Benchmarking the Robustness of Neural Network-based Partial Differential Equation Solver

Jiaqi Gu[1], Mohit Dighamber[2],
Zhengqi Gao[2] , Duane S. Boning[2]
[1]Arizona State University, [2]MIT
http://jqgu.net , jiaqigu@asu.edu |
https://scopex-asu.github.io

November 2, 2023

Machine Learning for Science

- AI shows its power in scientific computing
- ML can learn how to solve partial differential equations (PDEs)
  - Neural PDE solver Ψ_θ
  - Learn a mapping from PDE variables $\mathcal{A}$ to PDE solution $\mathcal{U}$
- Example models
  - CNNs, PINN, Fourier neural operators
- Scientific applications
  - Physical simulation, flow prediction, weather forecast…

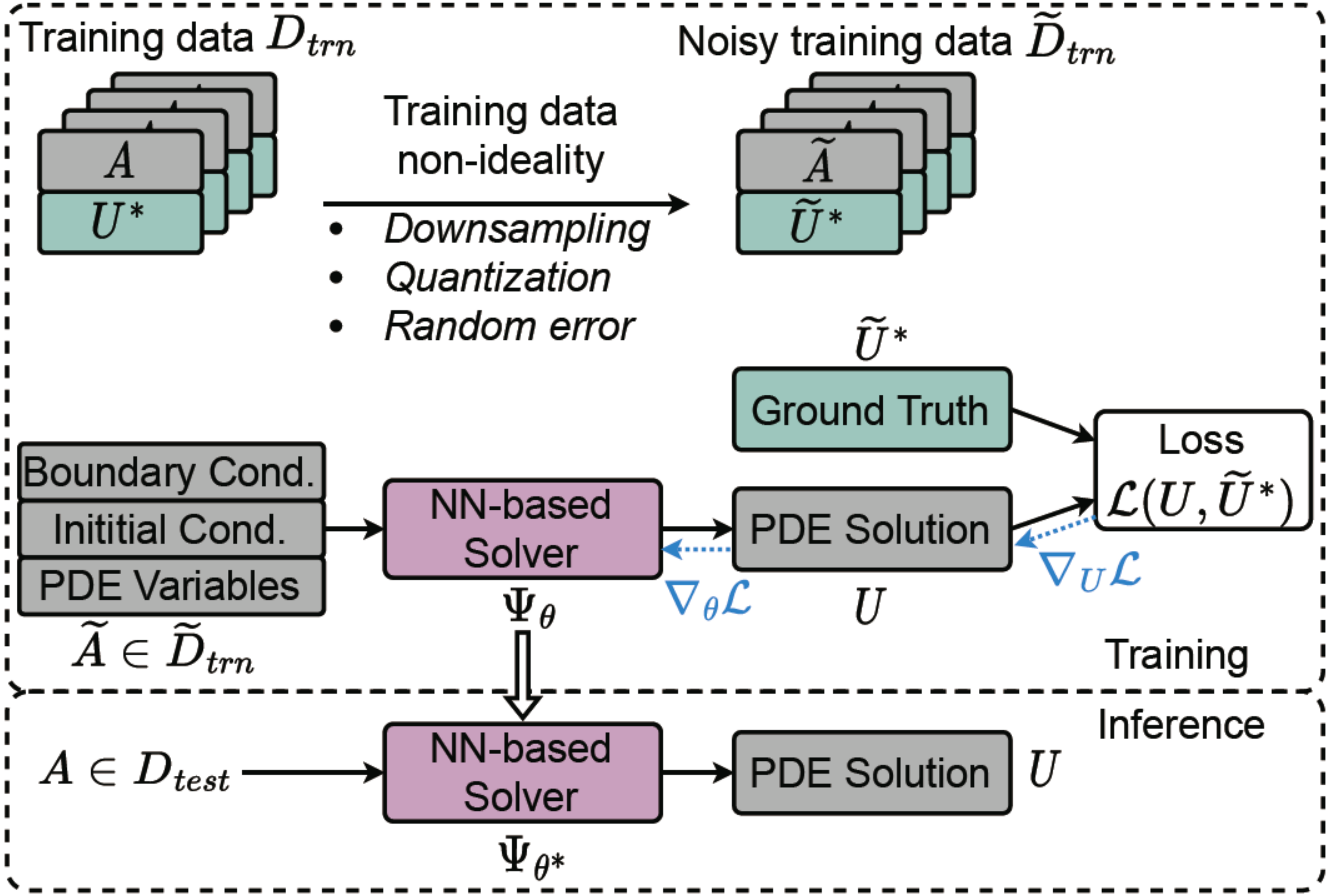
The fundamental driving force is ***High-Quality Data***

However…

High-resolution, high-fidelity data is hard/costly to collect

Training Data Robustness for Neural PDE Solver

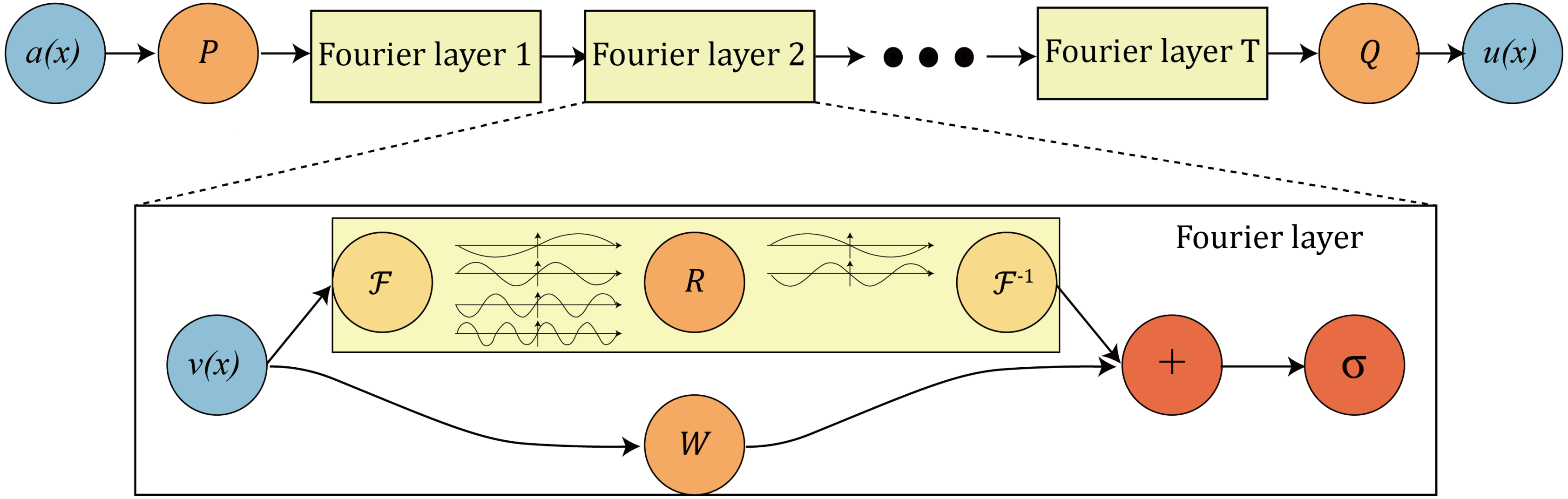
- What if the train data quality is degraded
- Train data: $\mathcal{D}_{trn} = A, U^*$
- Noisy data: $\widetilde{\mathcal{D}}_{trn} = \tilde{A}, \widetilde{U}^*$
- Model trained under perturbed PDE variables, supervised with corrupted solutions



How robust is Neural PDE solver to noisy training data?

Case Studies for This Topic – Neural PDE Solver

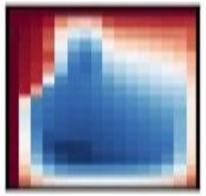
- Model: State-of-the-art Fourier Neural Operator (FNO)



- Data-driven
- N-dimensional FFT + complex matrix multiplication + iFFT

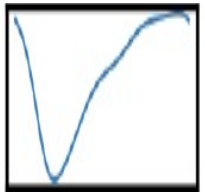
Case Studies for This Topic -- Benchmarks

- Tasks



2D Darcy flow: flow of a fluid through a porous medium

$$-\nabla \cdot (a(x)\nabla u(x)) = f(x) \qquad x \in (0,1)^2$$
$$u(x) = 0 \qquad x \in \partial(0,1)^2$$



1D Burgers equation: one dimensional flow of a viscous fluid

$$\partial_t u(x,t) + \partial_x (u^2(x,t)/2) = \nu \partial_{xx} u(x,t),$$
$$u(x,0) = u_0(x),$$

2D Navier-stokes equation: viscous, incompressible fluid in vorticity form

$$\partial_t w(x,t) + u(x,t) \cdot \nabla w(x,t) = \nu \Delta w(x,t) + f(x)$$
$$\nabla \cdot u(x,t) = 0,$$
$$w(x,0) = w_0(x),$$

2D frequency-domain Maxwell equations: photonic device simulation

$$\left(\nabla \times (\boldsymbol{\epsilon}_r^{-1}(\boldsymbol{r})\nabla\times) - \omega^2 \mu_0 \epsilon_0\right)\mathbf{H}(\boldsymbol{r}) = j\omega \mathbf{J}_m(\boldsymbol{r})$$

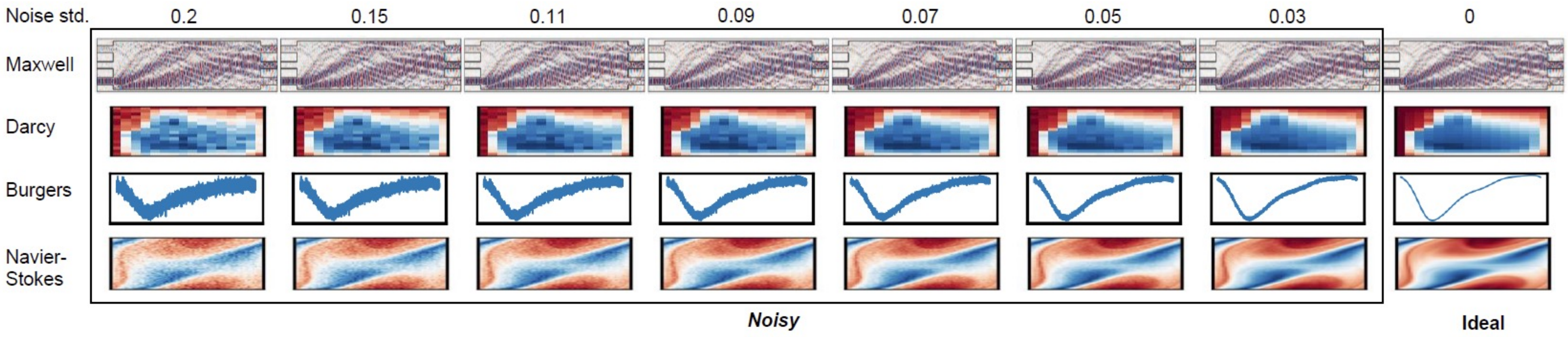
[Z. Li+, FNO, *ICLR*'2021]
[J. Gu+, NeurOLight, *NeurIPS*'2022]

Case Studies for This Topic – Error Settings

- Random noises
  - Emulate independent, uniform, high-frequency errors
  - $\tilde{\mathcal{A}} \leftarrow \mathcal{A} + \epsilon,\ \ \epsilon \sim \mathcal{N}(0, \sigma^2); \tilde{\mathcal{U}} \leftarrow \mathcal{U} + \epsilon,\ \ \epsilon \sim \mathcal{N}(0, \sigma^2)$
- Data down-sampling errors
  - Down-sample huge simulation dataset
  - $\tilde{\mathcal{U}} \leftarrow Interp_{1/s}\,(Interp_s\,(\mathcal{U}))$
- Numerical quantization errors
  - Happens in high-precision, high-dynamic range simulation tasks
  - Data compression to low-bit introduces more errors
  - $\tilde{\mathcal{U}} \leftarrow Q(U; \mathcal{U}_{min}, \mathcal{U}_{max})$

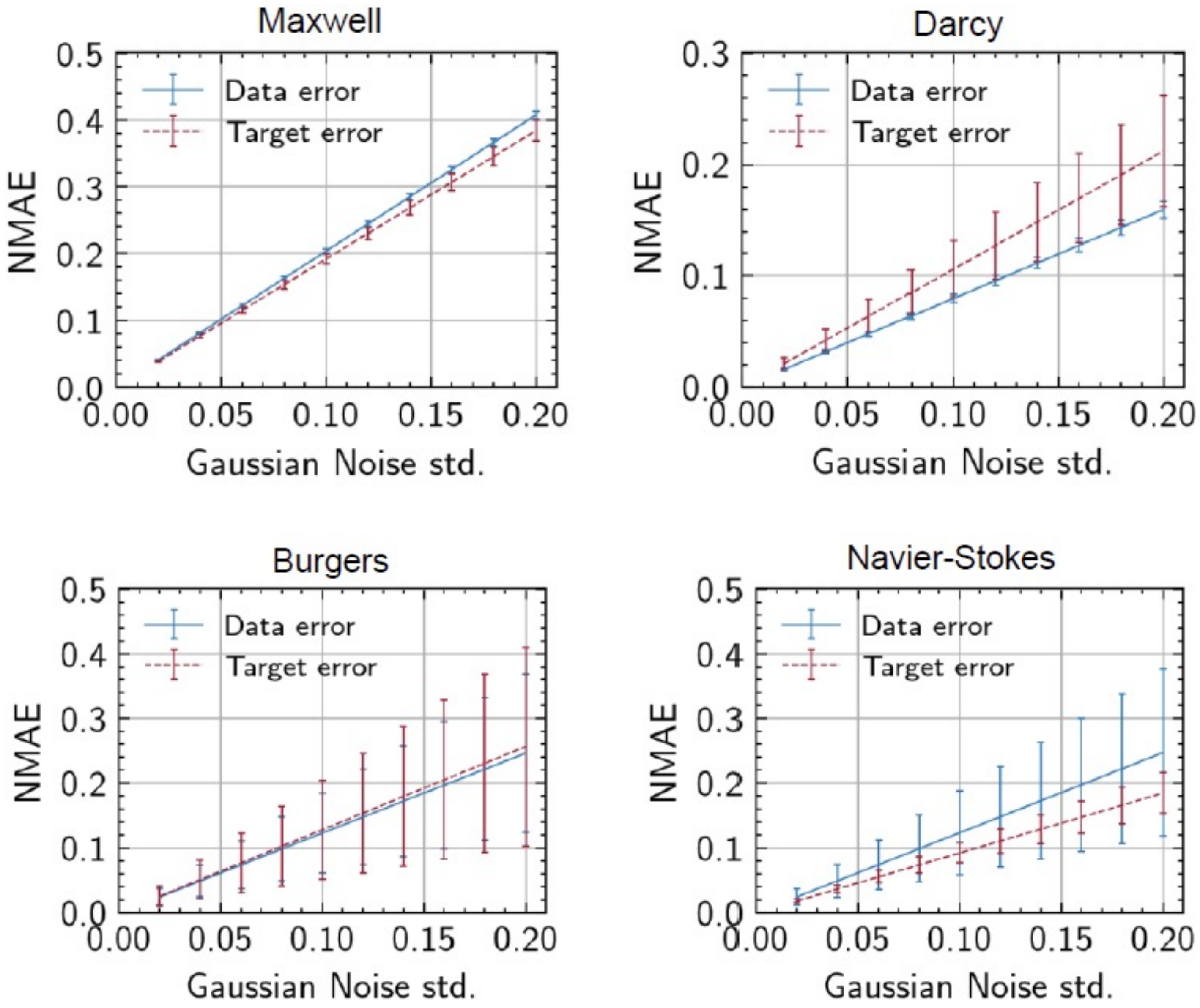
Random Noises on Training Data

- Emulate independent, uniform, high-frequency errors
- $\tilde{\mathcal{A}} \leftarrow \mathcal{A} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2); \tilde{\mathcal{U}} \leftarrow \mathcal{U} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$
- Global pattern does not change significantly
- Local fine-grained features are corrupted



Random Noises: Training Data Error

- As the first-order effect, let's look at how much errors on the data
- NMAE= $|U - \widetilde{U}|_1 / |U|_1$
- In general it leads to 2%-30% errors on input and target
- Much higher errors on sparse fields, e.g., optical fields with low light intensities

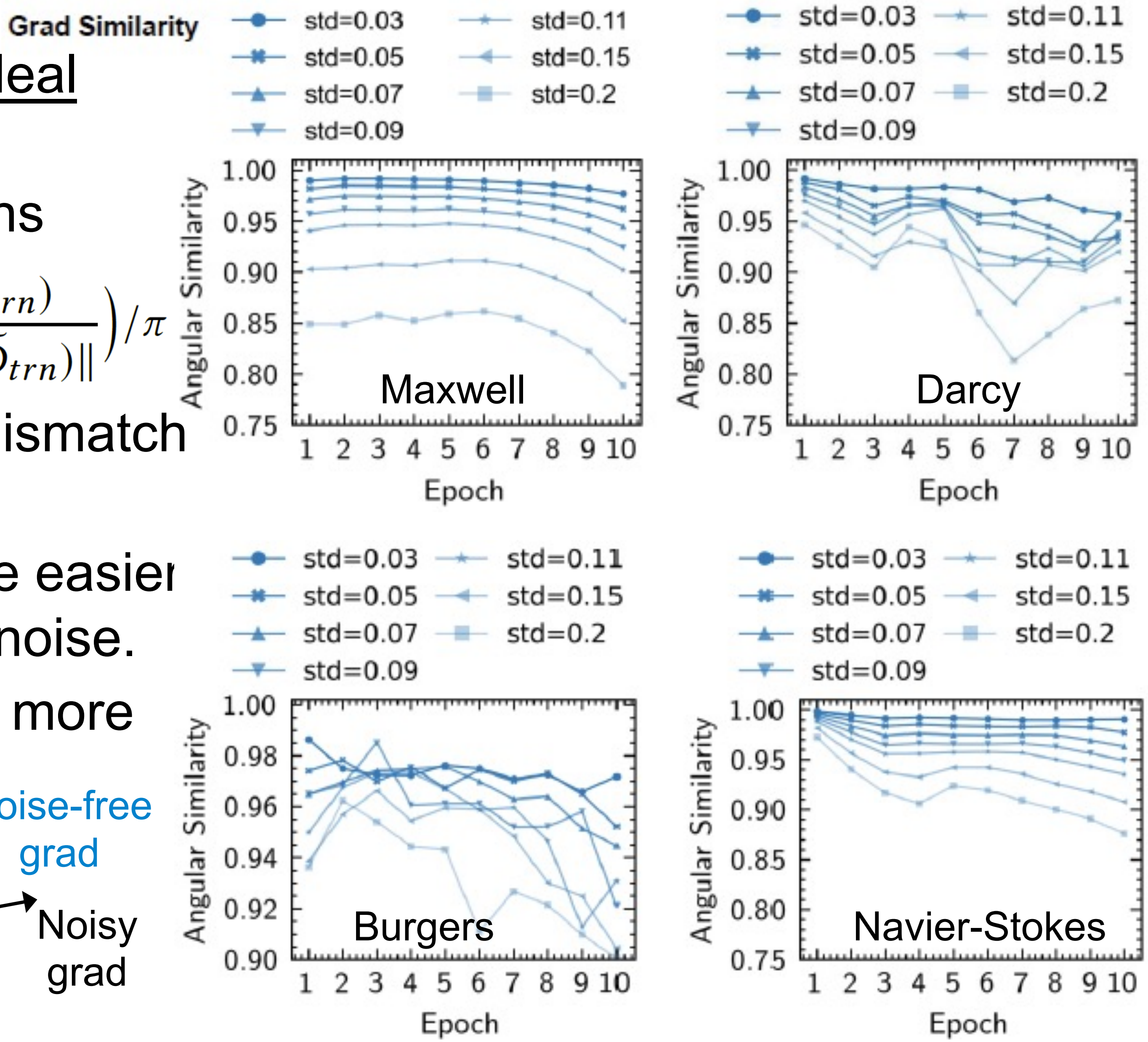


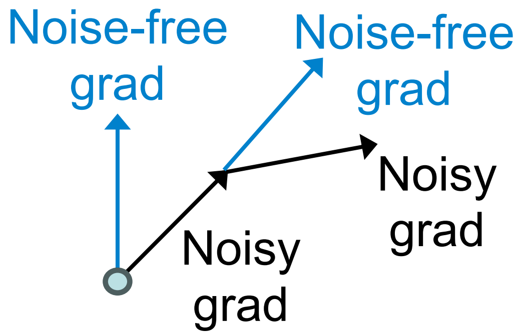
Random Noises: Training Dynamics

- Evaluate alignment between <u>ideal gradients</u> and <u>noisy gradients</u>
- Angular similarity across epochs

$$\texttt{Similarity} = 1 - \arccos\left(\frac{\nabla_\theta \mathcal{L}(\mathcal{D}_{trn}) \cdot \nabla_\theta \mathcal{L}(\widetilde{\mathcal{D}}_{trn})}{\|\nabla_\theta \mathcal{L}(\mathcal{D}_{trn})\| \cdot \|\nabla_\theta \mathcal{L}(\widetilde{\mathcal{D}}_{trn})\|}\right)/\pi$$

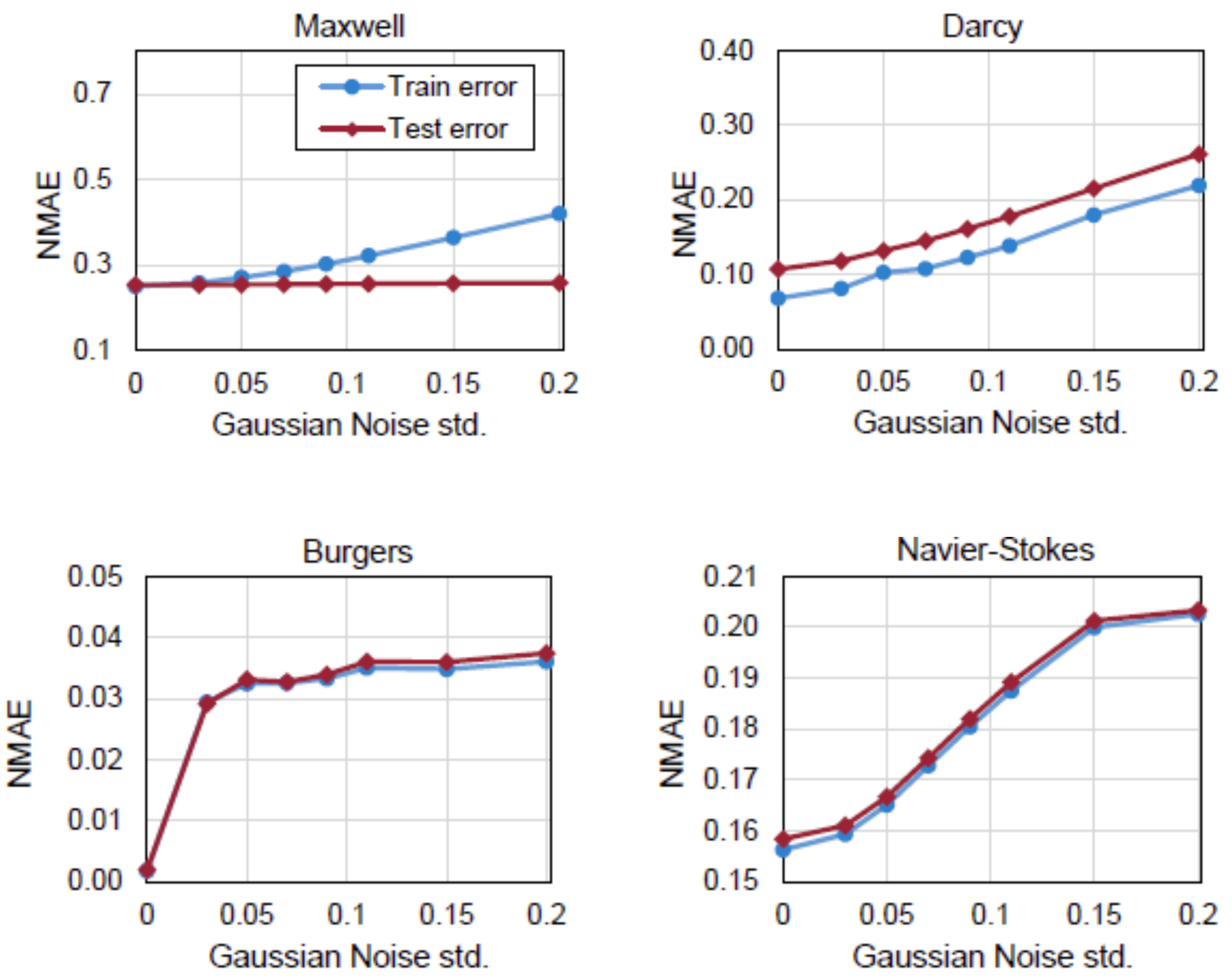
- Align at the beginning, more mismatch later
- Smooth functions (Burgers) are easier to learn, more tolerant to data noise.
- High-frequency Maxwell Eq. is more sensitive to noises





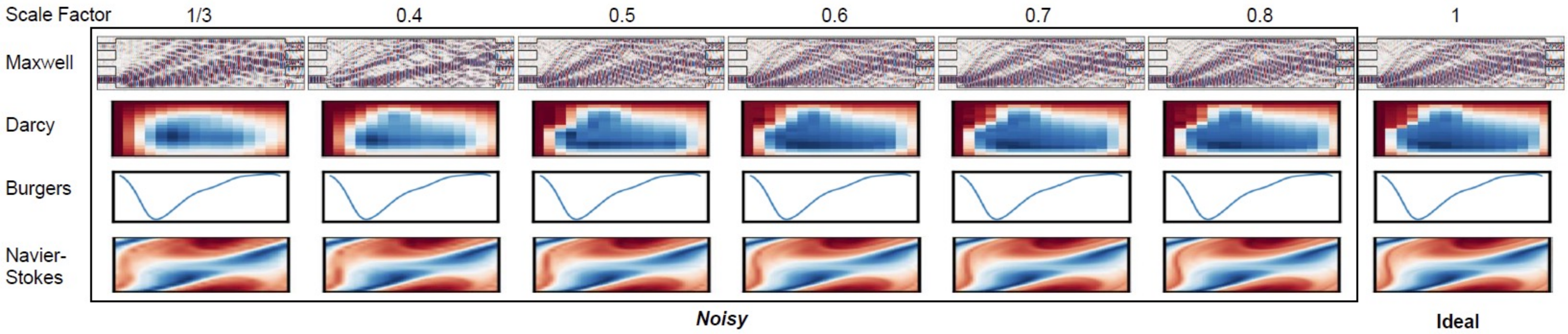
Random Noises: Model Robustness

- Training/Test NMAE
- Train/Test error increase simultaneously with larger noise
- Burgers equation significantly degrades with even small noise
- Maxwell equation shows the same test error immune to train noises
  - *Extra regularization and data augmentation helps improve noise tolerance even with large gradient mismatch*



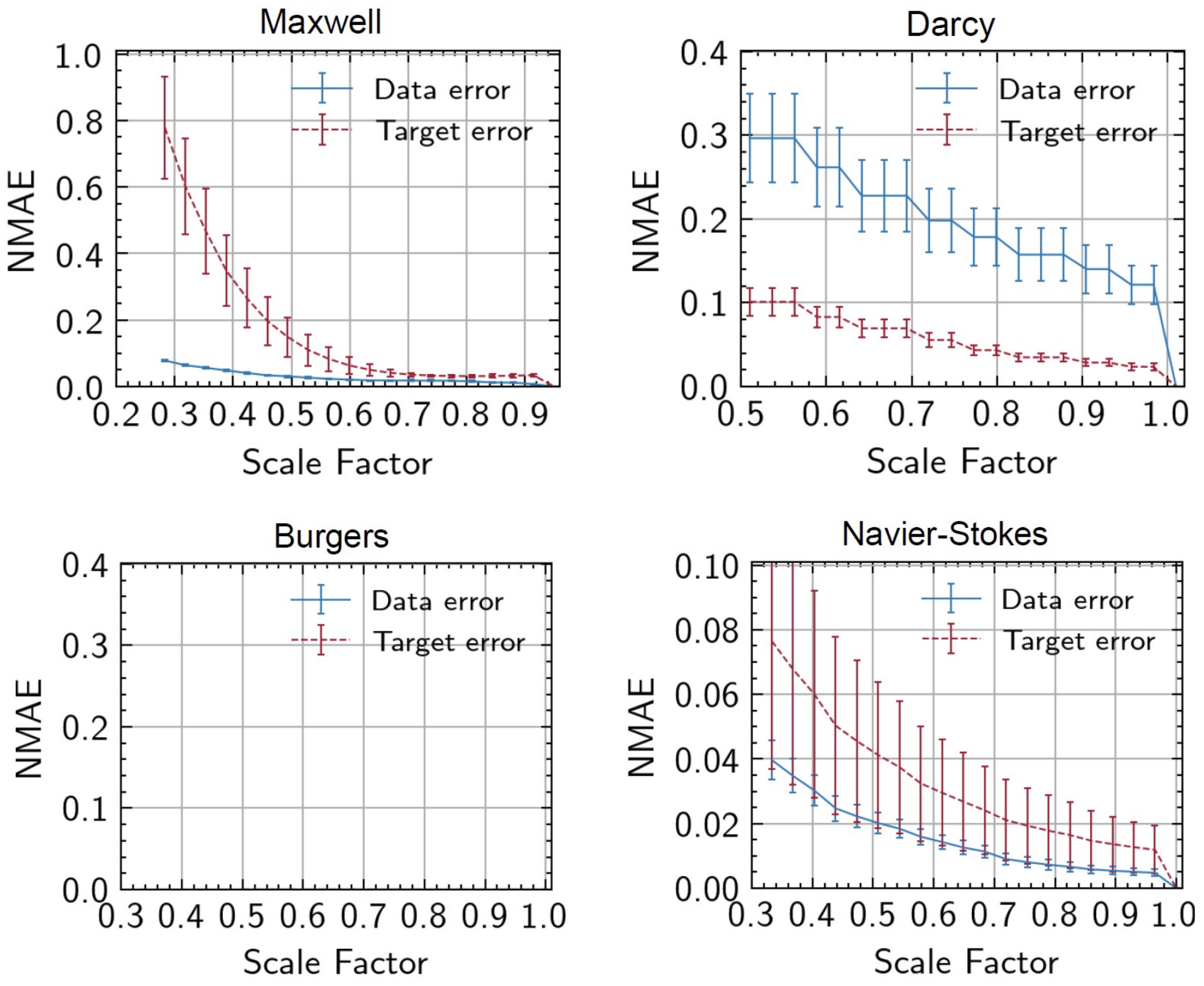
Data Downsampling on Training Data

- Compress high-res raw data with downsampling to save cost
- $\tilde{\mathcal{U}} \leftarrow Interp_{1/s}\left(Interp_s\left(\mathcal{U}\right)\right)$
- We inject bicubic/linear resizing errors to input/target
- Structural errors and related to local field/flow patterns
  - E.g., Light waves in Maxwell is severely distorted



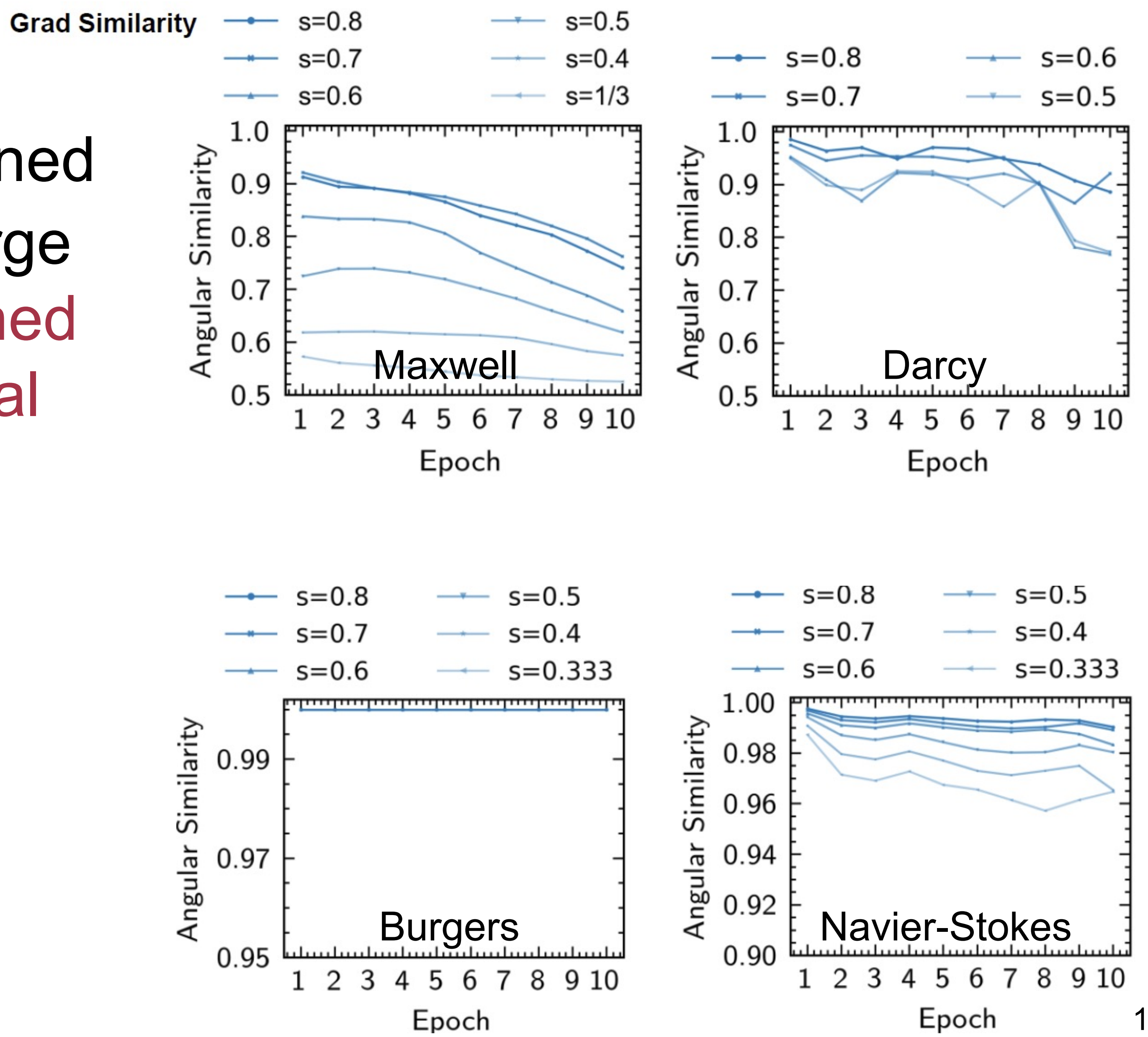
Data Downsampling: Training Data Error

- Downsampling is disastrous for Maxwell, with 0.2 (5x) downsampling, almost 80% error on waves
- No errors on Burgers with very smooth patterns
- Navier-stokes: relatively smooth flow, robust to downsampling (4%-8% error)



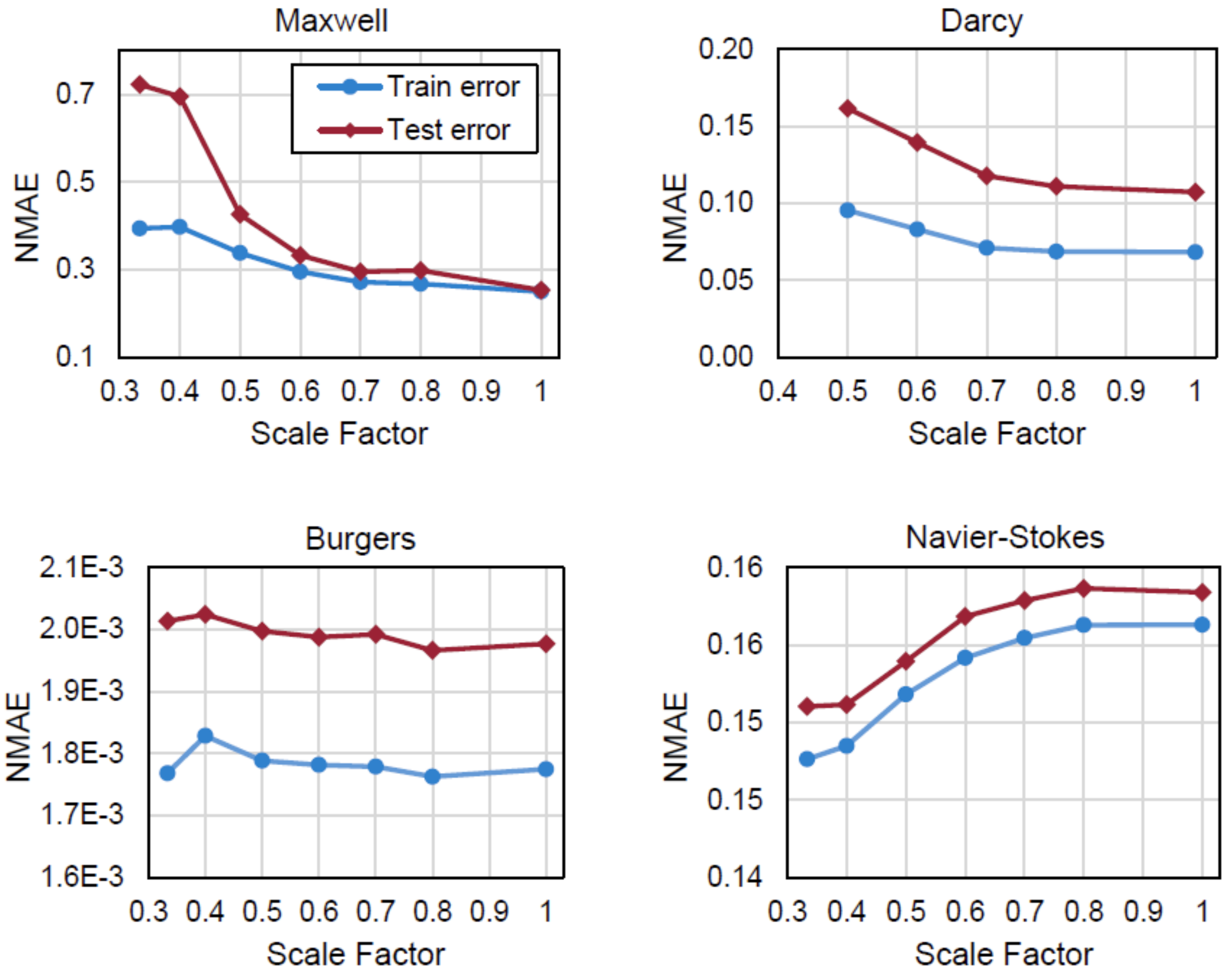
Data Downsampling: Training Dynamics

- Gradients for Burgers and Navier-Stokes are well-aligned
- Maxwell equations have large distortion, severe mismatched gradients (almost orthogonal with 3x resizing)
- For smooth patterns, the gradients are insensitive to data compression error.



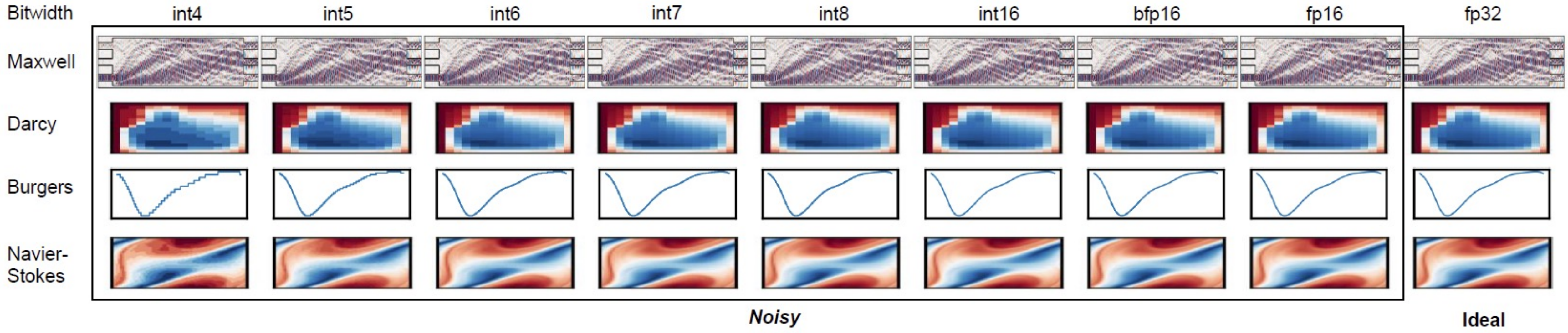
Data Downsampling: Model Robustness

- Rapid degradation on Maxwell equation
- Regional correlated errors cause a systematic bias on data distribution. Regularization cannot counter it.
- Navier-Stokes: both train/test errors are improved with small errors (smoothing effects help better converge)



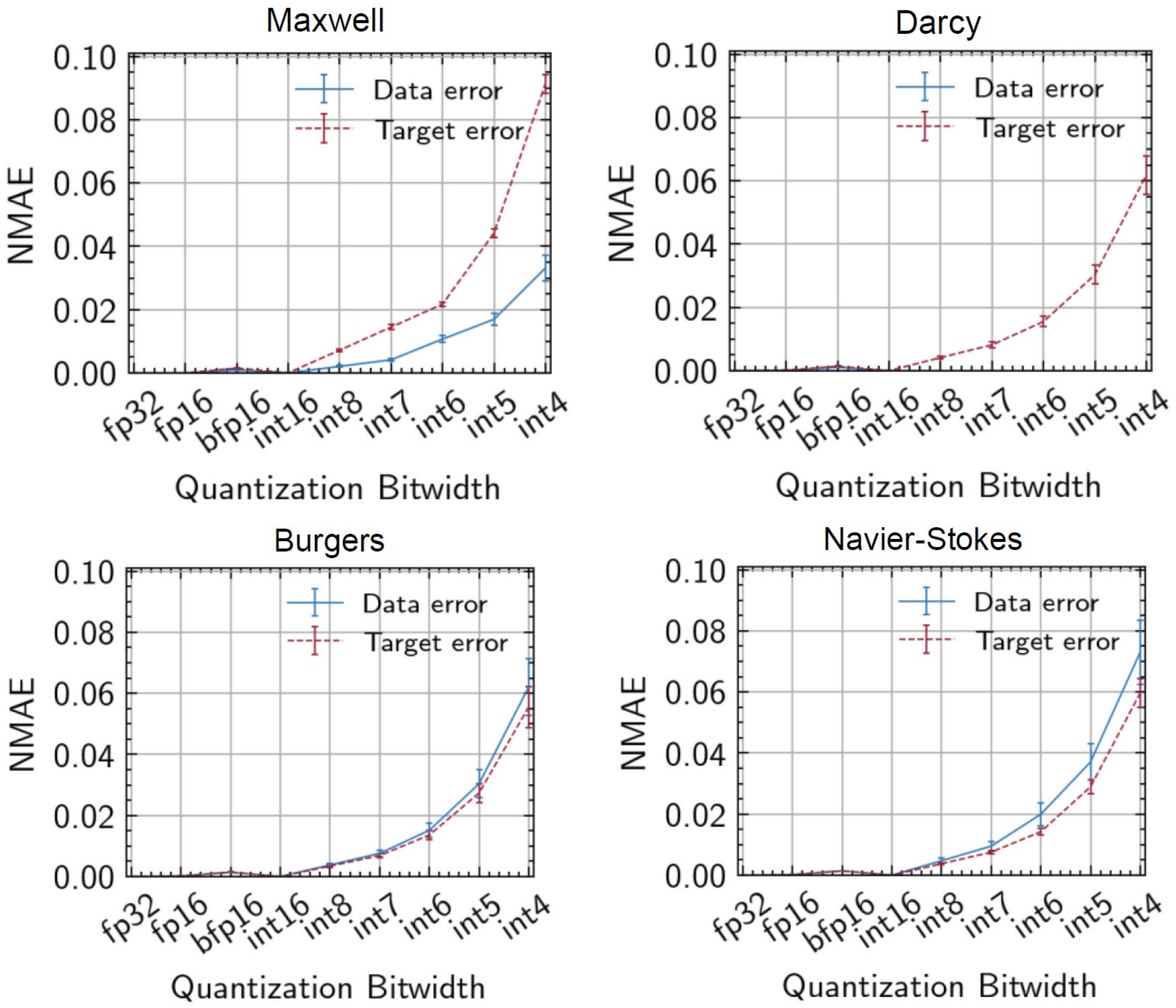
Numerical Quantization on Training Data

- Compress double/complex128 to low-bit data (INT4-16, BFP16, FP16) to save cost
- $\tilde{\mathcal{U}} \leftarrow Q(U; \mathcal{U}_{min}, \mathcal{U}_{max})$ quantize after min max scaling
- Still good visualization quality
- Subtle impact on global patterns, maintains relative magnitude ordering for local data



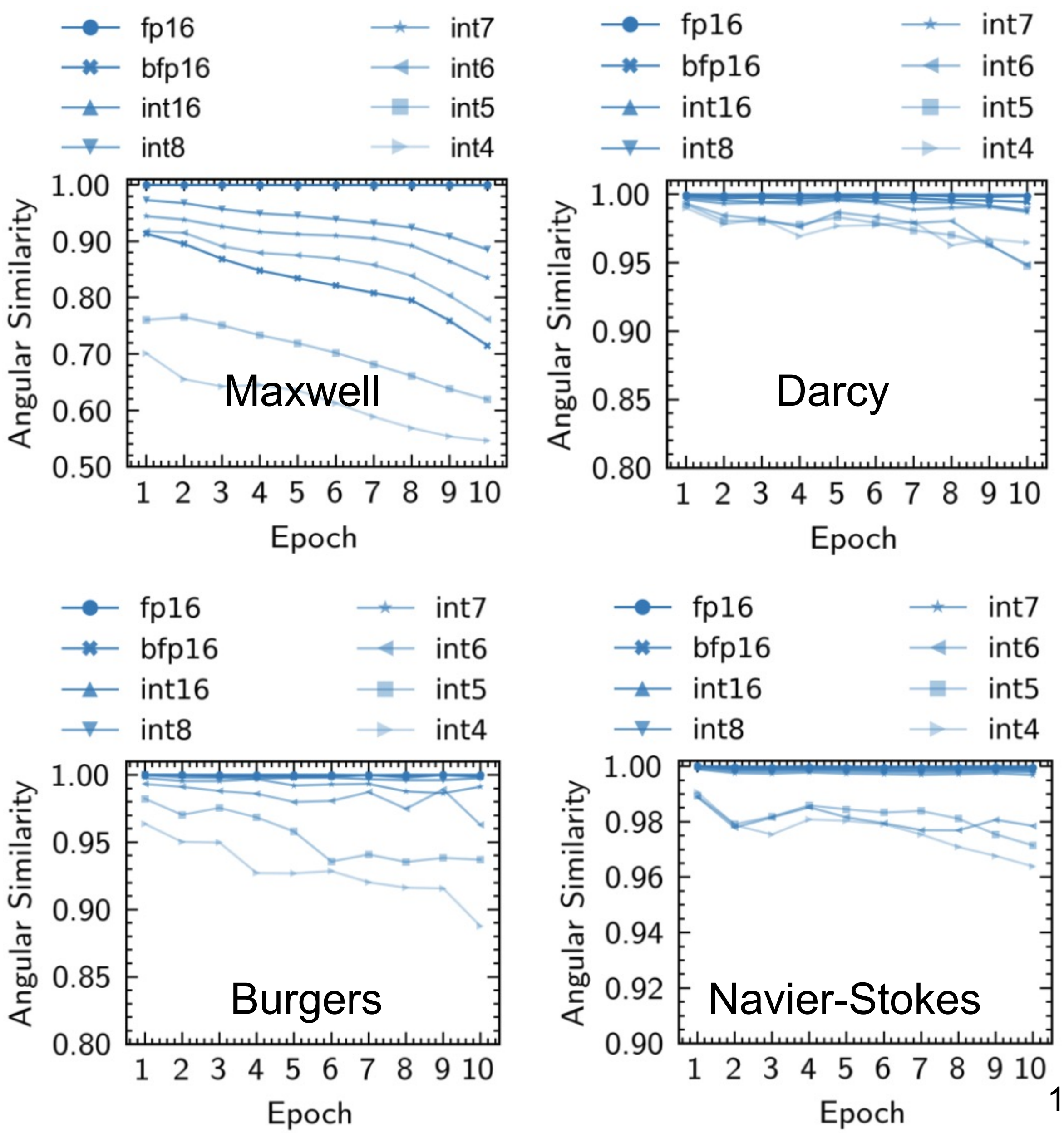
Numerical Quantization: Training Data Error

- No significant difference across 4 benchmarks
- 4-8% relative absolute errors
- INT4 have relatively large errors



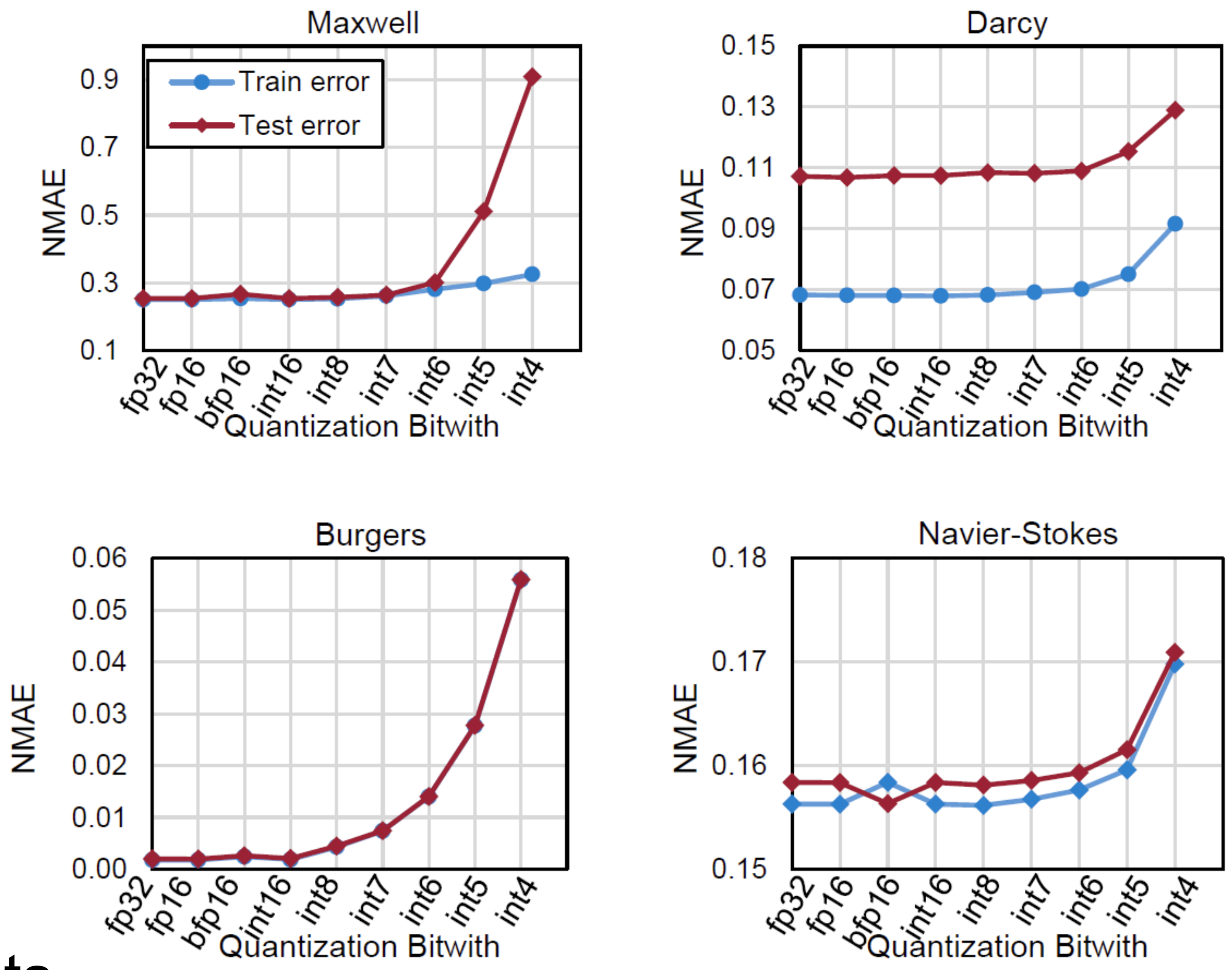
Numerical Quantization: Training Dynamics

- Only Maxwell benchmarks shows high sensitivity to quantization errors on gradients.
  - Quantizing Real/Imaginary part separately lead to significant phase rotation -> large angles in gradient mismatch
- BFloat16 [E8M7] has larger range and fewer fraction bits, shows more errors than FP16 [E5M10]



Numerical Quantization: Model Robustness

- Significant impacts on training error on Maxwell with 4-bit.
  - Gradient misalignment
  - Intrinsic sensitivity to input permittivity ϵ
  - Large quant error makes it hard to learn.
- Maxwell maintains high inference fidelity
  - Good robustness from regularization
- >8-bit has negligible impacts



Conclusion & Future Directions

- We evaluate the training data robustness of Neural PDE solver (FNO) on Burgers equation, Darcy flow, Navier-Stokes equations, and Maxwell equations
- We benchmark random errors, data downsampling, and numerical quantization and investigate data error, training dynamics (gradients), and generalization
- *Conclusion*
- High-res data with low-freq field/flow patterns demonstrate better tolerance, especially for downsampling errors
- Regularization helps enhance the resilience, but cannot counter systematic bias from regional errors
- *Future*: Compare data-driven/physics-informed, explore more equations, propose data quality metrics