DRIFT-NET: A SPECTRAL–COUPLED NEURAL OPERATOR FOR PDES LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning PDE dynamics with neural solvers can significantly improve wall-clock efficiency and accuracy compared with classical numerical solvers. In recent years, foundation models for PDEs have largely adopted multi-scale windowed self-attention, with the scOT backbone in POSEIDON serving as a representative example. However, because of their locality, truly globally consistent spectral coupling can only be propagated gradually through deep stacking and window shifting. This weakens global coupling and leads to error accumulation and drift during closed-loop rollouts. To address this, we propose **DRIFT-Net**. It employs a dual-branch design comprising a spectral branch and an image branch. The spectral branch is responsible for capturing global, large-scale low-frequency information, whereas the image branch focuses on local details and nonstationary structures. Specifically, we first perform controlled, lightweight mixing within the low-frequency range. Then we fuse the spectral and image paths at each layer via bandwise weighting, which avoids the width inflation and training instability caused by naive concatenation. The fused result is transformed back into the spatial domain and added to the image branch, thereby preserving both global structure and high-frequency details across scales. Compared with strong attention-based baselines, DRIFT-Net achieves lower error and higher throughput with fewer parameters under identical training settings and budget. On Navier–Stokes benchmarks, the relative L_1 error is reduced by 7%–54%, the parameter count decreases by about 15%, and the throughput remains higher than scOT. Ablation studies and theoretical analyses further demonstrate the stability and effectiveness of this design. The code is available at `https://anonymous.4open.science/r/DRIFT-Net-26D6`.

1 INTRODUCTION

Partial differential equations (PDEs) underpin science and engineering. Repeated high-accuracy simulations remain costly at scale (Trefethen, 2000; Benner et al., 2015). Neural operators address this challenge by learning mappings directly between function spaces. This enables fast inference across resolutions and inputs and supports cross-mesh generalization (Kovachki et al., 2023). Representative models include the Fourier Neural Operator (FNO) (Li et al., 2021) and DeepONet (Lu et al., 2021). Building on these advances, PDE foundation models adopt multi-scale windowed self-attention. POSEIDON with its SCOT backbone is a representative example.

Nevertheless, windowed self-attention is local. Global dependencies emerge only gradually with depth and shifted windows. This weakens *globally consistent spectral coupling* and can induce error accumulation and drift during closed-loop autoregressive rollouts (Lippe et al., 2023). In practice, naive cross-scale or cross-branch concatenation inflates channel width and destabilizes training. Purely spectral operators are global but often overemphasize low-frequency structure and underfit nonstationary local details.

Our approach. We introduce DRIFT-NET, a dual-branch neural operator. The spectral branch performs controlled low-frequency global mixing. The image branch handles local interactions. The two branches are fused bandwise through a non-expansive mechanism. This avoids width inflation and preserves high-frequency detail. Implementation details, including the low-frequency mixing and bandwise fusion, are presented in §4.

Contributions.

- **Modular operator unit.** DRIFT-NET provides a dual-branch unit with controlled low-frequency mixing and bandwise non-expansive fusion. It enhances global coupling, local-detail fidelity, and training stability. The unit can be swapped in for windowed self-attention blocks in multi-scale operator backbones.
- **Performance and efficiency.** Under matched training schedules and hardware, DRIFT-NET reduces final-time relative L_1 error by 7% to 54% on Navier–Stokes benchmarks. It uses about 15% fewer parameters and achieves higher throughput than SCOT. See §5.
- **Mechanism and reusability.** Ablations and spectral analyses show how non-expansive fusion and controlled low-frequency mixing support stable training and improved generalization. The design is reusable and modular for stronger PDE foundation models.

2 RELATED WORK

Neural operators. Neural operators learn PDE solution mappings directly between function spaces. The Fourier Neural Operator (FNO) introduced spectral-domain convolutions to efficiently capture global interactions (Li et al., 2021), while DeepONet adopts a branch and trunk factorization grounded in operator approximation theory (Lu et al., 2021). Building on these paradigms, several variants have broadened the applicability of neural operators. AFNO leverages low-rank factorization for high-resolution efficiency (Guibas et al., 2022). U-NO deepens the U-Net style encoder–decoder hierarchy for multi-scale representations (Rahman et al., 2022). Geo-FNO extends spectral operators to irregular geometries (Li et al., 2023), and CNO targets robust multi-scale approximation in challenging settings (Raonić et al., 2023). Beyond single-task settings, recent work explores foundation models for scientific ML to improve scaling and transfer (Subramanian et al., 2023).

Foundation models with multi-scale attention. Within PDE foundation models, POSEIDON employs a multi-scale windowed self-attention operator Transformer (SCOT) with a U-Net hierarchy and time-conditioned normalization. It demonstrates strong transfer performance across heterogeneous equation families (Herde et al., 2024; Liu et al., 2021; 2022). Windowed self-attention partitions the domain into local regions to compute attention efficiently, but its locality implies that domain-wide dependencies are established only gradually via deep stacking and window shifts. Empirically, this can weaken globally consistent spectral coupling and induce rollout drift over long horizons (Lippe et al., 2023). In practice, naive cross-scale or cross-branch concatenation may inflate channel width and destabilize training, whereas purely spectral operators, while global, often underfit nonstationary local details.

Positioning. DRIFT-Net augments a multi-scale attention backbone with an explicit spectral path for controlled low-frequency mixing. It also introduces a non-expansive, bandwise fusion mechanism that integrates the spectral and image branches without inflating the feature width. Methodological details and quantitative comparisons appear in Sections 4 and 5.

3 PROBLEM SETUP

Problem formulation. We consider a generic time-dependent partial differential equation on a spatial domain $D \subset \mathbb{R}^d$ and time horizon $T > 0$:

$$\begin{aligned} \partial_t u(x,t) + \mathcal{L}\big(u, \nabla_x u, \nabla_x^2 u, \dots\big) = 0, \qquad & \forall\, x \in D \subset \mathbb{R}^d,\ t \in (0,T), \\ \mathcal{B}(u) = 0, \qquad & \forall\, (x,t) \in \partial D \times (0,T), \\ u(x,0) = a(x), \quad & \forall\, x \in D. \end{aligned} \tag{3.1}$$

Here L and B denote the differential and boundary operators, and $a(x)$ is the initial datum. Time-independent problems are also covered by this formulation. If a steady state exists, the long-time limit ($t \to \infty$) yields the steady PDE

$$L\big(u(x), \nabla_x u(x), \nabla_x^2 u(x), \dots\big) \;=\; 0, \qquad B(u) = 0, \qquad x \in D, \tag{3.2}$$

which is the time-independent counterpart of equation 3.1.

Solution operator. Let X denote the state space, for example a suitable function space on D. The solution can be described by a map $S : [0, T] \times X \to X$ such that, for any $t \in [0, T]$ and $a \in X$, $u(t) = S(t, a)$. Equivalently, for each fixed t we define the flow map $S_t : X \to X$ with $S_t(a) = S(t, a)$.

Underlying operator learning task. Given a distribution μ over initial conditions $a \in X$, the goal is to learn an approximate solution operator $S^\star$ that closely reproduces the true operator S. For any $a \sim \mu$, the learned operator should generate the trajectory $\{S^\star(t, a)\}_{t\in[0,T]}$ that approximates $\{S(t, a)\}_{t\in[0,T]}$ for all t. The model is expected to produce the time evolution from a, given boundary conditions, without relying on intermediate information, analogous to a classical solver.

Learning objective. On a discrete grid of size $H \times W$ with C components, let $u_t \in \mathbb{R}^{C\times H\times W}$ denote the state at discrete time t. We learn a one-step operator $F_\theta : u_t \mapsto u_{t+1}$ with teacher forcing and a relative L_p objective with $p \in \{1, 2\}$. At test time, F_θ is composed autoregressively to produce a full trajectory $\{\hat{u}_t\}_{t=1}^T$. Details of the training schedule, data splits, rollout horizon, and evaluation metrics are specified in Sec. 5.

4 METHOD

Existing neural operators suffer from weak global coupling and the loss of high-frequency detail in long-horizon prediction (Lippe et al., 2023). We propose DRIFT-NET, a U-Net–style encoder–decoder (Ronneberger et al., 2015) with two parallel branches: a frequency path for cross-scale global interaction and an image path for local, nonstationary structures (Wen et al., 2022). At each scale, we fuse the branches by inverse-transforming the frequency output to the spatial domain and adding it to the image output. The model hinges on three mechanisms: (1) controlled low-frequency mixing to strengthen long-range dependencies without disturbing high-frequency modes; (2) bandwise fusion with radial gating for smooth cross-band transitions (Rahaman et al., 2019; Xu et al., 2020); and (3) a frequency-weighted loss to counter spectral bias. We first outline the architecture and then detail each component.

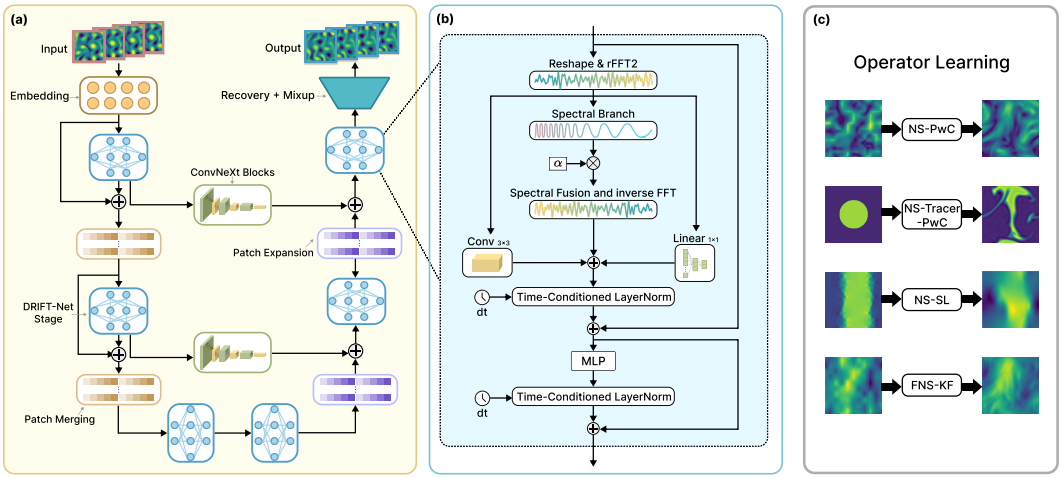


Figure 1: **Overall architecture of DRIFT-NET.** (a) U-Net style encoder–decoder with ConvNeXt blocks and patch merging/expansion. (b) Spectral branch (rFFT2, spectral fusion with radial gating) fused additively with image branch via inverse FFT. (c) Example operator-learning tasks: NS-PwC, NS-Tracer-PwC, NS-SL, FNS-KF.

4.1 ARCHITECTURE OVERVIEW

DRIFT-NET follows a hierarchical encoder–decoder and augments each scale with an explicit spectral path (Fig. 1). In the *image branch*, ConvNeXt-style blocks extract local and nonstationary structures; down/up-sampling is realized by patch merging/expansion to form a U-shaped hierarchy. In

parallel, the *spectral branch* converts features at the same scale to the Fourier domain via rFFT2, applies a controlled transformation only on low-frequency modes (Sec. 4.2), and fuses them back by a smooth bandwise mechanism (Sec. 4.3). After iFFT2, the spectral output is *added* to the image branch feature at that scale.

Two design choices are key. First, the spectral path provides *immediate* global receptive fields at every resolution, so globally consistent coupling does not rely on deep stacking or large kernels. Second, the additive cross-branch fusion is *non-expansive* in width (no concatenation), which avoids channel inflation and empirically stabilizes optimization. Across scales, the network thus propagates coarse global structure while preserving high-frequency details refined by local convolutions. This decomposition mirrors the physical intuition that low-k modes control large-scale dynamics, whereas high-k modes encode fine structures, discontinuities, and small eddies. For the architecture pseudocode content, please refer to Appendix E.

4.2 Controlled Low-Frequency Mixing

In complex physical scenarios, achieving global-range feature interaction is crucial for capturing long-range correlated dynamic behavior. However, indiscriminately applying global convolutions or frequency-domain operations across the entire frequency spectrum may overly amplify high-frequency noise and disrupt local details, leading to model instability. Therefore, the frequency-domain branch of DRIFT-Net performs controlled global mixing operations only on low-frequency modes, focusing global coupling on large-scale structures. Low-frequency components represent the field's overall shape and long-term evolution trends and play a decisive role in the global dynamics. In contrast, high-frequency components carry fine-grained local structures. By introducing global mixing only in the low-frequency band, the model can effectively propagate large-scale information at each layer to directly couple distant spatial locations, while maximally preserving high-frequency details and avoiding unnecessary interference with small-scale structures.

Fourier-domain decomposition and motivation. At the feature level, low spatial frequencies correspond to smooth, domain-wide basis functions whose coefficients modulate global structures; high frequencies capture localized variations. Modifying only a bounded set of low-k coefficients yields a global yet parsimonious interaction pattern that respects the separation of scales. This aligns with classical spectral methods and with the observation that neural networks tend to fit low-frequency components first (spectral bias) (Rahaman et al., 2019; Xu et al., 2020). Our design *uses* this bias constructively by letting the network explicitly *learn* how to mix low-k channels, while leaving high-k content intact.

Block computation. In each DRIFT-Net block, we apply a 2D real FFT (rFFT2) to the input feature X_{in} to obtain its frequency-domain representation $\hat{X}(k)$; by Hermitian symmetry of real-valued inputs, only half of the spectrum needs to be represented. We then use a *learnable rectangular* low-frequency mask $M_{\text{low}}(k_x, k_y)$ to split the spectrum into a low-frequency part $\hat{X}_{\text{low}}$ and a high-frequency residual $\hat{X}_{\text{high}} = \hat{X} - \hat{X}_{\text{low}}$. Within the low-frequency range $k \in M_{\text{low}}$, we introduce a learnable channel-wise complex linear transformation W (acting per frequency, without cross-frequency coupling) to mix the corresponding spectral coefficients (Rao et al., 2021):

$$\begin{aligned} \hat{V}(k) &= W\,\hat{X}(k)\,, \quad k \in M_{\text{low}}\,, \\ \hat{V}(k) &= \hat{X}(k)\,, \quad k \notin M_{\text{low}}\,. \end{aligned} \tag{4.1}$$

For later use we denote $\hat{V}_{\text{low}}(k) = \mathbb{1}_{k \in M_{\text{low}}} \hat{V}(k)$ and keep $\hat{X}_{\text{high}}(k) = \hat{X}(k) - \mathbb{1}_{k \in M_{\text{low}}} \hat{X}(k)$. In practice, W is unconstrained (no explicit spectral-norm clipping), which keeps the transform expressive. We use rFFT2/iFFT2 and the inverse transform yields a real-valued field by construction (Cooley & Tukey, 1965; Frigo & Johnson, 2005). This design focuses global mixing on large scales while leaving high-frequency content intact. For implementation details, refer to Appendix C.

4.3 Bandwise Fusion with Radial Gating

After the above low-frequency mixing, we need to fuse the modified low-frequency information from the frequency branch with the complementary content to produce the final spectrum for the

inverse transform back to the spatial domain. If one simply performs a hard band replacement or a direct addition, discontinuities may occur at the frequency band boundaries or amplitude overshoot may be introduced in certain bands. Therefore, we design a smooth and stable frequency-band fusion mechanism that uses a radial gating coefficient varying with frequency to weight and combine the two frequency-domain signals.

Specifically, we assign each frequency k in the spectrum a weight $\alpha(k) \in [0,1]$ to balance the contributions of the low-frequency mapped component and the high-frequency residual. We compute $\alpha(k)$ as a function of the frequency magnitude (radial frequency) using lightweight per-band processing and expand it to per-frequency weights so that in the low-frequency region $\alpha(k) \approx 1$ (favoring the mixed global component), whereas in the high-frequency region $\alpha(k) \approx 0$ (preserving fine local details). The fusion is then computed as

$$\hat{Y}(k) \;=\; \alpha(k)\,\hat{V}_{\text{low}}(k) \;+\; \big(1-\alpha(k)\big)\,\hat{X}_{\text{high}}(k)\,, \tag{4.2}$$

and since $\alpha(k) \in [0,1]$, by convexity we have the pointwise magnitude bound

$$\big|\hat{Y}(k)\big| \;\le\; \max\Big\{\,\big|\hat{V}_{\text{low}}(k)\big|,\; \big|\hat{X}_{\text{high}}(k)\big|\,\Big\}\,. \tag{4.3}$$

This bandwise "clamping" effect avoids introducing energy larger than either source at any frequency and empirically stabilizes training. Finally, we apply iFFT2 to obtain the spectral-path output in the spatial domain and add it to the image branch at the same scale.

Why radial and why additive. Choosing $\alpha(k)$ as a radial function enforces isotropy in fusion and prevents directional artifacts around the mask boundary. The convex combination above yields a non-expansive operator in the frequency domain (no overshoot), which directly translates to fewer ringing artifacts after iFFT2. In the spatial domain, we merge the spectral and image signals by *addition* rather than concatenation. This preserves feature dimensionality and makes the fusion behave as a residual correction that injects globally coupled information while letting the image branch keep full control over high-frequency refinement. See Appendix C for details.

4.4 Frequency-Weighted Loss

Despite the above architectural improvements in global coupling and local detail fidelity, the model training stage still needs to address the issue of spectral bias. Conventional losses tend to be dominated by low-frequency errors during optimization, causing the model to preferentially fit large-scale structures while converging slowly on smaller-amplitude yet physically important high-frequency details. As a result, fine structures in the predicted field may be underfitted and gradually become blurred over time.

To mitigate spectral bias in a simple and stable manner, we add a *frequency-weighted* auxiliary term in the Fourier domain. Let $E = u_\theta - u$ be the prediction error and $\widehat{E}$ its rFFT2 under the same normalization. We reweight the error spectrum by a radial weight $w(r) \propto r^\alpha$ (with r the normalized frequency magnitude) and minimize

$$L \;=\; L_{\text{base}} \;+\; \lambda\,\mathbb{E}\big[\,w(r)\,\big|\widehat{E}(k)\big|^2\big]\,, \tag{4.4}$$

where L_{base} is the standard L_p loss and (λ, α) are scalars. This radial weighting increases sensitivity to high-$|k|$ components and complements the bandwise fusion used in the frequency path, thereby improving the fidelity of multi-scale structures in long-horizon predictions (Fuoli et al., 2021; Jiang et al., 2021). From a functional-analytic perspective, this amounts to emphasizing higher-order (roughness-related) components of the error—akin to moving from a pure L_p metric toward a Sobolev-like metric—so that small-scale discrepancies are not overshadowed by low-frequency energy during optimization. Practically, tuning λ controls the balance between coarse and fine accuracy, while α modulates how aggressively high frequencies are emphasized; we find moderate values suffice to counter underfitting of fine structures without degrading large-scale fidelity. See Appendix B for details.

5 Experiments

Protocol and metrics. We follow the POSEIDON evaluation protocol Herde et al. (2024) and compare DRIFT-Net against two strong baselines: SCOT (multi-scale windowed self-attention)

and FNO (a Fourier neural operator). All models use the same training/validation/test split and preprocessing. They are trained with an identical data budget (approximately 20,000 trajectories over 40 epochs) and identical optimization settings (the same optimizer, learning-rate schedule, batch size, numerical precision, and hardware). On a discrete grid with C channels and $H \times W$ spatial points, we train a one-step operator $F_\theta : u_t \mapsto u_{t+1}$ using teacher forcing. At test time, we apply F_θ autoregressively in a closed loop to reach the common target time T^*. We report the test-set mean relative L^1 error at the final time:

$$e_{\text{rel-L1}} = \frac{\|\hat{u}_{T^*} - u_{T^*}\|_1}{\|u_{T^*}\|_1}, \qquad \|v\|_1 := \frac{1}{CHW} \sum_{c=1}^{C} \sum_{i=1}^{H} \sum_{j=1}^{W} |v_{c,i,j}|.$$

For tasks with multiple quantities of interest (QoIs)—for example, NS-Tracer-PwC predicts (u_x, u_y, c) while NS-PwC, NS-SL, and FNS-KF output only (u_x, u_y)—we compute the relative error for each QoI and then take an unweighted average:

$$e_{\text{task}} = \frac{1}{|Q|} \sum_{q \in Q} \frac{\|\hat{u}^{(q)}_{T^*} - u^{(q)}_{T^*}\|_1}{\|u^{(q)}_{T^*}\|_1}.$$

Benchmark tasks. We evaluate DRIFT-NET on four canonical unsteady Navier–Stokes benchmarks from the POSEIDON suite Herde et al. (2024). NS-SL (shear layer) tests vortex roll-up and mixing from a perturbed interface. NS-PwC (piecewise-constant vorticity) stresses the advection of sharp discontinuities. NS-Tracer-PwC extends NS-PwC by introducing a passive scalar c, which requires accurate coupling between the velocity and tracer fields. FNS-KF (forced Kolmogorov flow) sustains two-dimensional turbulence via steady forcing, challenging long-horizon stability and multi-scale fidelity.

In addition to these four benchmarks, we also consider two *ApeBench*-generated forced Kolmogorov variants constructed using a high-accuracy pseudo-spectral solver Koehler et al. (2024). These two variants share the same PDE setup and preprocessing but differ in physical parameters: one is more turbulent, whereas the other is comparatively smoother. To emphasize long-horizon robustness, we evaluate on both variants with a closed-loop rollout length of $T = 100$ and, for these two datasets, compare DRIFT-NET against SCOT under identical settings.

Main results. Table 1 reports the final-time relative L^1 errors under matched training conditions and evaluation protocol. On all four POSEIDON benchmarks, DRIFT-NET attains the lowest error among the compared methods. The table also includes the two long-horizon *ApeBench*-based Kolmogorov variants (rollout to T=100): on the turbulent variant, DRIFT-NET achieves a lower final error than SCOT; on the smoother variant, DRIFT-NET again yields a lower final error, indicating improved robustness across physical regimes. For completeness, we further summarize the mean error across the rollout and the linear growth rate in the paragraph following the table, and present the corresponding error–time curves in Figure 2.

Table 1: Final-time relative L^1 error (lower is better). Closed-loop rollouts to the common target time T^* for POSEIDON benchmarks, and to T=100 for our two ApeBench-based long-horizon Kolmogorov datasets. Values are *test-set means*. **Best in bold**.

Task	SCOT	FNO	DRIFT-NET
NS-SL (POSEIDON)	3.96	3.69	**3.40**
NS-PwC (POSEIDON)	2.35	4.57	**1.09**
NS-Tracer-PwC (POSEIDON)	5.18	9.46	**4.19**
FNS-KF (POSEIDON)	4.65	4.43	**4.32**
KF-Long (turbulent, ApeBench, T=100)	114.14	–	**110.87**
KF-Long (smoother, ApeBench, T=100)	62.99	–	**57.17**

For the two *ApeBench*-based datasets, we also report the mean relative L^1 error across the rollout and a linear growth slope obtained by a least-squares fit of error versus time. On the turbulent variant, DRIFT-NET attains a mean error of 69.89 compared with 74.76 for SCOT, and an error growth

slope of 1.154 compared with 1.185 for SCOT. On the smoother variant, the mean error is 31.07 compared with 35.41 for SCOT, and the slope is 0.581 compared with 0.641 for SCOT. Figure 2 depicts the corresponding error–time curves (left: turbulent; right: smoother), showing consistently lower curves and flatter tails for DRIFT-NET.

Long-horizon behavior. Although final-time metrics are informative, the full closed-loop trajectory over a long horizon provides additional insight into cumulative drift. Figure 3 presents the test-set mean relative L^1 error at each time step for NS-Tracer-PwC. DRIFT-NET achieves a final-time error of 4.193, whereas SCOT achieves 5.182 (a reduction of 19.1%). The average error across the rollout is 1.99 for DRIFT-NET and 2.46 for SCOT, and the linear growth slopes are 0.176 and 0.229, respectively. These results are consistent with the objective of strengthening global low-k coupling while maintaining high-k fidelity.

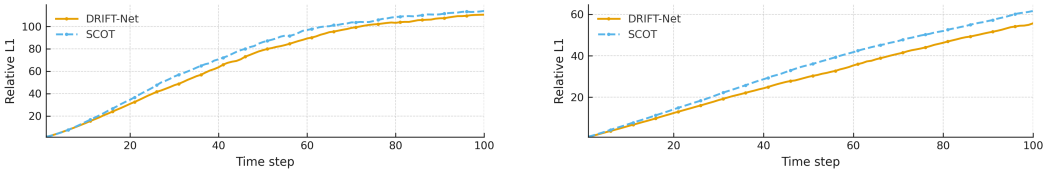


Figure 2: **Error vs. time on ApeBench-based long-horizon Kolmogorov datasets** (T=100)**.** Left: turbulent; Right: smoother. Solid line: DRIFT-NET; dashed line: SCOT.

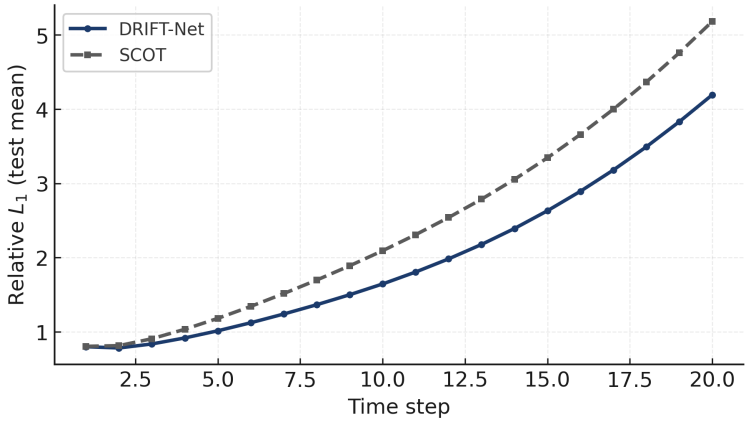


Figure 3: **Error vs. time on NS-Tracer-PwC.** Solid line with circles: DRIFT-NET; dashed line with squares: SCOT. We plot the test-set mean relative L^1 at each time step under closed-loop rollouts.

Additional final-time results (subset) and context. For the tasks where error time-series are available for both models, the final-time results confirm the advantage of DRIFT-NET over SCOT (Table 2). For completeness, we also include NS-PwC, on which DRIFT-NET achieves a final error of **1.090** compared with 2.350 for SCOT (as reported in prior results).

Table 2: **Additional final-time results (subset).** Final-time relative L^1 (lower is better) where both series are available, and percent improvement of DRIFT-NET vs. SCOT. **Best in bold**.

Task	DRIFT-NET ($e_{\text{rel-L1}}@T^*$)	SCOT ($e_{\text{rel-L1}}@T^*$)	% Improvement
NS-Tracer-PwC	**4.193**	5.182	19.1%
FNS-KF	**4.315**	4.647	7.2%
NS-SL	**3.402**	3.956	14.0%
NS-PwC	**1.090**	2.350	53.6%

Efficiency comparison. We also evaluate model size and inference speed under identical inference conditions (same hardware, numerical precision, batch size, and sequence length; excluding data-loading overhead). DRIFT-NET contains approximately 17 million parameters (about 15% fewer than SCOT's 20 million) and achieves roughly 158 time-step predictions per second, whereas SCOT achieves about 118 steps/s (Table 3). The Conclusion reports 158 vs. 118 steps/s under a different resolution and precision configuration; both measurements reflect a similar relative advantage for DRIFT-NET, and we include both for completeness.

Table 3: Model size and inference throughput (higher throughput is better). Throughput measured under identical inference settings.

Model	Parameters (M)	Throughput (steps/s)
DRIFT-NET	17	**158**
SCOT	20	118

Ablation studies. We assess the contribution of each major component of DRIFT-NET: Low-Frequency Mixing (LFM), Radial Gating (RG), and Frequency-Weighted Loss (FWL). We train ablated variants on NS-PwC and NS-Tracer-PwC, holding all other hyperparameters fixed and using the same training budget and evaluation protocol as the full model. Removing LFM or RG noticeably worsens final-time accuracy; removing FWL also degrades performance, although to a smaller degree (Table 4). Specifically, relative to the full model, eliminating LFM increases the final error by 0.56 on NS-PwC and by 2.13 on NS-Tracer-PwC. Removing RG increases the error by 0.61 and 2.36 on these tasks, respectively. Dropping FWL yields smaller increases of 0.27 and 1.17.

Table 4: Ablation of DRIFT-NET components. Numbers are final-time relative L^1 errors (lower is better) on NS-PwC and NS-Tracer-PwC under the same training and evaluation protocol. **Best in bold**.

Model variant	NS-PwC	NS-Tracer-PwC
Full DRIFT-NET (ours)	**1.09**	**4.19**
w/o Low-Frequency Mixing (LFM)	1.65	6.32
w/o Radial Gating (RG)	1.70	6.55
w/o Frequency-Weighted Loss (FWL)	1.36	5.36

Spectral analysis for ablations. To characterize scale-dependent effects, we analyze the evolution of bandwise errors over time (normalized RMSE per frequency band). Figure 4 compares four model variants (the full DRIFT-NET, no FWL, no RG, and no LFM), each showing error-versus-time curves for multiple wavenumber bands. The full model most effectively suppresses error growth in the mid- and high-frequency bands (those with $k \geq 16$). In contrast, removing RG leads to the earliest and most pronounced increase in the highest-frequency band. Removing FWL yields a marked late-stage increase in high-frequency error. Removing LFM increases both mid- and high-frequency errors, consistent with weakened low-frequency global coupling.

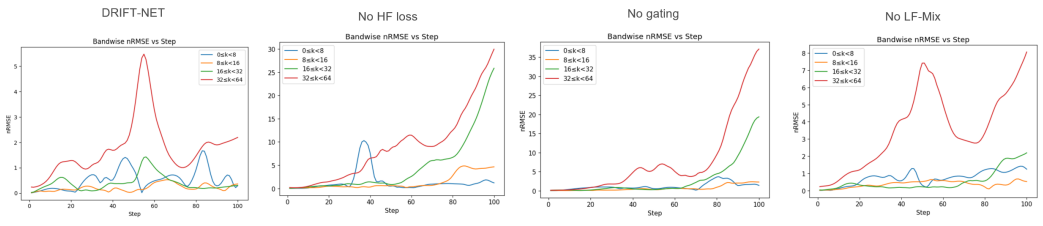


Figure 4: **Bandwise nRMSE vs. step for ablations.** Left to right: full DRIFT-NET, no HF loss, no gating, no LF-Mix.

6 Limitations and Future Work

DRIFT-Net has limitation. It uses a mask to select low frequencies. The initial band is hand tuned. The mask learns weights later, but the cutoff can be task specific. The model applies FFTs in many blocks. This adds memory traffic and latency on very large grids. Our tests target two dimensional flow. Three dimensional flow and coupled multi-physics may bring new training issues and higher cost. Future work will learn spectral partitions end to end. We will study 3D and multi-physics PDEs. We will pair the model with adaptive resolution and mesh refinement. We will also test irregular domains and complex boundary conditions.

7 Conclusion

We introduced DRIFT-Net for operator learning. The model couples a spectral branch with an image branch. It applies controlled low frequency mixing in the spectrum. It fuses branches with bandwise radial gating in a stable way. A frequency weighted loss reduces spectral bias. On Navier–Stokes benchmarks, DRIFT-Net lowers final time relative L1 error across all tasks. The reductions range from 7% to 54%. The model uses about 15% fewer parameters than SCOT and keeps higher throughput. For example, it reaches 158 steps per second while SCOT reaches 118 under the same setup. The unit is modular and can replace windowed attention blocks in existing backbones.

References

Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.

James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

Matteo Frigo and Steven G. Johnson. The design and implementation of fftw3. *Proceedings of the IEEE*, 93(2):216–231, 2005.

Dario Fuoli, Luc Van Gool, and Radu Timofte. Fourier space losses for efficient perceptual image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2360–2369, 2021.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive fourier neural operators: Efficient token mixers for transformers. In *International Conference on Learning Representations (ICLR)*, 2022.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13919–13929, October 2021.

Felix Koehler, Simon Niedermayr, Rüdiger Westermann, and Nils Thuerey. Apebench: A benchmark for autoregressive neural emulators of pdes, 2024. URL `https://arxiv.org/abs/2411.00180`.

Nikola B. Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Zongyi Li, Nikola B. Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations (ICLR)*, 2021.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 24(388):18593–18618, 2023.

Phillip Lippe, Bastiaan S. Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. PDE-Refiner: Achieving accurate long rollouts with neural pde solvers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022, 2021.

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12009–12019, 2022.

Lu Lu, Pengzhan Jin, and George Em. Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3): 218–229, 2021.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron C. Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Md Ashiqur Rahman, Zachary E. Ross, and Kamyar Azizzadenesheli. U-no: U-shaped neural operators. *arXiv preprint arXiv:2204.11127*, 2022.

Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Bogdan Raonić, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional neural operators for robust and accurate learning of pdes. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241. Springer, 2015.

Shashank Subramanian, Peter Harrington, Kurt Keutzer, Wahid Bhimji, Dmitriy Morozov, Michael W. Mahoney, and Amir Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Lloyd N. Trefethen. *Spectral Methods in MATLAB*. SIAM, 2000.

Gege Wen, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M. Benson. U-FNO—an enhanced Fourier Neural Operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, 163:104180, 2022. doi: 10.1016/j.advwatres.2022.104180.

Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *Communications in Computational Physics*, 28(5):1746–1767, 2020.

APPENDIX

A THEORETICAL PROOFS AND BOUNDS

A.1 SETUP AND NOTATION

Let $\Omega = \mathbb{T}^2$ be a periodic domain. Denote by $\mathcal{F}$ the 2-D rFFT with forward normalization (`norm=forward`). For a real-valued tensor $u \in \mathbb{R}^{H\times W\times C}$, write $\hat{u} = \mathcal{F}(u)$. Two learnable Nyquist-scaled cut-offs $\kappa_x, \kappa_y \in (0, 0.5]$ define a low-frequency rectangle

$$\hat{u}_{\text{low}} = \hat{u} \cdot \mathbf{1}_{\{\,|k_x|<\kappa_x,\ |k_y|<\kappa_y\,\}}, \qquad \hat{u}_{\text{high}} = \hat{u} - \hat{u}_{\text{low}},$$

with area $\sigma_{\text{LF}} = \kappa_x\kappa_y \le 0.25$. The spectral half-plane is partitioned into concentric radial bands $\{B_j\}_{j=0}^{J-1}$ for gating.

A.2 PER-BLOCK LIPSCHITZ UPPER BOUND

Core DRIFT operator. Inside the low rectangle a shared complex matrix $W \in \mathbb{C}^{C\times C}$ mixes channels:

$$\hat{u}_{\text{low-mix}}(k) = W\,\hat{u}_{\text{low}}(k), \quad k \in M_{\text{low}}.$$

Let the three parallel branches of a DRIFT layer be: (i) low-band mixer S, (ii) depth-wise 3×3 convolution C, (iii) 1×1 linear map L. With forward-normalized FFT, Parseval on the low rectangle yields a fixed scaling absorbed into operator constants.

Lemma A.1 (Core DRIFT operator bound). For the residual-free core $R_{\text{DRIFT}} = S + C + L$,

$$\|R_{\text{DRIFT}}\|_{\text{Lip}} \le \sqrt{\sigma_{\text{LF}}}\,\rho_W + K_{\text{conv}} + K_{\text{lin}},$$

where $\rho_W = \|W\|_2$, and $K_{\text{conv}}, K_{\text{lin}}$ are operator norms of the depth-wise conv and the 1×1 linear, respectively. If the block is wrapped by an outer residual $I + R_{\text{DRIFT}}$, then $\|I + R_{\text{DRIFT}}\|_{\text{Lip}} \le 1 + \|R_{\text{DRIFT}}\|_{\text{Lip}}$.

Swin-style reference bound. A Swin-style block with identity shortcut decomposes as $x \mapsto x + A_{\text{attn}}(x) + A_{\text{mlp}}(x)$, hence

$$\|F_{\text{SWIN}}\|_{\text{Lip}} \le 1 + \|A_{\text{attn}}\|_2 + \|A_{\text{mlp}}\|_2.$$

A.3 RELATIVE NETWORK-LEVEL BOUND

Proposition A.2 (Tighter cumulative bound). Suppose both DRIFT and Swin-style blocks use the same outer residual $I + \cdot$. If for every depth

$$\sqrt{\sigma_{\text{LF}}}\,\rho_W + K_{\text{conv}} + K_{\text{lin}} < \|A_{\text{attn}}\|_2 + \|A_{\text{mlp}}\|_2,$$

then for any number of layers L,

$$\prod_{\ell=1}^{L} \|F^{(\ell)}_{\text{DRIFT}}\|_{\text{Lip}} < \prod_{\ell=1}^{L} \|F^{(\ell)}_{\text{SWIN}}\|_{\text{Lip}}.$$

Consequently, DRIFT-Net admits a strictly smaller worst-case gain than an equally deep Swin-style stack without requiring either network to be contractive (< 1).

Discrete Grönwall implication. Let $e_{m+1} \le \bar{K}_\theta\, e_m + \eta_m$ with one-step defect η_m. Replacing $\bar{K}_\theta^{\text{SWIN}}$ by $\bar{K}_\theta^{\text{DRIFT}} < \bar{K}_\theta^{\text{SWIN}}$ flattens the geometric factor in the discrete Grönwall inequality:

$$e_m \le \bar{K}_\theta^{\,m} e_0 + \frac{1 - \bar{K}_\theta^{\,m}}{1 - \bar{K}_\theta}\,\bar{\eta}, \qquad \bar{\eta} = \max_{i<m} \eta_i.$$

A.4 RADIAL-BAND GATING IS ENERGY NON-EXPANSIVE

Lemma A.3 (Pointwise amplitude bound). For any fixed $\alpha \in [0, 1]$ and Fourier location k,

$$\hat{v}(k) = \alpha\,\hat{u}_{\text{low-mix}}(k) + \big(1-\alpha\big)\,\hat{u}_{\text{high}}(k) \Rightarrow |\hat{v}(k)| \le \max\{\,|\hat{u}_{\text{low-mix}}(k)|,\ |\hat{u}_{\text{high}}(k)|\,\}.$$

Proof. Convexity: $|\alpha a + (1-\alpha)b| \le \alpha|a| + (1-\alpha)|b| \le \max\{|a|, |b|\}$. □

Remark (input-dependent gates). When α depends on the input via a band-statistics MLP, the amplitude bound still holds pointwise; the Lipschitz constant additionally includes a term from $\partial\alpha/\partial u$. In practice, one may detach gradients through α in the stability analysis or constrain the gate MLP by spectral normalization.

A.5 Operator-norm controls for conv/linear

Depth-wise 3×3 conv. On a periodic grid, the depth-wise conv is block-circulant and diagonalized by the DFT. Hence $\|T_k\|_2 = \max_{h,w} |\widehat{k}_{h,w}|$. Ensuring $\sum_{i,j} |k_{ij}| \le 1$ or projecting to $\max_{h,w} |\widehat{k}_{h,w}| \le 1$ yields $K_{\text{conv}} \le 1$.

1×1 **linear.** If $W \in \mathbb{R}^{C\times C}$ is orthogonally initialized and projected each step to the spectral ball of radius $c_L \le 1$, then $\|W\|_2 \le c_L$, giving $K_{\text{lin}} \le c_L$.

B Sobolev-weighted One-step Defect Bound

Theorem B.1 (Sobolev-closed defect). Fix $s > 0$, $\lambda_{\text{hf}} > 0$ and define $L_{\text{Sob}}(u, \hat{u}) = \lambda_{\text{hf}} \|\Lambda^s (u - \hat{u})\|_2^2$ with $\Lambda^s = (I - \Delta)^{s/2}$. If $\mathbb{E}\, L_{\text{Sob}} \le \varepsilon$, then the expected one-step defect $\eta_m = \| \hat{u}_{m+1} - \Phi_\theta(\hat{u}_m) \|_2$ satisfies

$$\mathbb{E}\, \eta_m \le \lambda_{\text{hf}}^{\frac{1}{2+s}}\, \varepsilon^{\frac{1}{2+s}} =: \bar{\eta}.$$

Proof sketch. Parseval (with `norm=forward`) gives $L_{\text{Sob}} = \lambda_{\text{hf}} \sum_k (1 + \|k\|^2)^s |e_k|^2$, $e_k = u_k - \hat{u}_k$. Split the spectrum at radius r: $\eta_m^2 = \sum_{\|k\| \le r} |e_k|^2 + \sum_{\|k\| > r} |e_k|^2$; the HF term $\le (1 + r^2)^{-s} \lambda_{\text{hf}}^{-1} L_{\text{Sob}}$. Optimizing r yields the stated bound; insert into the discrete Grönwall inequality.□

C Spectral Pipeline Details

Low/high split (implementation). Given input size (H, W), the rFFT has size $(H_{\text{fft}}, W_{\text{fft}}) = (H,\ W/2 + 1)$. Two learnable scalars $\kappa_x = \sigma(\theta_x)$, $\kappa_y = \sigma(\theta_y)$ define

$$M_{\text{low}} = \{\, |k_x| < \kappa_x H_{\text{fft}},\ |k_y| < \kappa_y W_{\text{fft}} \,\}, \quad \hat{u}_{\text{low}} = \hat{u} \cdot \mathbf{1}_{M_{\text{low}}}, \quad \hat{u}_{\text{high}} = \hat{u} - \hat{u}_{\text{low}}.$$

Radial gate (energy-fraction driven). Let $r_{ij} = \sqrt{(i/(H_{\text{fft}} - 1))^2 + (j/(W_{\text{fft}} - 1))^2}$ and bands $B_j = \{(i, j) : \lfloor J r_{ij} \rfloor = j\}$, $j = 0, \ldots, J - 1$. Define a high-frequency energy fraction

$$E_{\text{HF}}(k) = \frac{|\hat{u}_{\text{high}}(k)|^2}{|\hat{u}_{\text{low}}(k)|^2 + |\hat{u}_{\text{high}}(k)|^2 + \varepsilon}.$$

Average within each band to obtain $\bar{f}_{n,c}(j)$, pass through a two-layer MLP with sigmoid output, and broadcast back: $\alpha_{n,c}(k) = \sigma(\text{MLP}(\bar{f}_{n,c}(\lfloor J r_{ij} \rfloor))) \in (0, 1]$.

Spectral fusion and inference-only taper. Blend as

$$\hat{v}(k) = \alpha(k)\, \hat{u}_{\text{low-mix}}(k) + \big(1 - \alpha(k)\big)\, \hat{u}_{\text{high}}(k), \qquad v = \mathcal{F}^{-1}(\hat{v}).$$

At evaluation time, a lightweight outer-band taper may be applied on the outermost ring: $\hat{v}(k) \leftarrow (1 - \beta\, \bar{\alpha}(k))\, \hat{v}(k)$ with $\beta \in (0, 1/2]$ and $\bar{\alpha}$ the channel-wise mean gate. Because $0 \le \alpha \le 1$ and the taper factor ≤ 1, every Fourier mode is non-expansive.

D Complexity and Throughput Protocol

Asymptotic costs per DRIFT block. rFFT/iFFT pair: $\mathcal{O}(HW \log(HW))$; band statistics & broadcast: $\mathcal{O}(HW)$; low-band mixing: $\mathcal{O}(|M_{\text{low}}|\, C^2)$ with $|M_{\text{low}}| \ll HW$. Thus DRIFT-Net matches the asymptotic complexity of window attention while eliminating dense projections in attention blocks.

Algorithm 1 DRIFT-NET one-step forward and training loss (code-aligned)

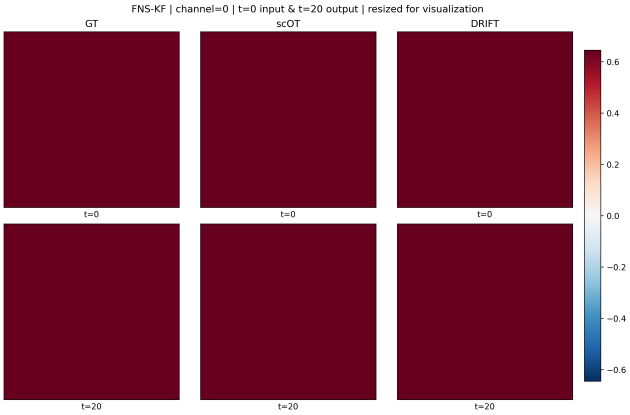
Require: current field $u_t \in \mathbb{R}^{C\times H\times W}$; #scales L; learnable LF mask params (k_x, k_y); complex mixers $\{W_\ell\}_{\ell=1}^{L}$; radial band gates $\{\mathrm{BandGate}_\ell\}_{\ell=1}^{L}$; base loss L_{base}; spectral weights $w(r)$; coefficient λ

Ensure: predicted next field $\hat{u}_{t+1}$; loss L (if training)

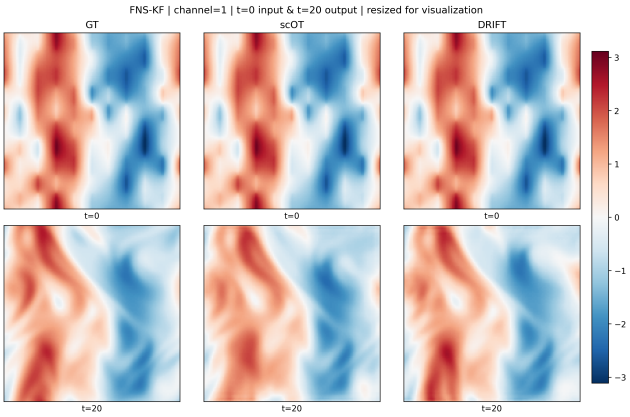
$x \leftarrow \mathrm{Embed}(u_t)$
for $\ell = 1$ to L **do**
  $\hat{X} \leftarrow \mathrm{rFFT2}(x)$
  $(H_{\text{fft}}, W_{\text{fft}}) \leftarrow \mathrm{shape}(\hat{X})$; $k_x \leftarrow \lfloor \sigma(\theta_x)\, H_{\text{fft}} \rfloor$, $k_y \leftarrow \lfloor \sigma(\theta_y)\, W_{\text{fft}} \rfloor$
  $\hat{X}_{\text{low}} \leftarrow \hat{X} \odot \mathbf{1}_{[:k_x,:k_y]}$; $\hat{X}_{\text{high}} \leftarrow \hat{X} - \hat{X}_{\text{low}}$
  $\hat{V}_{\text{low}}(k) \leftarrow W_\ell\, \hat{X}(k)$ for $k \in [:k_x, :k_y]$; $\hat{V}_{\text{low}}(k) \leftarrow 0$ otherwise
  $\mathrm{feat} \leftarrow \big||\hat{X}_{\text{high}}| - |\hat{X}_{\text{low}}|\big|$
  $\alpha(k) \leftarrow \mathrm{BandGate}_\ell\big(\mathrm{Pool}_r(\mathrm{feat}), \|k\|\big)$
  $\hat{Y}(k) \leftarrow \alpha(k)\, \hat{V}_{\text{low}}(k) + \big(1 - \alpha(k)\big)\, \hat{X}_{\text{high}}(k)$
  $y_{\text{spec}} \leftarrow \mathrm{iFFT2}(\hat{Y})$
  $y_{\text{local}} \leftarrow \mathrm{DWConv}_\ell(x) + \mathrm{PointwiseLinear}_\ell(x)$
  $z \leftarrow \mathrm{Norm}_\ell\big(y_{\text{spec}} + y_{\text{local}},\ \mathrm{time}\big)$
  $x \leftarrow x + z$
  if $\ell < L$ **then**
    $x \leftarrow \mathrm{Downsample}(x)$
for $\ell = L$ down to 1 **do**
  if $\ell < L$ **then**
    $x \leftarrow \mathrm{Upsample}(x)$
  $x \leftarrow \mathrm{ConvNeXtBlock}_\ell(x)$
$\hat{u}_{t+1} \leftarrow \mathrm{Recover}(x)$
if training **then**
  $E \leftarrow \hat{u}_{t+1} - u_{t+1}$; $\widehat{E} \leftarrow \mathrm{rFFT2}(E)$
  $L_{\text{base}} \leftarrow \|E\|_p$ ▷ $p \in \{1, 2\}$
  $L_{\text{freq}} \leftarrow \lambda \cdot \mathbb{E}_k\big[w(\|k\|)\, |\widehat{E}(k)|^2\big]$
  $L \leftarrow L_{\text{base}} + L_{\text{freq}}$
return $\hat{u}_{t+1}$ (and L if training)

E PSEUDOCODE

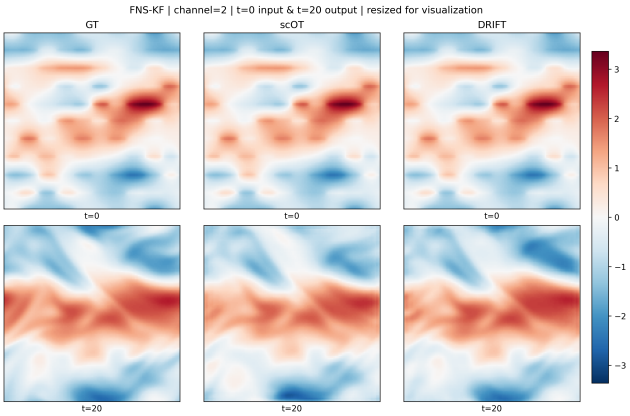
F FORCED KOLMOGOROV FLOW (KF) VISUALIZATION



Channel 0



Channel 1



Channel 2

Figure 5: **FNS-KF qualitative visualization.** Each panel shows GT / scOT / DRIFT at $t{=}0$ (top) and $t{=}20$ (bottom) for a single channel. Example is illustrative (not a main result).