

## A Experimental details.

### A.1 Further experiments.

Rollout steps	1	25	50	75	100
Baseline : $L = 1$	5.60e-04 (1.15e-04)	8.04e-02 (3.45e-02)	3.12e-01 (1.15e-01)	6.19e-01 (2.08e-01)	9.38e-01 (2.88e-01)
FNO [22] (4x Params.)	1.58e-03 (2.82e-04)	5.22e-02 (3.12e-02)	1.94e-01 (9.78e-02)	4.22e-01 (1.95e-01)	7.12e-01 (2.24e-01)
Spatial Hierarchical	5.15e-04 (1.14e-04)	6.59e-02 (3.59e-02)	2.69e-01 (1.31e-01)	5.77e-01 (2.39e-01)	9.28e-01 (3.08e-01)
History Hierarchy	5.11e-04 (1.14e-04)	6.41e-02 (4.03e-02)	2.36e-01 (1.23e-01)	4.99e-01 (2.29e-01)	7.79e-01 (2.70e-01)
2-step Ahead	1.06e-03 (2.18e-04)	4.22e-02 (1.87e-02)	1.79e-01 (7.70e-02)	4.11e-01 (1.69e-01)	7.17e-01 (2.40e-01)
2-step History [19]	4.84e-04 (1.09e-04)	5.21e-02 (2.76e-02)	2.02e-01 (8.99e-02)	4.42e-01 (1.86e-01)	7.67e-01 (2.71e-01)
3-step History [19]	<b>4.53e-04 (9.65e-05)</b>	4.28e-02 (2.10e-02)	1.81e-01 (8.71e-02)	3.96e-01 (1.57e-01)	6.81e-01 (2.39e-01)
Ours: $L = 2$	5.25e-04 (1.15e-04)	4.02e-02 (1.92e-02)	1.61e-01 (6.50e-02)	3.73e-01 (1.51e-01)	6.55e-01 (2.44e-01)
Ours: $L = 3$	5.50e-04 (1.20e-04)	<b>3.37e-02 (1.41e-02)</b>	<b>1.40e-01 (6.16e-02)</b>	<b>3.24e-01 (1.18e-01)</b>	<b>5.92e-01 (1.84e-01)</b>

Table 6: **Comparisons to other methods.** We compare to other methods using mean squared error (MSE) for autoregressive roll-out across different lengths. We report the results using average and standard deviation, in parentheses, and demonstrate that ours ( $L = 3$ ) outperforms others in all steps above 1-step MSE. We also show that building models with both spatial and temporal hierarchies enhances the stability of the estimation.

Rollout steps	1	25	50	75	100
Baseline: $L=1$	8.02e-07 (4.89e-07)	9.25e-06 (4.33e-06)	1.49e-05 (5.56e-06)	2.16e-05 (7.40e-06)	2.93e-05 (1.05e-05)
FNO [22] (4x Params.)	<b>4.90e-07 (1.91e-07)</b>	4.93e-06 (1.61e-06)	9.68e-06 (2.59e-06)	1.52e-05 (3.71e-06)	2.14e-05 (4.97e-06)
Spatial-Hierarchy	6.28e-07 (3.96e-07)	1.03e-05 (4.83e-06)	1.87e-05 (7.54e-06)	2.65e-05 (9.94e-06)	3.48e-05 (1.14e-05)
History-Hierarchy	8.52e-07 (6.56e-07)	8.95e-06 (5.34e-06)	1.33e-05 (6.66e-06)	1.84e-05 (8.33e-06)	2.42e-05 (9.50e-06)
2-step Ahead	5.89e-07 (4.10e-07)	<b>5.88e-06 (3.29e-06)</b>	9.94e-06 (4.42e-06)	1.49e-05 (5.71e-06)	2.10e-05 (7.75e-06)
2-step History [19]	9.83e-07 (6.22e-07)	8.11e-06 (4.29e-06)	1.13e-05 (4.90e-06)	1.58e-05 (5.87e-06)	2.18e-05 (7.06e-06)
3-step History [19]	5.65e-07 (4.40e-07)	7.05e-06 (3.93e-06)	1.09e-05 (5.08e-06)	1.55e-05 (6.30e-06)	2.12e-05 (8.01e-06)
Ours: $L=2$	7.86e-07 (4.75e-07)	9.42e-06 (4.91e-06)	1.24e-05 (5.59e-06)	1.62e-05 (6.14e-06)	2.10e-05 (6.24e-06)
Ours: $L=3$	6.37e-07 (3.80e-07)	6.28e-06 (2.90e-06)	<b>9.29e-06 (3.72e-06)</b>	<b>1.28e-05 (4.52e-06)</b>	<b>1.76e-05 (5.44e-06)</b>

Table 7: **Energy spectrum error comparison.** Our  $L = 3$  model achieves significant reduction in energy spectrum error during short-term rollouts (up to 200 steps), outperforming all comparison methods - including a baseline with  $4\times$  more parameters - for all prediction horizons beyond single-step forecasting.

**Experimenting with FNO.** All experiments thus far have utilized a UNet architecture with Fourier layers, as described in Section 5. To provide additional comparison, we run  $L = 1$  with Fourier Neural Operator (FNO) [55]. Key adaptations include: (1) Using the  $\ell_1 + \ell_2$  loss (consistent with our other experiments in Table 6) instead of the default Sobolev-norm objective [25], as the latter caused rapid prediction divergence (even within 200 steps); (2) For  $256 \times 256$  resolution data, we choose the Fourier mode number through grid search over  $\{64, 96, 128\}$ ; (3) We use 3 FNO layers, which results in a model with 130M parameters —  $4\times$  larger than our  $L = 3$  configuration.

**Model Efficiency.** Our model utilizes the UNet’s hierarchical framework to efficiently process multi-scale data. As shown in Table 8, compared to the baseline  $L = 1$ , ours  $L = 3$  adds only a few convolutional heads for handling and outputting latent variables  $z$ , resulting in minimal parameter and inference time overhead.

	Param Counts (M)	Forward Time (seconds)
Baseline: $L = 1$	34.88	0.030
Ours: $L = 3$	36.67	0.031

Table 8: **Model Running Time and Parameters comparison.** On input with  $256 \times 256$  resolution, our design ( $L = 3$ ) leverages the inherent hierarchical representation of UNet to process hierarchical latent variables, leading to minimal computational overhead than the baseline ( $L = 1$ ).

### A.2 Evaluation metrics.

**Mean squared error (MSE).** We use

$$\text{MSE}(\mathbf{u}_n, \hat{\mathbf{u}}_n) = \|\mathbf{u}_n - \hat{\mathbf{u}}_n\|_2^2$$

as our primary metric to quantify the prediction error for short-term predictions.

**Long-term stability.** To evaluate the long-term stability of predictions, we leverage the system’s conserved energy, defined as  $E = \frac{1}{2}(v_x^2 + v_y^2)$ . As discussed in Section 5.1, we calculate the standard deviation of the ground truth energy values. A trajectory is deemed stable if its energy remains within 5 standard deviations of this reference. This threshold is informed by the observation that all training data lie within 4 standard deviations, while even the true dynamics extrapolated over ten times the training horizon (simulating extreme long-term behavior) do not exceed 5 standard deviations, as shown in Fig. 7. Appendix B.2 provides a detailed visual comparison when the predicted dynamics exceed  $\pm 5$  standard deviation range.

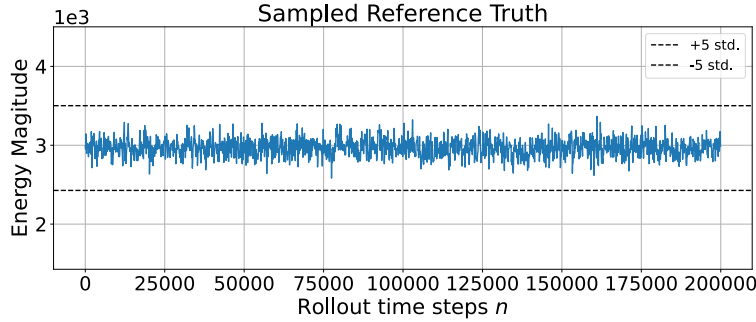


Figure 7: **Energy of true dynamics along extended timescale.** We compute the energy evolution of the true dynamics over 10-times the training dataset length ( $2 \times 10^5$  time steps). We show that energy along the sampled true dynamics remains within the  $\pm 5$  standard deviation of its mean.

**Fourier spectrum error.** We follow the implementation in py2d to compute the energy spectrum<sup>2</sup>. For short-term evaluation, we compute the mean absolute error of the energy spectrum—the spatial FFT  $\mathcal{F}[\mathbf{u}_n]$ , averaged over  $N$  timesteps:

$$\frac{1}{N} \sum_{n=1}^N \|\mathcal{F}[\mathbf{u}_n] - \mathcal{F}[\hat{\mathbf{u}}_n]\|_1.$$

For long-term evaluation (e.g.,  $N = 2 \times 10^5$ ), where a ground truth trajectory is unavailable for every new initial condition, and the system loses memory of initial conditions showing an invariant spectrum due to its ergodic properties, we instead compare against a fixed reference spectrum  $\mu[\mathcal{F}[\mathbf{u}]] := \frac{1}{N} \sum_{n=1}^N \mathcal{F}[\mathbf{u}_n]$ , computed from an extremely long true trajectory. The error is then:

$$\left\| \frac{1}{N} \sum_{n=1}^N \mathcal{F}[\mathbf{u}_n] - \mu[\mathcal{F}[\mathbf{u}]] \right\|_1.$$

**Zonal mean.** The zonal mean of vorticity is computed by averaging vorticity perpendicular to the jets (i.e., along the  $x$ -direction), shown as:

$$\frac{1}{NN_x} \sum_{n=1}^N \sum_{x=1}^{N_x} \mathbf{u}_{n,x}.$$

$\mathbf{u}_{n,x} \in \mathbb{R}^d$  denotes the vector of the vorticity at  $x$ -axis.  $N_x$  denotes the number of discretization points.

<sup>2</sup><https://github.com/envfluids/py2d/blob/main/py2d/spectra.py>

### A.3 Experimental setup.

**Model Architecture** We adopt the UNet architecture following the design in [53], with several customizations. For the experiments on  $256 \times 256$  resolution data, our encoder consists of 5 groups with latent channel sizes of [16, 32, 64, 128, 128, 128]. The spatial resolution is halved after each encoder group, resulting in a  $8 \times 8$  resolution at the bottleneck for  $256 \times 256$  input images. Each encoder group contains two convolutional blocks, which is made up of two convolutional layers with group normalization and residual connections. The decoder mirrors the encoder and incorporates skip connections from corresponding encoder layers. To improve the model’s ability to capture long-range spatial dependencies and multi-frequency signals, we add a Fourier layer to each convolutional block. To balance between accuracy and computational efficiency, we limit the number of frequency modes to 96 for  $256 \times 256$  data and apply Fourier convolutions in a depth-wise manner, *i.e.*, without inter-channel communication.

For  $512 \times 512$  resolution inputs, we use a similar architecture but add an extra encoder group, resulting in latent channel sizes of [16, 32, 32, 64, 128, 128, 128]. In the  $L = 3$  setup, we set  $r^1 = 8$  and  $r^2 = 32$ , the same rates used for our  $256 \times 256$  resolution experiments. Using these rates, we inject latent codes  $z_1^{(1)}$  and  $z_2^{(2)}$  into encoder groups with feature resolutions of  $64 \times 64$  and  $32 \times 32$ .

**Data preprocessing.** To stabilize training, we normalize the data using a constant scaling factor such that the resulting values have an approximate standard deviation of one. Specifically, for the dataset with Reynolds number 10000, we apply a dividing factor of 10, while for the dataset with Reynolds number 5000, we use a dividing factor of 6.

**Corner cases.** For models that take multiple temporal frames as input, we simulate the initial rollout setting during training by randomly zeroing out early history frames with a probability of 15%, approximating the absence of pre-initial frames.

**Upsampling projections.** Prior to injection, we upsample the latent variables and apply convolutional layers to match the corresponding encoder channels. Decoding is performed at the same spatial resolutions as the injected latent codes. The same architectural setup as  $L = 3$  is used to construct baseline models for Spatial Hierarchy and History Hierarchy. In the  $L = 2$  case, we use  $r^1 = 32$  and inject  $z_1^{(1)}$  into the encoder group at the  $64 \times 64$  resolution level.

**Training details.** We process the raw data generated from “py2d” [52] solver. Our emulator predicts vorticity at 0.05 time intervals, representing a  $500\times$  coarser temporal resolution than the numerical solver’s 0.0001 timestep. We normalize the raw simulation data to approximate a standard normal distribution. Since the data is naturally zero-centered, we simply scale it by constant factors (10 for jet-containing flows and 6 for jet-free flows in Section 5.3) to achieve unit standard deviation.

For all experiments, we use the AdamW optimizer with learning rate at  $3 \times 10^{-4}$ . We conduct experiments on NVIDIA A100, H100, and L40S GPUs.

## B Additional visualizations.

### B.1 Further visualizations.

We provide additional visualization of short-term rollout in Fig. 8, Fig. 9, Fig. 10, and Fig. 11. The comparison across various datasets and methods show consistent improvements of our methods ( $L = 2$  and  $L = 3$ ).

### B.2 Stability.

We present four trials of long-term rollouts across all methods, each with distinct initial conditions. The energy evolution over  $2 \times 10^5$  timesteps is shown in Figures 12 and 15, while the corresponding dynamic visualizations appear in Figures 13 through 17.

Our analysis reveals that deviations beyond the  $\pm 5$  standard deviations range of the ground truth energy distribution consistently correlate with unstable, exploding dynamics or overly smoothed, averaged patterns. While baseline methods exhibit persistent failures once instability occurs, our  $L = 2$  model demonstrates a unique recovery pattern, where the dynamics (and associated energy values) can return to physically plausible states after temporary deviations. This robustness stems

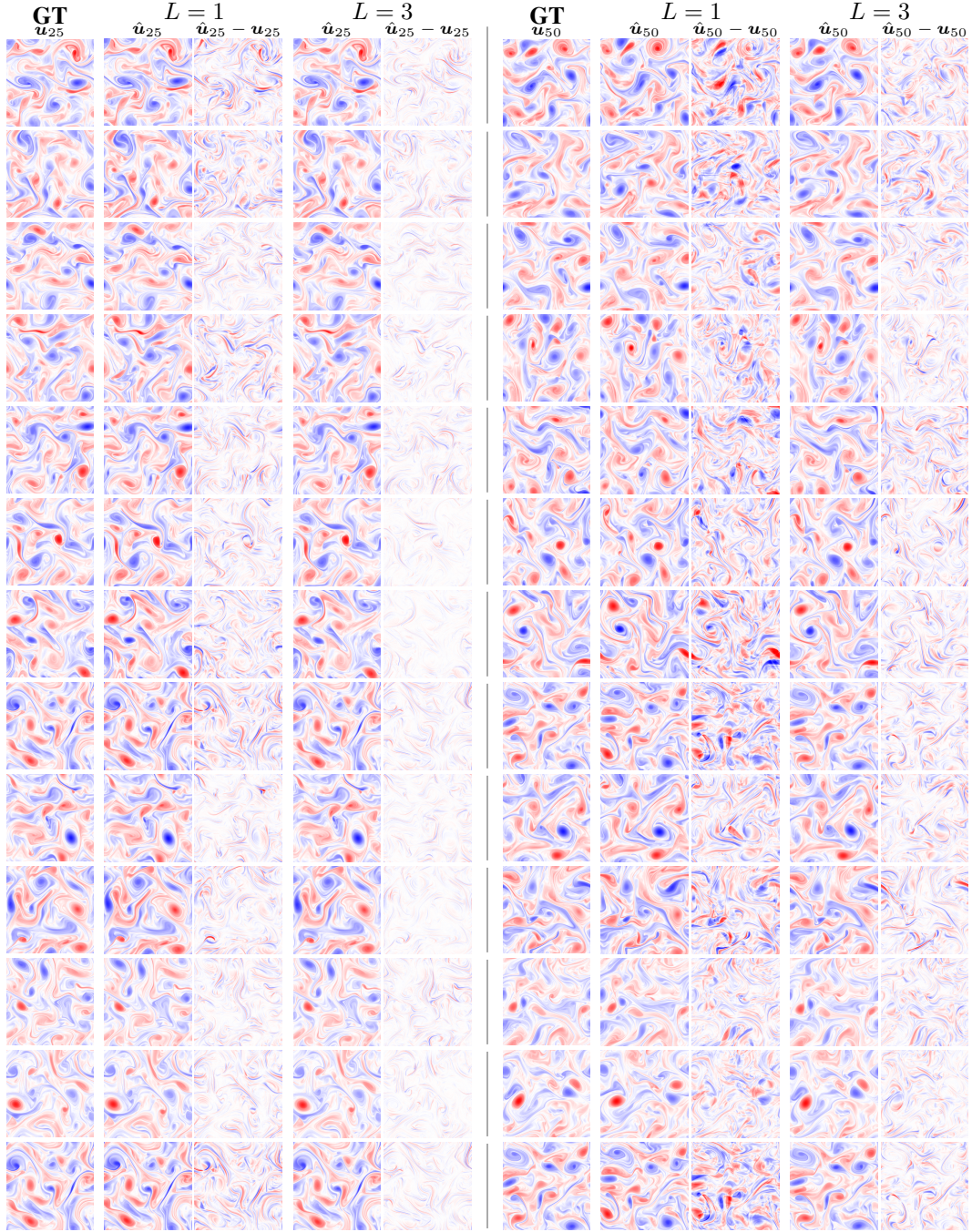


Figure 8: We apply our approach to flow dataset of  $Re = 5 \times 10^3$ ,  $256 \times 256$  resolution without zonal jets. Our method ( $L = 3$ ) gives more accurate predictions with lower associated residuals.



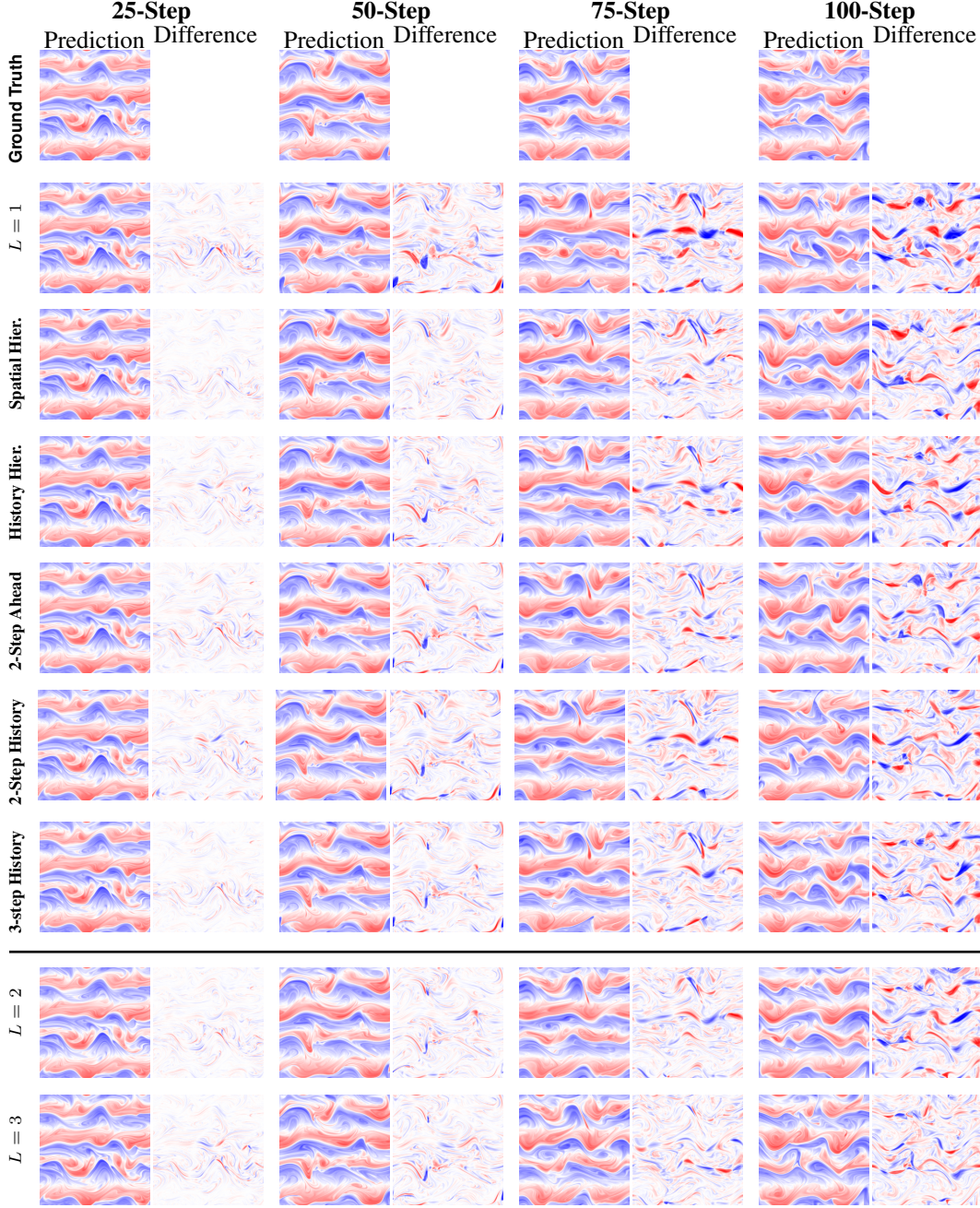


Figure 9: Visualization of prediction and residual to ground truth across methods and prediction steps. Our methods  $L = 2$ ,  $L = 3$  consistently outperform all compared methods.

from the guidance provided by the top-level compressed variables, which help correct errors in fine-grained details.

For stability rate calculations, we conservatively classify a trajectory as unstable if it ever exceeds the predefined  $\pm 5$  standard deviations' bounds, regardless of subsequent recovery. This ensures fair comparison across methods.

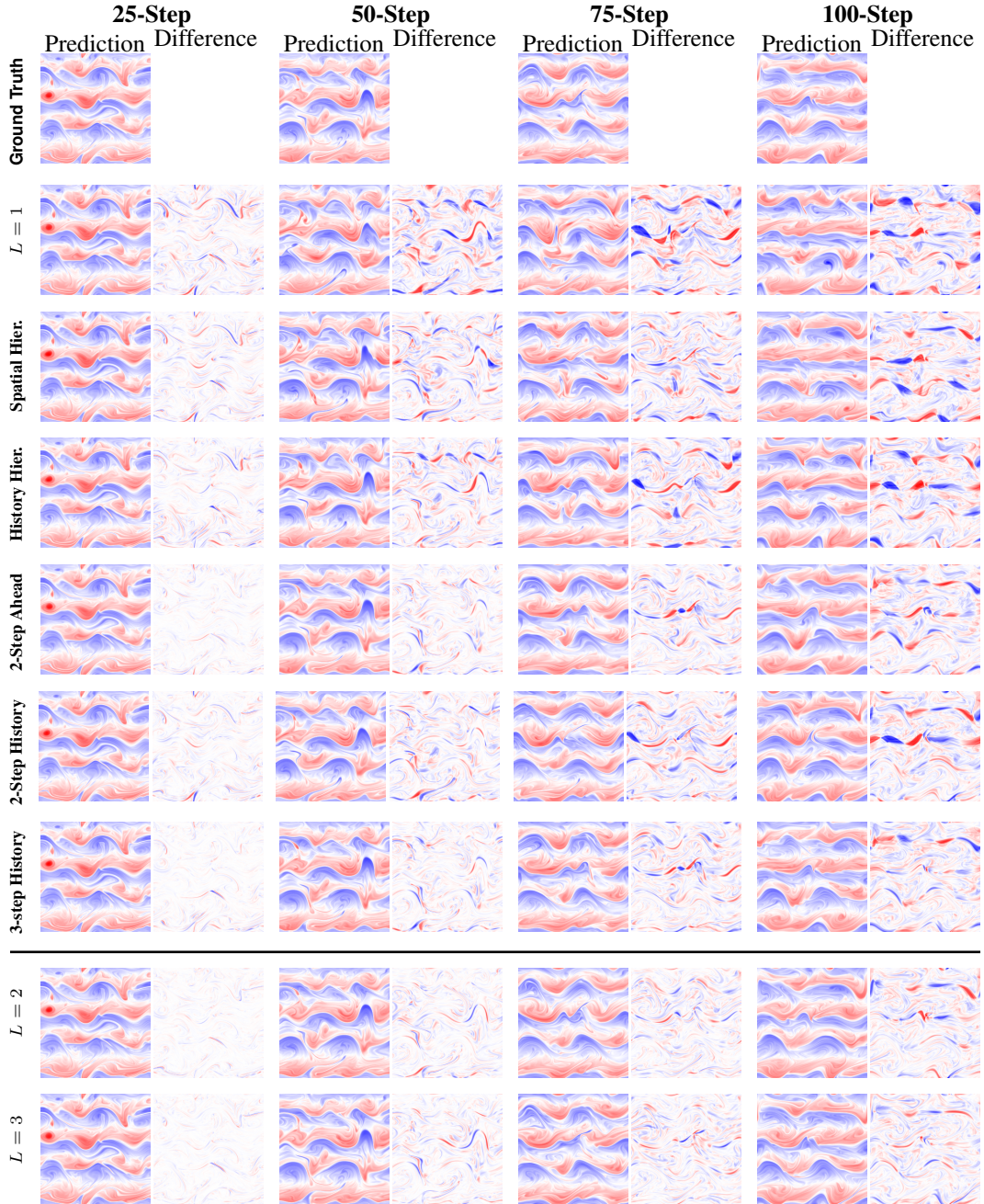


Figure 10: Visualization of prediction and residual to ground truth across methods and prediction steps. Our methods  $L = 2, L = 3$  consistently outperform all compared methods.



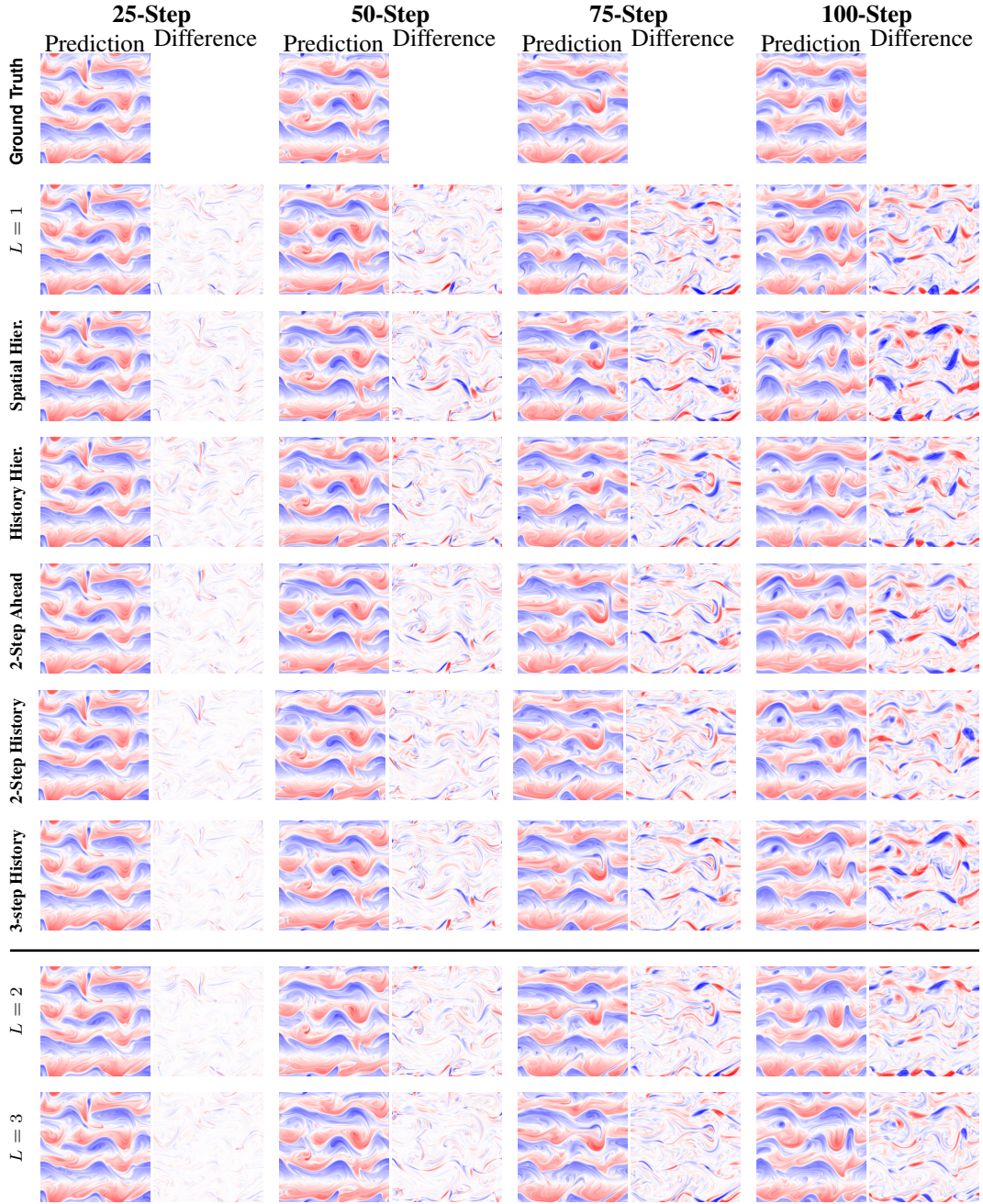


Figure 11: Visualization of prediction and residual to ground truth across methods and prediction steps. Our methods  $L = 2, L = 3$  consistently outperform all compared methods.

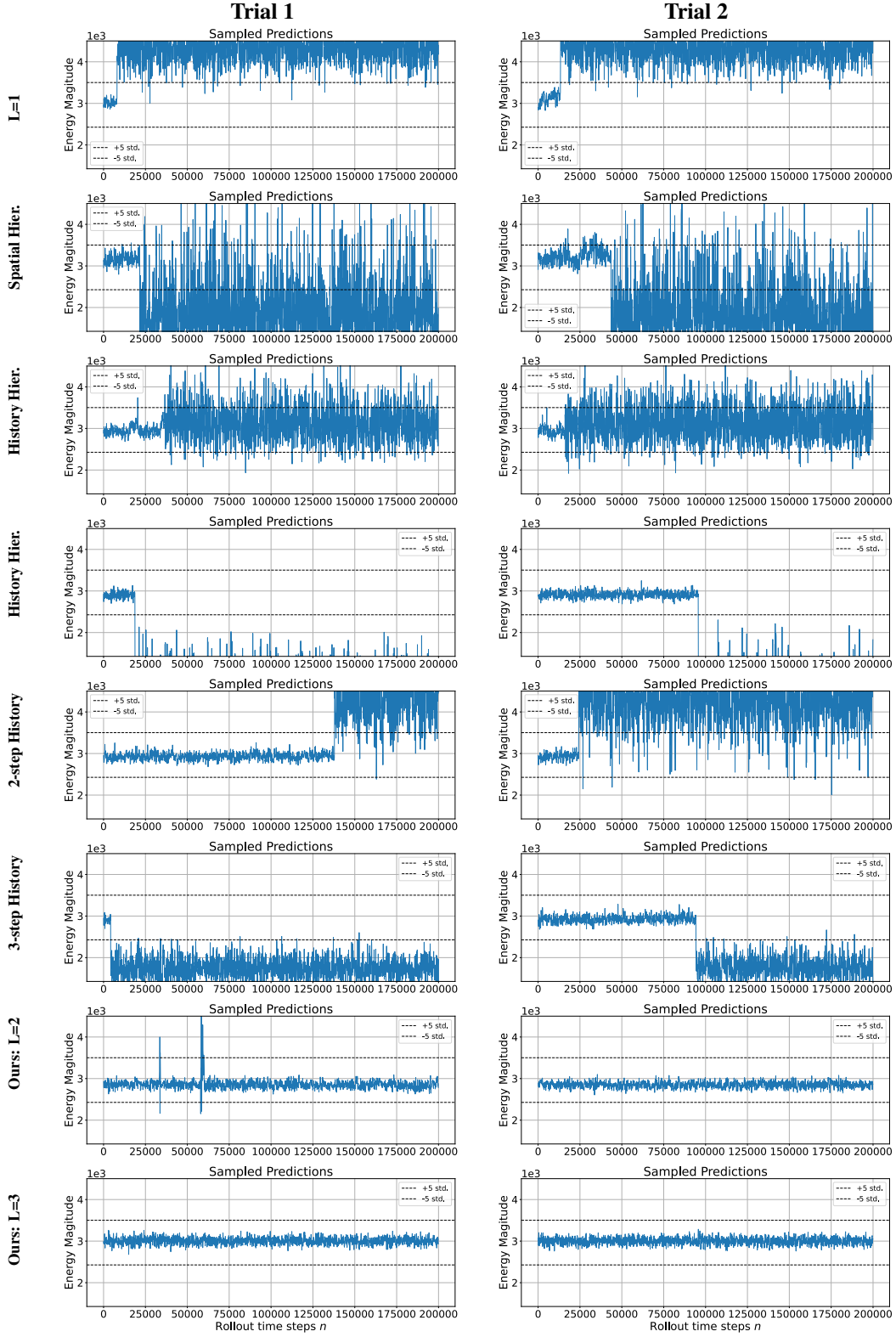


Figure 12: **Energy v.s. rollout time steps of the predicted dynamics.** The dashed horizontal line shows the maximum and minimum values computed using 5 standard deviations from the mean of the true dynamics. Our  $L = 3$  method is able to maintain energy along the long-term predictions.



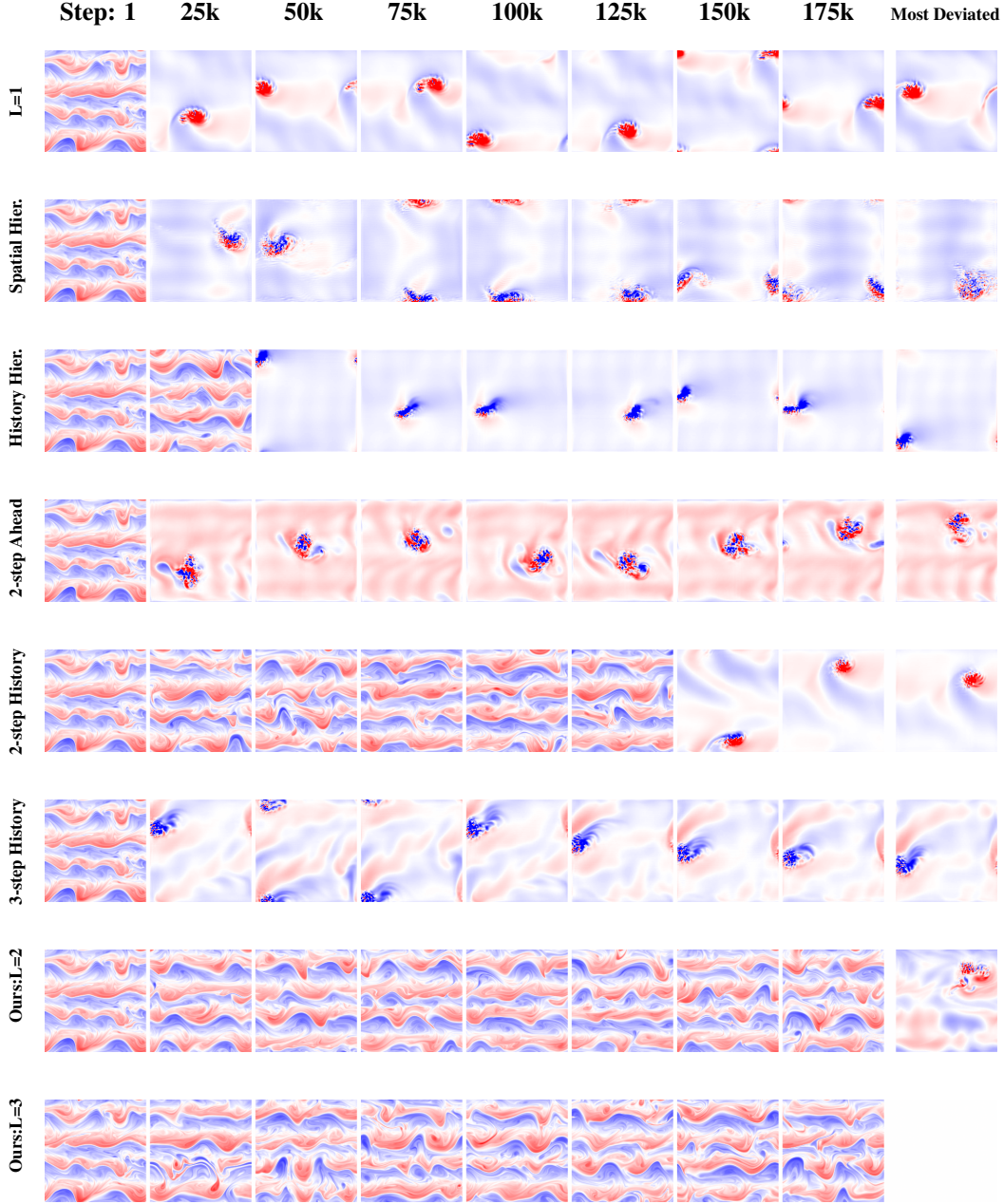


Figure 13: **Long-Term Rollout Visualization.** Long-term dynamics corresponding to Trial 1 in Fig. 12. The rightmost column shows the frame with the maximum deviation—exceeding  $\pm 5$  standard deviations from the reference truth statistics, with a blank block indicating that all states along the sampled trajectory remain within this range. When energy predictions fall outside the reference distribution, non-physical dynamics emerge (exploding/averaging artifacts; blank blocks indicate predictions within the  $\pm 5$  std. range). Our  $L = 3$  model demonstrates superior stability across all comparisons, while  $L = 2$  exhibits deviations correlated with energy prediction errors.

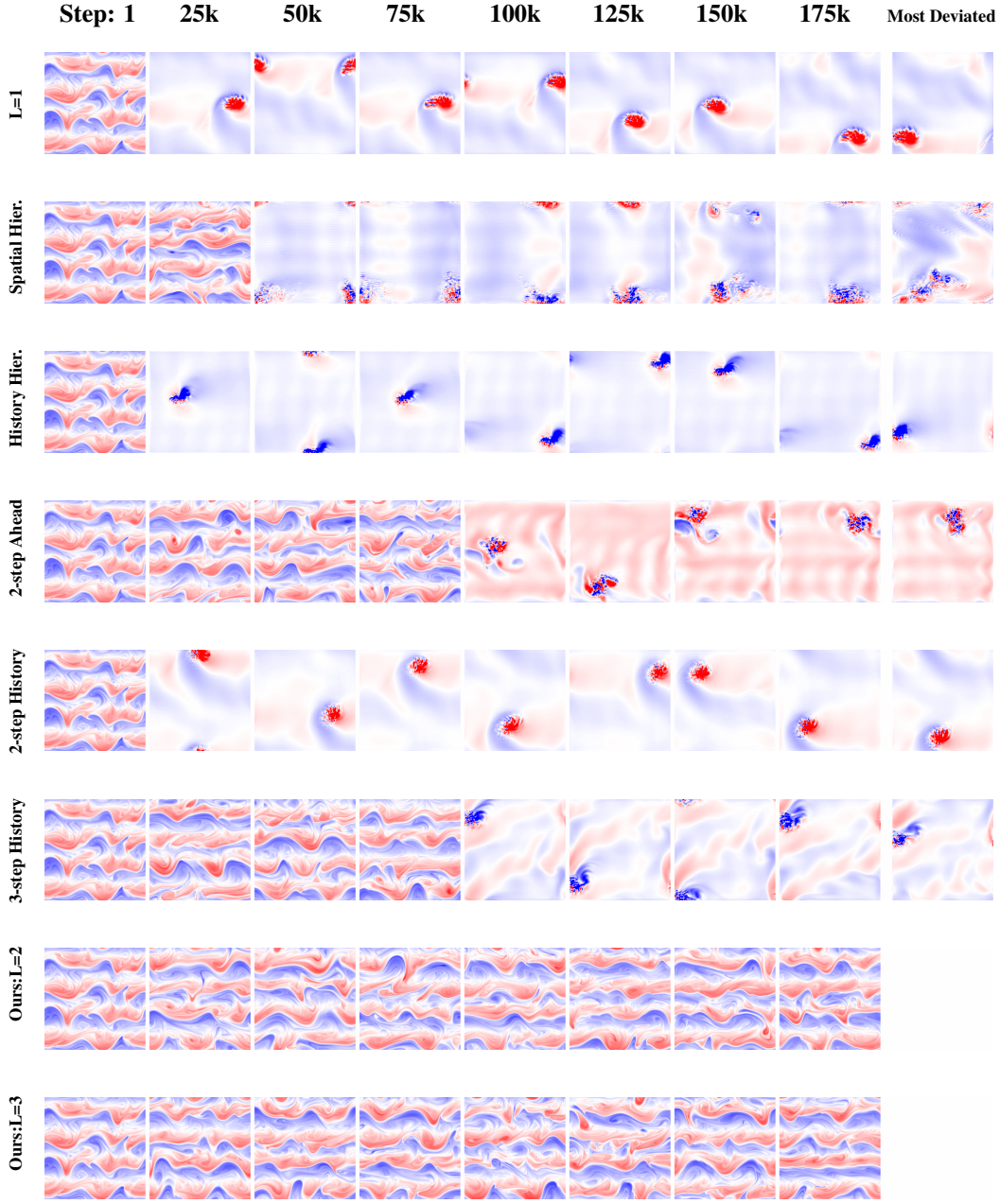


Figure 14: **Visualization of long-term rollout.** We visualize the long-term dynamics corresponds to trial 2 in Fig. 12. The rightmost column shows the frame with the maximum deviation—exceeding  $\pm 5$  standard deviations from the reference truth statistics, with a blank block indicating that all states along the sampled trajectory remain within this range. The results show that when the energy is deviated from the reference truth distribution, the dynamics exhibit exploding or averaging non-physical dynamics. Among all comparison methods, our  $L = 3$  gives the most stable predictions.

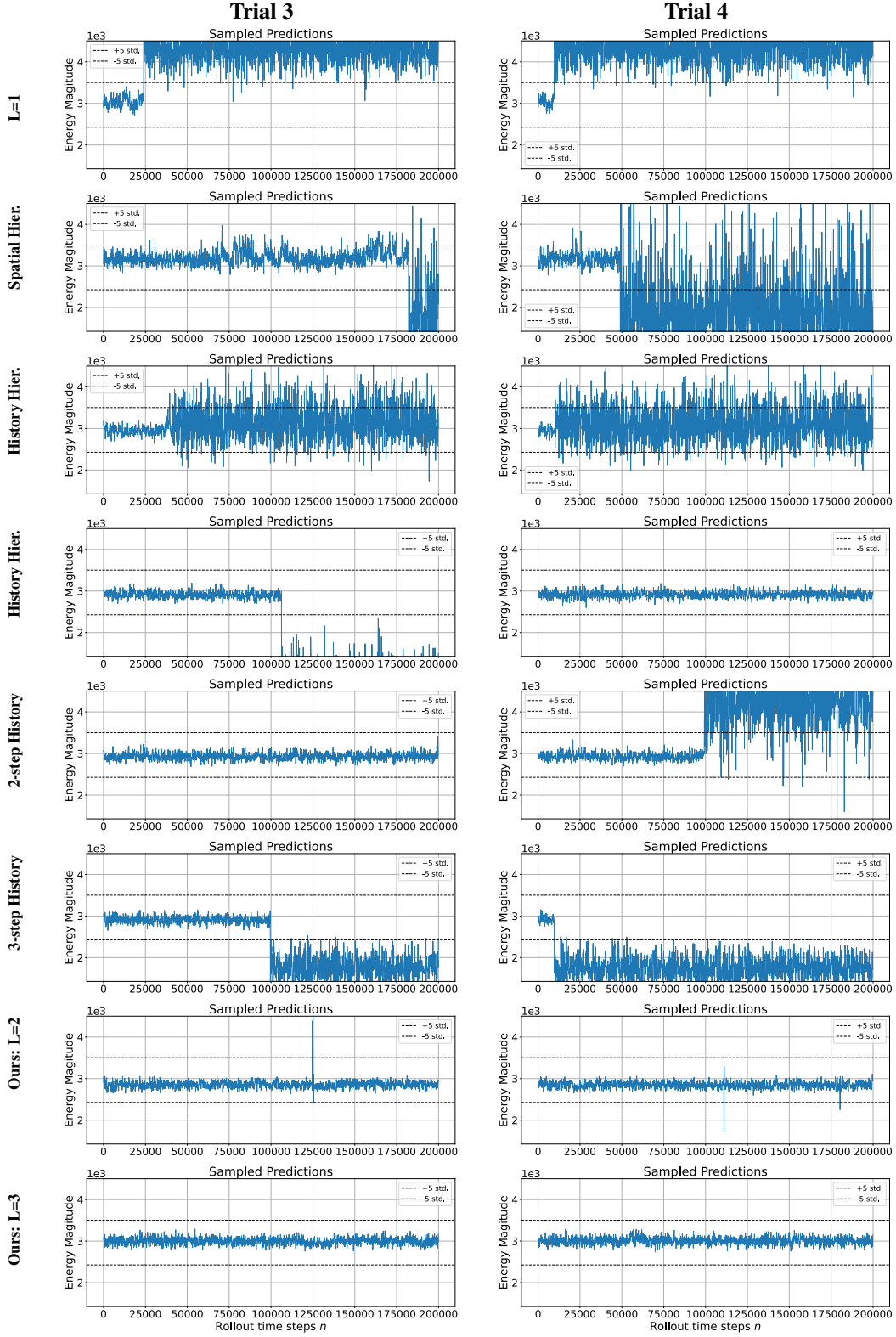


Figure 15: **Energy v.s. rollout time steps of the predicted dynamics.** The dashed horizontal line shows the maximum and minimum values computed using 5 standard deviations from the mean of the true dynamics. Our  $L = 3$  method is able to maintain energy along the long-term predictions.

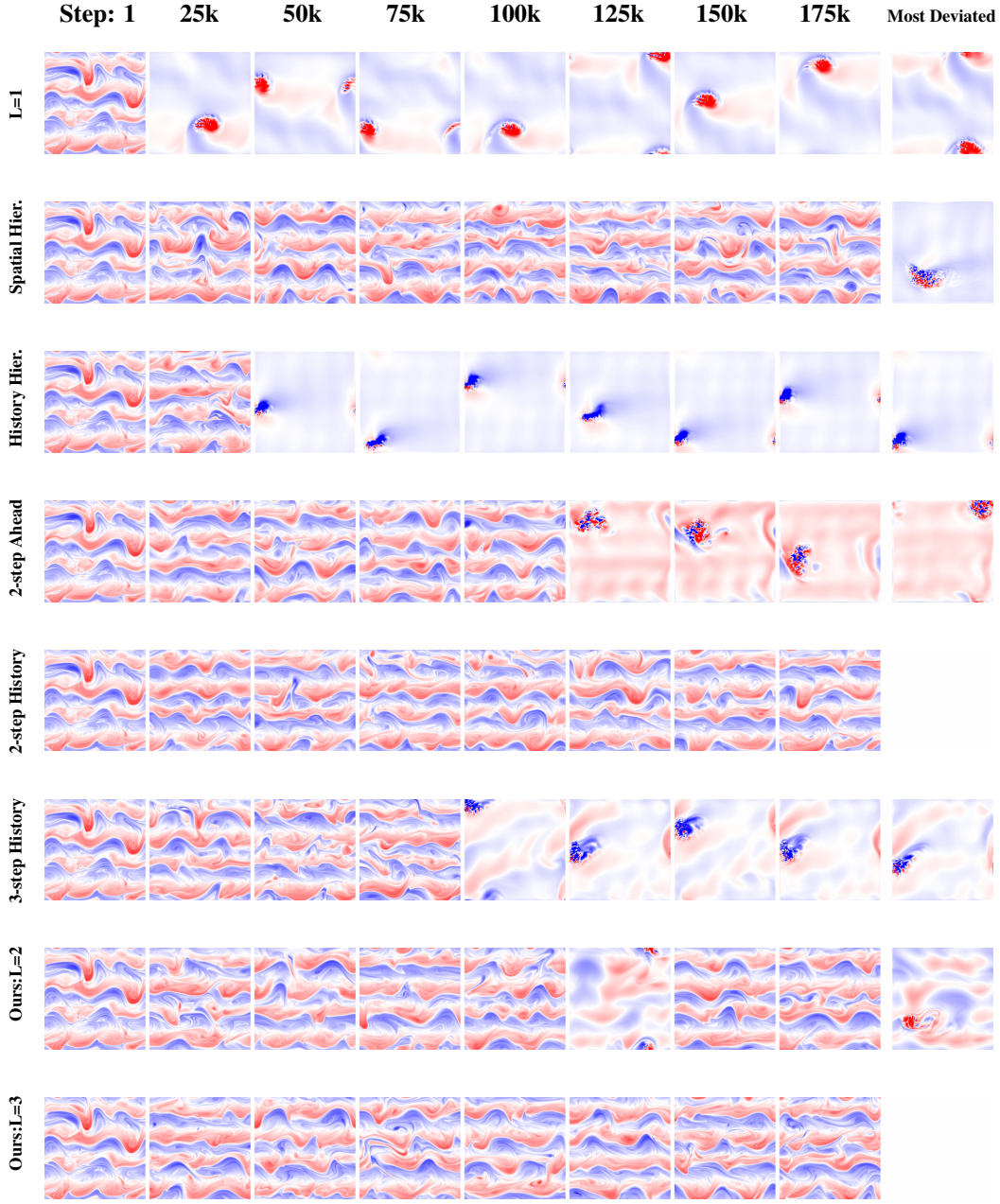


Figure 16: **Visualization of long-term rollout.** We visualize the long-term dynamics corresponds to trial 3 in Fig. 15. The rightmost column shows the frame with the maximum deviation—exceeding  $\pm 5$  standard deviations from the reference truth statistics, with a blank block indicating that all states along the sampled trajectory remain within this range. The results show that when the energy is deviated from the reference truth distribution, the dynamics exhibit exploding or averaging non-physical dynamics. Among all comparison methods, our  $L = 3$  gives the most stable predictions.



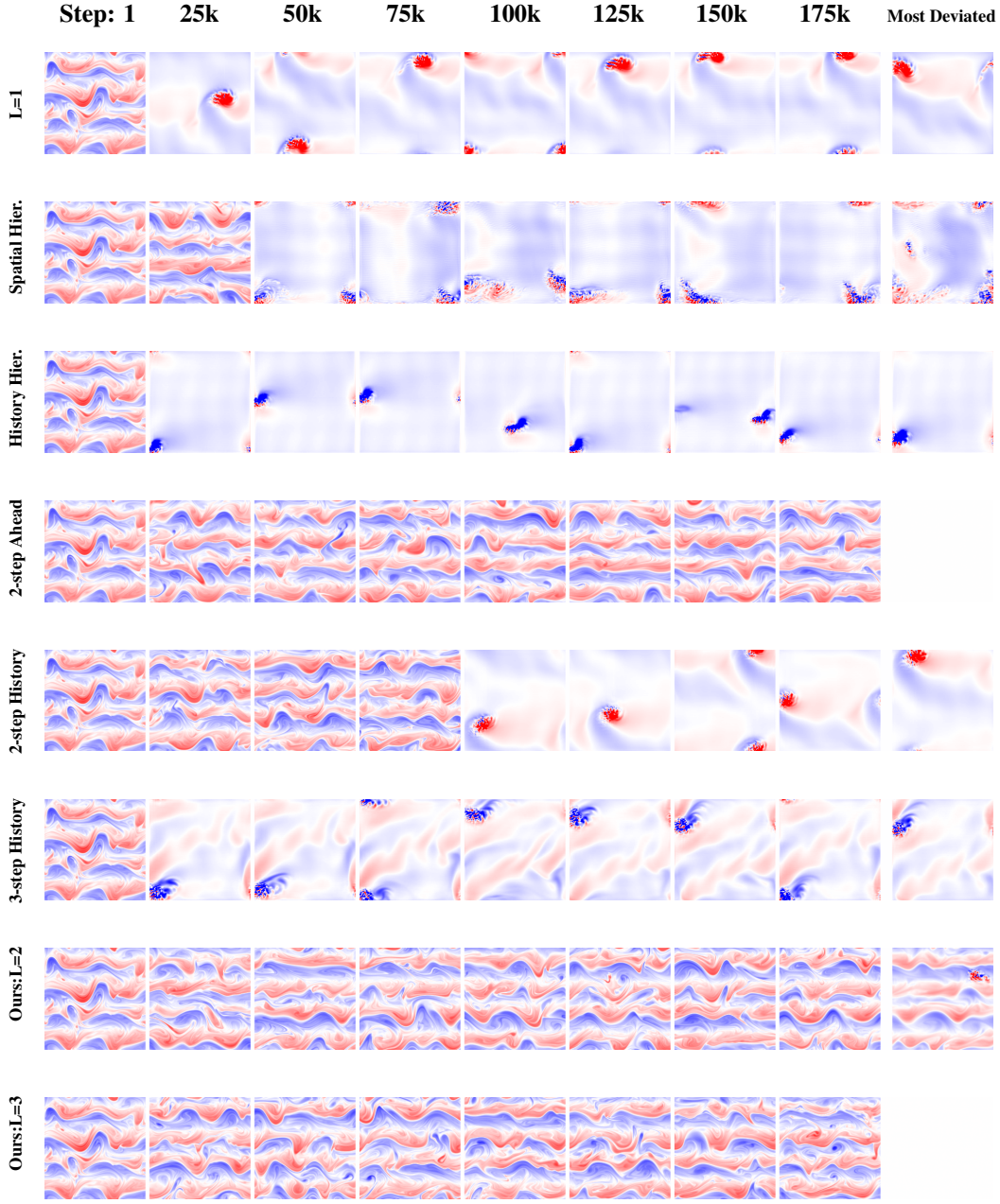


Figure 17: **Visualization of long-term rollout.** We visualize the long-term dynamics corresponds to trial 4 in Fig. 15. The rightmost column shows the frame with the maximum deviation—exceeding  $\pm 5$  standard deviations from the reference truth statistics, with a blank block indicating that all states along the sampled trajectory remain within this range. The results show that when the energy is deviated from the reference truth distribution, the dynamics exhibit exploding or averaging non-physical dynamics. Among all comparison methods, our  $L = 3$  gives the most stable predictions.