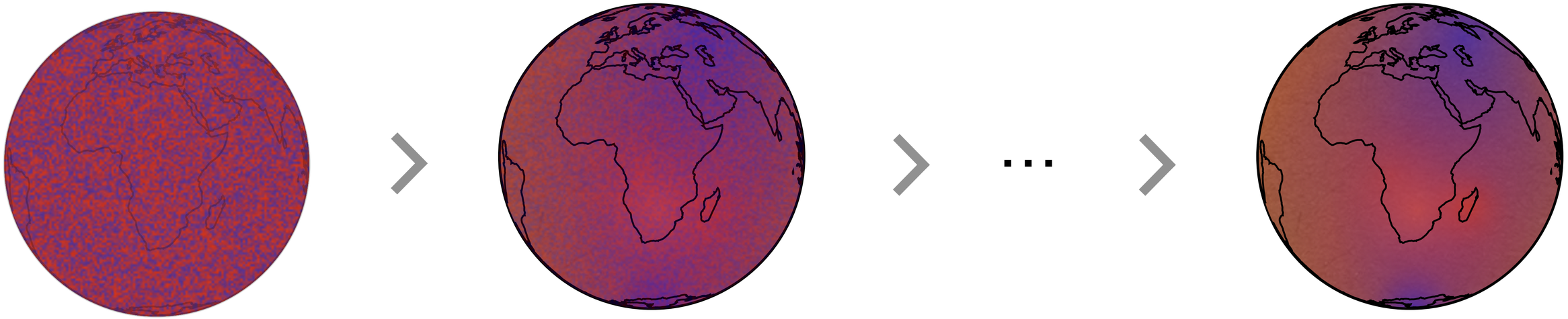
- Sample $k < N$ noise levels $\sigma_0, \ldots, \sigma_{n-1} \sim \left(1, \frac{k-1}{k}\right], \ldots, \left(\frac{1}{k}, 0\right]$

Approximated Diffusion Inference

- Sample $k < N$ noise levels $\sigma_0, \ldots, \sigma_{n-1} \sim \left(1, \frac{k-1}{k}\right], \ldots, \left(\frac{1}{k}, 0\right]$
- Choose k so that gradient calculation is feasible

Approximated Diffusion Inference

- Sample $k < N$ noise levels $\sigma_0, \ldots, \sigma_{n-1} \sim \left(1, \frac{k-1}{k}\right], \ldots, \left(\frac{1}{k}, 0\right]$
- Choose k so that gradient calculation is feasible



σ_0 σ_1 $\sigma = 0$

Evaluation

Evaluation

- Targeted attack

Evaluation

- Targeted attack
- Two scenarios
  - Fabricate extreme forecasts in normal weather

Evaluation

- Targeted attack
- Two scenarios
  - Fabricate extreme forecasts in normal weather
  - Manipulate forecasts of existing extreme weather

Fabricating Extreme Forecasts

Fabricating Extreme Forecasts

- Maximize target *variable* in *region* at *time*

Fabricating Extreme Forecasts

- Maximize target *variable* in *region* at *time*
- Baselines adapted from other domains[1][2]

[1] Liang et al. “Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples”, ICML’23
[2] Chen et al. “Diffusion policy attacker: Crafting adversarial attacks for diffusion-based policies”, NeurIPS’24

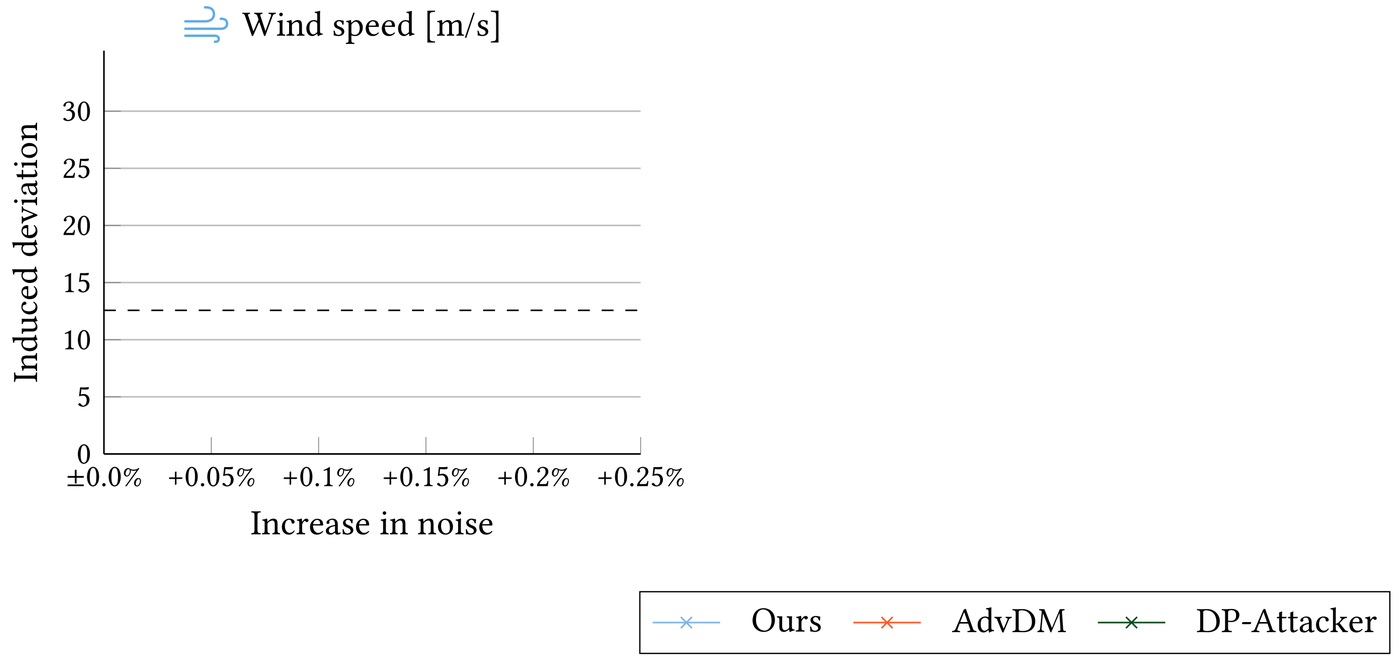
Fabricating Extreme Forecasts

- Maximize target *variable* in *region* at *time*
- Baselines adapted from other domains[1][2]
- Varying strength of perturbation up to single satellite

[1] Liang et al. “Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples”, ICML’23
[2] Chen et al. “Diffusion policy attacker: Crafting adversarial attacks for diffusion-based policies”, NeurIPS’24

Fabricating Extreme Forecasts

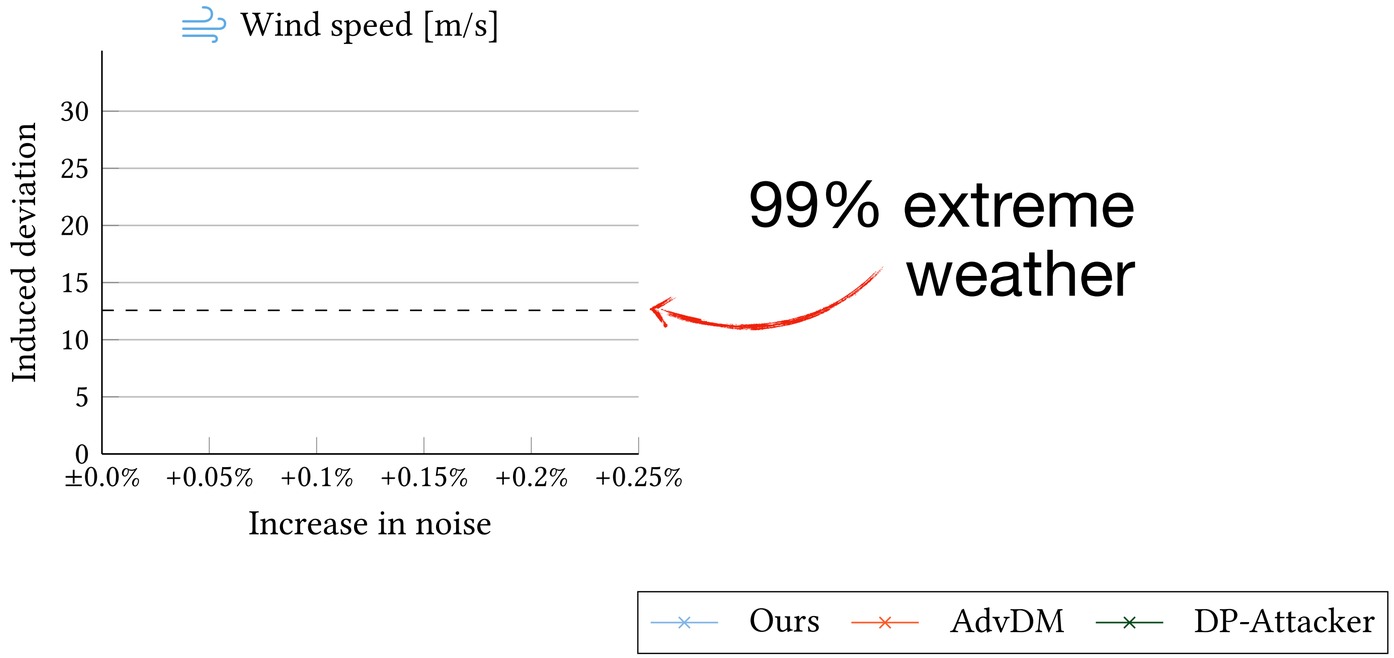
- Maximize target *variable* in *region* at *time*
- Baselines adapted from other domains[1][2]
- Varying strength of perturbation up to single satellite



[1] Liang et al. “Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples”, ICML’23
[2] Chen et al. “Diffusion policy attacker: Crafting adversarial attacks for diffusion-based policies”, NeurIPS’24

Fabricating Extreme Forecasts

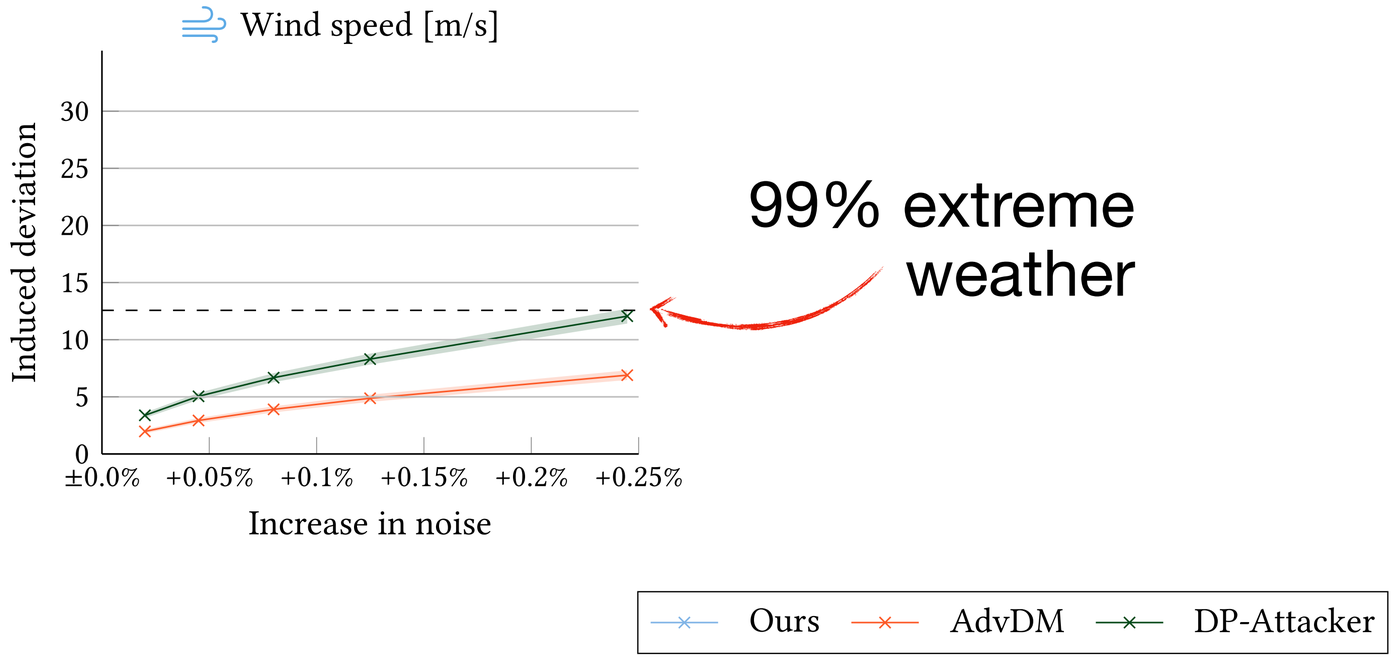
- Maximize target *variable* in *region* at *time*
- Baselines adapted from other domains[1][2]
- Varying strength of perturbation up to single satellite



[1] Liang et al. “Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples”, ICML’23
[2] Chen et al. “Diffusion policy attacker: Crafting adversarial attacks for diffusion-based policies”, NeurIPS’24

Fabricating Extreme Forecasts

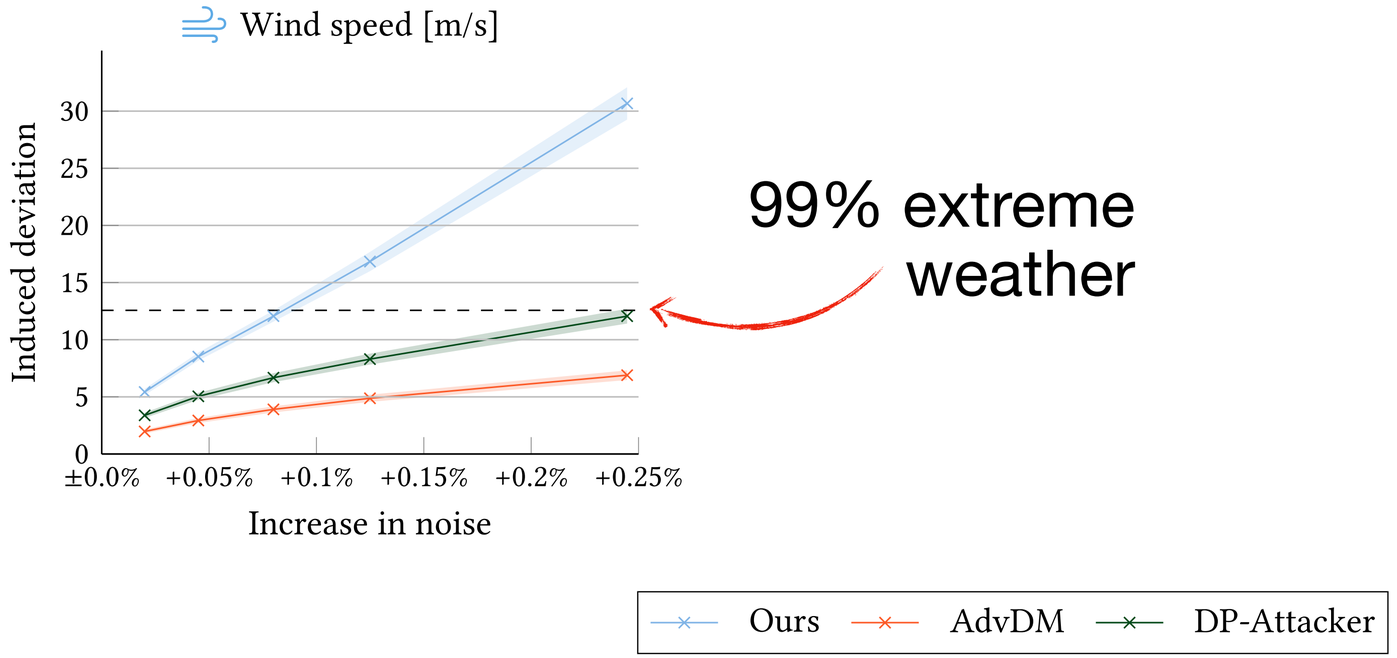
- Maximize target *variable* in *region* at *time*
- Baselines adapted from other domains[1][2]
- Varying strength of perturbation up to single satellite



[1] Liang et al. “Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples”, ICML’23
[2] Chen et al. “Diffusion policy attacker: Crafting adversarial attacks for diffusion-based policies”, NeurIPS’24

Fabricating Extreme Forecasts

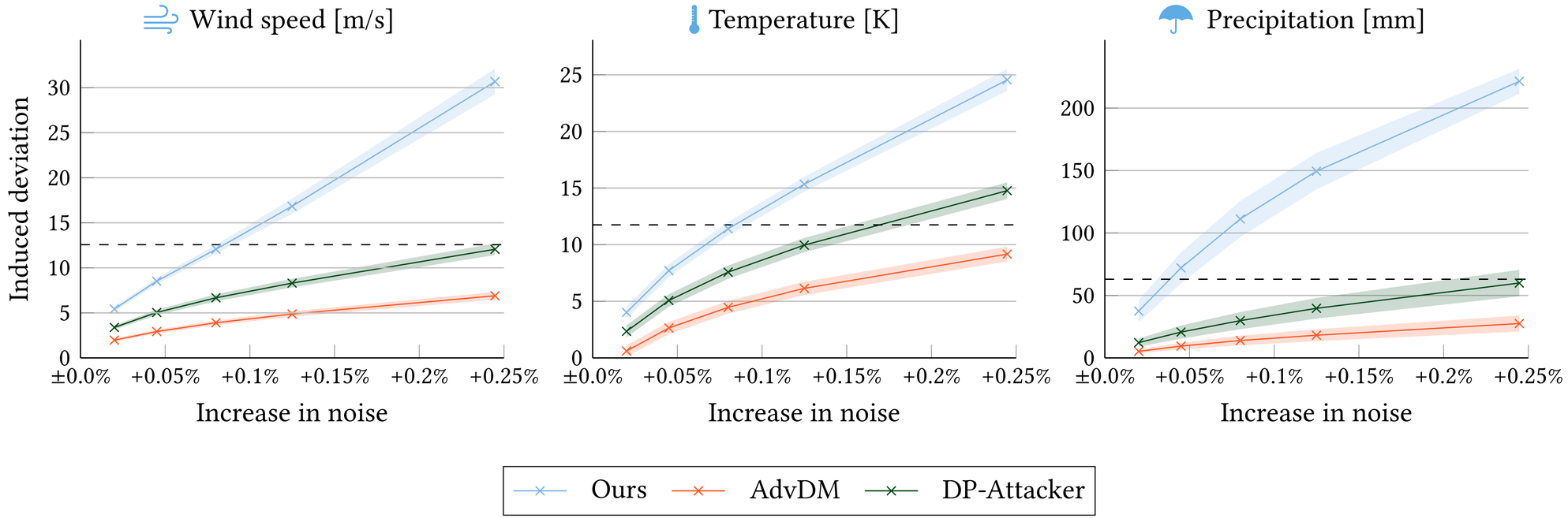
- Maximize target *variable* in *region* at *time*
- Baselines adapted from other domains[1][2]
- Varying strength of perturbation up to single satellite



[1] Liang et al. “Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples”, ICML’23
[2] Chen et al. “Diffusion policy attacker: Crafting adversarial attacks for diffusion-based policies”, NeurIPS’24

Fabricating Extreme Forecasts

- Maximize target *variable* in *region* at *time*
- Baselines adapted from other domains[1][2]
- Varying strength of perturbation up to single satellite

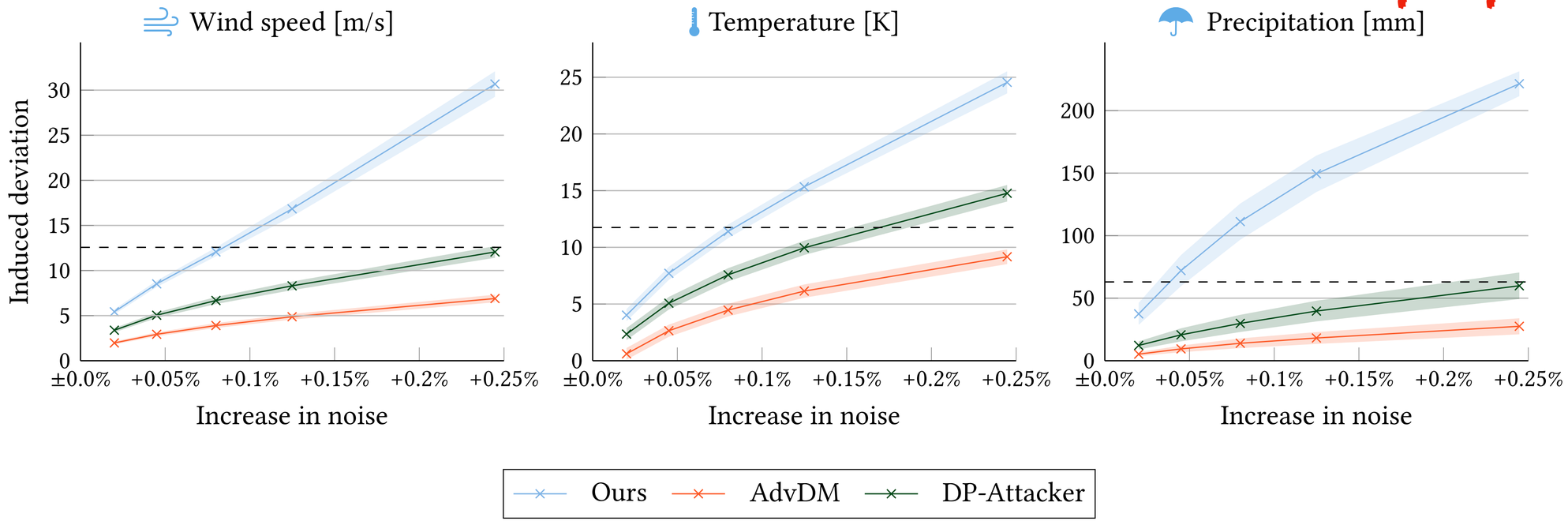


[1] Liang et al. “Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples”, ICML’23
[2] Chen et al. “Diffusion policy attacker: Crafting adversarial attacks for diffusion-based policies”, NeurIPS’24

Fabricating Extreme Forecasts

- Maximize target *variable* in *region* at *time*
- Baselines adapted from other domains[1][2]
- Varying strength of perturbation up to single satellite

More results in the paper!



[1] Liang et al. “Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples”, ICML’23
[2] Chen et al. “Diffusion policy attacker: Crafting adversarial attacks for diffusion-based policies”, NeurIPS’24

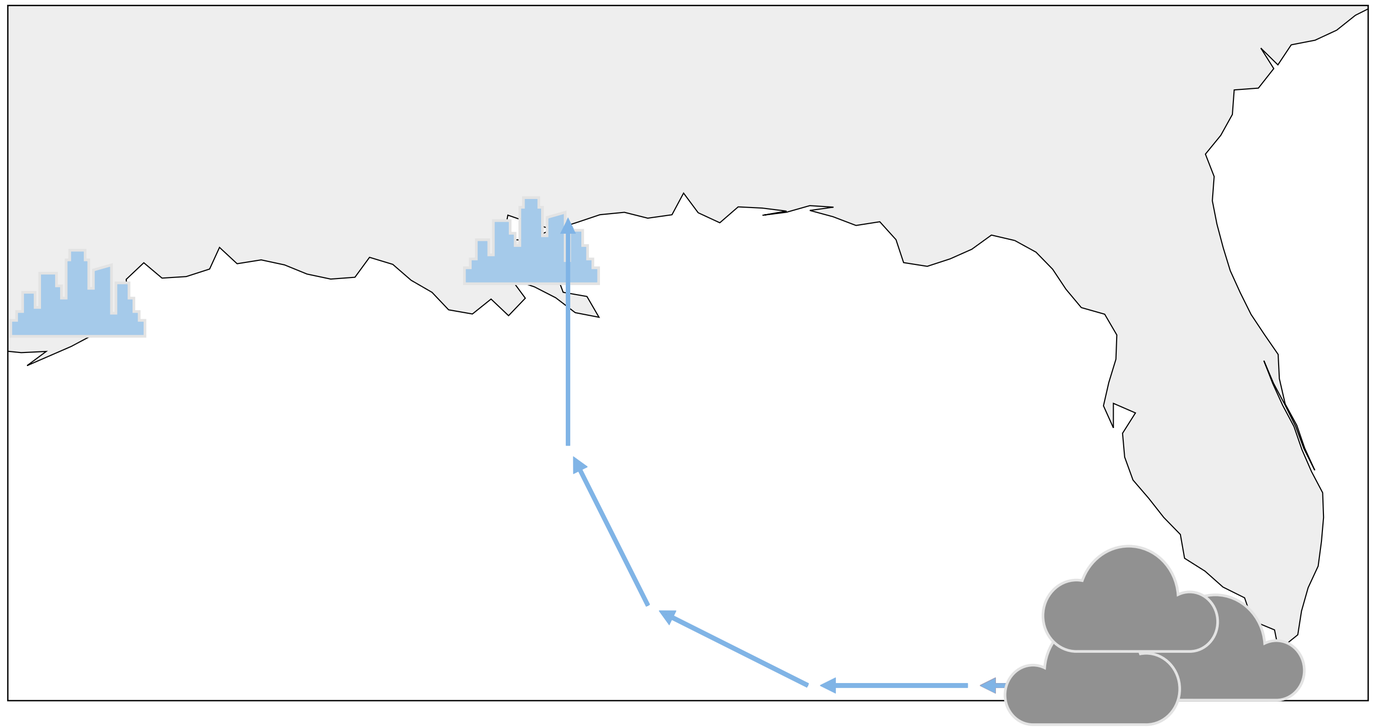
Manipulating Extreme Forecasts

Manipulating Extreme Forecasts

- Apply SotA to historical data (Hurricane Katrina)

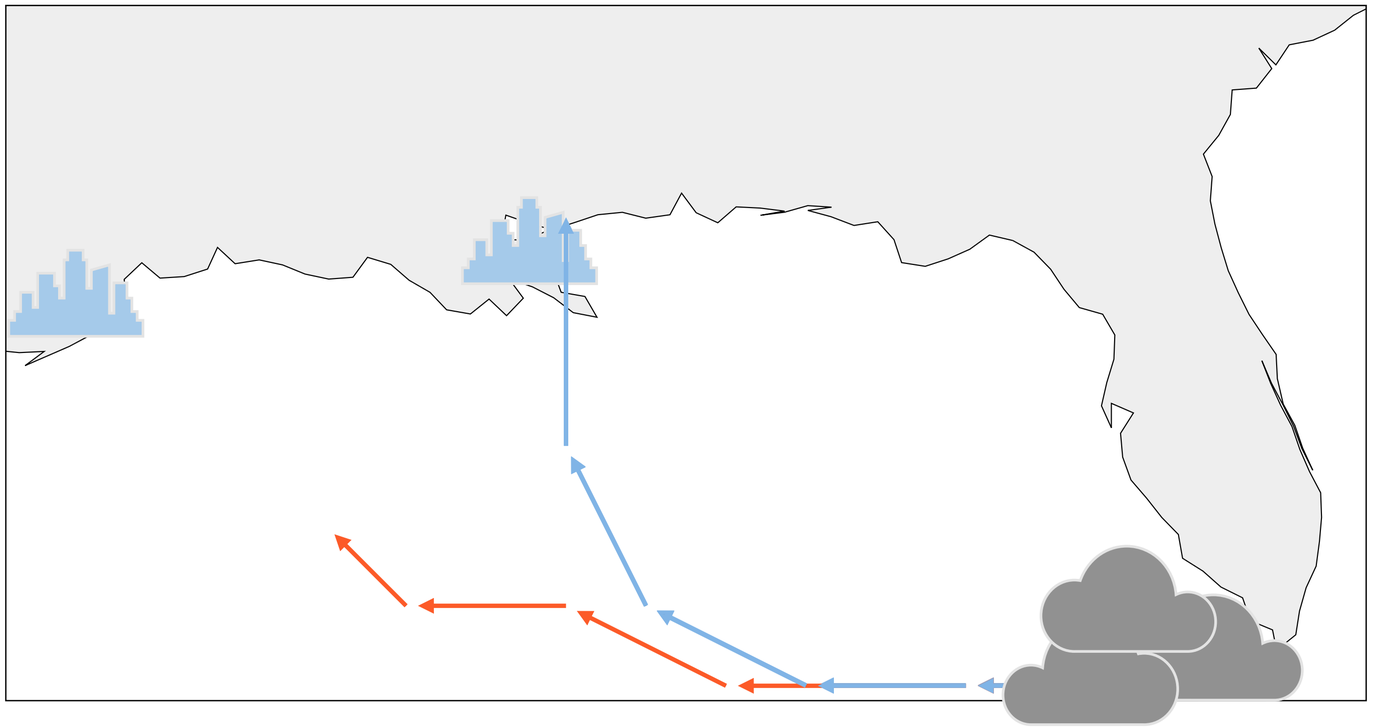
Manipulating Extreme Forecasts

- Apply SotA to historical data (Hurricane Katrina)
- Unperturbed data correctly predicts storm path



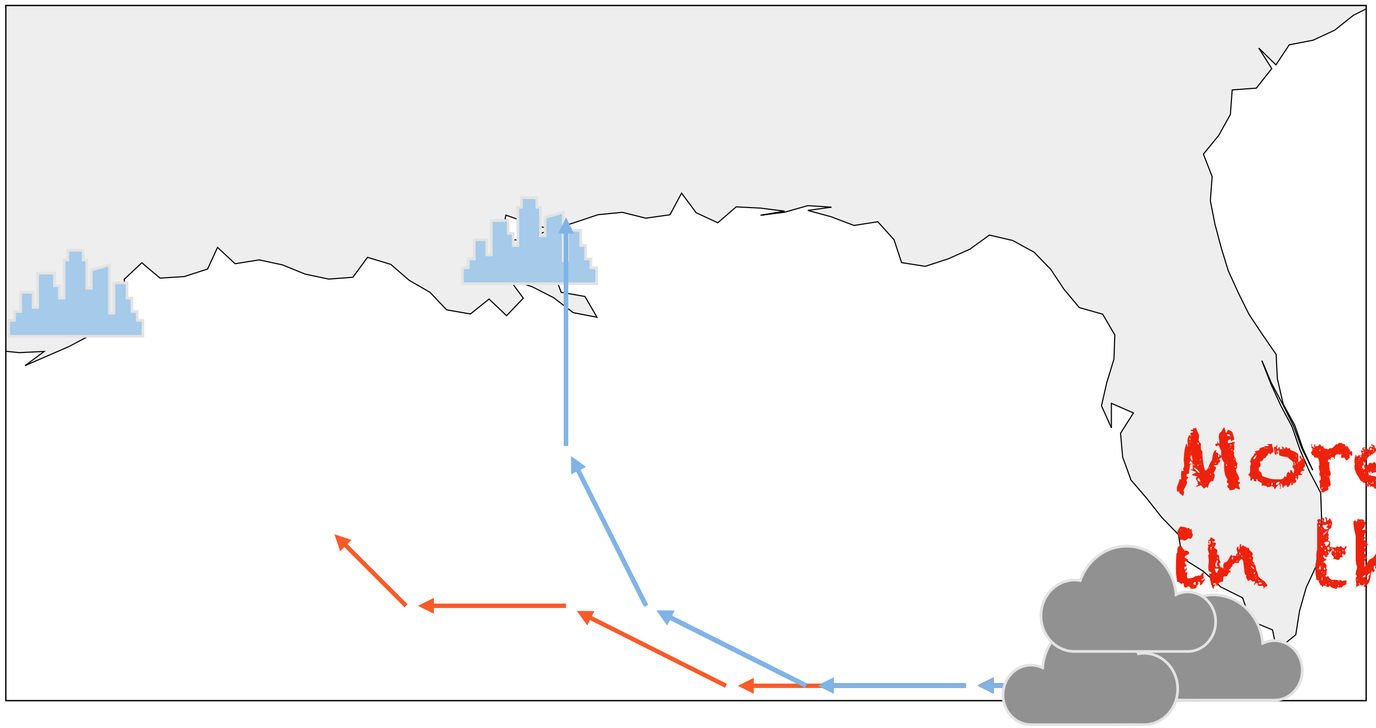
Manipulating Extreme Forecasts

- Apply SotA to historical data (Hurricane Katrina)
- Unperturbed data correctly predicts storm path
- Attacker computes perturbation to change predicted path



Manipulating Extreme Forecasts

- Apply SotA to historical data (Hurricane Katrina)
- Unperturbed data correctly predicts storm path
- Attacker computes perturbation to change predicted path



More results in the paper!

Detection

Detection

$$\overline{X} = X + \underbrace{\mathcal{N}\left(0, \sigma^2\right)}_{\text{Natural Noise}} + \underbrace{\mathcal{N}\left(0, \epsilon^2\right)}_{\text{Attacker}}$$

Detection

$$\overline{X} = X + \mathcal{N}\left(0, \sigma^2\right) + \mathcal{N}\left(0, \epsilon^2\right)$$
$$= X + \mathcal{N}\left(0, \sigma^2 + \epsilon^2\right)$$

Detection

$$\overline{X} = X + \mathcal{N}\left(0, \sigma^2\right) + \mathcal{N}\left(0, \epsilon^2\right)$$

$$= X + \mathcal{N}\left(0, \sigma^2 + \epsilon^2\right)$$

- Assume knowledge of measurement noise

Detection

$$\overline{X} = X + \mathcal{N}\left(0, \sigma^2\right) + \mathcal{N}\left(0, \epsilon^2\right)$$

$$= X + \mathcal{N}\left(0, \sigma^2 + \epsilon^2\right)$$

- Assume knowledge of measurement noise
- Chi-square test to detect increased variance

Detection

$$\overline{X} = X + \mathcal{N}\left(0, \sigma^2\right) + \mathcal{N}\left(0, \epsilon^2\right)$$
$$= X + \mathcal{N}\left(0, \sigma^2 + \epsilon^2\right)$$

- Assume knowledge of measurement noise
- Chi-square test to detect increased variance

Method	Wind Speed	Temperature	Precipitation
AdvDM	>99.99%	99.92%	>99.99%
DP-Attacker	95.04%	45.85%	95.33%
Ours			

Detection Probabilities

Detection

$$\overline{X} = X + \mathcal{N}\left(0, \sigma^2\right) + \mathcal{N}\left(0, \epsilon^2\right)$$
$$= X + \mathcal{N}\left(0, \sigma^2 + \epsilon^2\right)$$

- Assume knowledge of measurement noise
- Chi-square test to detect increased variance

Method	Wind Speed	Temperature	Precipitation
AdvDM	>99.99%	99.92%	>99.99%
DP-Attacker	95.04%	45.85%	95.33%
Ours	**3.07%**	**2.96%**	**0.20%**

Detection Probabilities

Take Aways

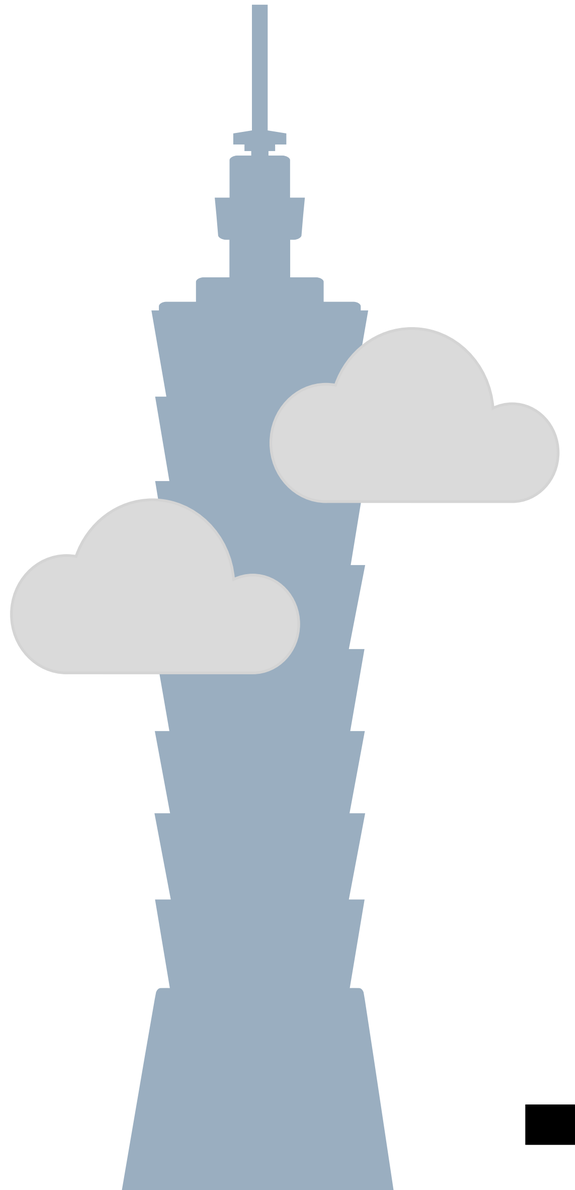
AI-based weather forecasting is vulnerable

- One satellite is enough to control the forecast
- Detection is non-trivial

Diffusion models are vulnerable

- Existing attacks do not suffice
- Better approximation of diffusion process

Paper and code
github.com/mlsec-group/adversarial-observations



Fin