

783 A Detailed model and training descriptions

784 A.1 Additional training details

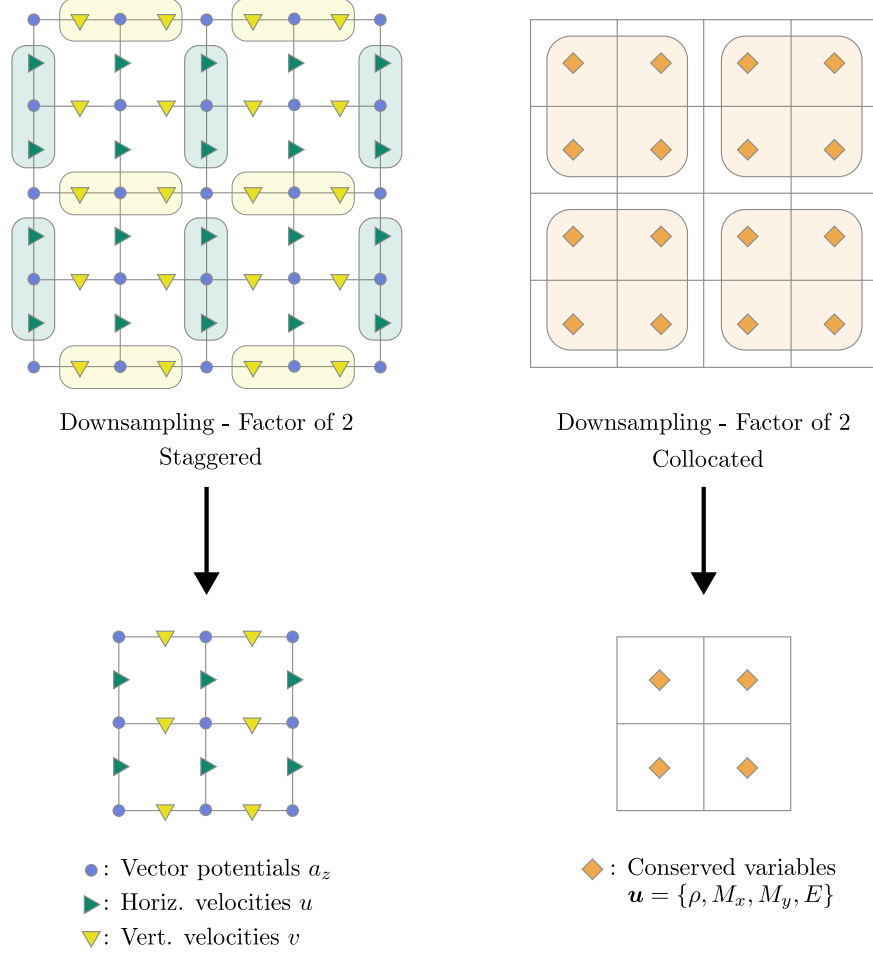


Figure 6: Coarsening operations for staggered and collocated grids. For staggered grids, we employ the face averaging approach for velocities described in [Kochkov et al., 2021]. This method conserves momentum and maintains incompressibility. In contrast, for collocated grids, we apply non-overlapping averaging that adheres to conservation laws.

785 We provide the summary of the training parameters in Table 1.

786 **Training of our decoder and Super-resolution (SR) models** We train the SR models separately
 787 for upsampling tasks without unrolling in time. Since the rolled-out auxiliary variables \tilde{c}_{t+k} are not
 788 expected to match the encoded variables $\mathcal{E}_\phi(X_{t+k})$, our decoder is trained on rolled-out trajectories
 789 of $(\tilde{x}_t, \tilde{c}_t)$. However, we stop gradient propagation to both the processor and encoder during decoder
 790 training.

791 A.2 Additional model details

792 In this section, we give additional model details that are not present in the main text or Table 1. We
 793 also describe our modifications to the official implementations of FactFormer and DINO.

794 **mUnet** Our implementation of mUnet uses an initial convolution layer to lift the channel dimension
 795 to 32. At each level of the downsampling, two consecutive ConvNeXt layers [Woo et al., 2023]

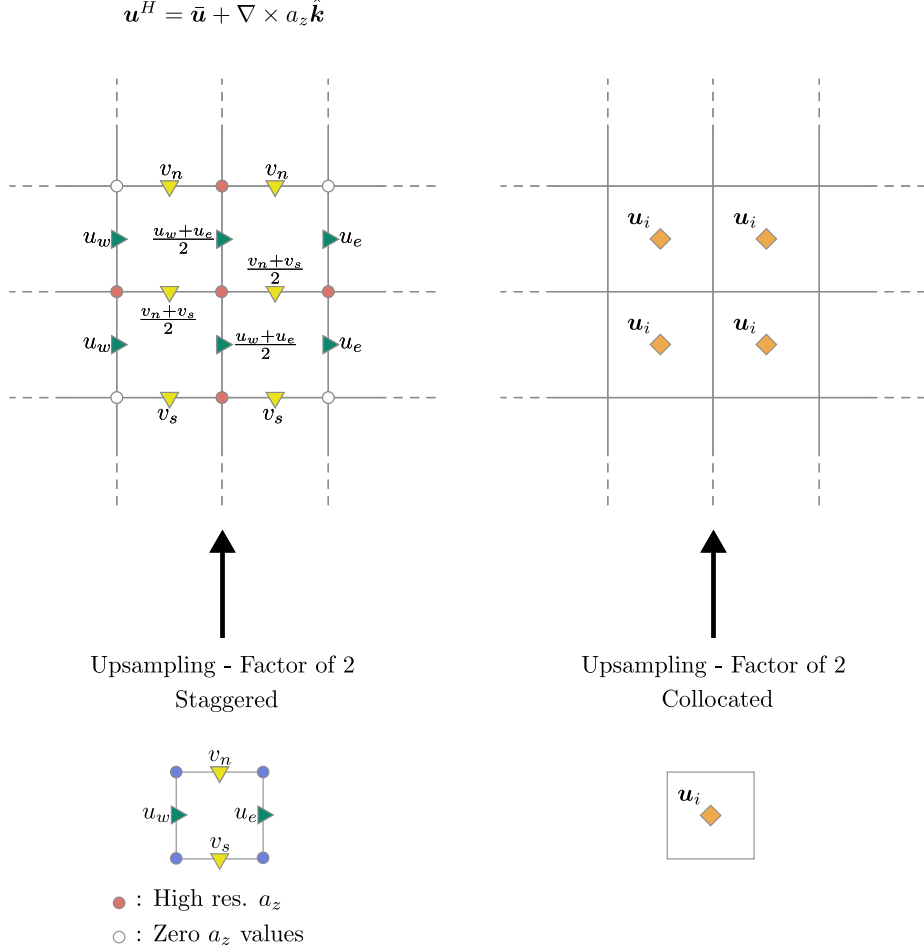


Figure 7: Upsampling methods for staggered variables and collocated grids. For staggered variables, face values are copied and velocity values in inner cells are linearly interpolated, preserving momentum conservation and incompressibility. Corrections are learned through a vector potential a_z . We either fix the corner values of a_z to satisfy Equation 14 or relax this assumption to obtain continuous upsampled velocity fields. For collocated grids, upsampling is performed by copying pixels from the low-resolution input or by bilinearly interpolating them.

are followed by a 2D convolution using a stride 2 and a group norm with 32 groups. The channel dimensions for each level of downsampling are [32, 64, 128, 256]. Each skip connection incorporates linear self-attention operations. The bottleneck of the mUnet consists of a ConvNeXt layer followed by full self-attention and another ConvNeXt layer. The upsampling blocks are designed to be symmetric to the down blocks (We use transpose convolutions for upsampling).

HRE ours Our implementation uses the up and down blocks of the mUnet without the skip connections. The encoder \mathcal{E}_ϕ 's channel counts are [32, 64, 128, 16] with corresponding norm groups of [32, 32, 32, 8]. The processor maintains the same hyperparameters as the mUnet but uses a different initial lifting convolution that operates on (x_t, c_t) pairs. c_t fields are passed through ConvNeXt layers before and after the processor, while x_t fields are passed through a physical constraint layer after the processor. The decoder is symmetric to the encoder except that the inputs are (x_t, c_t) pairs and the first group normalization uses 9 groups for ID, KF and KFHR tasks and 10 groups for the CD task.

FactFormer We increase the input channel dimensions to 2. For rollout resolutions of 256^2 and 512^2 , to prevent out-of-memory errors, we decrease the latent steps from 4 to 2, the output window

from 16 to 10 and the batch size from 32 to 4. We also decrease the maximum rollout length for the pushforward trick in our training curriculum from 4 to 2.

DINo We change the number of input channels. We observed that the performance of DINo after 12K epochs is not competitive. Therefore, instead of training DINo for 12K epochs, we train it for 60K epochs with early stopping patience of 20K epochs.

A.3 Physics-constrained decoders for hybrid representations

Here we describe the construction of decoders with hard physical constraints. We consider two cases: conservation of multiple quantities on a collocated grid, and incompressibility with conservation of momentum on a staggered grid. In each case, we learn a correction \mathcal{B}_c that is added to an interpolation operator \mathcal{D}_x . \mathcal{D}_x is the coarse-to-fine counterpart of \mathcal{E}_x (Fig. 7), and obeys all relevant physical constraints for each PDE.

A.3.1 Collocated grid

Our decoder on a collocated grid is defined as:

$$\mathcal{B}_c(\tilde{x}_t, \tilde{c}_t) = \mathcal{B}_\psi(\tilde{x}_t, \tilde{c}_t) - \mathcal{D}_x \circ \mathcal{E}_x \circ \mathcal{B}_\psi(\tilde{x}_t, \tilde{c}_t) \quad (13)$$

Here \mathcal{D}_x , a fixed interpolation operator respecting physical constraints (e.g., bilinear or nearest neighbor interpolation), is the coarse-to-fine counterpart of \mathcal{E}_x (See Fig. 6, 7-right). \mathcal{B}_ψ is an unconstrained mapping from $(\tilde{x}_t, \tilde{c}_t)$ onto an additive correction for \tilde{X}_t . In eq. 13 the second term ensures that \mathcal{B}_c corrects the physically constrained quantity (e.g. density, momentum or energy) in a way that respects physical constraints, and ensures the decoder satisfies the consistency condition:

$$\forall x_t, c_t : \quad \mathcal{E}_x \circ \mathcal{D}_c(x_t, c_t) = x_t \quad (14)$$

For example, if \mathcal{E}_x averages a density field over nonoverlapping $r \times r$ pixel squares, and \mathcal{D}_x copies each pixel $r \times r$ times, then \mathcal{B}_c has 0 mean on each $r \times r$ square of outputs.

A.3.2 Staggered grid

For cases where the divergence-free condition must be satisfied, the decoder ensures this by representing the correction operator as the curl of a learned high-resolution vector potential:

$$\mathcal{D}_c(\tilde{x}_t, \tilde{c}_t) = \nabla \times (D_\psi(\tilde{x}_t, \tilde{c}_t) \cdot \delta_r(i, j)) \quad (15)$$

Here, i and j are pixel indices of the high-resolution output. Function $\delta_r(i, j)$ fixes the corners of each $r \times r$ square on the learned vector potential to zero (Fig. 7-left). This guarantees that the decoder satisfies the eq. 14 and is the counterpart of face-averaging \mathcal{E}_x in Fig. 6-left. The function and $\delta_r(i, j)$ is defined as:

$$\delta_r(i, j) = \begin{cases} 0 & \text{if } i \bmod r = 0 \text{ and } j \bmod r = 0, \\ 1 & \text{otherwise.} \end{cases} \quad (16)$$

We observe that fixing the corner vector potential values introduces discontinuities that the correction term cannot rectify. Therefore, we relax the assumption of Equation 14 and allow the decoder to output continuous vector potential fields. This relaxation does not violate the conditions of incompressibility or momentum conservation. However, the encoding and decoding processes compromise the immutability of the coarse field variables, meaning that repeated encoding-decoding of X_t changes the coarsened fields x_t .

B Measurement of inference speed

We analyze inference times across varying batch sizes for the ID, KF, and KFHR cases at rollout resolutions of 32^2 and 64^2 on an A100 GPU with 40GB memory. We exclude the data transfer and encoding costs for DINo and our model to obtain the marginal rollout cost. To obtain the scaling behavior, we fit a linear regression using least squares to the largest three batch size measurements for each model and calculate the resulting slope (However, we use the inference speed for batch

size of 32 in Figure 1 due to GPU memory limitations). This slope approximates the per-batch-element inference time under GPU saturation conditions (considering that the last three points lie approximately on a line), while the y-intercept represents the computational overhead inherent to each model. The complete results are presented in Figures 23 and 24. In both cases, DilResNet is the slowest and DINO is the fastest model. Also, DINO has the smallest overhead while DilResnet has the highest. Our model is slightly slower than mUnet and slightly faster than FactFormer. All models are faster than the numerical solver operating in the native resolution, which takes ≈ 19.5 seconds to solve the system. However, the numerical solver operating in 512^2 input size takes ≈ 0.6 seconds, meaning that models operating at 64^2 rollout resolution with batch size 1 can be slower than this low-resolution numerical solver.

C Additional dataset details

In this section we give full details on the datasets used for each emulation task. All datasets are stored in float32 precision. Additional information and dataset parameters are summarized in Table 3. Initial conditions are taken from [Kochkov et al., 2021] for ID, and from [Takamoto et al., 2023] for CD. For KF and KFHR, we take independent samples from the stationary distribution.

C.1 2D Kolmogorov Flow (KF)

We use the staggered conservative DNS solver from the jax-cfd package [Kochkov et al., 2021] to integrate this system with periodic boundary conditions. We generate 128 experiments using 2048×2048 cells for the solver. We set the numerical solver’s CFL (Courant-Friedrichs-Levy) [Courant et al., 1928] condition to 0.5 and then uniformly subsample its solution, obtaining $n_t = 300$ points in the interval $[40, 80]$. We then coarsen the data (velocities) to a 512×512 grid using the face-averaging approach to maintain incompressibility and momentum conservation. This 512×512 data then comprises the high-resolution inputs X_t' for HREs and other methods operating at on inputs at 512×512 resolution for our emulation tasks, or are coarsened further as needed by other emulators. These spatial and temporal coarsening steps result in a CFL number of 9.4 for the training dataset, far exceeding that of the solver.

C.2 2D Kolmogorov Flow with $\text{Re} = 4000$ (KFHR)

The Reynolds number of KF scales as $\text{Re} \propto \sqrt{\chi} L^{2/3} / \nu$. χ is the forcing strength scale, L is the domain size and ν is the kinematic viscosity. Following [Kochkov et al., 2021] we scale $L \rightarrow 2L$, $\nu \rightarrow \nu/2$ and $\chi \rightarrow \chi/2$ to increase the Reynolds number to 4000. Since the smallest eddy size scales with $1/\sqrt{\text{Re}}$, we get smaller eddy structures in the solution. We use a 4096×4096 grid for the solver and downsample the solution to a 512×512 grid. All other parameters match KF.

C.3 2D Incompressible Decaying Turbulence (ID)

This dataset, like the Kolmogorov flow, is governed by the incompressible Navier-Stokes equations and solved with the same jax-cfd package. We generate 200 experiments using the same parameters and spatial resolutions as KF except for the forcing term, which is set to $\mathbf{f} = \mathbf{0}$. Time is uniformly discretized to $n_t = 166$ points in $[4.5, 25]$.

C.4 Compressible Decaying Turbulence (CD)

We generate the dataset using direct numerical simulation (DNS) software provided by [Takamoto et al., 2023]. Time integration uses a mass, momentum and energy-conserving finite volume scheme, whose implementation we slightly modified to avoid rounding errors when converting between floating point representations of momentum and velocity. Our dataset consists of 200 different simulations having varying initial conditions of density, momentum and energy. Space is uniformly discretized to $n_x = n_y = 512$ cells in $[0, 1] \times [0, 1]$ with periodic boundary conditions and time is uniformly discretized to $n_t = 100$ points in $[0, 5]$.

894 C.5 1D Kuramoto-Sivashinsky Equation

895 We generate the dataset using the `jax-cfd` package. The spatial domain (with length $L = 256$) is
 896 discretized into 2048 points, while the temporal domain $t \in [0, 150]$ is discretized into $n_t = 3000$
 897 time steps. Initial conditions are specified as $u_0(x) = \sum_{m=1}^5 A_m \sin(2\pi\ell_m x/L + \phi_m)$, where
 898 $\{A_m, \ell_m, \phi_m\}$ are random parameters sampled as in [Lippe et al., 2023]. Boundary conditions are
 899 specified as periodic. We use a spectral solver with an RK4 time integrator. To ensure statistical
 900 stationarity, we discard solutions up to $t = 80$ as a burn-in period. We generate 1000 trajectories and
 901 downsample the solutions by a factor of 5 along the temporal axis. Rollout resolution is chosen as
 902 32^2 meaning that $r = 64$.

903 D Supplementary experimental results

904 **KS and ID with multiple seeds** To test the robustness of our HRE method to different initial seeds,
 905 we initialized the network used in the KS case on a coarse version of the dataset using 5 different
 906 random seeds and trained it on a next-step prediction task. We observed that the model’s performance
 907 was consistent over seeds, with mean validation MSE of $5e - 8$ and standard deviation of $1e - 8$. We
 908 also repeated the ID experiment on a rollout resolution of 32^2 with an additional seed, resulting in
 909 similar accuracy. The additional experiment is conducted with seed 44 and resulted in a validation
 910 rollout loss of 0.29. In comparison, the experiment we report was run with the seed 43 and achieved
 911 a rollout error of 0.31.

912 **TKE spectra** We report the TKE spectra of PDE fields predicted by the trained models at a rollout
 913 resolution of 32^2 in Figures 18, 19. We observe that the high-resolution information encoded in c_t
 914 fields enables our model to capture high-frequency features in the flow field. On the KF task, our
 915 model consistently outperforms others in reproducing the TKE spectrum of the reference solver.
 916 However, it occasionally produces unwanted oscillations at high frequencies for the ID case.

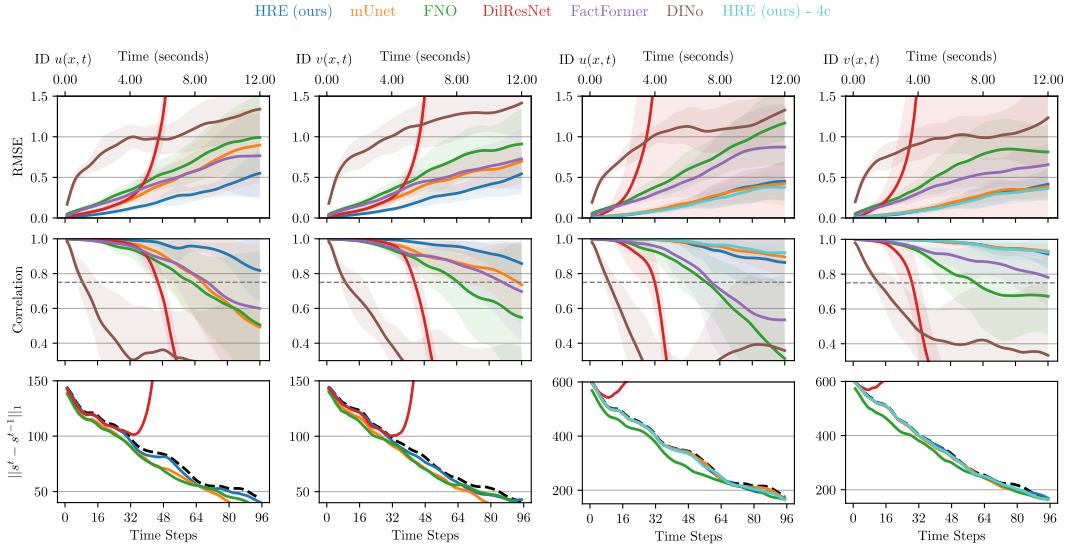


Figure 8: Rollout performance 32^2 (left), 64^2 (right) ID.

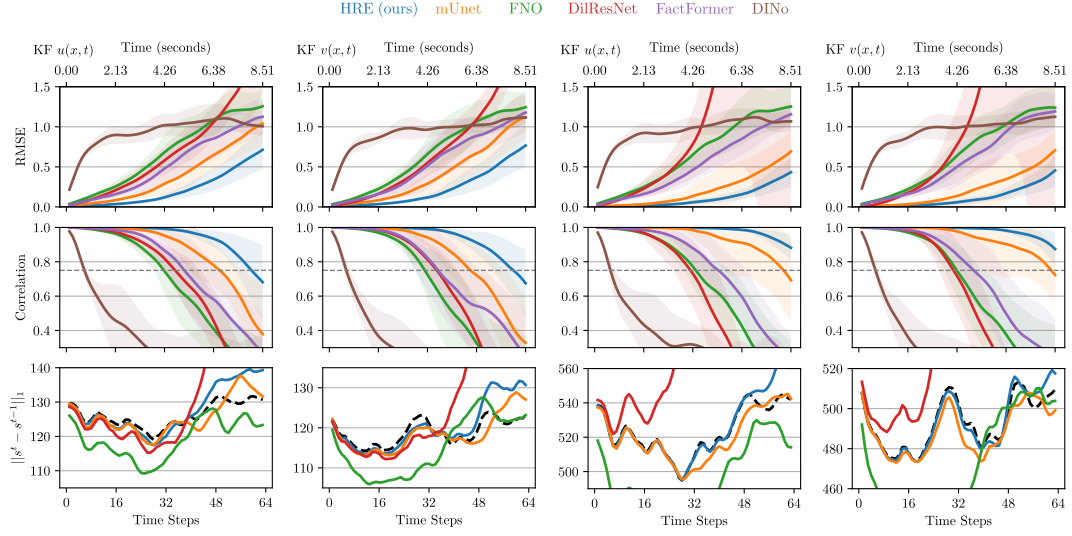


Figure 9: Rollout performance on the KF task at rollout/encoding resolutions 32^2 (left) and 64^2 .

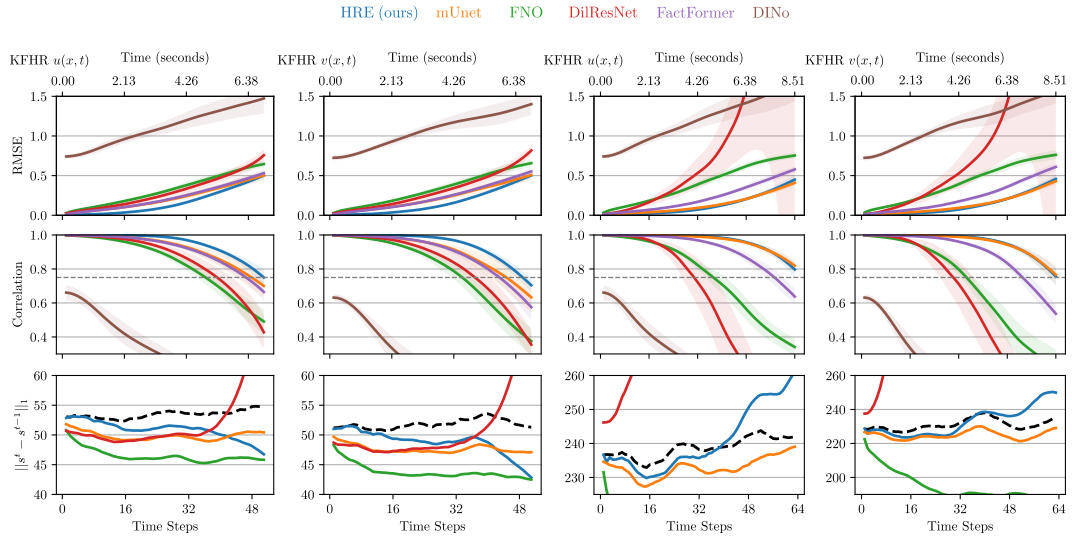


Figure 10: Rollout performance on the KFHR task at rollout/encoding resolutions 32^2 (left) and 64^2 .

Table 1: Training details.

	HRE (ours)		MUNET		FNO		DILRESNET		FACTFORMER		DINO	
	KF/KFHR/ID/CD		KF/KFHR/ID/CD		KF/KFHR/ID/CD		KF/KFHR/ID/CD		KF/ID		KF/ID	
BATCH SIZE (PER GPU)	8		32		32		32		32		32	
INITIAL LEARNING RATE	$1e-4$		$1e-4$		$1e-4$		$1e-4$		$5e-4$		$1e-2$	
OPTIMIZER	ADAMW		ADAMW		ADAMW		ADAMW		ADAMW		ADAM	
WEIGHT DECAY	0.01		0.01		0.01		0.01		$1e-4$		-	
LR SCHEDULER	COSINE $\rightarrow 1e-6$		COSINE $\rightarrow 1e-6$		COSINE $\rightarrow 1e-6$		COSINEWR $\rightarrow 1e-6$		ONECYCLE		-	
RESOURCES	8 A100 GPUS		4 A100 GPUS		4 A100 GPUS		4 A100 GPUS		1 H100 GPU		1 V100 GPU	
CURRICULUM	[1, 2, 4, 8, 16]		[1, 2, 4, 8, 16]		[1, 2, 4, 8, 16]		-		PUSHFORWARD [1, 2, 3, 4]		ESS [16]	
TRAINING NOISE	-		-		-		$\mu = 0, \sigma = 10^{-4}$		-		-	
TRAINING EPOCHS	100×5		100×5		100×5		300		50		60000	
PADDING	CIRCULAR		CIRCULAR		CIRCULAR		CIRCULAR		-		-	
LOSS	MSE		MSE		MSE		MSE		RELATIVE MSE		MSE	
ACTIVATION	SiLU		SiLU		SiLU		SiLU		GELU		SiLU	
EARLY STOPPING (25 Ep.)	YES		YES		YES		YES		No		YES(20000 Ep.)	
SEED	42		42		42		42		1234		1	
DEC./SR DEC. EPOCHS	100		100		-		-		-		-	

Table 2: Number of parameters per model.

	KF/ID/KFHR	CD
OUR PROCESSOR	19.492M	19.493M
MUNET	19.430M	19.433M
FNO	18.974M	18.974M
DILRESNET	583680	588096
FACTFORMER	3464834	-
DINO	769420	-
OUR ENCODER (64^2)	2.156M	2.173M
OUR ENCODER (32^2)	8.780M	8.798M
OUR DECODER (64^2)	2.204M	2.226M
OUR DECODER (32^2)	8.833M	8.860M
SR DECODER (64^2)	2.153M	2.158M
SR DECODER (32^2)	8.743M	8.753M

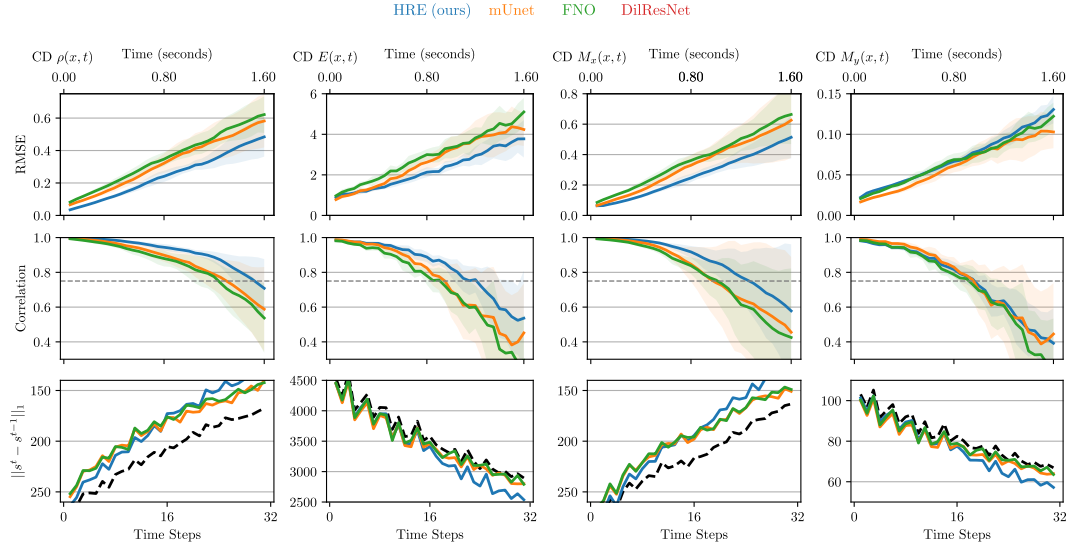


Figure 11: Rollout performance for 32^2 CD.

Table 3: Dataset details.

	COMP. DECAYING	KOLMOGOROV	KOLMOGOROV (RE=4000)	INCOMP. DECAYING
# SPATIAL DIMENSIONS	2	2	2	2
# FEATURES	4	2	2	2
FEATURES	ρ, M_x, M_y, E	v_x, v_y	v_x, v_y	v_x, v_y
PARAMETERS	$(\zeta, \eta, M) = (10^{-8}, 10^{-8}, 0.4)$	$(\rho, \text{Re}, \nu) = (1, 10^3, 10^{-3})$	$(\rho, \text{Re}, \nu) = (1, 4 \times 10^3, 5 \times 10^{-4})$	$Re_{\text{init}} = 1000$
FORCING	0	$\sin(4y)\hat{\mathbf{e}}_1 - 0.1\mathbf{u}$	$0.5 \sin(4y)\hat{\mathbf{e}}_1 - 0.1\mathbf{u}$	0
DOMAIN SIZE	$[0, 1] \times [0, 1]$	$[0, 2\pi] \times [0, 2\pi]$	$[0, 4\pi] \times [0, 4\pi]$	$[0, 2\pi] \times [0, 2\pi]$
SOLVER GRID SIZE	1/512	$2\pi/2048$	$2\pi/4096$	$2\pi/2048$
DATASET GRID SIZE	1/512	$2\pi/512$	$2\pi/512$	$2\pi/512$
SPATIAL COARSENING	8, 16	1, 8, 16	8, 16	1, 8, 16
SOLVER TIMESTEP	$\approx 1.1 \times 10^{-4}$	2.191×10^{-4}	2.191×10^{-4}	3.652×10^{-4}
EMULATOR TIMESTEP	0.05	0.133	0.133	0.123
TEMPORAL COARSENING	$\approx 450 \times$	$\approx 600 \times$	$\approx 600 \times$	$\approx 340 \times$
SOLVER CFL	0.3	0.5	0.5	0.5
COARSE DATASET CFL	28.125/56.25	9.375/18.75	4.687/9.375	6.25/12.5
BURN-IN TIME	1	40	40	4.5
TERMINAL TIME	5	80	80	25
TRAJECTORY LENGTH	100	300	300	166
# TRAJECTORIES TRAIN	200	128	128	200
VALIDATION FRACTION	0.1	0.1	0.1	0.1
# TRAJECTORIES TEST	20	16	16	16

Table 4: Rollout performance across emulators and experiments. We report MSE and correlation (Corr.) of field variable u for each method after 64 rollout steps (52 steps for KFHR at 32^2). Best model performance in **bold** and second underlined.

	KF		ID		KFHR	
	RMSE \downarrow	CORR. \uparrow	RMSE	CORR.	RMSE	CORR.
ROLLOUT RESOLUTION 32×32						
OURS	$0.71_{\pm.23}$	$0.68_{\pm.21}$	$0.29_{\pm.08}$	$0.95_{\pm.03}$	$0.50_{\pm.04}$	$0.74_{\pm.05}$
MUNET	$1.02_{\pm.24}$	<u>$0.39_{\pm.25}$</u>	<u>$0.58_{\pm.34}$</u>	$0.78_{\pm.25}$	<u>$0.52_{\pm.05}$</u>	<u>$0.69_{\pm.05}$</u>
FNO	$1.25_{\pm.20}$	$0.16_{\pm.18}$	$1.32_{\pm.21}$	$0.14_{\pm.23}$	$0.66_{\pm.05}$	$0.48_{\pm.08}$
DILRESNET	$2.89_{\pm 1.47}$	$0.05_{\pm.10}$	$26.83_{\pm 9.98}$	$0.09_{\pm.05}$	$0.75_{\pm.07}$	$0.43_{\pm.10}$
FACTFORMER	$1.12_{\pm.21}$	$0.28_{\pm.02}$	<u>$0.58_{\pm.25}$</u>	<u>$0.79_{\pm.51}$</u>	$0.25_{\pm.05}$	$0.66_{\pm.05}$
DINO	<u>$1.00_{\pm.1}$</u>	$0.20_{\pm.17}$	$1.10_{\pm.27}$	$0.30_{\pm.30}$	$0.99_{\pm.04}$	$0.21_{\pm.04}$
ROLLOUT RESOLUTION 64×64						
OURS	$0.43_{\pm.17}$	$0.88_{\pm.07}$	$0.28_{\pm.15}$	<u>$0.93_{\pm.12}$</u>	<u>$0.44_{\pm.05}$</u>	$0.79_{\pm.04}$
MUNET	<u>$0.69_{\pm.25}$</u>	<u>$0.69_{\pm.24}$</u>	$0.28_{\pm.13}$	$0.95_{\pm.04}$	$0.41_{\pm.04}$	<u>$0.81_{\pm.03}$</u>
FNO	$1.25_{\pm.22}$	$0.17_{\pm.16}$	$0.78_{\pm.26}$	$0.67_{\pm.20}$	$0.76_{\pm.05}$	$0.35_{\pm.07}$
DILRESNET	$144.6_{\pm 157.2}$	$0.01_{\pm.03}$	$1710_{\pm 4513.8}$	$-0.01_{\pm.08}$	$2.12_{\pm 1.20}$	$0.16_{\pm.14}$
FACTFORMER	$1.15_{\pm.15}$	$0.23_{\pm.01}$	<u>$0.66_{\pm.36}$</u>	$0.71_{\pm 0.3}$	$0.41_{\pm.04}$	$0.82_{\pm.04}$
DINO	$1.07_{\pm.12}$	$0.15_{\pm.12}$	$1.09_{\pm.28}$	$0.30_{\pm.33}$	$1.47_{\pm.19}$	$0.13_{\pm.06}$

Table 5: Rollout performance across different experiments for the field variable v for each method. Model parameters are the same as Table 4.

METHOD	KF		ID		KFHR	
	RMSE \downarrow	CORR. \uparrow	RMSE	CORR.	RMSE	CORR.
ROLLOUT RESOLUTION 32×32						
OURS	$0.76_{\pm.27}$	$0.67_{\pm.20}$	$0.30_{\pm.11}$	$0.96_{\pm.03}$	$0.50_{\pm.04}$	$0.70_{\pm.04}$
MUNET	<u>$1.10_{\pm.27}$</u>	<u>$0.36_{\pm.25}$</u>	<u>$0.51_{\pm.21}$</u>	<u>$0.86_{\pm.12}$</u>	<u>$0.53_{\pm.05}$</u>	<u>$0.62_{\pm.07}$</u>
FNO	$1.24_{\pm.17}$	$0.18_{\pm.17}$	$1.26_{\pm.32}$	$0.26_{\pm.29}$	$0.68_{\pm.05}$	$0.37_{\pm.08}$
DILRESNET	$2.99_{\pm.71}$	$0.04_{\pm.08}$	$20.54_{\pm 5.30}$	$0.02_{\pm.06}$	$0.81_{\pm.06}$	$0.37_{\pm.08}$
FACTFORMER	$1.17_{\pm.23}$	$0.25_{\pm.03}$	$0.54_{\pm.23}$	$0.85_{\pm.13}$	$0.55_{\pm.05}$	$0.58_{\pm.08}$
DINO	$1.11_{\pm.1}$	$0.00_{\pm.12}$	$1.27_{\pm.31}$	$0.05_{\pm.36}$	$0.98_{\pm.02}$	$0.06_{\pm.08}$
ROLLOUT RESOLUTION 64×64						
OURS	$0.45_{\pm.19}$	$0.87_{\pm.10}$	$0.27_{\pm.14}$	$0.97_{\pm.03}$	<u>$0.46_{\pm.06}$</u>	<u>$0.76_{\pm.05}$</u>
MUNET	<u>$0.70_{\pm.24}$</u>	<u>$0.72_{\pm.16}$</u>	<u>$0.29_{\pm.10}$</u>	$0.97_{\pm.02}$	$0.43_{\pm.43}$	$0.77_{\pm.05}$
FNO	$1.23_{\pm.24}$	$0.18_{\pm.25}$	$0.79_{\pm.38}$	$0.72_{\pm.23}$	$0.79_{\pm.05}$	$0.21_{\pm.10}$
DILRESNET	$94.24_{\pm 103.27}$	$0.00_{\pm.09}$	$875.12_{\pm 2223.6}$	$-0.02_{\pm.20}$	$10.04_{\pm 10.46}$	$0.02_{\pm.04}$
FACTFORMER	$1.19_{\pm 0.2}$	$0.23_{\pm.02}$	$0.54_{\pm.20}$	<u>$0.87_{\pm.08}$</u>	$0.43_{\pm.04}$	$0.77_{\pm 0.04}$
DINO	$1.13_{\pm.12}$	$0.04_{\pm.16}$	$0.99_{\pm.27}$	$0.42_{\pm.30}$	$1.40_{\pm.14}$	$0.04_{\pm.05}$

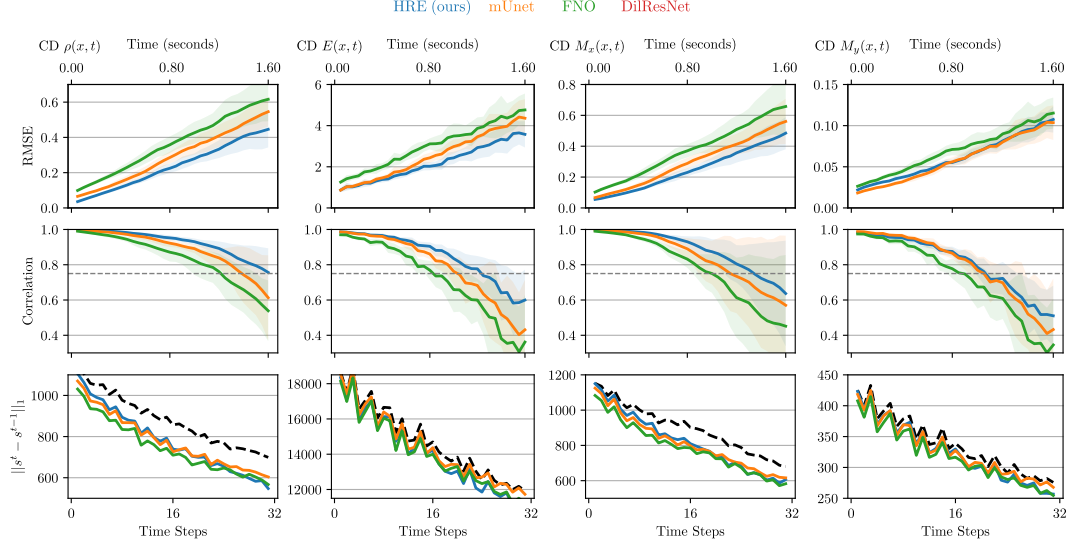


Figure 12: Rollout performance for 64^2 CD.

Table 6: Rollout performance across different numbers of learned latent fields n and coarsening factors r . We report MSE and correlation (Corr.) of field variable u for each method. The rollout length is 96 steps for KF and 128 for ID. Values in **bold** show the best result at each rollout resolution.

	KF		ID	
	RMSE ↓	CORR. ↑	RMSE	CORR.
ROLLOUT RESOLUTION 32×32				
$n = 4$	1.263 ± 0.23	0.143 ± 0.157	0.6 ± 0.319	0.766 ± 0.332
$n = 16$	1.295 ± 0.188	0.147 ± 0.118	0.192 ± 0.227	0.658 ± 0.33
ROLLOUT RESOLUTION 64×64				
$n = 4$	1.23 ± 0.21	0.214 ± 0.215	1.17 ± 0.26	0.868 ± 0.192
$n = 16$	0.69 ± 0.25	0.69 ± 0.24	0.28 ± 0.126	0.95 ± 0.039

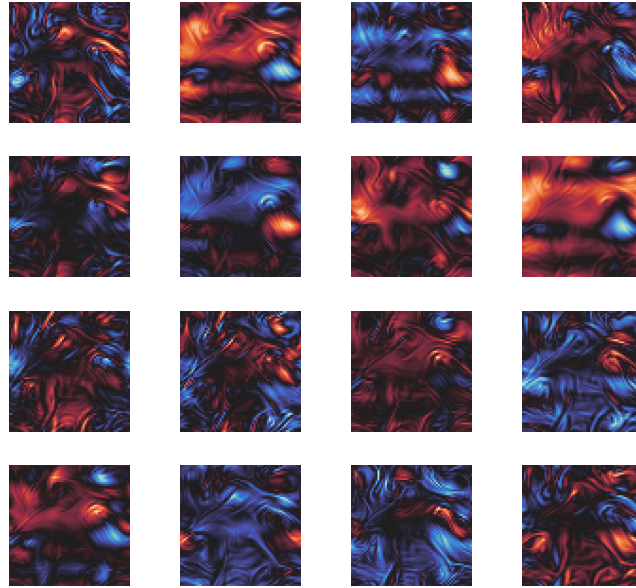


Figure 13: A snapshot of extracted c_t fields for KF.

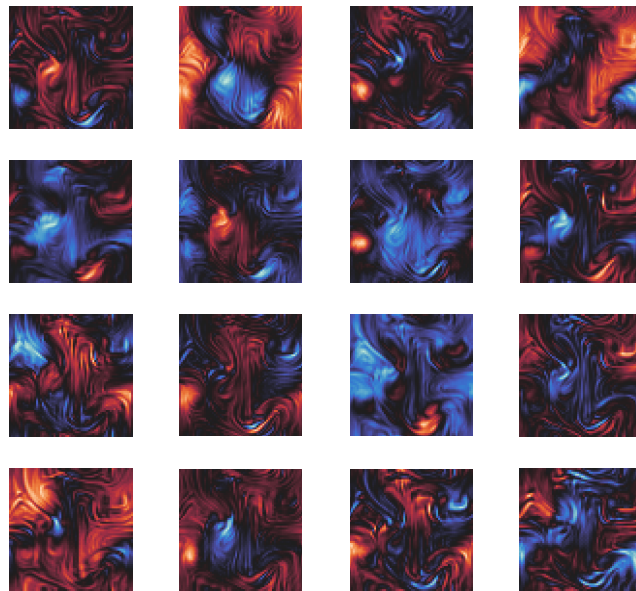


Figure 14: A snapshot of extracted c_t fields for ID.

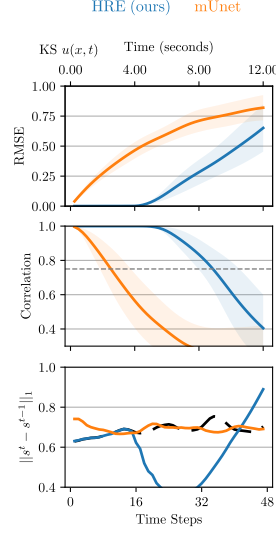


Figure 15: Comparison of mUnet emulators and HREs on the KS task. The reference resolution was 2048, and we used $r = 64$ for a rollout resolution of 32. We used $r' = 1$, meaning our HREs encoded full-resolution PDE fields from the reference simulation.

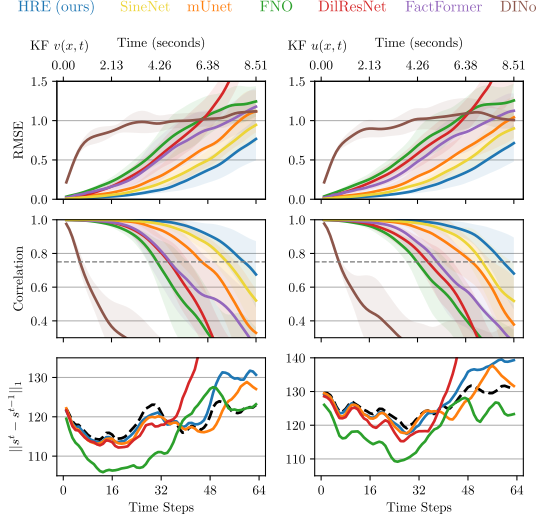


Figure 16: Comparison of SineNet, mUnet and HRE emulators on on KF (32^2). We use the SineNet-8 model with the same training parameters as for their INS task [Zhang et al., 2024]. We excluded this baseline from further analysis due to its unfavorable cost-accuracy trade-off. While it provides a slight improvement in accuracy, it is several orders of magnitude slower than mUnet.

Table 7: Physics and c_t field ablations. We measure the rollout performance of the models on KF (32^2) for 64 timesteps and compare the final RMSE and correlation values.

	KF	
	RMSE	CORR.
OUR MODEL	0.71 ± 0.23	0.68 ± 0.21
MUNET W PHYSICS	0.956 ± 0.206	0.475 ± 0.212
OURS W/O PHYSICS	0.881 ± 0.21	0.563 ± 0.215
MUNET	1.1 ± 0.271	0.355 ± 0.254

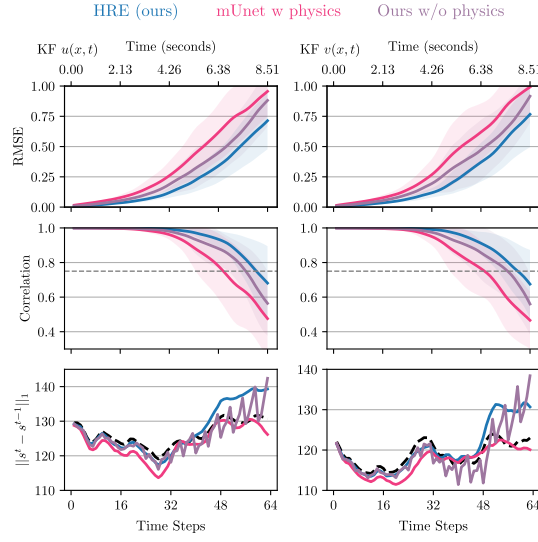


Figure 17: Performance of ablated models on KF 32^2 .

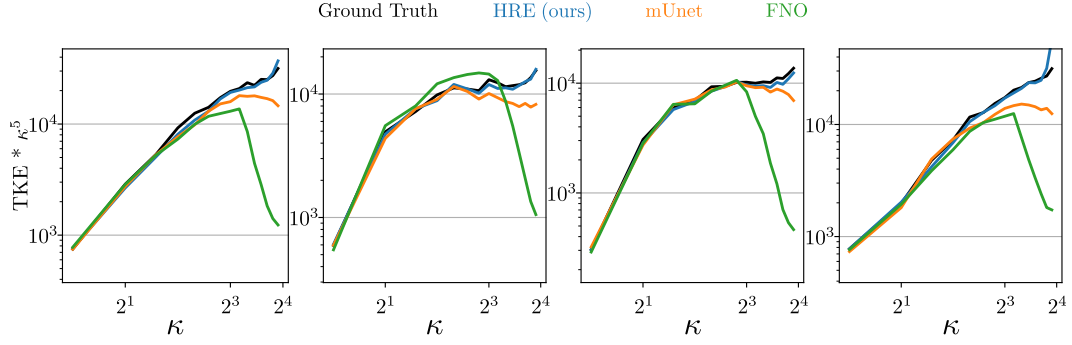


Figure 18: Additional TKE analysis for ID rollouts of 96 (Resolution 32^2).

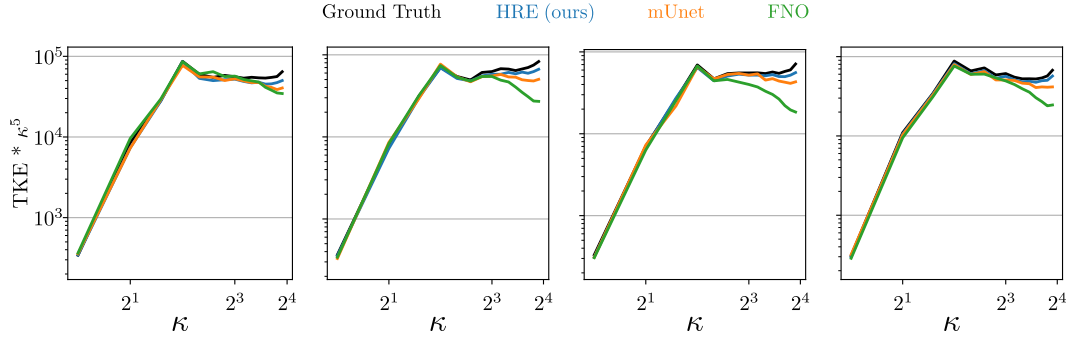


Figure 19: Additional TKE analysis for KF rollouts of 64 (Resolution 32^2).

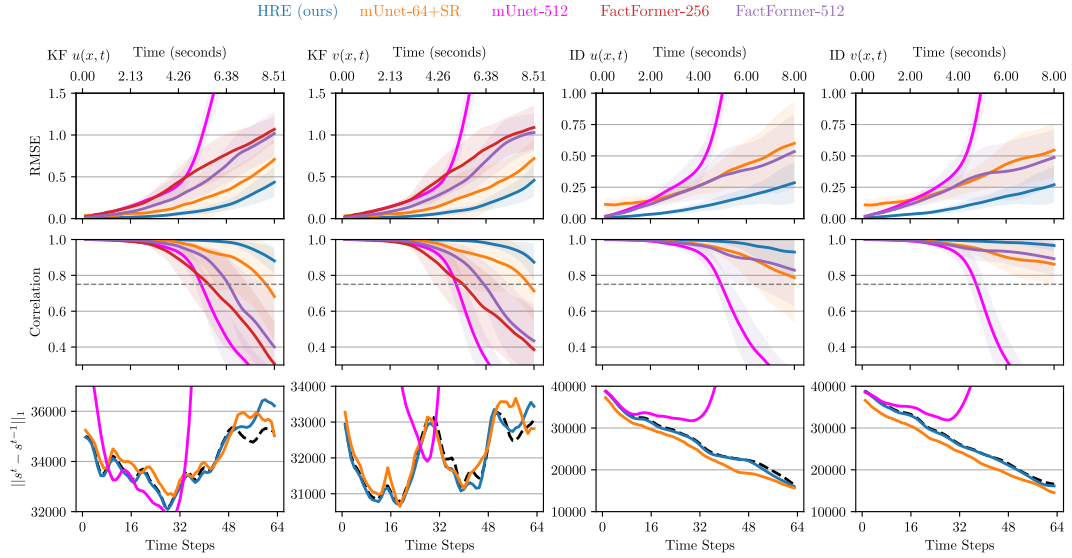


Figure 20: Comparison of our HREs with mUnet and FactoFormer operating on high-resolution data and mUnet with a super-resolution decoder for the KF (left) and ID (right) tasks, rollout resolution 64^2 .

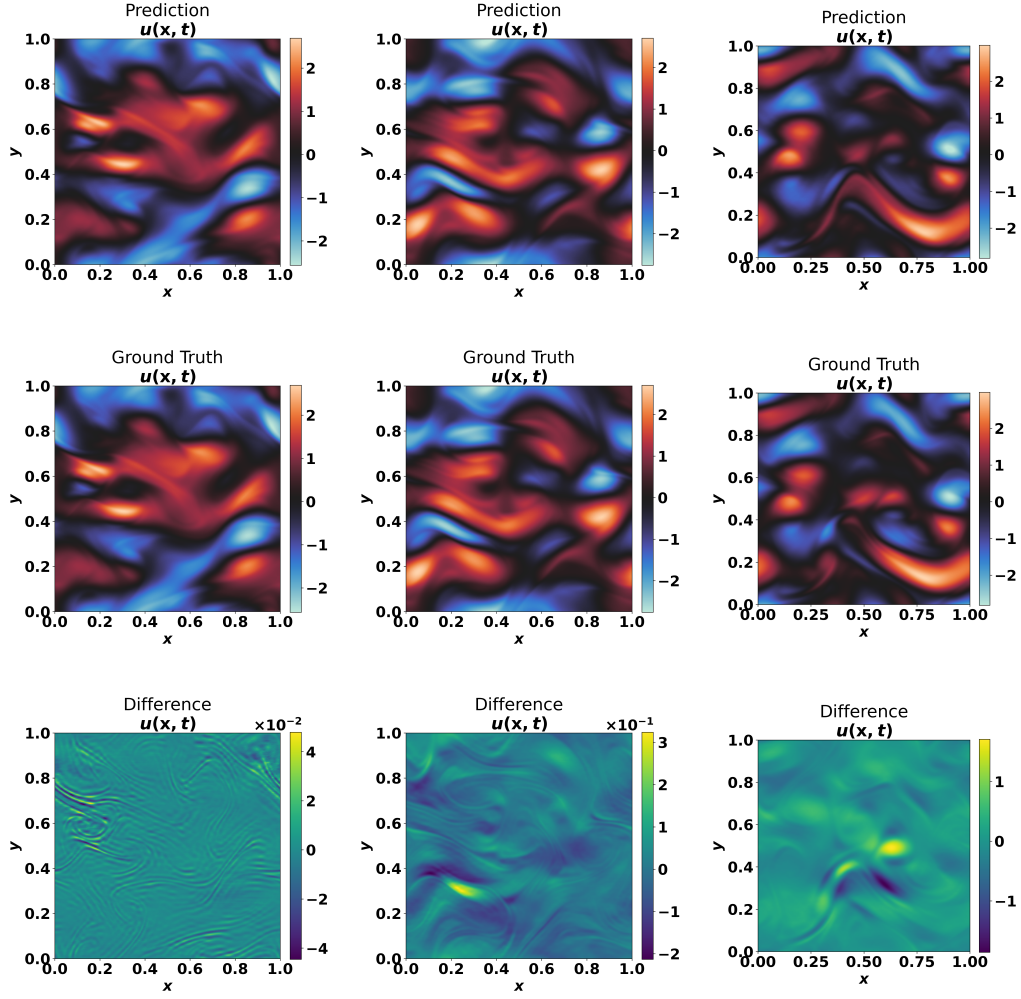


Figure 21: Our High-resolution predictions of KF with rollout resolution 64^2 and target resolution 512^2 with 32 timestep intervals.

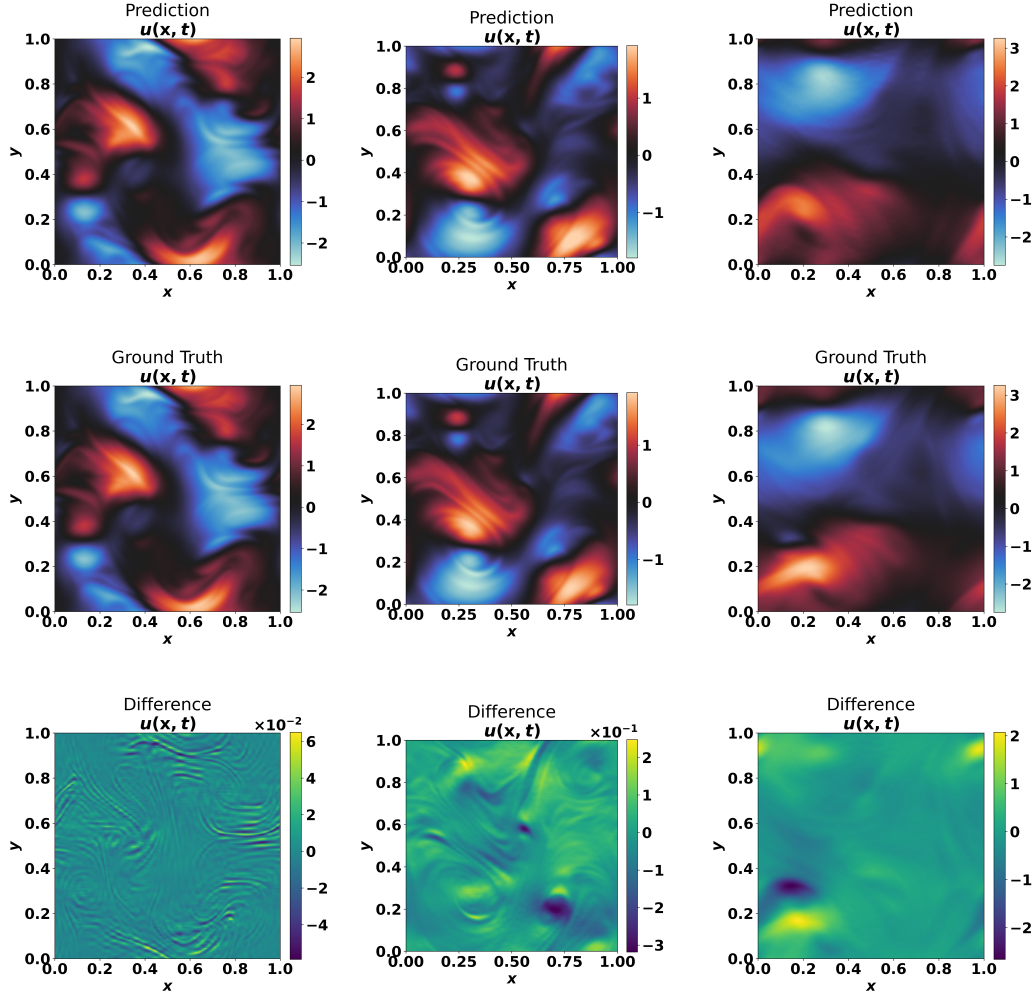


Figure 22: Our High-resolution predictions of ID with rollout resolution 64^2 and target resolution 512^2 with 32 timestep intervals.

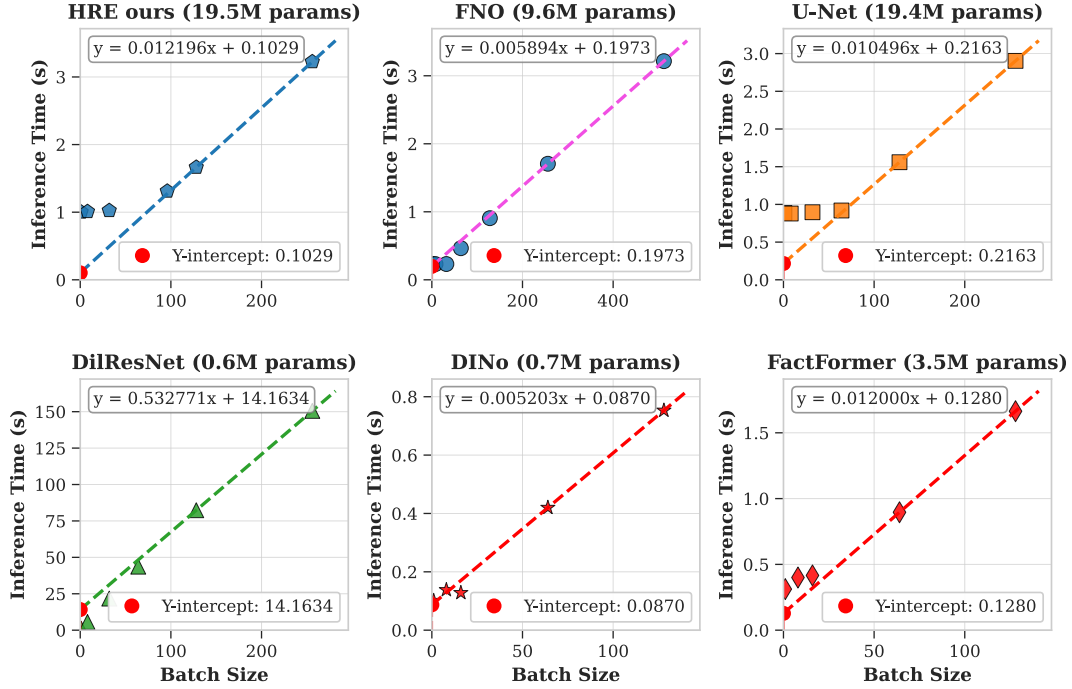


Figure 23: Inference times for varying batch sizes. We choose rollout resolution (32^2) for ID, KF and KFHR tasks and computed rollouts over 64 time steps.

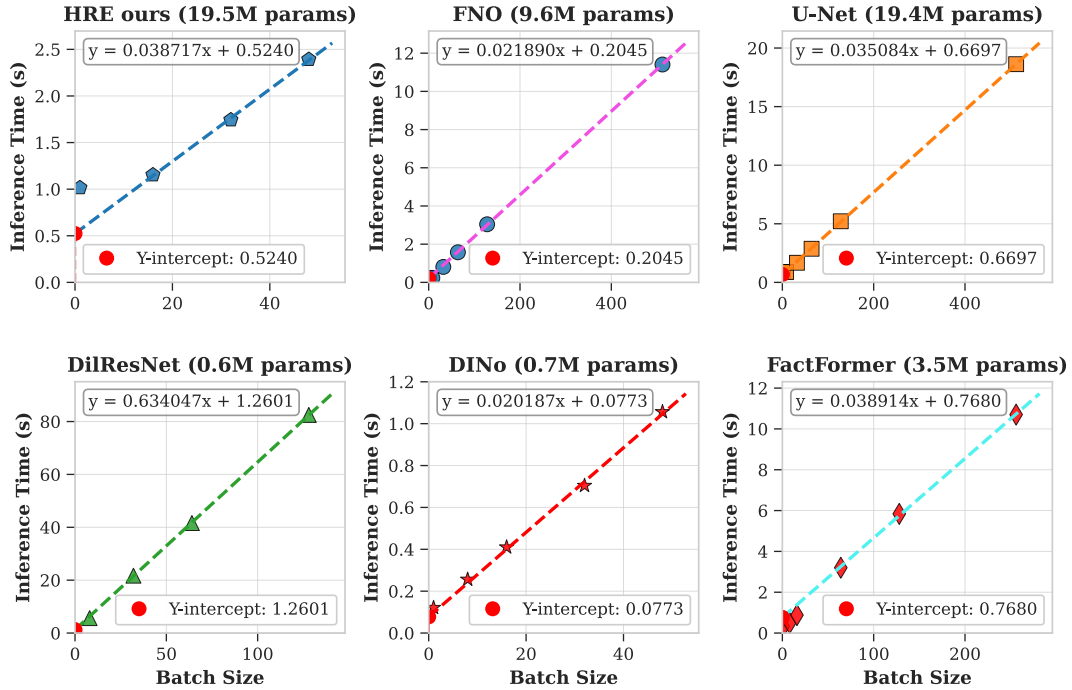


Figure 24: Inference times for varying batch sizes. We choose rollout resolution (64^2) for ID, KF and KFHR tasks and computed rollout over 64 time steps.