in hanfengzhai
hzhai@stanford.edu

Scalable and generalizable dynamics predictions of dislocation Nye tensor fields

Hanfeng Zhai[1], Myung Chul Kim[1], Wei Cai[1]

[1]Department of Mechanical Engineering, Stanford University

Summary

- Discrete dislocation dynamics (DDD) is accurate but computationally expensive for long-time evolution [1].
- Nye tensor $\boldsymbol{\alpha}$ stores geometrically necessary dislocation content that is useful for continuum models [4, 3].
- We convert DDD graphs on the glide plane to $\boldsymbol{\alpha}$ field, then learn the dynamical evolution instead of rerunning DDD.
- A fully convolutional CNN–ConvGRU, trained only for 32×32, can be transferred to much larger scales (64×64 & 128×128).
- Currently constrained to 2D dislocation motion on the glide plane.

Dislocations

A dislocation is a line defect that lets a crystal deform by slip. An edge dislocation inserts an extra half-plane of atoms. A screw dislocation shears the lattice along the line and leaves a helical step. (Figure 1)

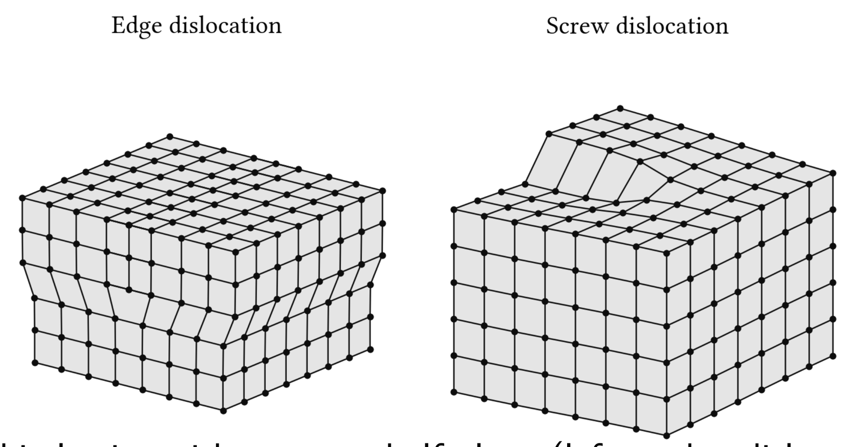

Figure 1: Cubic lattice with an extra half-plane (left – edge dislocation); and a helical surface step (right – screw dislocation).

From DDD lines to a voxelized Nye field

The Nye tensor $\alpha_{ij}(\mathbf{r})$ is a continuum measure of geometrically necessary dislocations [4]. For a discrete network, each edge k with Burgers vector $\mathbf{b}^{(k)}$ and unit tangent $\hat{\mathbf{t}}^{(k)}$ carries Nye tensor:

$$\alpha_{ij}^{(k)}(\mathbf{r}) = b_i^{(k)}\, \hat{t}_j^{(k)}\, \hat{\rho}^{(k)}(\mathbf{r}). \tag{1}$$

Superposition over M edges gives the voxelized field used for learning [3]:

$$\alpha_{ij}(\mathbf{r}) = \sum_{k=1}^{M} b_i^{(k)}\, \hat{t}_j^{(k)}\, \hat{\rho}^{(k)}(\mathbf{r}). \tag{2}$$

Each voxel stores smeared Burgers content, not a sharp line. We smear each segment as a Gaussian tube of width half a voxel, smeared over 32×32.

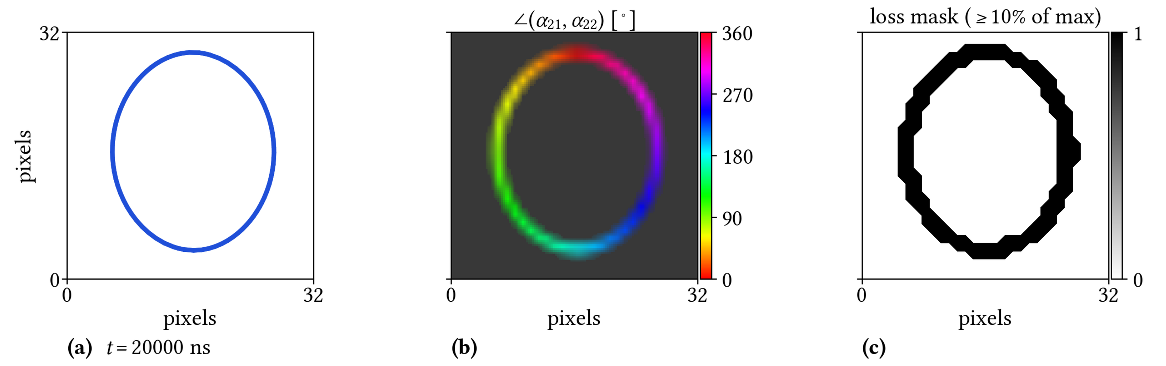

Figure 2: One expanding loop on 32×32 at $t = 2 \times 10^4$ ns. (a) DDD graph. (b) Midplane $\angle(\alpha_{21}, \alpha_{22})$ for $\mathbf{b} = [010]$. (c) Dislocation core loss mask m_n.

(a) is the discrete line network. (b) is the Gaussian-smeared α to be learnt. (c) is the core mask used in the data loss, $s_n \geq 0.10 \max_m s_m$. In the continuum limit, α is divergence-free, and its cell integral is fixed by the initial network. Those two identities are why training penalizes $\nabla \cdot \hat{\boldsymbol{\alpha}}$ and the midplane integral of $\hat{\boldsymbol{\alpha}}$.

Learning a next-step Nye map

On a glide plane we learn the mapping:

$$\hat{\boldsymbol{\alpha}}_{t+1} = f_\theta(\boldsymbol{\alpha}_{t-T+1:t},\ \boldsymbol{\sigma}_{t-T+1:t}), \qquad \text{with } T = 10. \tag{3}$$

Each frame has four Nye channels and nine stress channels. The target is the Nye field after $\Delta t = 400\,$ns. Evaluation feeds $\hat{\boldsymbol{\alpha}}$ back as input and steps forward in time. Stress is an input only. The decoder outputs the four Nye channels.

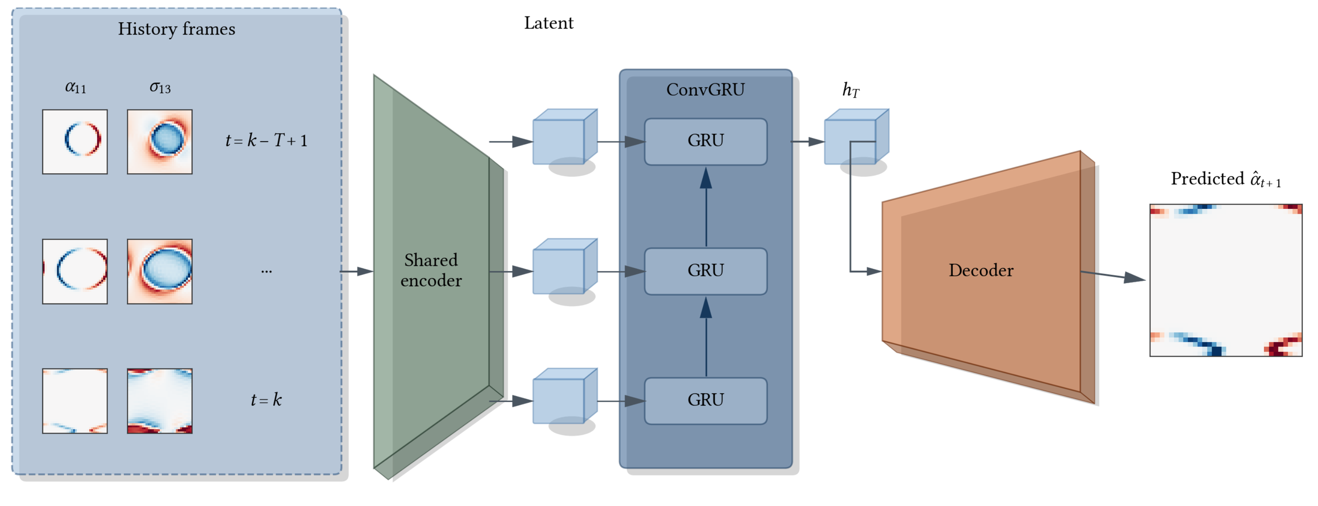

Figure 3: CNN2D–ConvGRU [5, 2]. Shared encoder, ConvGRU at half resolution, decoder to $\hat{\boldsymbol{\alpha}}_{t+1}$.

Each midplane frame concatenates four Nye channels with nine stress channels. A shared encoder maps the history of frames to latent features.

A ConvGRU updates h_t over T steps from $h_0 = 0$. The decoder maps h_T to $\hat{\boldsymbol{\alpha}}_{t+1}$. All layers are fully convolutional, so H and W may change at evaluation. The ConvGRU cell, with encoded features x_t and hidden state h_{t-1}, is

$$r_t, z_t = \sigma\big(\mathrm{Conv}([x_t, h_{t-1}])\big), \tag{4}$$

$$\tilde{h}_t = \tanh\big(\mathrm{Conv}([x_t, r_t \odot h_{t-1}])\big), \tag{5}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \tag{6}$$

Training is done on voxel fields with size 32×32. Core voxels satisfy $s_n \geq 0.10 \max_m s_m$. The data term and physics regularizers are

$$\mathcal{L}_{\text{data}} = \frac{\sum_{n,c} w_n (\hat{\alpha}_n^{(c)} - \alpha_n^{(c)})^2}{\sum_{n,c} w_n + \varepsilon}, \tag{7}$$

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \lambda_{\text{div}} \langle \|\nabla \cdot \hat{\boldsymbol{\alpha}}\|^2 \rangle + \lambda_{\text{int}} \Big(\int \hat{\boldsymbol{\alpha}}\, \mathrm{d}A - \int \boldsymbol{\alpha}\, \mathrm{d}A \Big)^2, \tag{8}$$

with $w_{\text{fg}} = 100$, $w_{\text{bg}} = 1$, and $\lambda_{\text{div}} = \lambda_{\text{int}} = 10^{-2}$. These two penalties encode $\nabla \cdot \alpha = 0$ and conserved Burgers vectors over the whole field. Input Nye may include noise; the target stays clean.

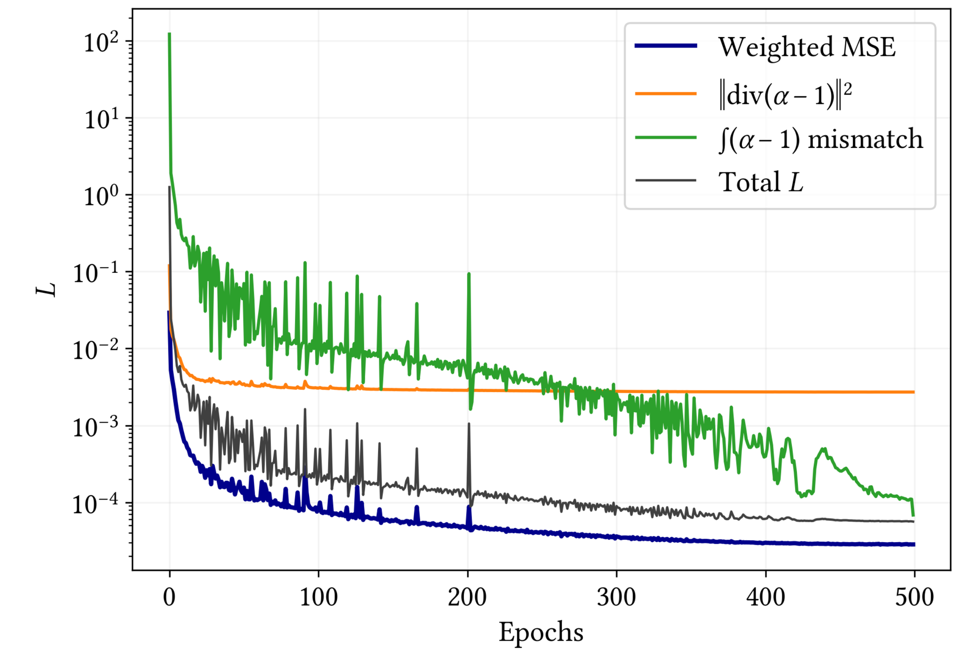

Figure 4: Training loss over 500 epochs (log scale). Weighted MSE and total $\mathcal{L}$ fall below 10^{-4}.

Weighted MSE dominates after a short transient. The integral mismatch is noisier but trends down.

Evaluations on different physical scenarios

HSV color is $\angle(\alpha_{11}, \alpha_{12})$ on the training grid – visualize the full 2D Nye tensor field. We evaluate dislocation configurations at $t = 0$, 2000, and 4000 ns.

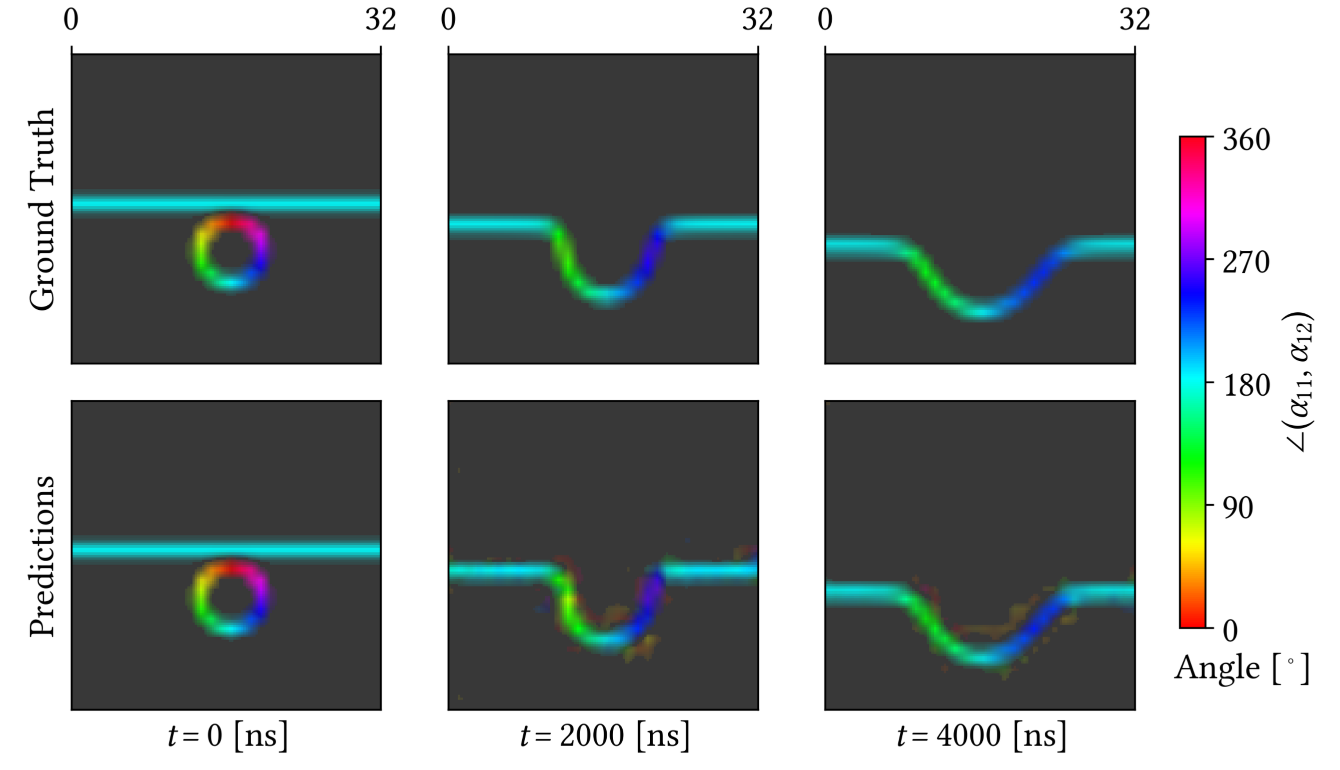

Figure 5: Line and loop on 32×32. Top: DDD. Bottom: CNN–ConvGRU.

The model keeps the line and loop cores and the in-plane orientation. The line stays nearly straight while the loop expands and leaves the slice.

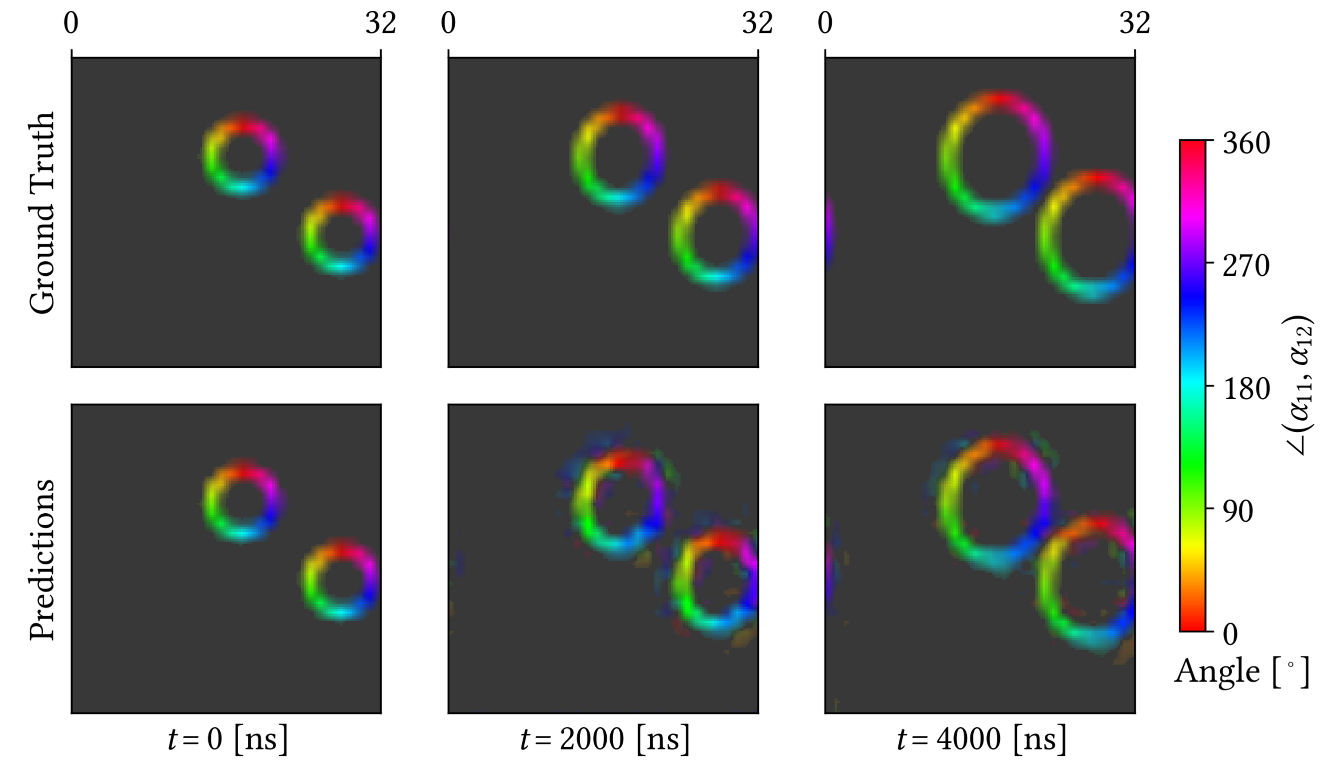

Figure 6: Two expanding loops on 32×32. Dislocation cores interact and annihilate.

Weak background noise appears at later steps; the cores remain localized. Both loops expand until they leave the midplane, and the model follows that exit. Pixel-wise parity on the training set gives $R^2 = 0.98$ (α_{11}), 0.94 (α_{12}), and 0.99 (α_{21}, α_{22}).

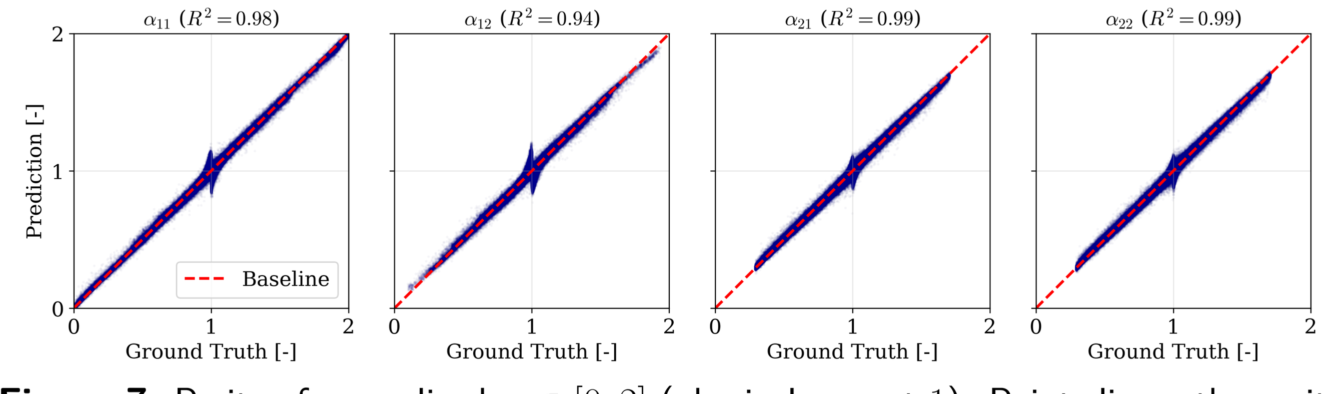

Figure 7: Parity of normalized $\alpha \in [0, 2]$ (physical zero at 1). Points lie on the parity benchmark line.

Agreement is tight on all four in-plane channels – for the 4 Nye tensor components.

Evaluate model at different fields of view

The same weights are run on six loops in a larger box. Voxel size matches training; only the field of view changed.

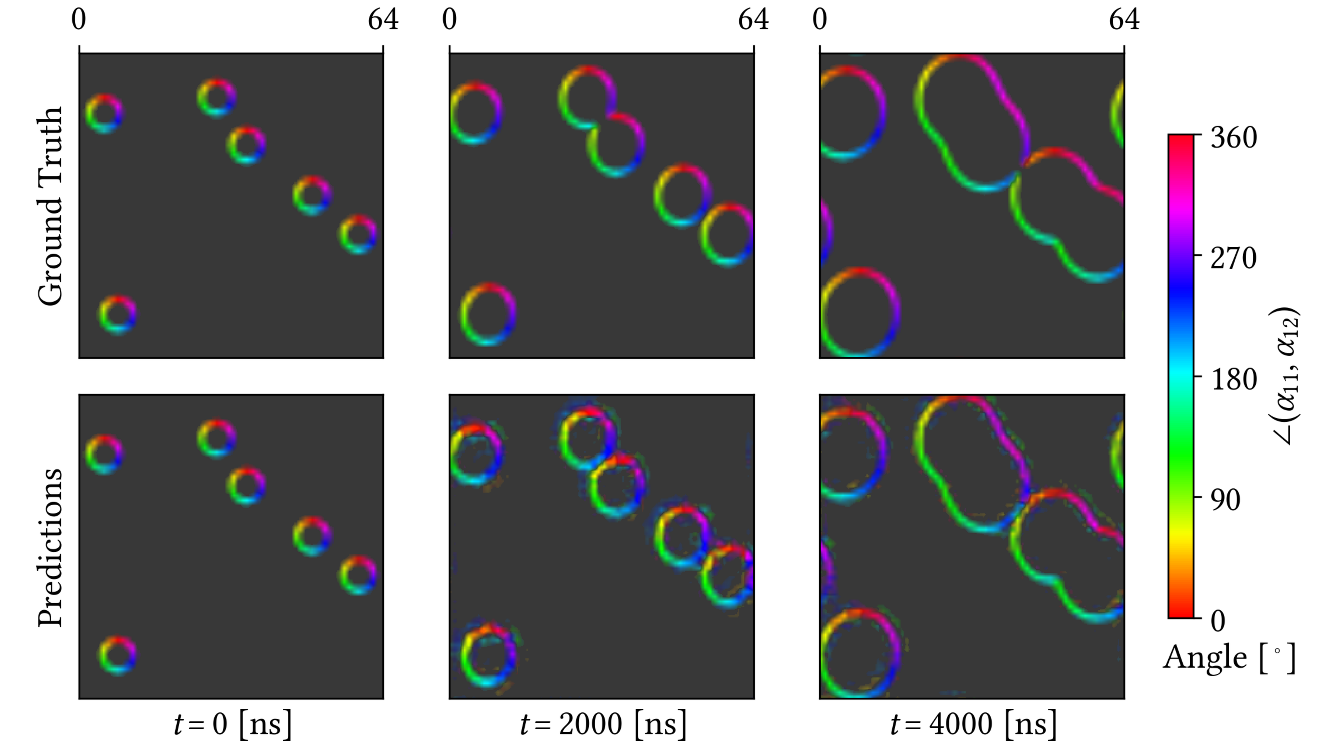

Figure 8: Six loops on 64×64 at time: $0/2000/4000\,$ns. Expansion and merging follow dislocation topological rules.

Neighboring loops that touch in DDD also merge in the predicted sequence. No weights are retrained.

Evaluation on twelve loops for a much larger box, without retraining the model.

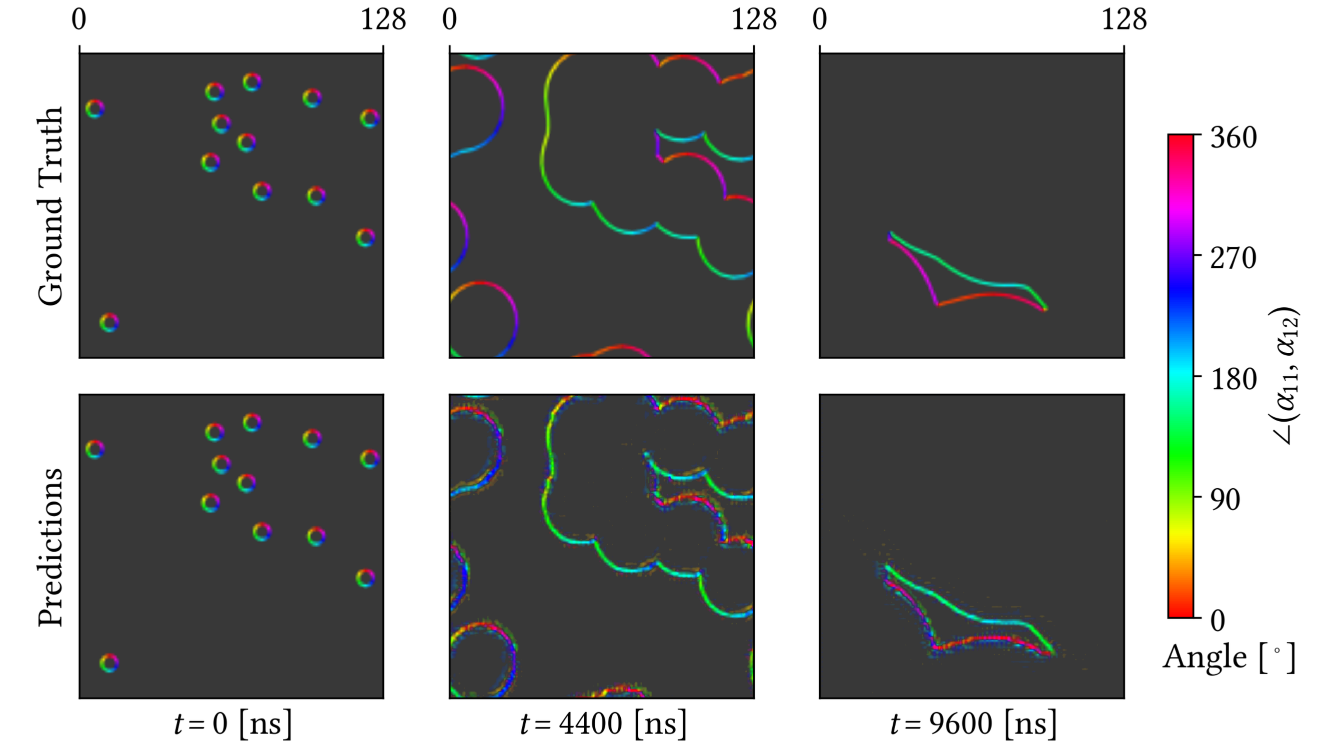

Figure 9: Twelve loops on 128×128 at time: $0/4400/9600\,$ns. Expansion and merging follow dislocation topological rules.

Dislocation cores stay localized after contact. This ML framework is based on a 2D formulation on the glide plane.

References

[1] A. Arsenlis, W. Cai, M. Tang, M. Rhee, T. Oppelstrup, G. Hommes, T. G. Pierce, and V. V. Bulatov. Enabling strain hardening simulations with dislocation dynamics. *Modelling and Simulation in Materials Science and Engineering*, 15(6):553–595, 2007.

[2] N. Ballas, L. Yao, C. Pal, and A. Courville. Delving deeper into convolutional networks for learning video representations. In *International Conference on Learning Representations*, 2016.

[3] N. Bertin. Connecting discrete and continuum dislocation mechanics: A non-singular spectral framework. *International Journal of Plasticity*, 122:268–284, 2019.

[4] J. F. Nye. Some geometrical relations in dislocated crystals. *Acta Metallurgica*, 1(2):153–162, 1953.

[5] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, volume 28, pages 802–810, 2015.

AESCAPE 2026, Dallas, TX, USA