Investigating the impacts of aerosol perturbations with a denoising diffusion model

Jatan Buch

jb4625@columbia.edu

w/ Kara Lamb, Pierre Gentine

Micro2Macro Workshop 2024

Jeff Schmaltz, MODIS, NASA/GSFC

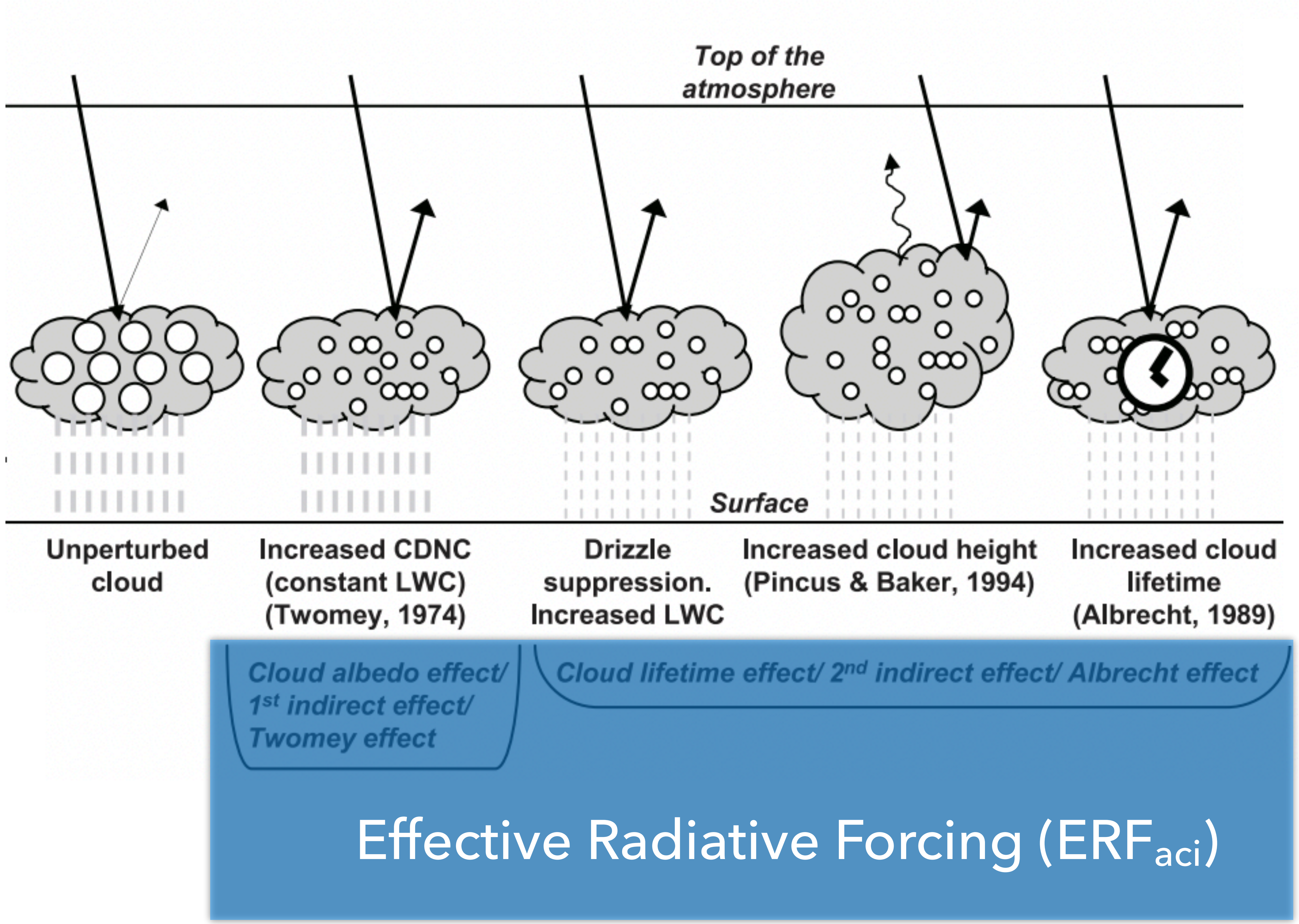
LEAP

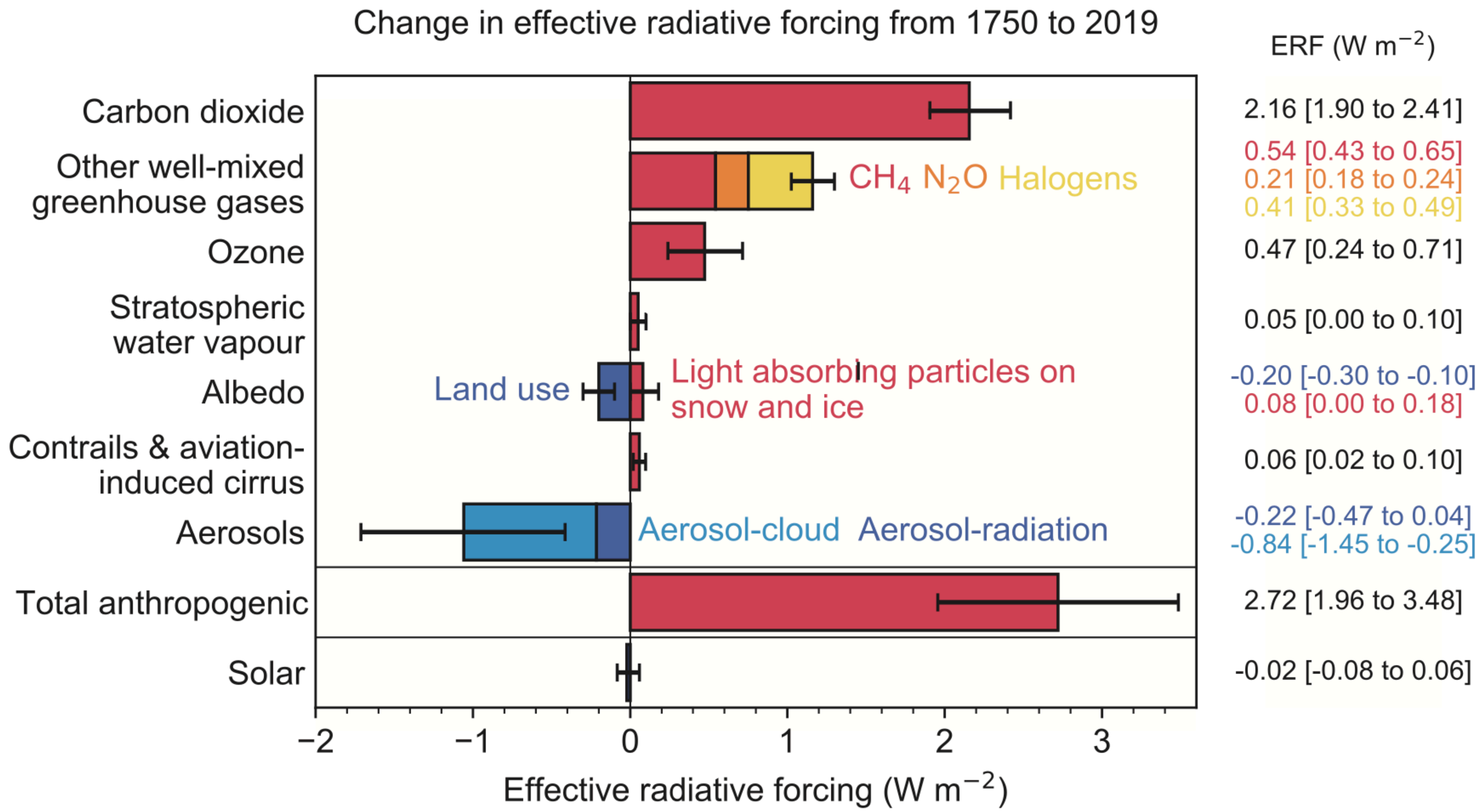
Columbia Engineering — The Fu Foundation School of Engineering and Applied Science

Zegar Family Foundation

Aerosol-cloud interactions (ACI)

- Uncertainty in effective radiative forcing (ERF) due to ACI is the largest source of uncertainty in the historical anthropogenic radiative forcing since the pre-industrial era





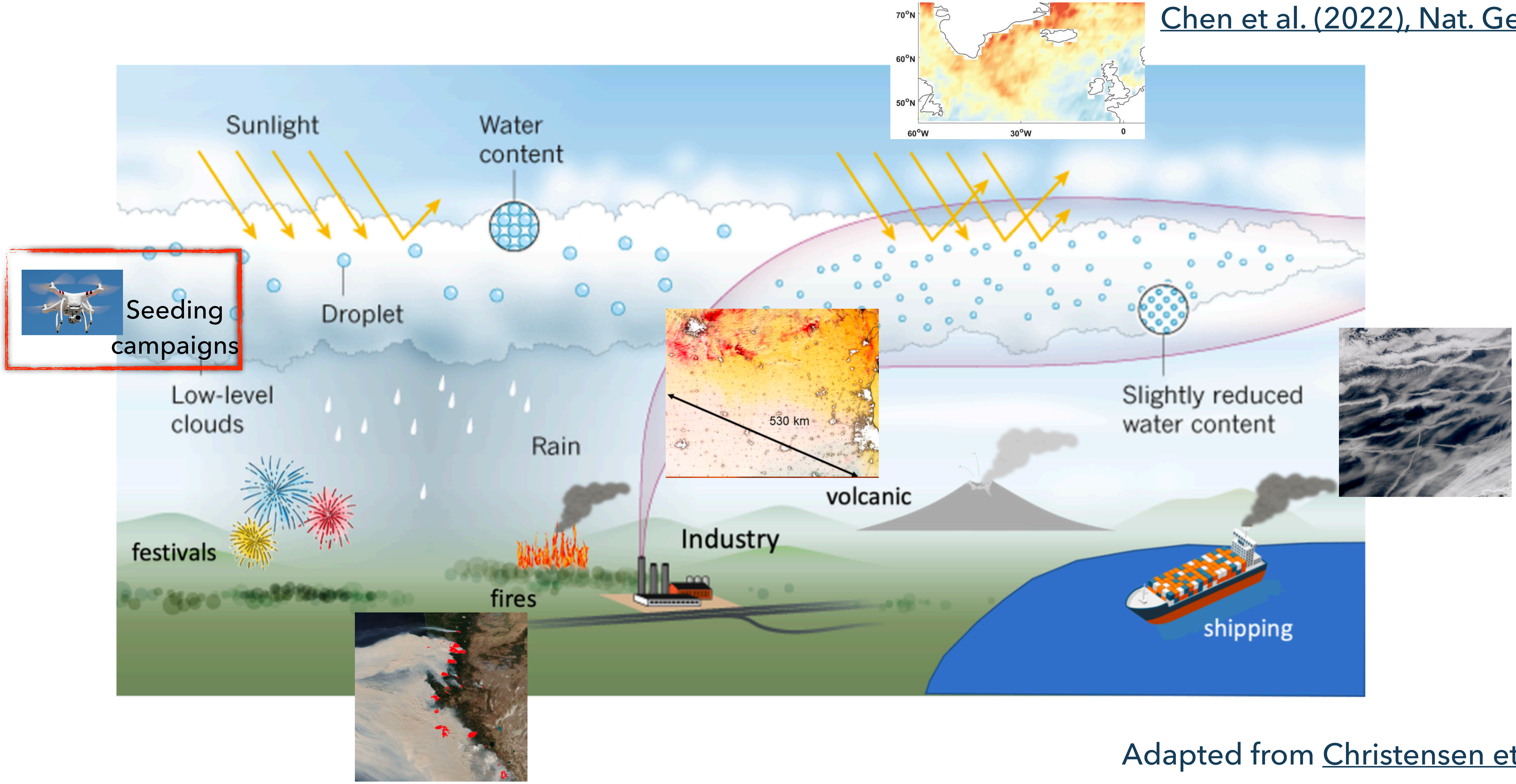
Aerosol perturbations as experiments of opportunity

- Natural and anthropogenic aerosol perturbations (or lack thereof due to COVID-19, say) could overcome the confounding due to meteorological co-variability of aerosols and cloud microphysical properties

Chen et al. (2022), Nat. Geosci.

SNOWIE

CLOUDLAB

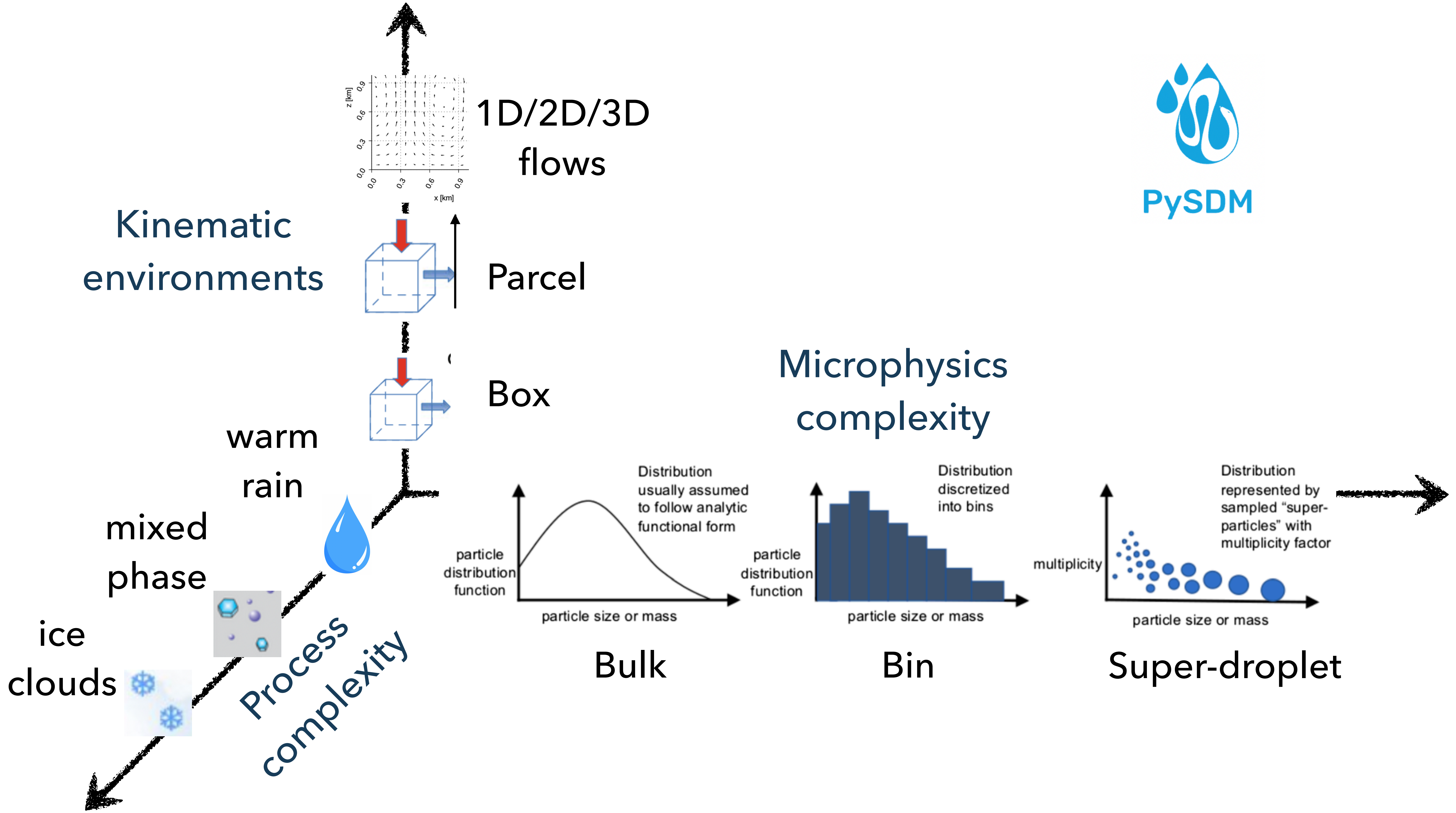


Adapted from Christensen et al. (2020), ACP

Traversing the model hierarchy

Jeevanjee et al. (2017), JAMES
Morrison et al. (2020), JAMES
Korolev et al. (2022), GRL

- Interplay between different scales: kinematic environments, representation of hydrometeor sizes and masses, process complexity, spatiotemporal resolution, as well as coupled processes



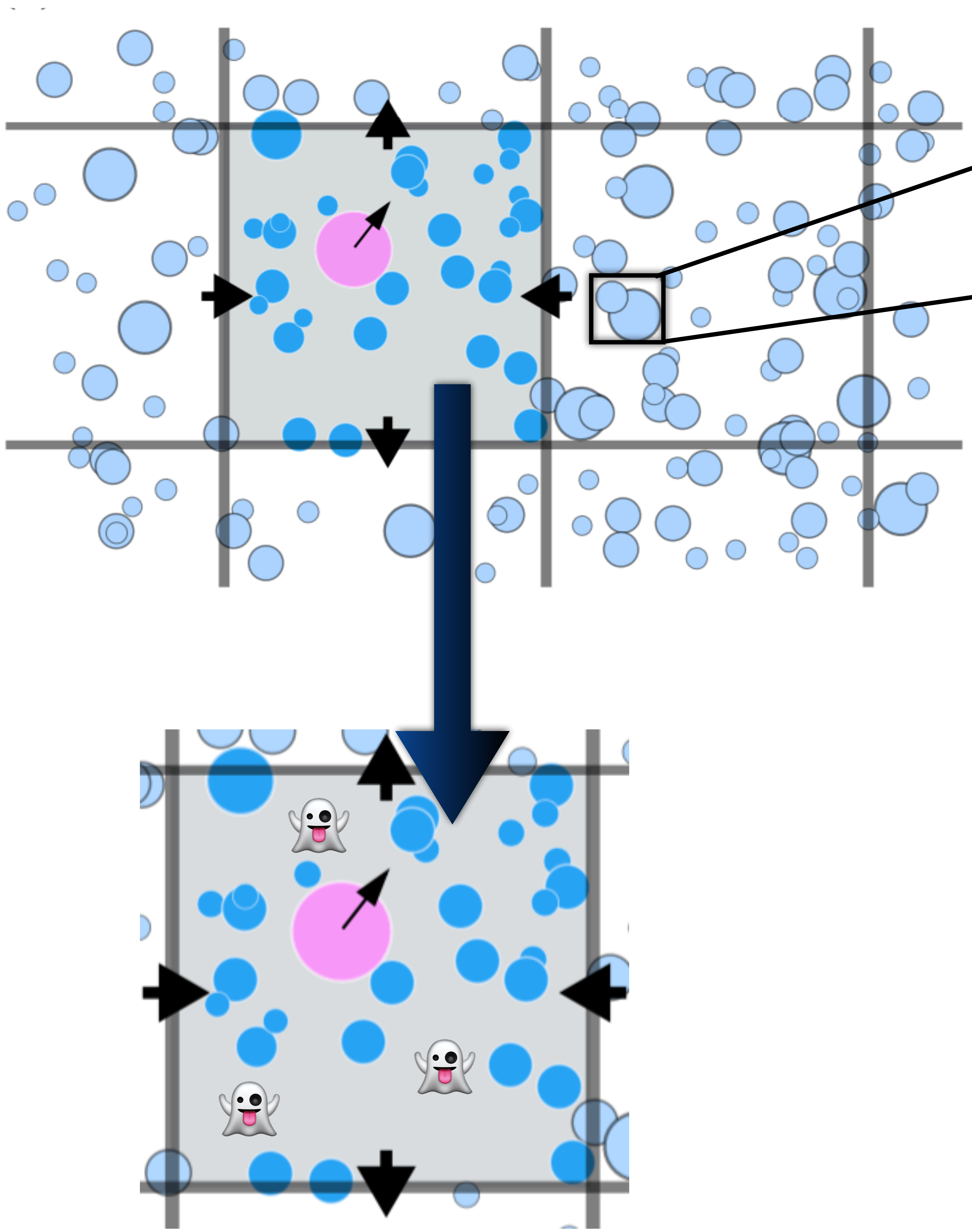
Two questions

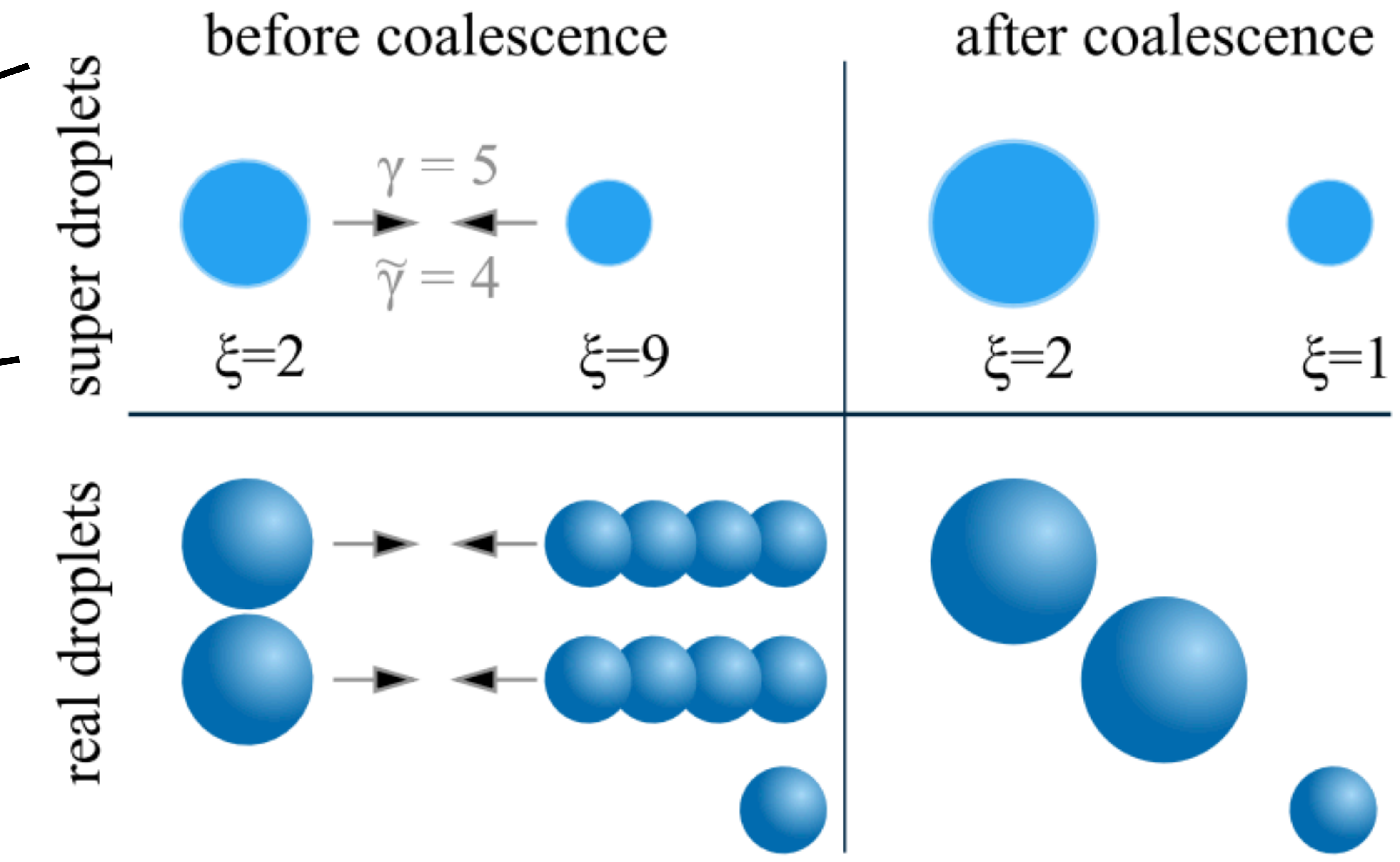
- How do we determine the trajectory, i.e. number of aerosol perturbation steps in space and time, that optimizes a given cloud property despite only having access to expensive numerical simulations or multimodal observations?
    - ➡ This talk: deriving optimal aerosol perturbation trajectories that maximize surface rainfall using warm rain super-droplet microphysics in an idealized 1D prescribed kinematic flow

- Can we use statistical and machine learning methods to design future (model and field) experiments for reducing ACI uncertainty due to microphysical processes?

PySDM

Seeding in super-droplet microphysics (SDM)



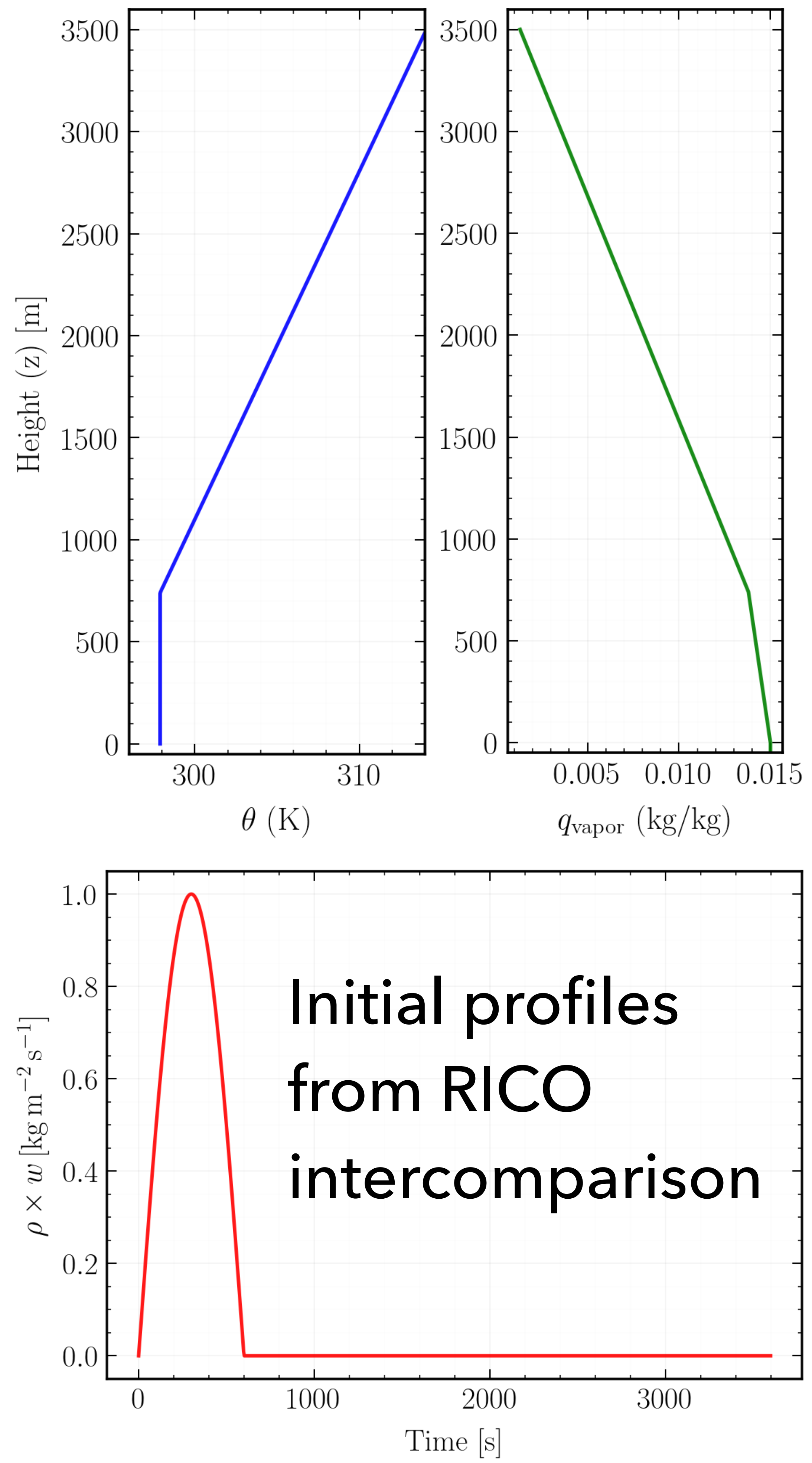


- Seeding is now incorporated in the latest PySDM update (h/t C. Singer and S. Arabas) for different kinematic environments

- 👻 –> ‘ghost’ super-droplets that can be initialized with the desired mulitplicity and radius as well as specific location and time

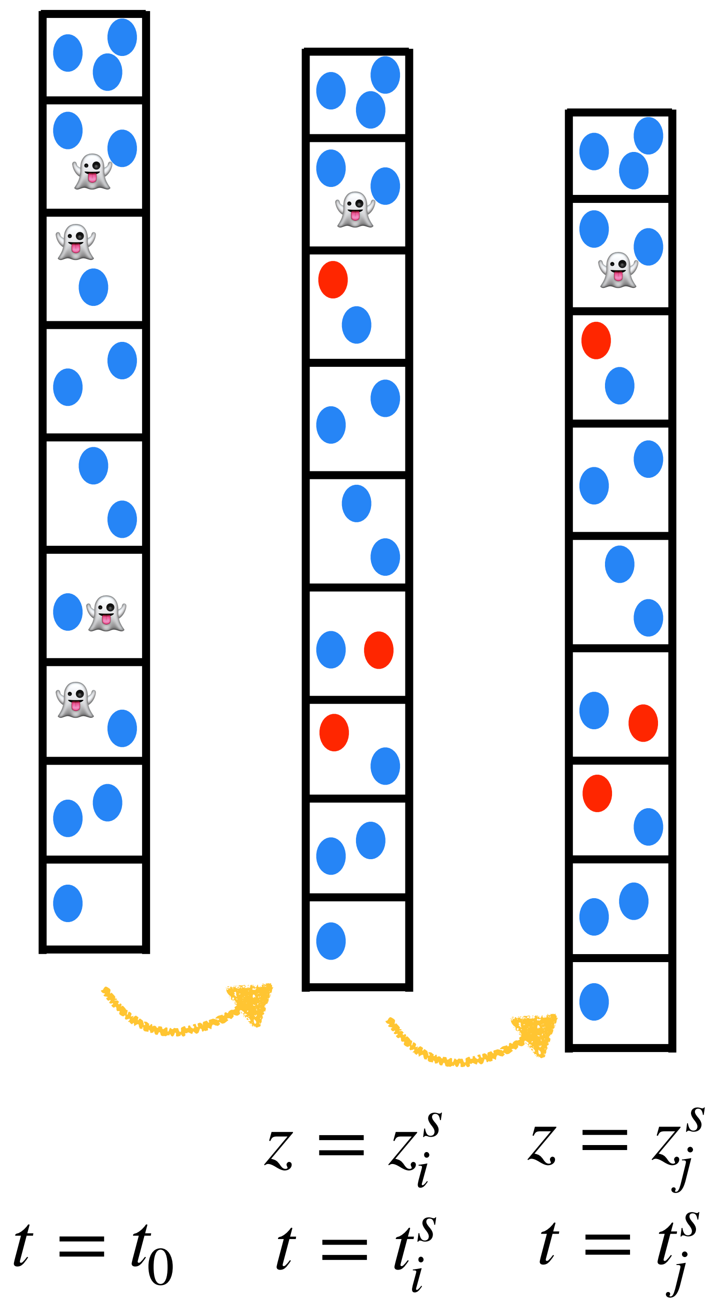
Seeding in the 1D Kinematic Driver

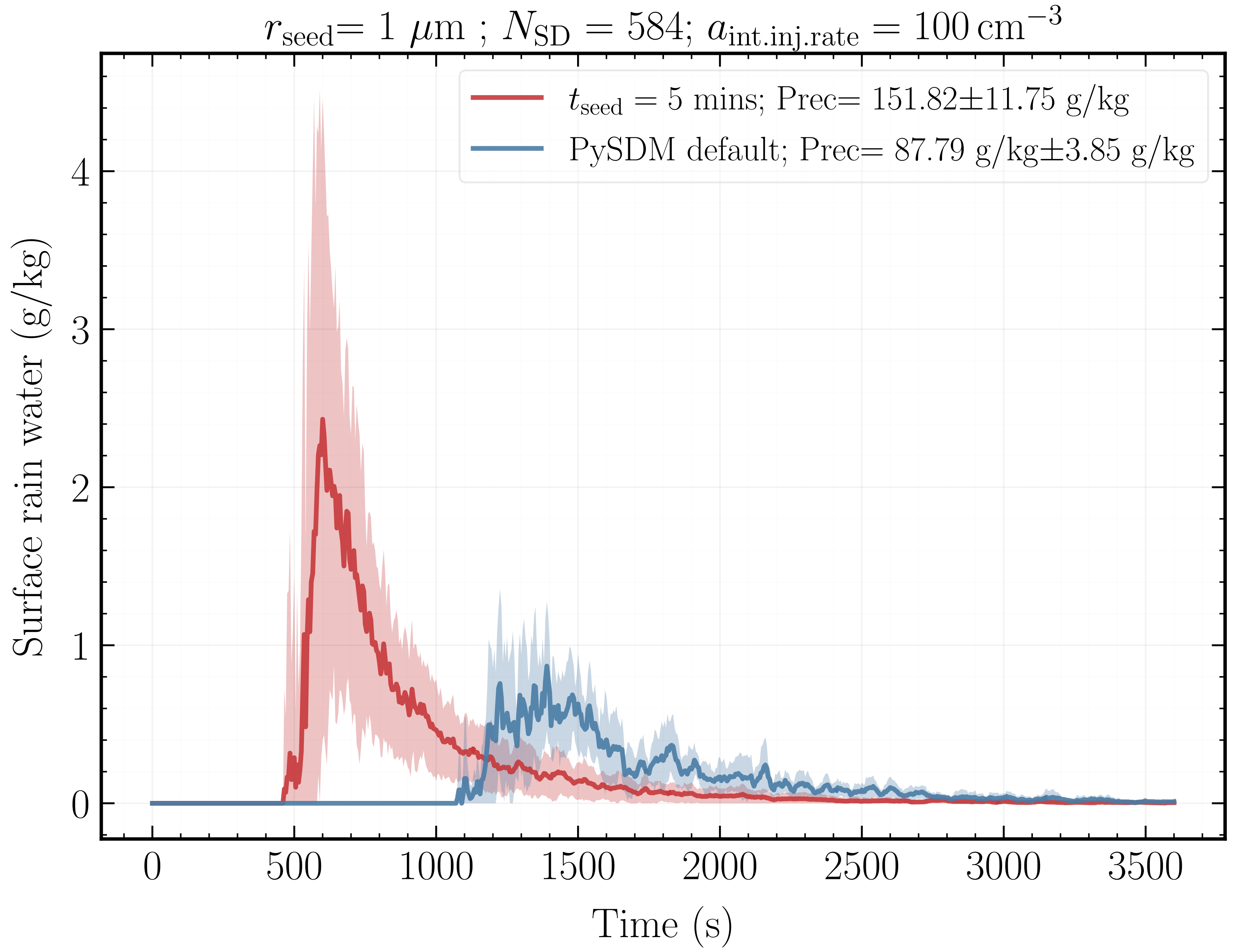
- Injecting larger, more hygroscopic seed aerosols after initializing the kinematic driver (KiD) model yields statistically significant rainfall over the benchmark case with only background aerosols



● $\sim \mathtt{Lognormal}(n_{\mathrm{bkg}} = 10\,\mathrm{cm}^{-3}, r_{\mathrm{bkg}} = 0.1\,\mathrm{nm}, \kappa = 0.3)$

● $\sim \mathtt{Lognormal}(n_{\mathrm{seed}} = 100\,\mathrm{cm}^{-3}, r_{\mathrm{seed}} = 1\,\mu\mathrm{m}, \kappa = 0.8)$



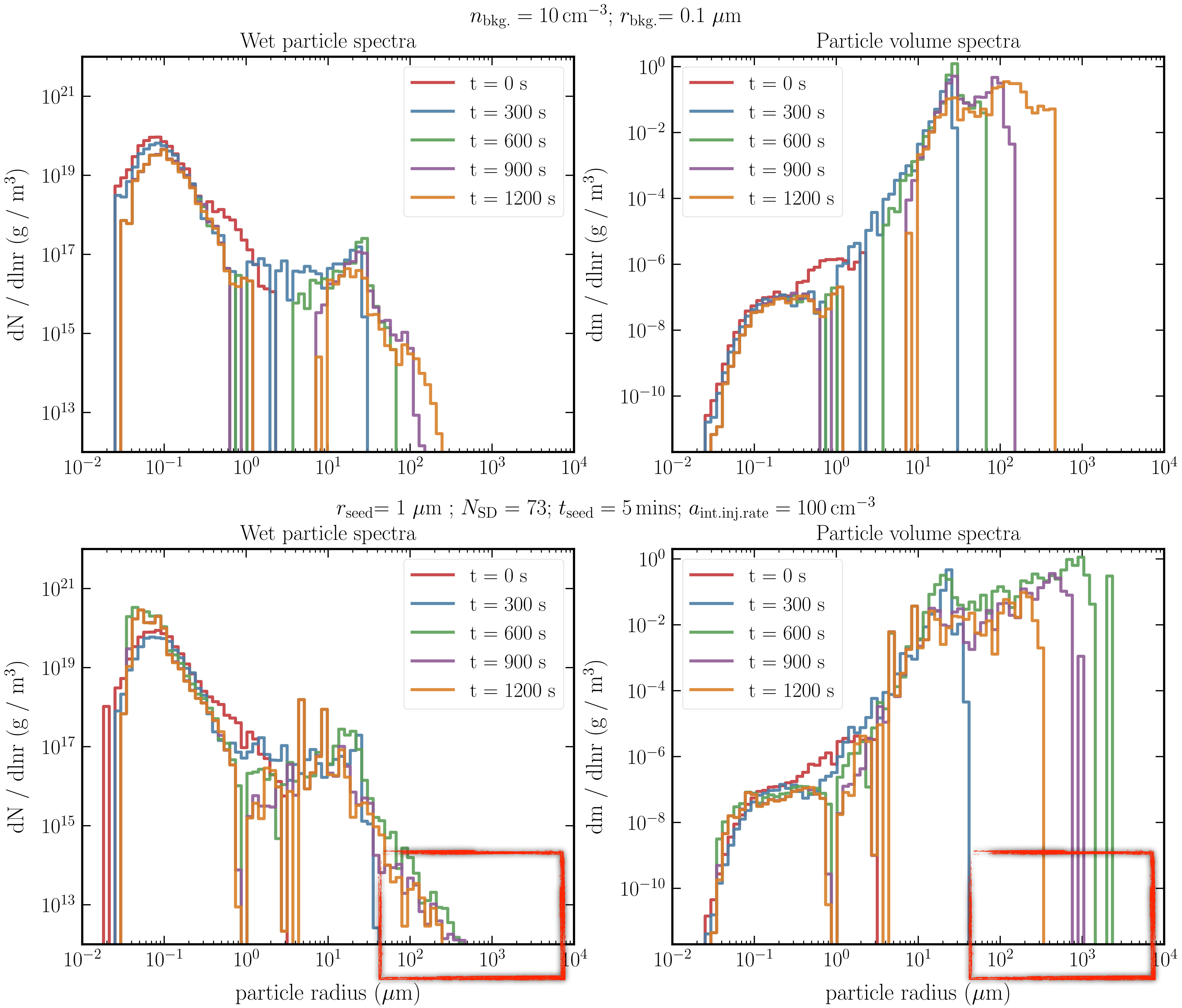


Code: https://github.com/jtbuch/PySDM

Broadening of droplet size distribution

Benchmark case (no seeding)

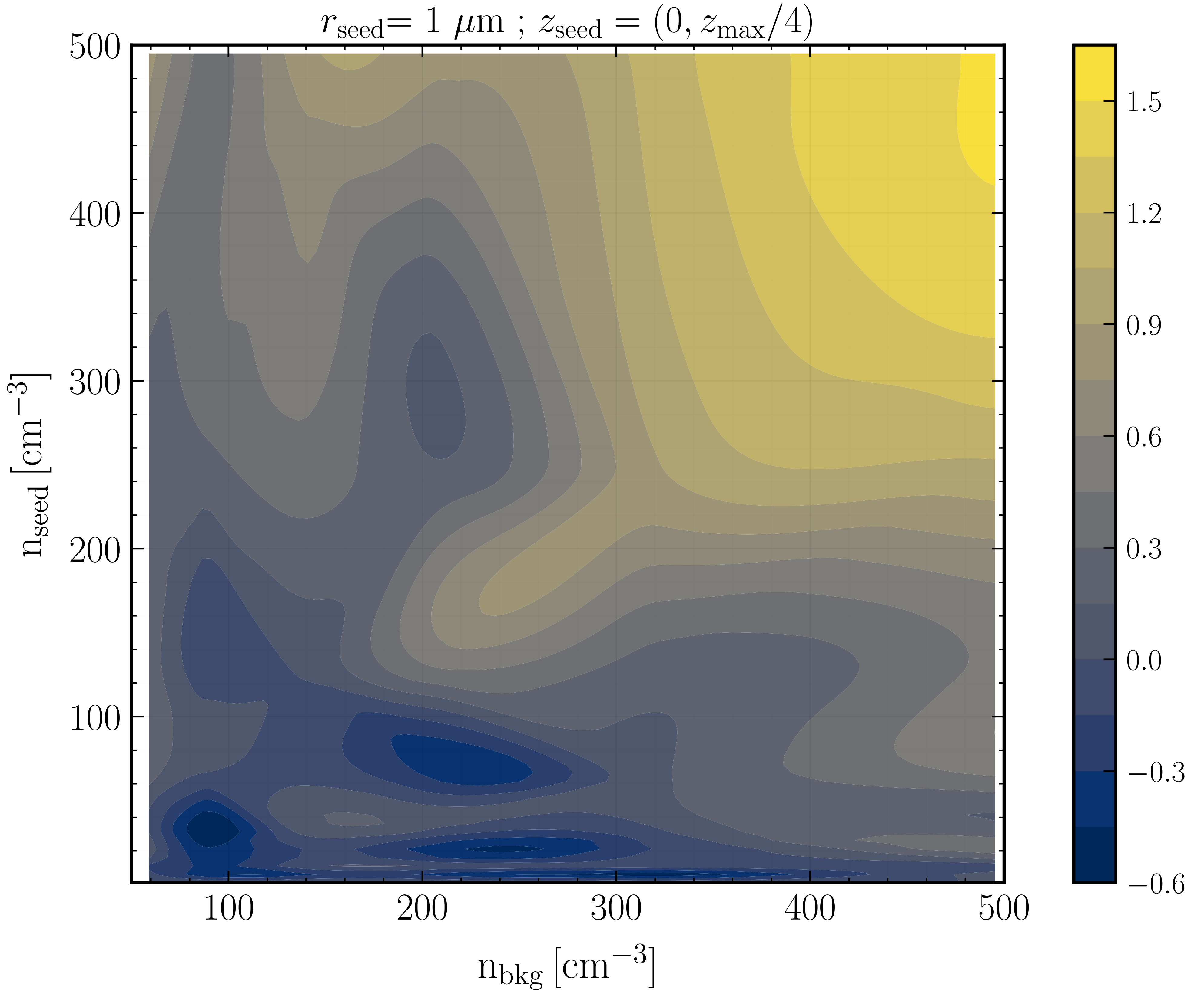
Seeding at t = 300s creates a increase in the higher radius bins at ~600 – 900s



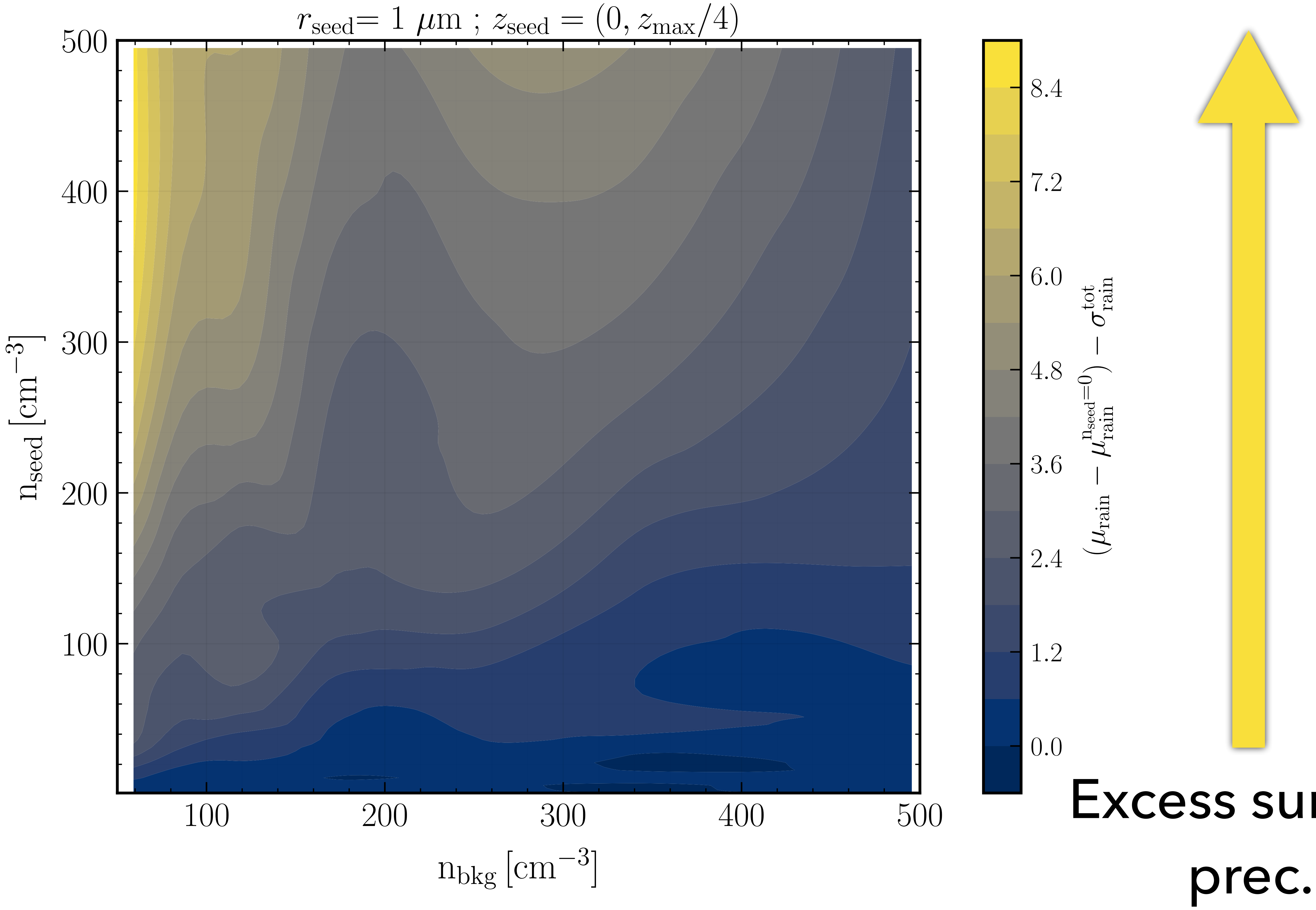
Code: https://github.com/jtbuch/PySDM

Visualizing seeding parameter space

Bimodal seed distribution at initialization



Seeding at $t_{seed} = 10\,\mathrm{mins}$

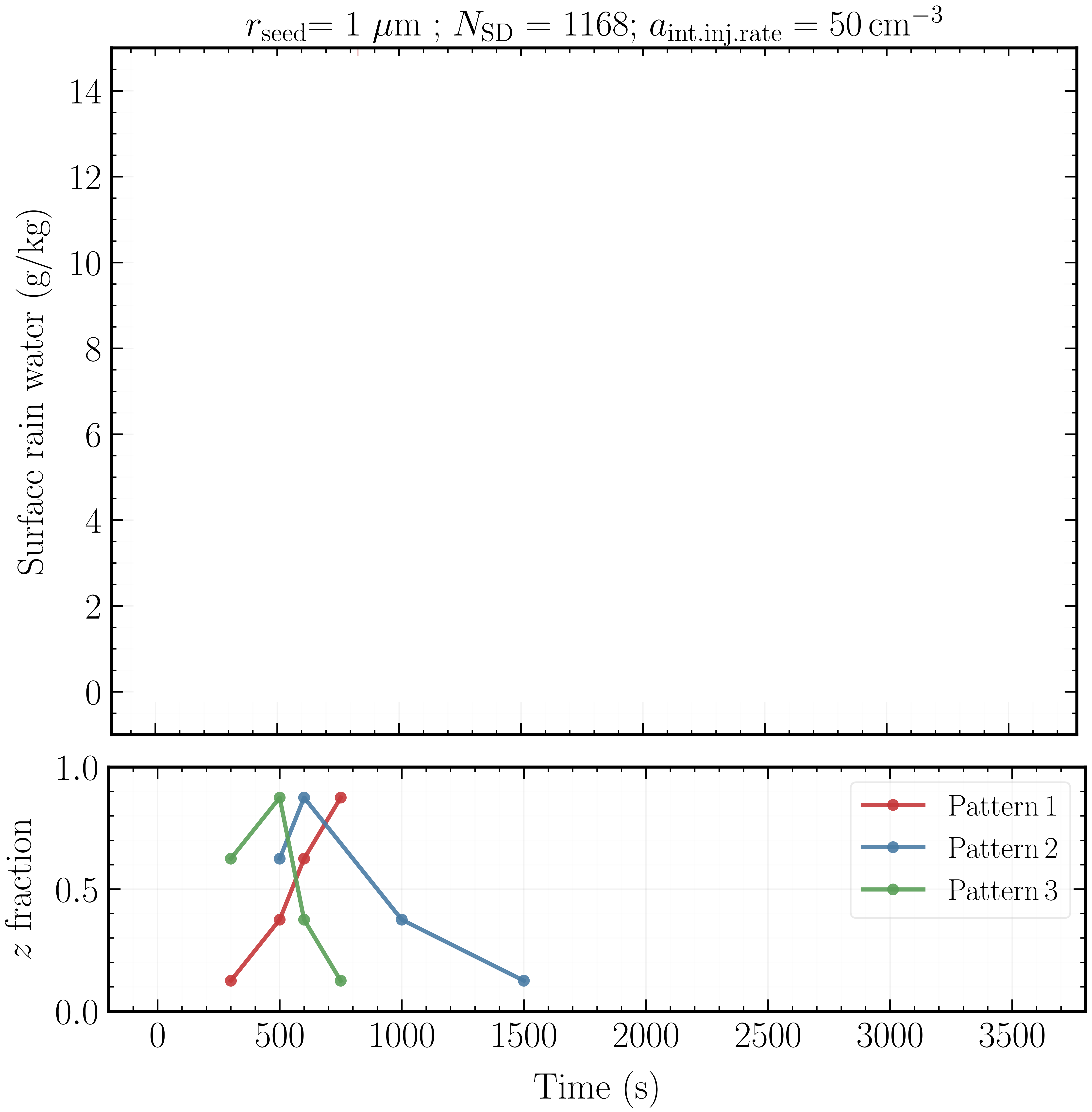


Excess surface prec.

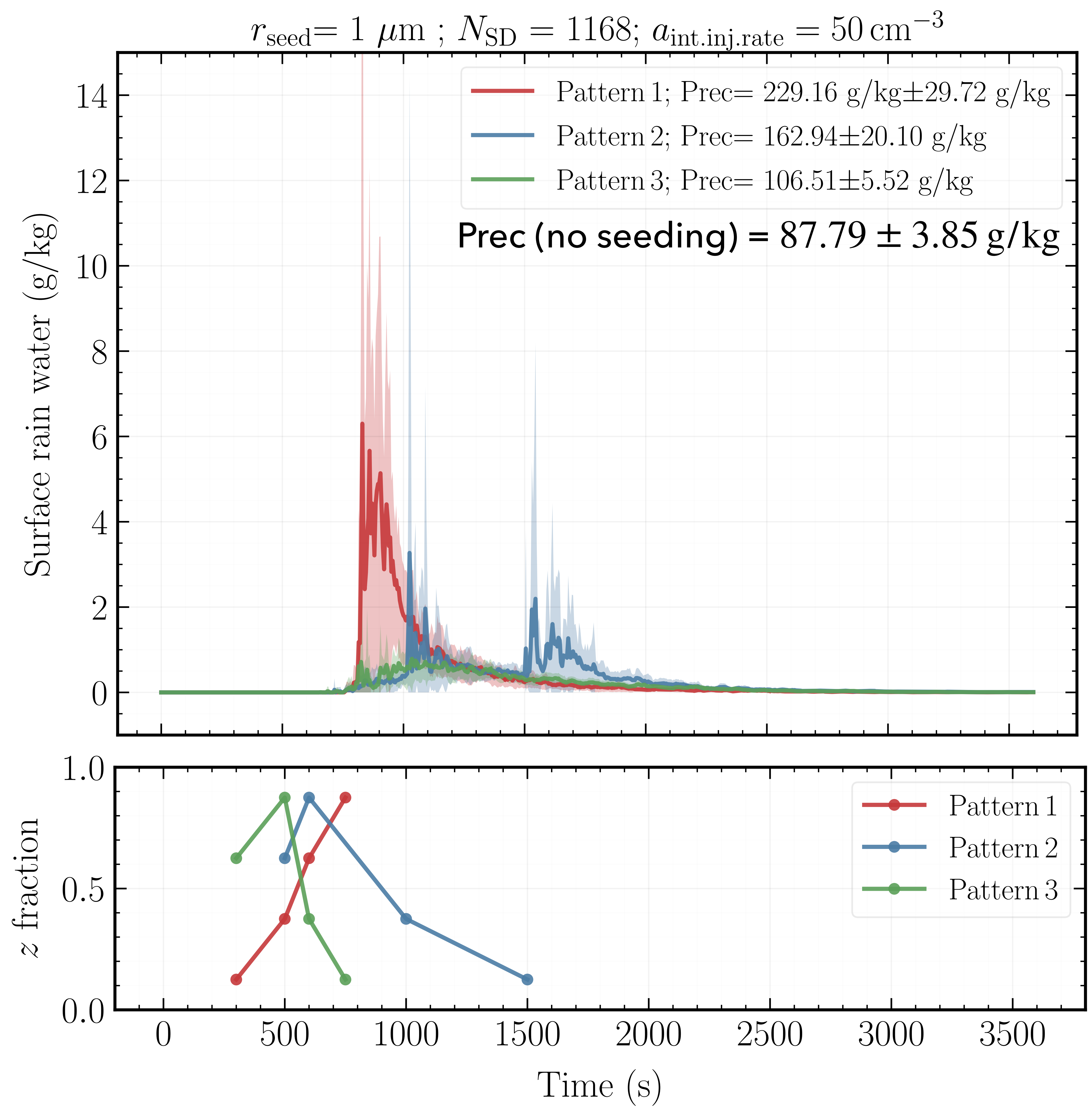
- Moreover, seeding after initialization yields significantly more precipitation as background droplets are growing through collision-coalescence

Code: https://github.com/jtbuch/PySDM

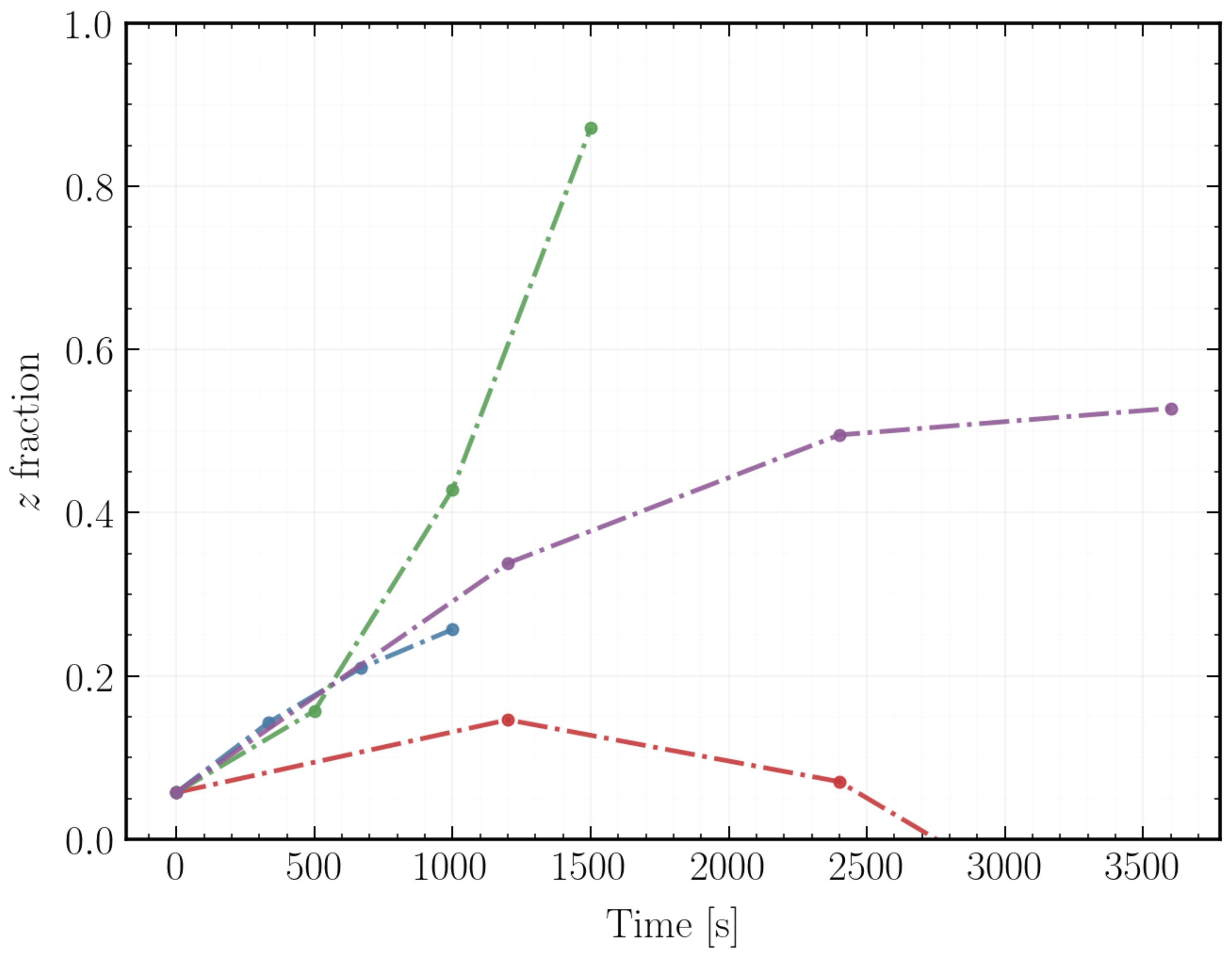
Surface rain is sensitive to perturbation trajectory



Surface rain is sensitive to perturbation trajectory



- To enable optimization without a reinforcement learning framework, we parameterize each trajectory as a function of height and time.
- That is, for n linearly separated time points,
$$t_n = t_0 + n\Delta t$$
we inject aerosols at the following heights,
$$z(t) = z_0 + z_1 t + z_2 t^2 + z_3 t^3 \,.$$



Offline black-box optimization (BBO)

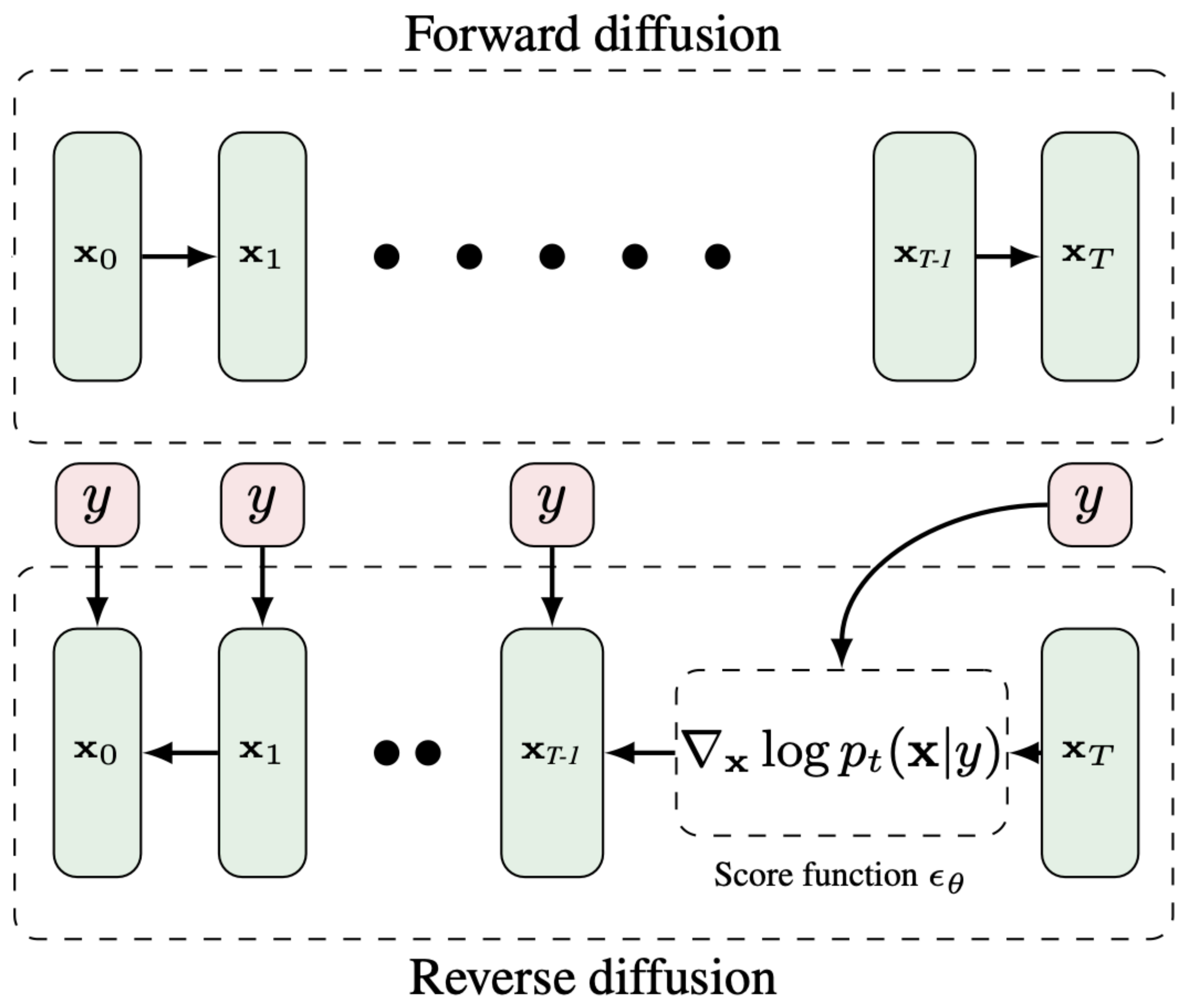
- Traditional BBO methods have relied on *forward* modeling, i.e. constructing a surrogate model followed by gradient-based optimization of its inputs/parameters

- Here, we utilize a novel *inverse* modeling approach to find a point in the high-dimensional input space, $\mathbf{x}$ that maximizes the black-box function,

$$\mathbf{x}^* = \text{argmax}_{\mathbf{x} \in \mathcal{X}} \, f(\mathbf{x})$$

- Step 1: Given a *offline* data set, $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots\}$, train a conditional diffusion model to learn the inverse map, $p(\mathbf{x} \,|\, y)$

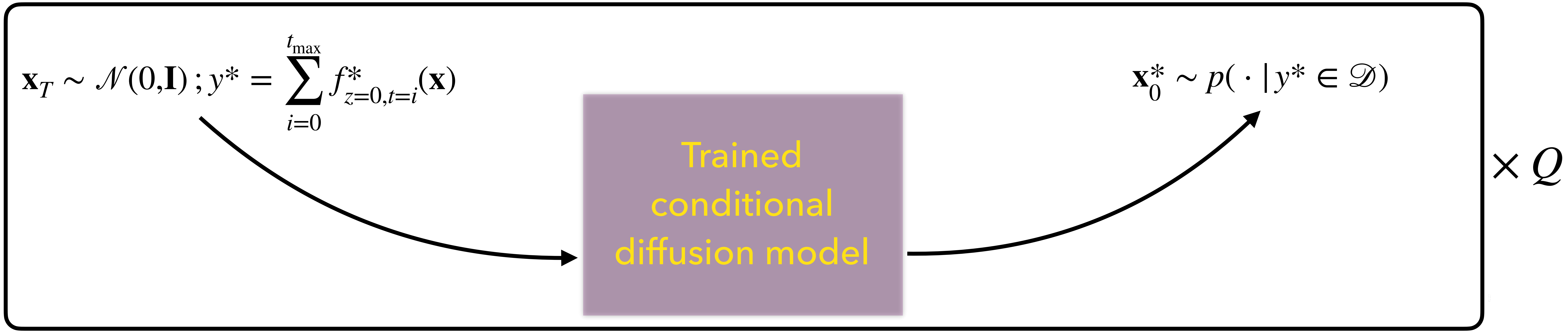
- Step 2: During testing, use a set of Q query points to sample optimal $\mathbf{x}$ values from the trained model



Score function is trained with weighted samples to emphasize higher y values,

$$\mathbb{E}_t \left[\lambda(t) \mathbb{E}_{\mathbf{x}_0, y} \left[w(y) \, \mathbb{E}_{\mathbf{x}_t|\mathbf{x}_0} \left[\left\| \epsilon_{\theta,\gamma}(\mathbf{x}_t, t, y) - \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t \,|\, \mathbf{x}_0) \right\|_2^2 \right] \right] \right]$$

Optimizing perturbation trajectories



- Cumulative surface rainfall is simulated using PySDM with background parameters set to:

$$\mathrm{n_{bkg}} = 100\,\mathrm{cm}^{-3}, r_{\mathrm{bkg}} = 0.1\,\mathrm{nm}, \mathrm{n_{seed}} = 200\,\mathrm{cm}^{-3}, r_{\mathrm{seed}} = 1\,\mu\mathrm{m}$$

- The diffusion-based BBO model is trained and validated on $N \sim 15000$ input-output pairs, $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots\}$, where $\mathbf{x} = (z_0, \ldots, z_3, t_0, \Delta t, n)$ and y is excess rainfall

- For inference and evaluation, Q = 100 points are sampled from the last step of the diffusion model conditioned on the maximum rainfall in $\mathcal{D}$

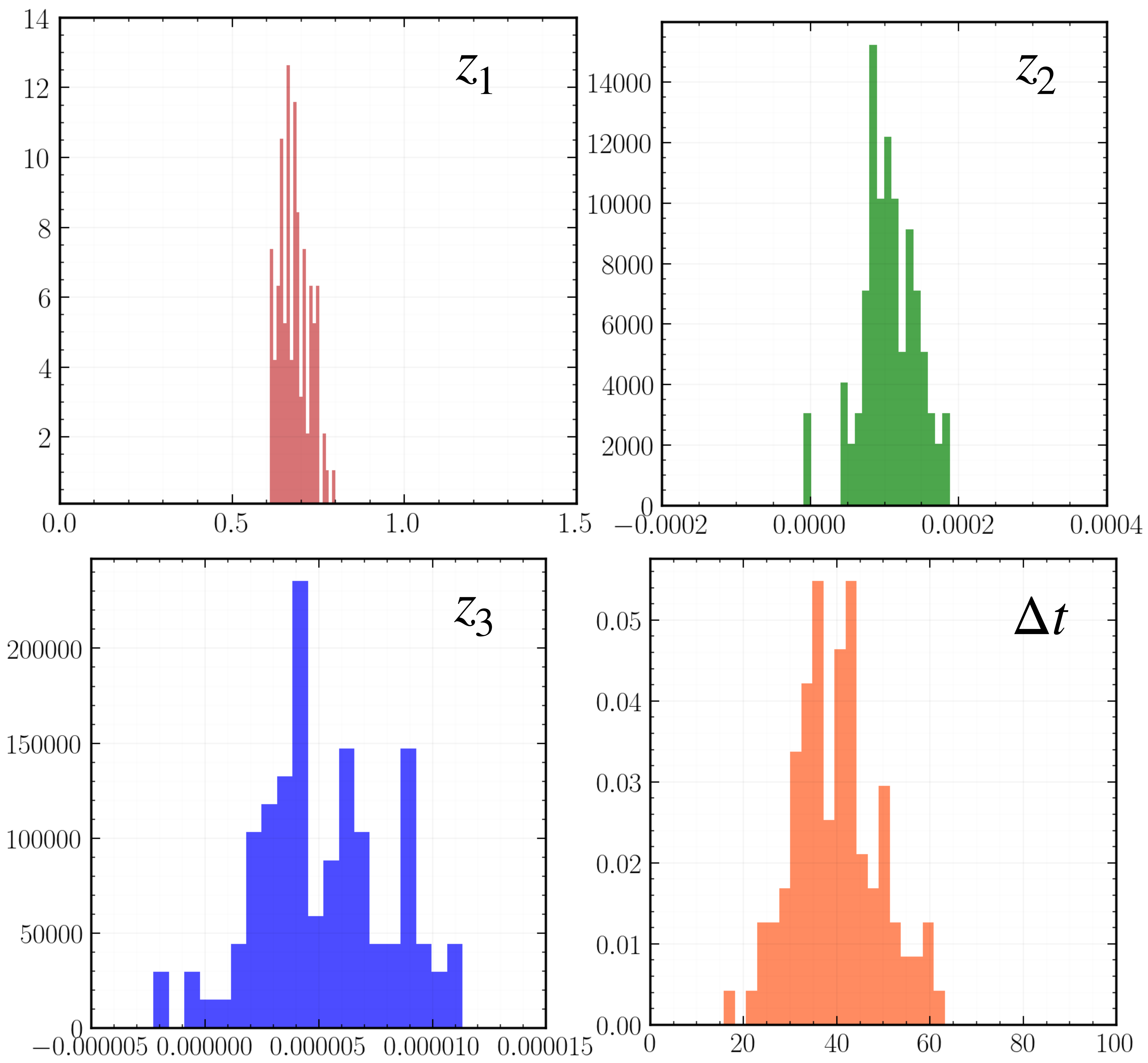
Optimization results for 1D case

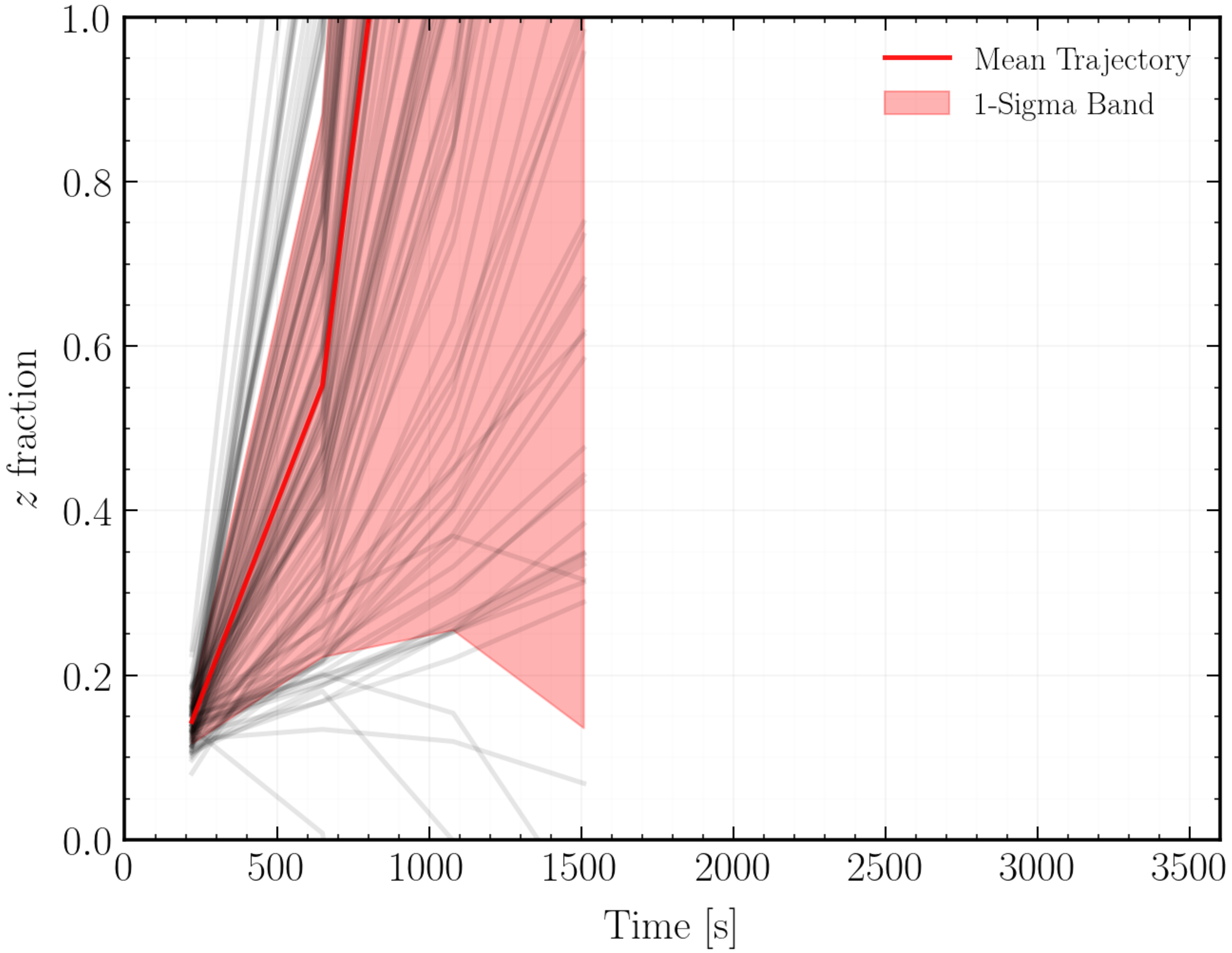
- We draw Q = 100 samples the trained model evaluated at the maximum function value in our offline dataset,

$$\mathbf{x}^* \sim p(\cdot \,|\, \max\{y \,|\, (x, y) \in \mathcal{D}\})$$

$$t_n = t_0 + n\Delta t$$

$$z(t) = z_0 + z_1 t + z_2 t^2 + z_3 t^3 \,.$$





Summary & Outlook

- Developed a seeding algorithm for super-droplet microphysics in a 1D kinematic flow and used a denoising diffusion model to optimize aerosol perturbation trajectories in $z - t$ space
  - ➡ Github: https://github.com/jtbuch/PySDM
- Ongoing work extends our framework to 2D kinematic flow in a stratocumulus cloud; preliminary results indicate that the algorithm scales with new parameters but requires additional training data
- Next steps: a) incorporate active learning in diffusion-based BBO (Wu et al. 2024) for online optimization of aerosol perturbation trajectories; b) apply the framework to LES model output with bin and super-droplet microphysics (!)
- If you have data from a past field campaign, I am interested in learning more about existing experimental design approaches to constrain microphysical parameterization with observations and how these could be optimized with ML

Join me on a saner social network!

Mail: jb4625@columbia.edu

Web: jatanbuch.com

Extra slides

Outline

- What is the problem: Why does constraining ACI affect uncertainty of future climate change uncertainty? Role of microphysics (Slide 1)

- Barrier: What are the limitations to these problems? Mostly covariability and observational uncertainty (Slide 2)

- Some ways to solve the problem: What are opportunistic natural experiments to constrain radiative effects of ACI? Show figuratively how these span multiple spatial and temporal scales (Slide 3)

- Focus of this talk: disentangle ACP pathways on a local scale using LCM and ML; utility for both method and scheme development (Slide 4); introduce 1D and 2D environments and SDM (Slide 5)

- Diffusion based BBO as an alternative to MCMC and PPE (Slide 6)

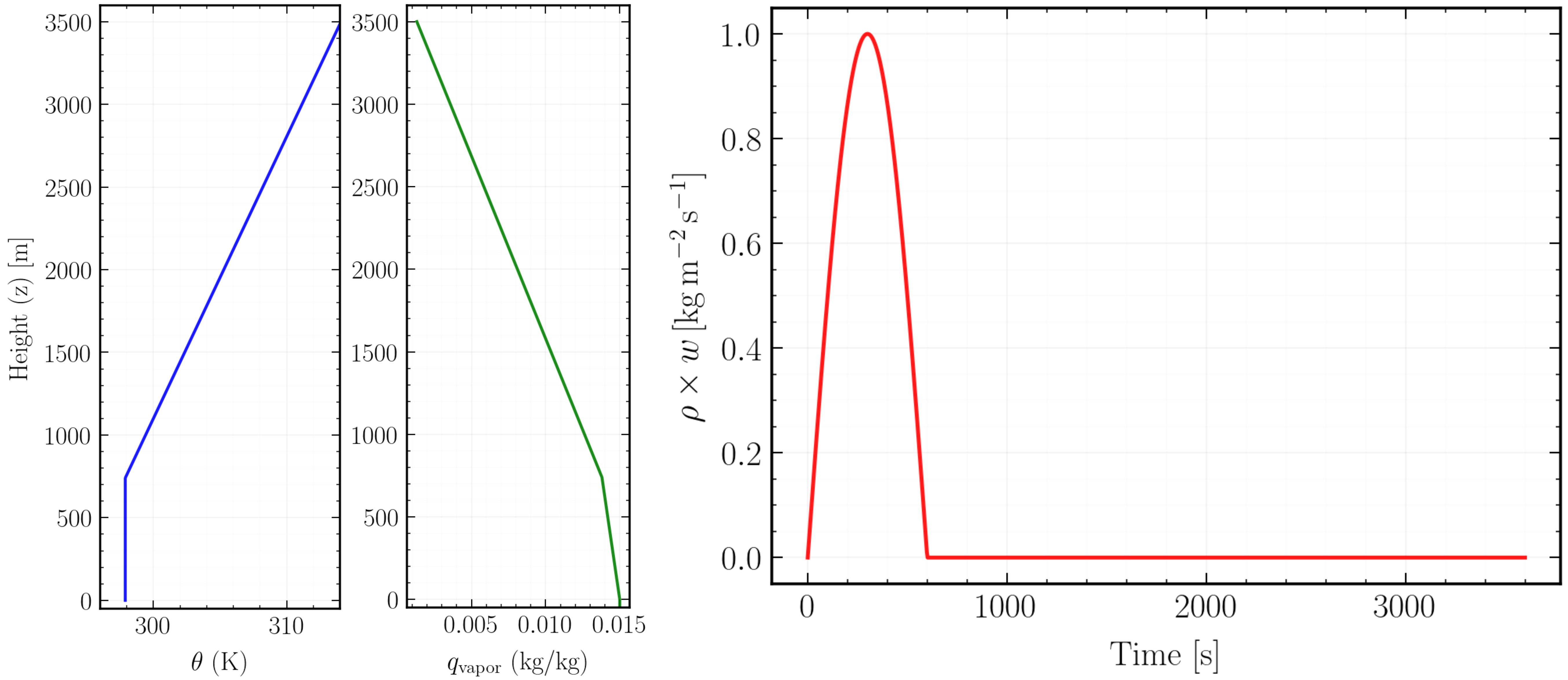
- Results: Rain as a function of N_d and r_d in 1D, 2D with diff. seeding locations; optimization of trajectory with diffusion model (Slide 7 – 10);

Shipway and Hill (2012), QJRMS
Morrison and Grabowski (2007), JAS

Kinematic environment

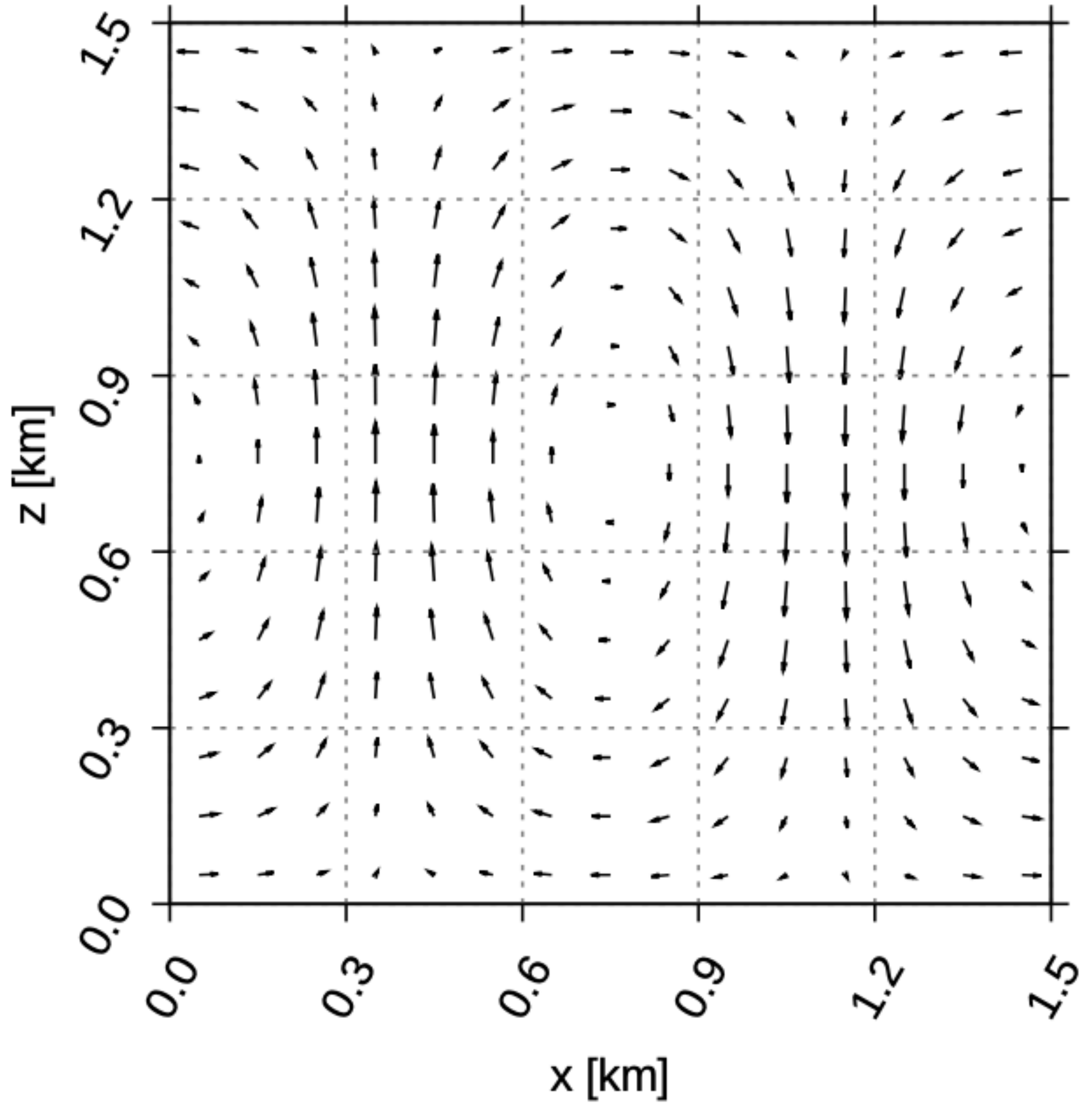
1D Kinematic driver (KiD)

- Temperature and vapor profiles are based on the RICO LES case with a sinusoidal updraft constant in z



2D Stratocumulus case

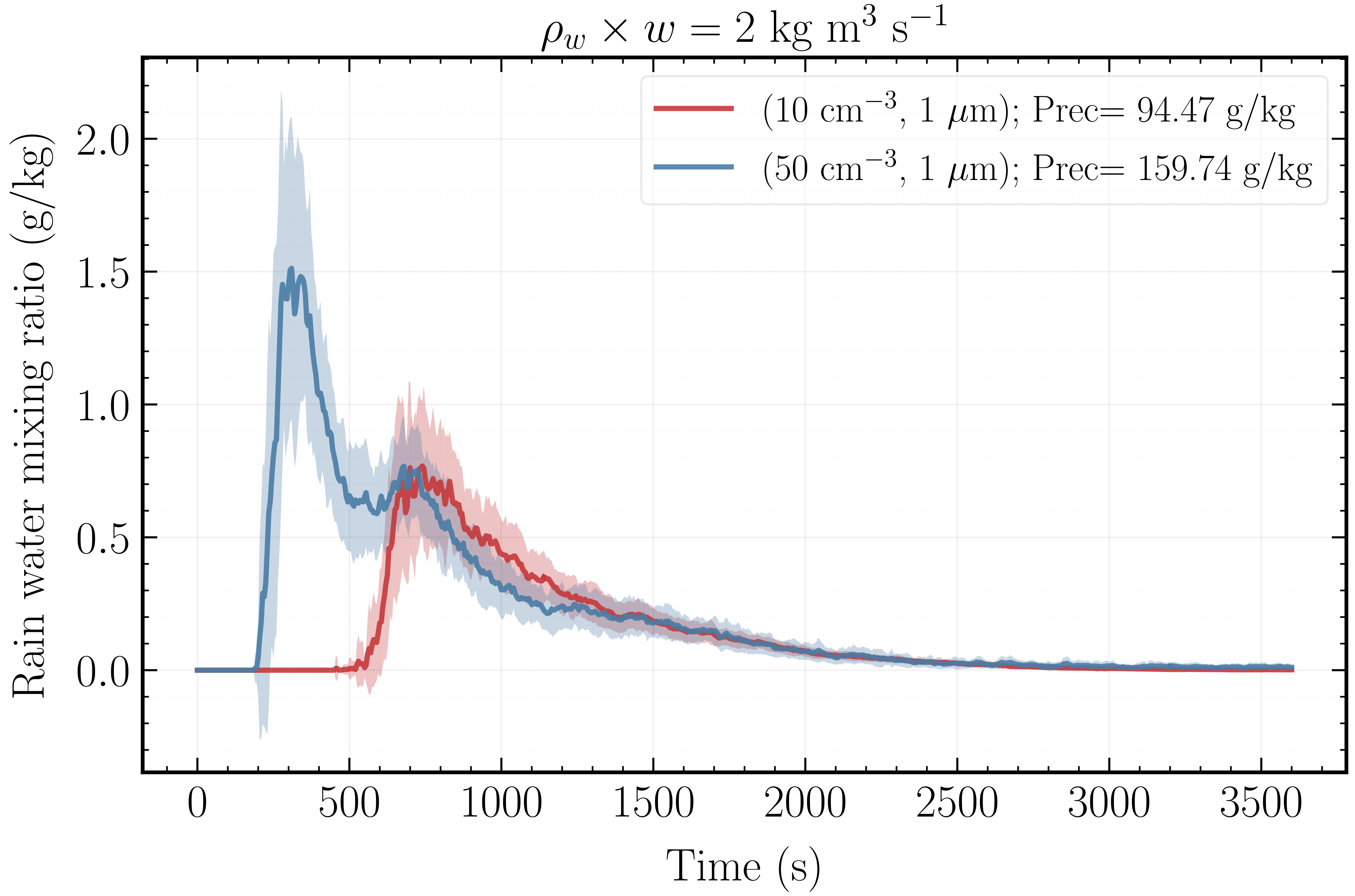
$$\psi(x,z) = -w_{\max}\frac{X}{\pi}\sin\left(\frac{\pi z}{Z}\right)\cos\left(\frac{2\pi x}{X}\right)$$

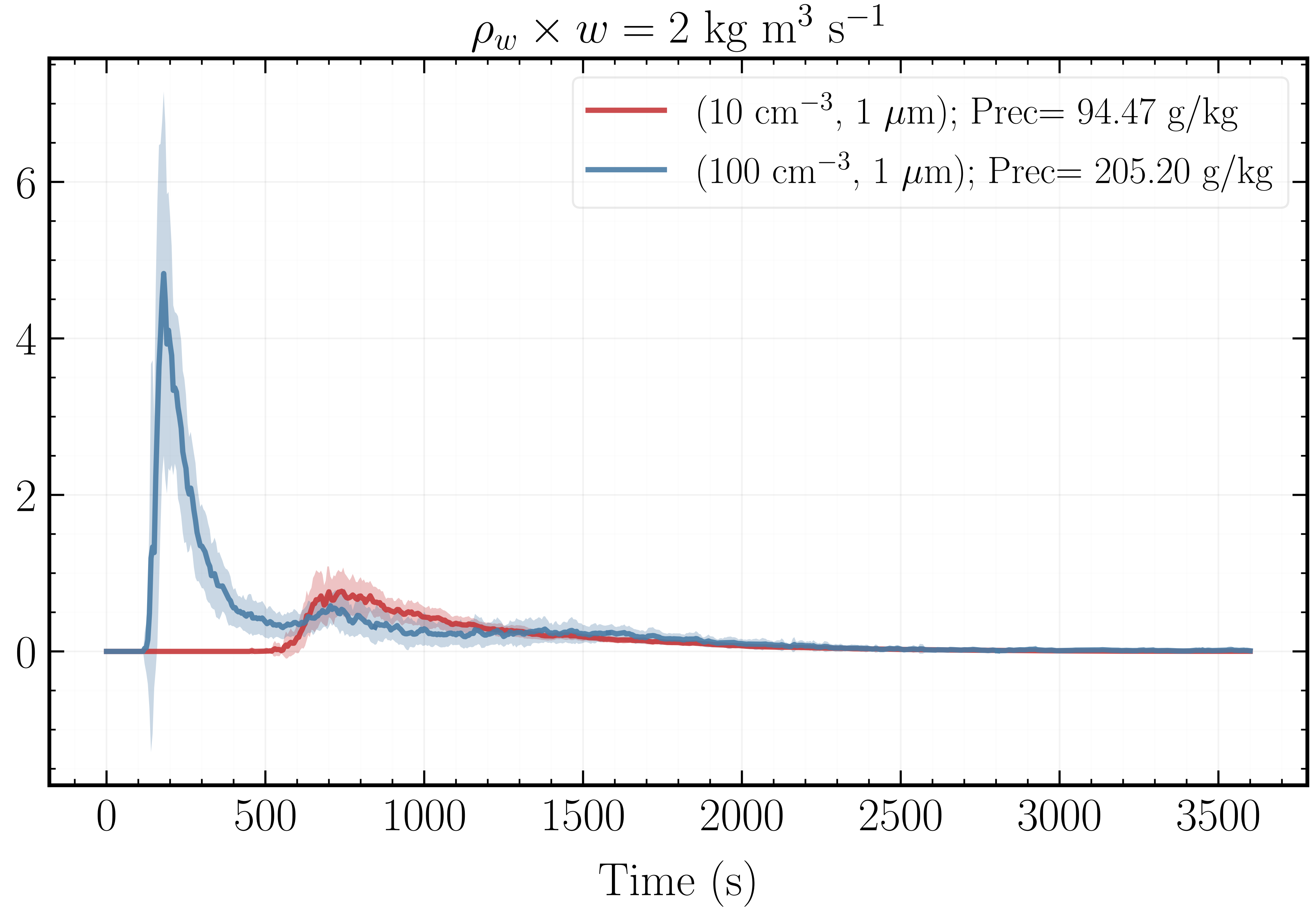


Effect of seeding concentration

- For a wide range of seeding aerosol concentration values, increasing the concentration uniformly in the domain yields higher surface rainfall

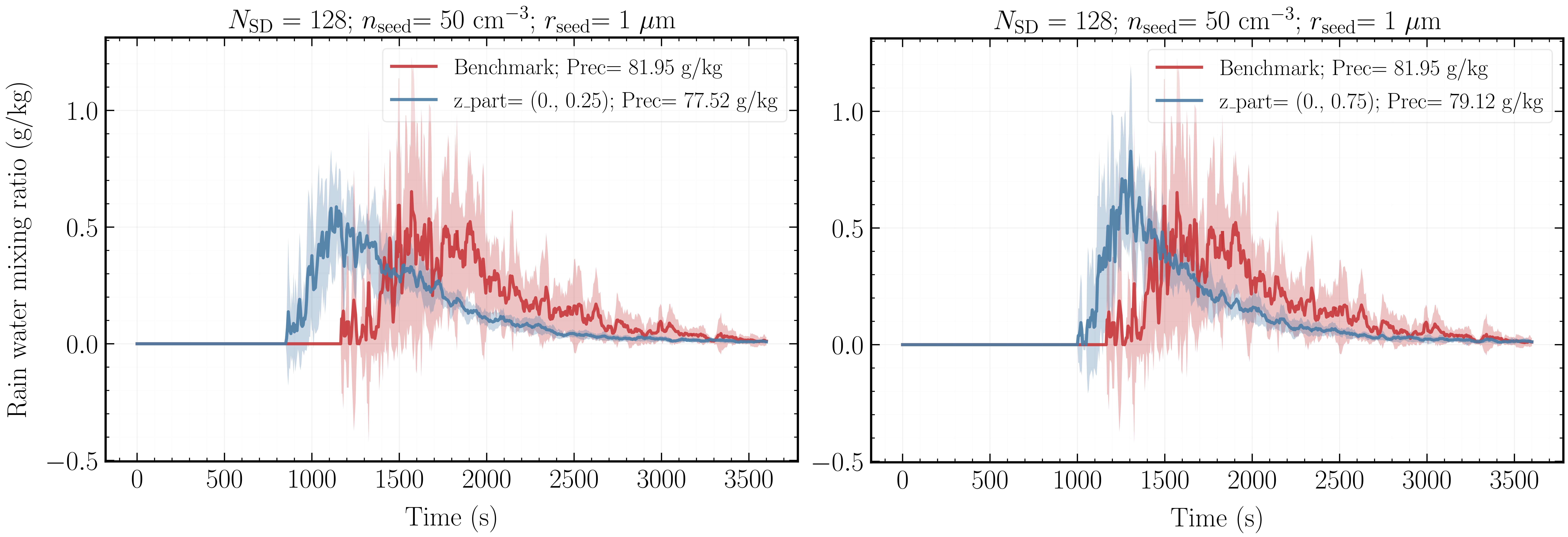
- However, for extremely high concentrations there is a suppression due to a “crowding out” effect





Effect of seeding location

- Seeding uniformly in a domain results in higher cumulative surface rainfall
- Meanwhile, seeding in the upper quadrant of the domain is preferable to seeding at the lower quadrant since it allows the background cloud droplets to grow larger in the updraft before colliding



Effect of seeding timing

$$z_{seed} = [0, n_z/4]$$

Later seeding injection time

Effect of seeding timing

$z_{seed} = [0, n_z/4]$

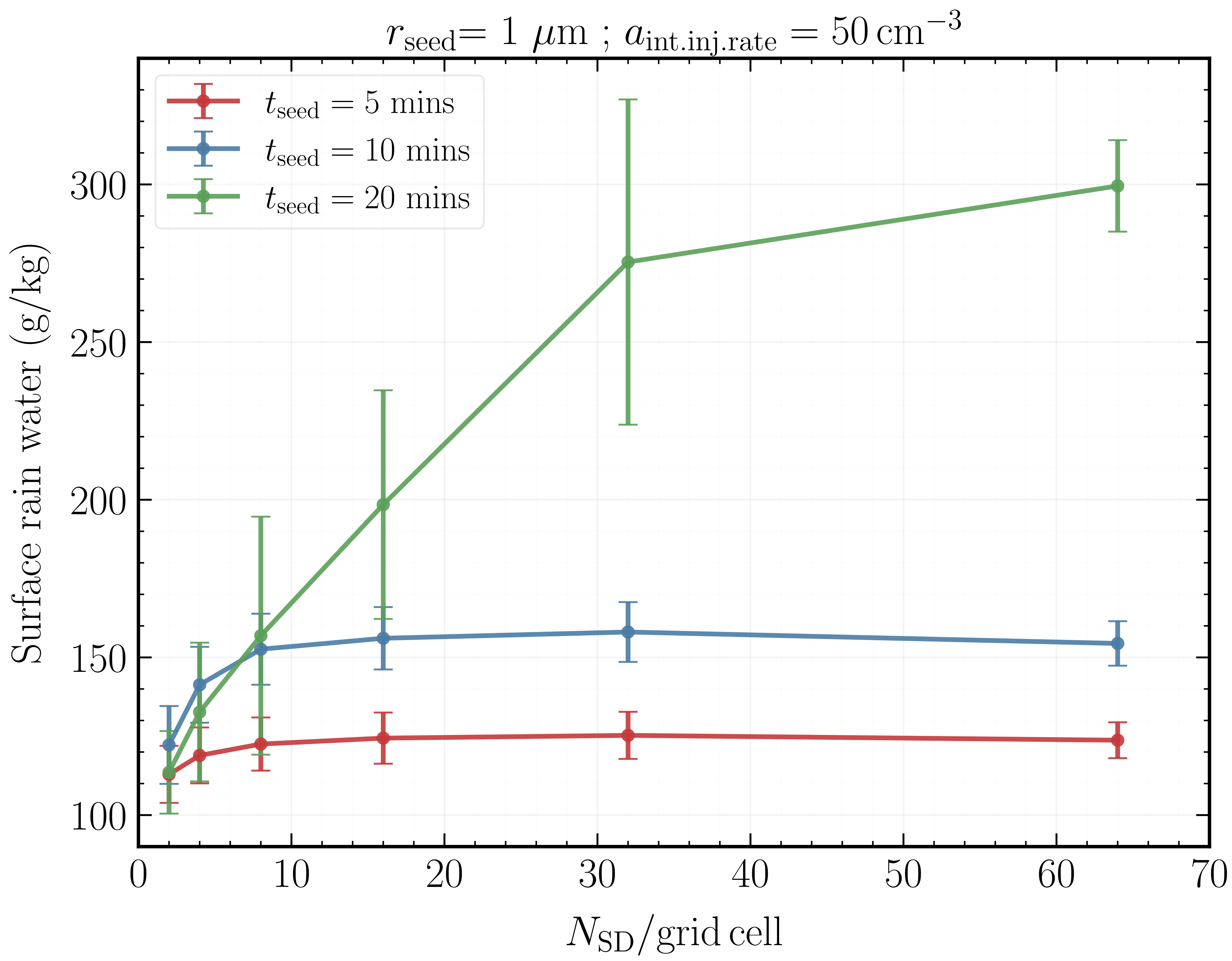
Increasing number of seeds for the same integrated injection rate

Effect of seeding timing

$z_{\text{seed}} = [0, n_z/4]$

Increasing number of seeds for the same integrated injection rate

Scaling law



$r_{\mathrm{seed}} = 1\ \mu\mathrm{m}$; $a_{\mathrm{int.inj.rate}} = 50\,\mathrm{cm}^{-3}$
$t_{\mathrm{seed}} = 5$ mins
$t_{\mathrm{seed}} = 10$ mins
$t_{\mathrm{seed}} = 20$ mins
Surface rain water (g/kg)
300
250
200
150
100
0
10
20
30
40
50
60
70
N_{SD}/grid cell