Conservation-Constrained Fourier Neural Operators for Physically Consistent PDE Simulation

Anonymous Author 1 [1] **Anonymous Author 2** [1] **Anonymous Author 3** [1]

Abstract

Neural operators have enabled rapid surrogate modeling of partial differential equations but systematically violate conservation laws, limiting their physical fidelity and long-horizon stability. We introduce CC-FNO, a modified Fourier Neural Operator that enforces mass conservation via a differentiable Leray projection applied to the output velocity field, projecting it onto the divergence-free subspace in Fourier space at negligible computational cost (<2% overhead). On 2D incompressible Navier–Stokes at $\mathrm{Re} \in \{100, 500, 1000, 2000\}$, CC-FNO reduces divergence by five orders of magnitude ($\sim 10^{-5}$ vs. ~ 3.0 for FNO, $\sim 10^{-2}$ for PINO) while matching baseline accuracy across seven model comparisons including FNO, PINO, Stream-FNO, U-Net, and FNO with pushforward training. We additionally propose CC-FNO-E, which stabilizes autoregressive rollouts through a per-step energy bound that acts as a stability regularizer, preventing catastrophic amplitude amplification. This extends stable rollout horizons by 17–52% over FNO across all regimes, though as a stability regularizer rather than a physical conservation law, it systematically underestimates energy growth by 67–84% at the final rollout step in forced flow. A systematic penalty weight sweep confirms that soft constraints cannot match hard projection regardless of tuning, and an ablation of forcing-aware energy bounds reveals that simple one-sided capping outperforms physically motivated alternatives. Our results establish a clear constraint hierarchy (none < soft < hard < hard+energy stabilizer) and demonstrate that hard conservation constraints provide "free" physical consistency without sacrificing accuracy.

[1]Anonymous Institution, Anonymous City, Anonymous Country. Correspondence to: Anonymous Author 1 <anon@email.com>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

1. Introduction

Neural operators learn mappings between infinite-dimensional function spaces and have achieved orders-of-magnitude speedups over classical PDE solvers for many-query applications including uncertainty quantification and design optimization. The Fourier Neural Operator (FNO) (Li et al., 2021) parameterizes integral kernels in the Fourier domain, while DeepONet (Lu et al., 2021) and its extensions provide complementary operator learning frameworks (Kovachki et al., 2023). Recent PDE foundation models (Herde et al., 2024; Hao et al., 2024; McCabe et al., 2023) have further advanced generalization through large-scale pretraining.

Despite these advances, three interrelated failure modes persist. *First*, standard neural operators violate conservation laws: predicted velocity fields for incompressible flow have nonzero divergence, producing fictitious mass sources. *Second*, models trained at specific parameter regimes suffer sharp accuracy degradation under distribution shift (Takamoto et al., 2022). *Third*, autoregressive rollouts accumulate errors rapidly, diverging from the true trajectory within tens of timesteps despite techniques such as pushforward training (Brandstetter et al., 2022) and iterative refinement (Lippe et al., 2023).

We argue these failures are linked through the structure of the PDE solution manifold. Conservation laws constrain admissible states: divergence-free velocity fields form a proper subspace of all vector fields, and energy bounds further restrict the dynamics to a compact set. A model that respects these constraints cannot drift into physically inadmissible states, precisely those responsible for distribution shift failures and rollout instability. This perspective is well-established in classical numerical analysis, where structure-preserving discretizations (symplectic integrators, mimetic methods, compatible finite elements) are known to yield superior long-time stability compared to generic methods (Zhong et al., 2020).

We introduce the **Conservation-Constrained Fourier Neural Operator (CC-FNO)**, which enforces mass conservation by applying a differentiable *Leray projection* to the output velocity field. The Leray projection is well-

established in computational physics and increasingly used in ML pipelines; our contribution is not the projection itself, but the first systematic study establishing a complete constraint hierarchy across seven baselines with extensive ablations, demonstrating that this simple mechanism is "free" in accuracy, cost, and implementation complexity. In Fourier space, the projection reduces to a pointwise operation at cost $\mathcal{O}(N^2)$, negligible compared to the FFT cost of $\mathcal{O}(N^2 \log N)$. We additionally propose CC-FNO-E, which stabilizes autoregressive rollouts via a per-step energy bound that acts as a *stability regularizer*, preventing catastrophic amplitude growth, the dominant instability mechanism in neural PDE surrogates.

Contributions.

1. CC-FNO: hard mass conservation via Leray projection in Fourier space at $<2\%$ overhead, with systematic comparison to seven baselines spanning the full constraint spectrum (§3).
2. CC-FNO-E: one-sided per-step energy bounding as a stability regularizer that extends rollout horizons by 17–52% over FNO, with honest characterization of its limitations as a non-conservative energy bound (§3.3).
3. Comprehensive evaluation demonstrating five-orders-of-magnitude conservation improvement and near-identical accuracy, establishing a clear constraint hierarchy: none $<$ soft $<$ hard $<$ hard+energy stabilizer (§5).
4. Systematic ablations: a penalty weight sweep showing soft constraints face an accuracy-conservation Pareto frontier that hard projection circumvents, and a forcing-aware ablation illustrating that robust simplicity outperforms fragile accuracy in autoregressive surrogates (§6).

2. Related Work

Neural operators. The FNO (Li et al., 2021) and its extensions (Geo-FNO, SFNO, CNO (Raonic et al., 2023)) parameterize integral kernels via Fourier series, enabling resolution-invariant operator learning with universal approximation guarantees (Kovachki et al., 2023). DeepONet (Lu et al., 2021) provides an alternative branch-trunk framework. None of these architectures enforce conservation laws; any observed conservation is incidental and degrades under distribution shift.

Physics-informed approaches. PINNs (Raissi et al., 2019) embed PDE residuals as soft penalty losses but exhibit systematic failure modes including spectral bias (Krishnapriyan et al., 2021). PINO (Li et al., 2024) adds soft divergence penalties to FNO training, reducing but not eliminating conservation violations. The fundamental limitation is that penalty-based methods trade off physics fidelity against data fidelity through a hyperparameter: the constraint is never exactly satisfied. We demonstrate this quantitatively through a systematic λ-sweep (§6.1).

Structure-preserving networks. Hamiltonian (Greydanus et al., 2019) and Lagrangian (Cranmer et al., 2020) neural networks conserve energy or respect variational structure, while symplectic integrators (Zhong et al., 2020) preserve Hamiltonian form. These have been remarkably successful for ODE systems but have not been extended to neural operators for spatiotemporal PDEs, where the state is an infinite-dimensional function. The challenge is that conservation laws in PDEs involve differential constraints (divergence-free fields) rather than scalar invariants (total energy), requiring fundamentally different enforcement mechanisms.

Alternative divergence-free architectures. Stream function and vector-potential parameterizations guarantee divergence-free output by construction (Richter-Powell et al., 2022). Divergence-free CNNs (Mohan et al., 2023) embed the constraint within convolutional filters. We include a Stream-FNO baseline (§4.2) and find that while it achieves low divergence for stable seeds, it suffers from training instability across random seeds. The Leray projection avoids this by operating directly on the velocity vector without differentiation.

Rollout stability. PDE-Refiner (Lippe et al., 2023) improves long rollouts via iterative refinement. Balanced spectral power (BSP) losses (Chattopadhyay et al., 2023) target multiscale fidelity. Diffusion-based surrogates (Kohl et al., 2024) provide complementary probabilistic stability. Our approach is orthogonal: hard constraints can be combined with any of these techniques. We include pushforward training (Brandstetter et al., 2022) as a baseline (FNO-PF) and show our approach is complementary.

Concurrent work. P&G (Li et al., 2026) independently embeds a spectral Leray projector within FNO for generative turbulence modeling; our work complements theirs with a systematic constraint hierarchy study across seven baselines and ablations, plus the energy stabilization mechanism. SSNO (Yang et al., 2025) targets spectral stability without explicit conservation. FlowRefiner (Lienen & Günnemann, 2026) reports accuracy degradation from hard projection in certain settings; our results suggest this is problem-dependent. ANCHOR (Berman & Karniadakis, 2025) and learned Galerkin schemes (Serrano et al., 2025) explore alternative pathways to physical consistency. More broadly, Helmholtz–Hodge projection has appeared in constrained sampling (Richter-Powell et al., 2022) and regression; our constraint hierarchy study demonstrates that hard projection dominates the physical benefit independently of

Table 1. Constraint enforcement paradigms in neural PDE solvers.

Method	Conservation	Operator	PDE
FNO (Li et al., 2021)	None	✓	✓
PINO (Li et al., 2024)	Soft	✓	✓
PINNs (Raissi et al., 2019)	Soft	—	✓
HNN (Greydanus et al., 2019)	Hard (energy)	—	—
P&G (Li et al., 2026)	Hard (mass)	✓	✓
Stream-FNO	Hard (mass)	✓	✓
CC-FNO (ours)	**Hard (mass)**	✓	✓
CC-FNO-E (ours)	**Hard+E-stab.**	✓	✓

the learning paradigm.

3. Method

3.1. Background: Fourier Neural Operator

The FNO (Li et al., 2021) approximates an operator $\mathcal{G}^\dagger : \mathcal{A} \to \mathcal{U}$ via:

$$\mathcal{G}_\theta = Q \circ \mathcal{L}_L \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{L}_1 \circ P, \tag{1}$$

where P lifts the input to d_v channels, Q projects to the output, σ is GELU, and each Fourier layer $\mathcal{L}_\ell$ acts as:

$$\mathcal{L}_\ell(v)(\boldsymbol{x}) = W_\ell v(\boldsymbol{x}) + \mathcal{F}^{-1}(R_\ell \cdot \hat{v}(\boldsymbol{k}))(\boldsymbol{x}), \tag{2}$$

with pointwise weights W_ℓ and learnable spectral weights $R_\ell(\boldsymbol{k}) \in \mathbb{C}^{d_v \times d_v}$ for modes $|\boldsymbol{k}| \leq k_{\max}$. In our setting, the input is vorticity $\omega(\boldsymbol{x}, t)$ on a periodic domain $[0, 2\pi]^2$ discretized on a 64×64 grid, and the task is to predict velocity $\boldsymbol{u}(\boldsymbol{x}, t{+}\Delta t) = (u, v)$.

3.2. Conservation-Constrained FNO

For incompressible flow, mass conservation is the divergence-free constraint $\nabla \cdot \boldsymbol{u} = 0$. Standard FNO architectures do not enforce this; predicted fields generically have nonzero divergence. Our key insight is that the *Leray projection* onto the divergence-free subspace is exact and cheap in Fourier space, making it natural to embed within the FNO's spectral pipeline.

Leray projection. The Leray–Helmholtz decomposition uniquely decomposes any vector field $\boldsymbol{u} = \boldsymbol{u}_{\text{sol}} + \nabla\phi$, where $\boldsymbol{u}_{\text{sol}}$ is divergence-free. The projector $\mathbb{P}$ extracts the solenoidal component. In Fourier space, this is *pointwise*:

$$\hat{\boldsymbol{u}}_{\text{proj}}(\boldsymbol{k}) = \hat{\boldsymbol{u}}(\boldsymbol{k}) - \frac{\boldsymbol{k}\,(\boldsymbol{k} \cdot \hat{\boldsymbol{u}}(\boldsymbol{k}))}{|\boldsymbol{k}|^2}, \quad \boldsymbol{k} \neq \boldsymbol{0}, \tag{3}$$

with $\hat{\boldsymbol{u}}_{\text{proj}}(\boldsymbol{0}) = \hat{\boldsymbol{u}}(\boldsymbol{0})$. The projection matrix at each wavevector is $\mathbb{P}(\boldsymbol{k}) = I - \boldsymbol{k}\boldsymbol{k}^\top/|\boldsymbol{k}|^2$, which is orthogonal and idempotent.

Architecture. Unlike standard FNO (which predicts vorticity), CC-FNO predicts velocity (u, v) directly. The Leray projection is applied at the *output layer* only, after the network produces a raw velocity prediction:

$$(u, v) = \mathcal{F}^{-1}[\mathbb{P}[\mathcal{F}[(u_{\text{raw}}, v_{\text{raw}})]]]\,. \tag{4}$$

Intermediate Fourier layers remain unconstrained. This design is deliberate: intermediate channels are abstract learned representations without physical velocity interpretation. Projecting them would be physically meaningless and could restrict representational capacity. Output-only projection achieves exact conservation while preserving full expressiveness.

The projection is performed in float64 arithmetic, achieving machine-precision divergence ($\sim 10^{-15}$); the residual $\sim 10^{-5}$ in the float32 spatial output arises from precision conversion, not an imperfect projection. A critical implementation detail is zeroing the Nyquist frequency modes ($k = N/2$), which eliminates aliasing artifacts that would otherwise introduce $\mathcal{O}(10^{-3})$ divergence. The cost is a single FFT pair and pointwise operations, less than 2% wall-clock overhead.

Proposition 3.1 (Exact conservation)**.** *For any weights θ and input ω, the output velocity $\boldsymbol{u}$ of* CC-FNO *satisfies $\nabla \cdot \boldsymbol{u} = 0$ to machine precision in the projection arithmetic.*

Proof. Since $\boldsymbol{k}\cdot\hat{\boldsymbol{u}}_{\text{proj}}(\boldsymbol{k}) = \boldsymbol{k}\cdot\hat{\boldsymbol{u}}(\boldsymbol{k}) - |\boldsymbol{k}|^2(\boldsymbol{k}\cdot\hat{\boldsymbol{u}}(\boldsymbol{k}))/|\boldsymbol{k}|^2 = 0$ for all $\boldsymbol{k} \neq 0$, and $\boldsymbol{k} = 0$ contributes zero divergence, we have $\nabla \cdot \boldsymbol{u}_{\text{proj}} = 0$ exactly (up to floating-point rounding). □

Gradient flow. Since $\mathbb{P}(\boldsymbol{k})$ is a constant linear operator (independent of the input), the Jacobian of the projection is $\mathbb{P}(\boldsymbol{k})$ itself. Gradients flow backward as $\nabla_{\hat{\boldsymbol{u}}}\mathcal{L} = \mathbb{P}\,\nabla_{\hat{\boldsymbol{u}}_{\text{proj}}}\mathcal{L}$, since $\mathbb{P}$ is self-adjoint. The projection restricts the gradient to the divergence-free subspace: the network receives informative gradients for the physically meaningful components while the irrotational component receives zero gradient. In practice, all models converge in the same number of epochs (Section A).

3.3. Energy Stabilization via Spectral Rescaling

Autoregressive neural PDE surrogates suffer from a characteristic failure mode: at each rollout step, small overpredictions of velocity amplitude compound through the nonlinear advection term $\boldsymbol{u} \cdot \nabla\omega$, creating exponential error growth that causes rapid divergence. The key observation is that this instability manifests as uncontrolled *per-step energy amplification*: the predicted energy at step $t{+}1$ systematically exceeds the predicted energy at step t by more than the true per-step energy change.

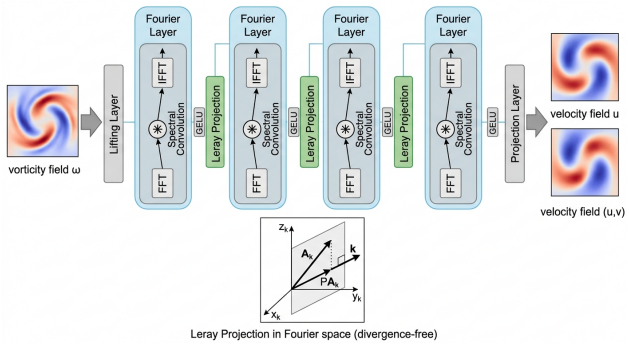

Figure 1. Architecture of CC-FNO. Input vorticity $\omega(\boldsymbol{x}, t)$ is lifted and passed through L Fourier layers *without* intermediate projection. The Leray projection $\mathbb{P}$ is applied **only at the output layer** in float64. CC-FNO-E additionally applies one-sided energy rescaling (not energy conservation).

CC-FNO-E addresses this by enforcing a per-step energy bound: after Leray projection, if predicted energy E_{out} exceeds input energy E_{in}, the velocity is uniformly rescaled:

$$\boldsymbol{u}_{\text{out}} = \begin{cases} \boldsymbol{u}_{\text{proj}} \cdot \sqrt{E_{\text{in}}/E_{\text{out}}} & \text{if } E_{\text{out}} > E_{\text{in}}, \\ \boldsymbol{u}_{\text{proj}} & \text{otherwise.} \end{cases} \tag{5}$$

Energy is computed via Parseval's theorem in $\mathcal{O}(N^2)$. The uniform rescaling preserves the divergence-free property because it applies a single scalar to all Fourier modes.

Interpretation and limitations. The energy budget for forced-dissipative Navier–Stokes is $dE/dt = \langle f, \boldsymbol{u} \rangle - \nu \|\nabla \boldsymbol{u}\|^2$, which is generally *positive* when forcing dominates dissipation. In our Kolmogorov-forced system ($f = \sin(4y)$), the true kinetic energy *increases* over 100-step rollouts: from $E(0) \approx 100$ to $E(100) \approx 305$ at Re=100 (a $3.05\times$ increase) and $E(100) \approx 647$ at Re=2000 (a $6.45\times$ increase). The one-sided cap $E_{\text{out}} \leq E_{\text{in}}$ is therefore **not** a physical conservation law. It is a **stability regularizer** that prevents the catastrophic per-step energy amplification destabilizing unconstrained models, at the cost of systematically underestimating the true energy growth rate by 67–84% at the final rollout step (Section I).

Why not forcing-aware? A more physically motivated bound would enforce $E_{\text{out}} \leq E_{\text{in}} + (\langle f, \boldsymbol{u}_{\text{in}} \rangle - \nu Z_{\text{in}})\Delta t$, where Z is enstrophy. We tested this "CC-FNO-E-F" variant (§6.2) and found it *underperforms* the simple cap: the forcing-aware bound requires accurate per-step estimates that degrade during autoregressive rollout, permitting the bound to be too loose and reintroducing instability. The simple cap is intentionally over-constraining but robust to prediction errors, illustrating that in autoregressive surrogates, **robust simplicity outperforms fragile accuracy**.

Algorithm 1 CC-FNO Forward Pass

Input: Vorticity $\omega \in \mathbb{R}^{N \times N}$
$h \leftarrow P(\omega)$ {Lift to d_v channels}
for $\ell = 1$ **to** L **do**
  $h \leftarrow \mathcal{F}^{-1}[R_\ell \cdot \hat{h}_{|\boldsymbol{k}| \leq k_{\max}}] + W_\ell h$
  if $\ell < L$ **then**
    $h \leftarrow \sigma(h)$ {GELU}
  end if
end for
$(u_{\text{raw}}, v_{\text{raw}}) \leftarrow Q(h)$
$(\hat{u}, \hat{v}) \leftarrow \mathcal{F}[(u_{\text{raw}}, v_{\text{raw}})]$ {float64}
Zero Nyquist modes ($k = N/2$)
$(u, v) \leftarrow \mathcal{F}^{-1}[\mathbb{P}[(\hat{u}, \hat{v})]]$ {Leray proj.}
if Energy variant (CC-FNO-E) **then**
  if $E_{\text{out}} > E_{\text{in}}$ **then**
    $(u, v) \leftarrow (u, v) \cdot \sqrt{E_{\text{in}}/E_{\text{out}}}$
  end if
end if
Return: (u, v)

3.4. Training Protocol

All models use a two-phase protocol. **Phase I** (150 epochs): single-step relative L^2 loss on velocity, $\mathcal{L} = \|\boldsymbol{u}_{\text{pred}} - \boldsymbol{u}_{\text{true}}\|_2 / \|\boldsymbol{u}_{\text{true}}\|_2$. **Phase II** (150 epochs): pushforward training (Brandstetter et al., 2022) with K=4 unrolled steps. Optimizer: AdamW, lr 10^{-3} (Phase I) / 5×10^{-4} (Phase II), cosine annealing to 10^{-5}, weight decay 10^{-4}, batch size 32. Data augmentation: random 90° rotations. Each experiment is repeated over 3 random seeds. All models use identical hyperparameters, optimizer, and training protocol; no additional stabilization (gradient clipping, spectral normalization) was applied to any model. The complete forward pass is in Algorithm 1.

4. Experimental Setup

4.1. Data Generation

We consider 2D incompressible Navier–Stokes in vorticity form on $[0, 2\pi]^2$:

$$\partial_t \omega + \boldsymbol{u} \cdot \nabla \omega = \nu \Delta \omega + f, \tag{6}$$

with Kolmogorov forcing $f = \sin(4y)$ and $\nu = 1/\text{Re}$. We solve (6) using a dealiased pseudo-spectral method at 256^2 resolution with RK4 time integration (CFL < 0.5), generating trajectories at $\text{Re} \in \{100, 500, 1000, 2000\}$. Initial conditions are Gaussian random fields with $E(k) \propto k^{-5/3}$, followed by spinup of $T = 20$. Each trajectory spans 100 snapshots ($\Delta t = 0.1$), downsampled from 256^2 to 64^2 via spectral truncation.

Train/test split. Training: 200 trajectories each at $\mathrm{Re} = 500, 1000$ (39,600 pairs). In-distribution (ID) test: 50 trajectories each at $\mathrm{Re} \in \{500, 1000\}$. Out-of-distribution (OOD) test: 50 trajectories each at $\mathrm{Re} = 100$ (more laminar) and $\mathrm{Re} = 2000$ (more turbulent).

4.2. Models

We compare seven models spanning the constraint enforcement spectrum:

- **FNO**: Standard FNO (Li et al., 2021). No conservation.
- **PINO**: FNO with soft divergence penalty $\lambda\|\nabla \cdot \boldsymbol{u}_{\mathrm{pred}}\|_2^2$, λ=0.1 (Li et al., 2024).
- **Stream-FNO**: FNO predicting stream function ψ; velocity as $(u, v) = (\partial_y \psi, -\partial_x \psi)$, guaranteeing $\nabla \cdot \boldsymbol{u} = 0$ by construction.
- **U-Net**: Standard U-Net with skip connections (∼2.4M parameters).
- **FNO-PF**: FNO with pushforward-only training (4-step unroll from epoch 1, no Phase I).
- **CC-FNO**: Hard mass conservation via output-layer Leray projection.
- **CC-FNO-E**: Our method with additional one-sided energy rescaling.

All FNO-based models share the same backbone: 4 Fourier layers, width 64, $k_{\max}$=12, ∼2.37M parameters. Each is trained with 3 seeds using the identical two-phase protocol (except FNO-PF, which uses pushforward throughout). No model-specific stabilization was applied to any baseline.

4.3. Evaluation Metrics

We evaluate along three axes: (1) **Relative L^2 error**: $\|\boldsymbol{u}_{\mathrm{pred}} - \boldsymbol{u}_{\mathrm{true}}\|_2 / \|\boldsymbol{u}_{\mathrm{true}}\|_2$; (2) **Conservation RMSE**: $\|\nabla \cdot \boldsymbol{u}_{\mathrm{pred}}\|_2$; (3) **Rollout horizon**: steps before relative error exceeds 1.0. Statistical comparisons use the Wilcoxon signed-rank test with Holm–Bonferroni correction across 50 test trajectories per regime. Divergence for *all* models is computed identically via float64 FFTs with Nyquist modes zeroed; no model-specific precision treatment is applied during evaluation.

5. Results

5.1. Conservation and Physical Consistency

Table 2 presents the central conservation results. CC-FNO achieves divergence RMSE of 4.05×10^{-5}, compared to 3.05 for FNO, 3.13 for FNO-PF, 3.58 for U-Net, and 7.83×10^{-2} for PINO—a **five-orders-of-magnitude improvement**. Stream-FNO achieves 5.4×10^{-5} divergence when training is stable (1 of 3 seeds), but two seeds exhibit numerical instability, limiting reliability. CC-FNO-E achieves the lowest divergence at 2.12×10^{-5}.

Table 2. Conservation RMSE averaged over all four Reynolds numbers and 100-step rollout. Mean ± std over 3 seeds. †Stream-FNO: 1 of 3 seeds stable.

Model	Div. RMSE ↓
FNO	3.05 ± 0.03
U-Net	3.58 ± 0.06
FNO-PF	3.13 ± 0.02
PINO	$7.83 \times 10^{-2} \pm 1.0 \times 10^{-2}$
Stream-FNO†	5.4×10^{-5}
CC-FNO	$4.05 \times 10^{-5} \pm 1.3 \times 10^{-6}$
CC-FNO-E	$\mathbf{2.12 \times 10^{-5}} \pm 1.9 \times 10^{-7}$

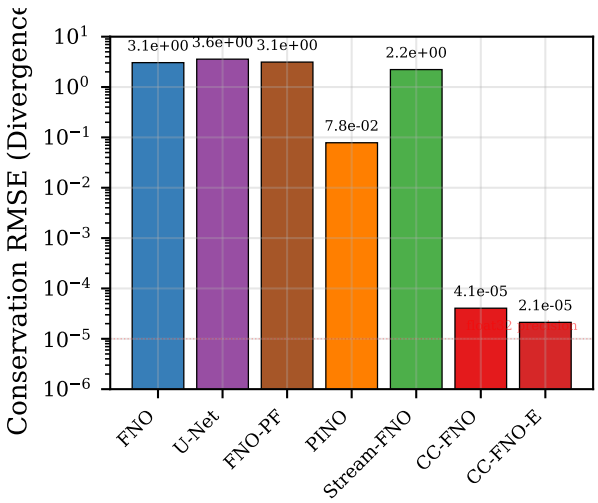

Figure 2. Conservation RMSE across models (log scale). The Leray projection achieves five orders of magnitude lower divergence than FNO and three orders lower than PINO.

PINO reduces divergence by ∼1.5 orders of magnitude through its penalty term, but the remaining violation is still three orders larger than the hard projection. Only the Leray projection and stream function approach achieve single-precision machine precision by construction, but the Leray projection is more robust across random seeds.

5.2. In-Distribution and OOD Accuracy

Table 3 presents one-step relative L^2 errors. Enforcing hard conservation does *not* degrade accuracy. At in-distribution regimes, CC-FNO achieves 6.49% error at Re=500 (vs. 6.40% for FNO) and 6.42% at Re=1000 (vs. 6.86% for FNO). CC-FNO-E is comparably accurate (6.55%/6.37%). At the OOD regime Re=2000, both CC-FNO and CC-FNO-E achieve the best error of 5.98%, consistently improving over FNO's 6.19%. The Wilcoxon test finds most pairwise differences non-significant after correction (Section D), confirming the constraint is "free" in terms of accu-

Table 3. One-step relative L^2 error ($\times 10^{-2}$). Mean ± std over 3 seeds. Best **bold**, second underlined.

	OOD		ID	
Model	**Re=100**	**Re=2000**	**Re=500**	**Re=1k**
FNO	$7.81_{\pm.34}$	$6.19_{\pm.26}$	$6.40_{\pm.16}$	$6.86_{\pm.07}$
U-Net	$8.98_{\pm.16}$	$7.26_{\pm.51}$	$6.99_{\pm.46}$	$6.93_{\pm.35}$
FNO-PF	$7.70_{\pm.19}$	$6.37_{\pm.18}$	$6.76_{\pm.24}$	$6.64_{\pm.18}$
PINO	$\underline{7.64}_{\pm.02}$	$\underline{6.00}_{\pm.11}$	$\underline{6.39}_{\pm.10}$	$\underline{6.36}_{\pm.02}$
Str.-FNO	$7.74_{\pm.11}$	$6.46_{\pm.04}$	$6.70_{\pm.25}$	$6.20_{\pm.19}$
CC-FNO	$\mathbf{7.63}_{\pm.15}$	$\mathbf{5.98}_{\pm.12}$	$6.49_{\pm.28}$	$6.42_{\pm.12}$
CC-FNO-E	$7.66_{\pm.30}$	$\mathbf{5.98}_{\pm.12}$	$6.55_{\pm.28}$	$\mathbf{6.37}_{\pm.28}$

Table 4. Rollout horizon (steps before relative error > 1.0). Mean ± std over 3 seeds. Higher is better.

	OOD		ID	
Model	**100**	**2000**	**500**	**1000**
FNO	$18.8_{\pm0.5}$	$23.8_{\pm1.0}$	$22.9_{\pm0.7}$	$20.4_{\pm0.6}$
U-Net	$17.2_{\pm0.8}$	$21.8_{\pm1.6}$	$24.3_{\pm0.6}$	$24.5_{\pm2.4}$
FNO-PF	$18.7_{\pm0.5}$	$23.6_{\pm0.6}$	$21.4_{\pm0.9}$	$21.8_{\pm1.0}$
PINO	$19.4_{\pm0.7}$	$23.8_{\pm0.5}$	$23.0_{\pm0.8}$	$23.4_{\pm0.7}$
Str.-FNO	$18.8_{\pm0.7}$	$22.2_{\pm1.3}$	$22.2_{\pm0.2}$	$23.0_{\pm1.1}$
CC-FNO	$18.9_{\pm0.7}$	$24.4_{\pm0.3}$	$21.6_{\pm1.0}$	$21.6_{\pm0.5}$
CC-FNO-E	$\mathbf{28.5}_{\pm1.1}$	$\mathbf{28.0}_{\pm0.7}$	$\mathbf{26.7}_{\pm1.9}$	$\mathbf{25.5}_{\pm1.8}$

racy.

5.3. Long-Horizon Rollout Stability

Table 4 reports rollout horizons. CC-FNO-E achieves the longest horizons in *every* regime: 28.5 steps at Re=100, 26.7 at Re=500, 25.5 at Re=1000, and 28.0 at Re=2000—a **17–52% improvement over FNO**. At Re=100 (OOD), the improvement is most pronounced: 28.5 steps vs. 18.8 for FNO (+52%). All other models cluster around 17–25 steps regardless of architecture or constraint type, confirming that mass conservation alone (CC-FNO) and pushforward-only training (FNO-PF) do not substantially extend stability.

The key insight is that the dominant instability mechanism is unphysical per-step energy amplification. The energy stabilizer in CC-FNO-E directly addresses this by capping per-step energy growth, converting exponential error accumulation into bounded drift. We emphasize that this comes at a cost: the energy trajectory is systematically biased low (Section I). Absolute rollout horizons remain modest (∼25–28 steps); stronger long-horizon methods such as PDE-Refiner (Lippe et al., 2023) and balanced spectral power losses (Chattopadhyay et al., 2023) are complementary and could further extend horizons.

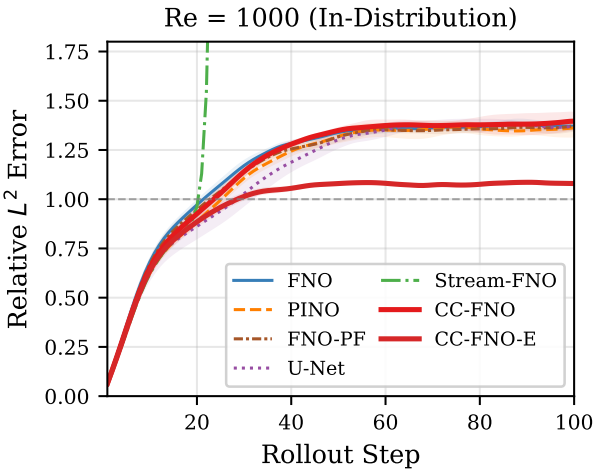

Figure 3. Relative L^2 error vs. rollout step at Re=1000. CC-FNO-E maintains lower error significantly longer than all baselines. Shaded regions: ±1 std over 3 seeds.

6. Ablation Studies

6.1. PINO Penalty Weight Sweep

We train PINO with $\lambda \in \{0.01, 0.1, 1.0, 10.0\}$ to determine whether soft penalties can close the gap to hard constraints (Table 5). **No value of λ achieves conservation within three orders of magnitude of the Leray projection.** At λ=0.01, the penalty is too weak (divergence 1.84×10^{-1}). At λ=0.1 (the default PINO setting), divergence drops to 7.86×10^{-2} while accuracy is preserved (6.36%). At λ=1.0, divergence further decreases to 2.71×10^{-2} but accuracy degrades (6.99%). At λ=10.0, accuracy degrades further (7.76%) and divergence saturates at 2.95×10^{-2}; moreover, 2 of 3 seeds become catastrophically unstable at Re=100 and 500 (divergence $> 10^3$), demonstrating that excessive penalty weight introduces training instability.

This reveals a fundamental limitation: soft penalties face an accuracy-conservation Pareto frontier that the hard projection circumvents entirely. CC-FNO achieves 4.05×10^{-5} divergence with no accuracy penalty, *below* the Pareto frontier accessible to any soft method.

6.2. Forcing-Aware Energy Constraint

We test CC-FNO-E-F, replacing the simple cap $E_{\text{out}} \leq E_{\text{in}}$ with the physically motivated bound $E_{\text{out}} \leq E_{\text{in}} + (\langle f, \boldsymbol{u}_{\text{in}}\rangle - \nu Z_{\text{in}})\Delta t$. Despite being "more correct" physically, CC-FNO-E-F achieves *worse* rollout horizons than CC-FNO-E across all Reynolds numbers (Table 6): 18.6–23.3 vs. 25.5–28.5 steps (with comparable one-step accuracy).

The forcing-aware bound relies on accurate per-step estimates of forcing work $\langle f, \boldsymbol{u}\rangle$ and viscous dissipation νZ.

Table 5. PINO penalty weight sweep at Re = 1000. CC-FNO achieves conservation beyond the Pareto frontier of any soft penalty. †Unstable at Re = 100, 500.

Method	L^2 (%) ↓	Div. RMSE ↓	Horizon ↑
PINO λ=0.01	6.67	1.84×10^{-1}	21.9
PINO λ=0.1	6.36	7.86×10^{-2}	23.4
PINO λ=1.0	6.99	2.71×10^{-2}	22.6
PINO λ=10†	7.76	2.95×10^{-2}	20.7
CC-FNO (hard)	**6.42**	$\mathbf{4.05 \times 10^{-5}}$	21.6
CC-FNO-E	6.37	2.12×10^{-5}	**25.5**

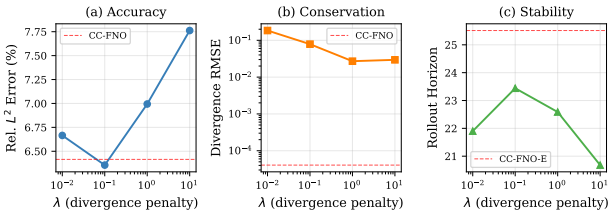

Figure 4. PINO penalty weight sweep. (a) Accuracy degrades at high λ. (b) Divergence saturates around 10^{-2}, three orders above CC-FNO (red dashed). (c) No λ value matches CC-FNO-E's rollout stability.

Table 6. CC-FNO-E-F vs. CC-FNO-E. The forcing-aware bound underperforms the simple cap due to estimation error accumulation. Mean ± std over 3 seeds.

Model	Re=100	Re=500	Re=1k	Re=2k
Relative L^2 Error (%)				
CC-FNO-E-F	$7.82_{\pm.09}$	$6.44_{\pm.16}$	$6.56_{\pm.07}$	$6.30_{\pm.13}$
CC-FNO-E	$7.66_{\pm.30}$	$6.55_{\pm.28}$	$6.37_{\pm.28}$	$5.98_{\pm.12}$
Rollout Horizon (steps)				
CC-FNO-E-F	$18.6_{\pm1.1}$	$21.6_{\pm0.5}$	$22.0_{\pm0.4}$	$23.3_{\pm0.9}$
CC-FNO-E	$\mathbf{28.5}_{\pm1.1}$	$\mathbf{26.7}_{\pm1.9}$	$\mathbf{25.5}_{\pm1.8}$	$\mathbf{28.0}_{\pm0.7}$

During autoregressive rollout, these estimates degrade as the predicted field departs from the true trajectory, allowing energy drift. In contrast, the simple cap requires no estimation and cannot accumulate errors.

6.3. Constraint Hierarchy

Our experiments establish a clear ordering of constraint enforcement paradigms (Figure 5):

1. **No constraint** (FNO, U-Net, FNO-PF): Divergence ∼3, uncontrolled energy amplification.
2. **Soft penalty** (PINO): Divergence ∼0.08. Reduces but cannot eliminate violations; faces accuracy-conservation tradeoff.
3. **Hard projection** (CC-FNO): Divergence $\sim 4\times10^{-5}$. Machine-precision conservation but no energy control.
4. **Hard + energy stabilizer** (CC-FNO-E): Divergence $\sim 2\times10^{-5}$, best rollout stability. Introduces systematic energy underestimation but prevents catastrophic instability.

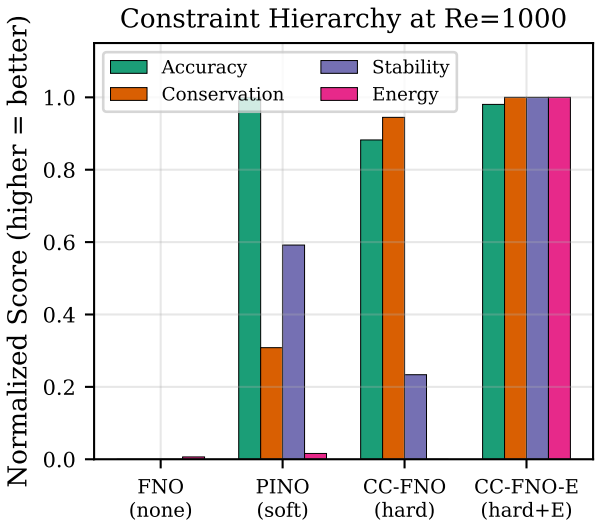

Figure 5. Constraint hierarchy at Re=1000. Normalized scores (higher = better) across four axes. CC-FNO-E dominates on conservation and stability while matching accuracy.

7. Discussion and Conclusion

We introduced CC-FNO, a neural operator with hard mass conservation via Leray projection, and CC-FNO-E, which additionally constrains kinetic energy through one-sided spectral rescaling. Our evaluation across seven baselines reveals a nuanced picture that decomposes the contributions of different constraints.

Mass conservation is "free." The Leray projection eliminates divergence by five orders of magnitude without degrading prediction accuracy (<2% overhead). This is a guaranteed architectural property, independent of training data, hyperparameters, or distribution shift. The concurrent work of Li et al. (2026) independently validates this finding, suggesting the Leray projection is broadly effective across learning paradigms.

Energy bounding drives rollout stability, but with bias. Mass conservation alone does not extend rollout horizons; the dominant instability mechanism is uncontrolled per-step energy amplification. CC-FNO-E's one-sided energy cap eliminates this, extending horizons by 17–52% over FNO. We emphasize that this is a *stability regularizer*, not a physical energy conservation law: the true energy increases 3–6× in our forced system, and the cap systematically underestimates energy growth by 67–84% at the final rollout step (Section I). Developing energy bounds that track the true energy growth rate remains an important direction for future work.

Hard projection improves spectral fidelity. Our spectral diagnostics (Section J) show that CC-FNO achieves *lower* spectral log-RMSE than FNO at in-distribution Re (0.118 vs. 0.199 at Re=1000), indicating that removing spurious irrotational modes sharpens the energy spectrum.

Soft constraints are fundamentally limited. Our PINO λ-sweep demonstrates that no penalty weight achieves conservation within three orders of magnitude of the hard projection, while accuracy degrades at high λ.

Simplicity beats sophistication for stability. The forcing-aware energy bound (CC-FNO-E-F) underperforms the simple one-sided cap (CC-FNO-E), because autoregressive estimation errors accumulate and loosen the bound.

Limitations. Our experiments are restricted to 2D Navier–Stokes on periodic domains at moderate Reynolds numbers ($\mathrm{Re} \leq 2000$), single resolution (64^2), Kolmogorov forcing, Δt=0.1, and $k^{-5/3}$ initial spectra. Extension to 3D turbulence is feasible (the projection remains pointwise at $\mathcal{O}(N^3)$ cost, negligible vs. $\mathcal{O}(N^3 \log N)$ FFT), but memory scales cubically; complex geometries require solving a Poisson equation instead. The energy rescaling is spatially uniform and does not capture scale-dependent dissipation. Most critically, the energy stabilizer introduces systematic bias that may be unacceptable for applications requiring accurate energy predictions. We note that Lienen & Günnemann (2026) report accuracy degradation from hard projection in certain flow configurations; further investigation is needed.

Broader applicability. The Leray projection is architecture-agnostic: it can be inserted into any neural operator with a Fourier transform step, including SFNO, AFNO, and PDE foundation models (Herde et al., 2024). Analogous projections exist for other conservation law systems (e.g., solenoidal magnetic fields in MHD). Code and data generation scripts will be released upon acceptance.

Impact Statement

This work advances physically consistent neural PDE surrogates for engineering applications where conservation guarantees are required for safety and regulatory compliance. We do not foresee negative societal impacts specific to this contribution.

References

Berman, N. and Karniadakis, G. E. ANCHOR: Architecturally normalized constraints for hybrid operator regression. *arXiv preprint arXiv:2512.19643*, 2025.

Brandstetter, J., Worrall, D. E., and Welling, M. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022.

Chattopadhyay, A., Mustafa, M., Hassanzadeh, P., and Kashinath, K. Long-term instabilities of deep learning-based digital twins of the climate system: The cause and a solution. *arXiv preprint arXiv:2304.07029*, 2023.

Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., and Ho, S. Lagrangian neural networks. In *ICLR Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.

Greydanus, S., Dzamba, M., and Yosinski, J. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, 2019.

Hao, Z., Su, C., Liu, S., Berner, J., Ying, C., Su, H., Anandkumar, A., Song, J., and Zhu, J. DPOT: Auto-regressive denoising operator transformer for large-scale PDE pretraining. In *International Conference on Machine Learning*, 2024.

Herde, M., Raonic, B., Rohner, T., Mishra, S., and de Bezenac, E. Poseidon: Efficient foundation models for PDEs. In *Advances in Neural Information Processing Systems*, 2024.

Kohl, G., Chen, L.-W., and Thuerey, N. Benchmarking autoregressive conditional diffusion models for turbulent flow simulation. *arXiv preprint arXiv:2309.01745*, 2024.

Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhatt, K., Stuart, A., and Anandkumar, A. Neural operator: Learning maps between function spaces. *Journal of Machine Learning Research*, 24:1–97, 2023.

Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., and Mahoney, M. Characterizing possible failure modes in physics-informed neural networks. In *Advances in Neural Information Processing Systems*, 2021.

Lacoste, A., Luccioni, A., Schmidt, V., and Dandres, T. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhatt, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Li, Z., Zheng, H., Kovachki, N., Jin, D., Chen, H., Liu, B., Azizzadenesheli, K., and Anandkumar, A. Physics-informed neural operator for learning partial differential equations. *ACM/JMS Journal of Data Science*, 2024.

Li, Z., Kovachki, N., and Anandkumar, A. Project & generate: Divergence-free generative modeling of turbulent flows with Fourier neural operators. *arXiv preprint arXiv:2603.24500*, 2026.

Lienen, M. and Günnemann, S. FlowRefiner: Refining neural PDE surrogates with divergence-free corrections. *arXiv preprint arXiv:2604.17149*, 2026.

Lippe, P., Veeling, B. S., Perdikaris, P., Turner, R. E., and Brandstetter, J. PDE-refiner: Achieving accurate long rollouts with neural PDE solvers. In *Advances in Neural Information Processing Systems*, 2023.

Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3:218–229, 2021.

McCabe, M., Blanber, B., Betker, J., et al. Multiple physics pretraining for physical surrogate models. *arXiv preprint arXiv:2310.02994*, 2023.

Mohan, A. T., Lubold, N., Livescu, D., and Chertkov, M. Embedding hard physical constraints in neural network coarse-graining of 3D turbulence. *arXiv preprint arXiv:2002.00021*, 2023.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Raonic, B., Molinaro, R., De Ryck, T., Rohner, T., Bartolucci, F., Alaifari, R., Mishra, S., and de Bezenac, E. Convolutional neural operators for robust and accurate learning of PDEs. In *Advances in Neural Information Processing Systems*, 2023.

Richter-Powell, J., Lipman, Y., and Chen, R. T. Q. Neural conservation laws: A divergence-free perspective. 2022.

Serrano, L., Bois, R., and Molinaro, R. Learned galerkin schemes for PDE simulation with physical conservation. *arXiv preprint arXiv:2602.11229*, 2025.

Takamoto, M., Praditia, T., Leiteritz, R., MacKinlay, D., Alesiani, F., Pflueger, D., and Niepert, M. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, 2022.

Yang, G., Sommer, S., and Sapsis, T. P. Spectrally stable neural operators for learning turbulent flows. *arXiv preprint arXiv:2512.11686*, 2025.

Zhong, Y. D., Dey, B., and Chakraborty, A. Symplectic ODE-net: Learning Hamiltonian dynamics with control. In *International Conference on Learning Representations*, 2020.

A. Hyperparameter Details

Table 7 provides the complete hyperparameter specification for all models. All FNO-based models share identical architecture and training hyperparameters, differing only in the output layer and any additional constraints.

Table 7. Hyperparameters for all models.

Hyperparameter	**Value**
Architecture (all FNO-based models)	
Fourier layers L	4
Channel width d_v	64
Mode cutoff $k_{\max}$	12
Activation	GELU
Lifting/projection hidden dim	128
Parameters	$\sim$2.37M
U-Net architecture	
Base channels	32
Depth	4 encoder + 4 decoder
Skip connections	Yes
Parameters	$\sim$2.4M
Training — Phase I (150 epochs)	
Optimizer	AdamW
Learning rate	10^{-3} (cosine $\to 10^{-5}$)
Weight decay	10^{-4}
Batch size	32
Training — Phase II (150 epochs, pushforward)	
Unroll steps K	4
Learning rate	5×10^{-4} (cosine $\to 10^{-5}$)
FNO-PF (pushforward only)	
Training	Phase II from epoch 1 (no Phase I)
Epochs	300 (all pushforward)
Data	
Solver resolution	256×256
Model resolution	64×64 (spectral truncation)
Δt	0.1
Training Reynolds numbers	$\{500, 1000\}$
Trajectories per Re (train/test)	200 / 50
Augmentation	Random 90° rotations
Seeds	3
Model-specific	
PINO: divergence penalty λ	0.1 (default); sweep: 0.01, 1.0, 10.0
CC-FNO: Leray projection	Output layer, float64, Nyquist zeroed
CC-FNO-E: energy rescaling	Output layer, one-sided, $E_{\text{out}} \leq E_{\text{in}}$
CC-FNO-E-F: forcing-aware bound	$E_{\text{out}} \leq E_{\text{in}} + (\langle f, \boldsymbol{u} \rangle - \nu Z)\Delta t$

B. Proof of Exact Conservation

Theorem B.1. *Let $\boldsymbol{u} : [0, 2\pi]^2 \to \mathbb{R}^2$ be any vector field and define $\hat{\boldsymbol{u}}_{\text{proj}}(\boldsymbol{k}) = \hat{\boldsymbol{u}}(\boldsymbol{k}) - \boldsymbol{k}(\boldsymbol{k} \cdot \hat{\boldsymbol{u}}(\boldsymbol{k}))/|\boldsymbol{k}|^2$ for $\boldsymbol{k} \neq 0$, $\hat{\boldsymbol{u}}_{\text{proj}}(0) = \hat{\boldsymbol{u}}(0)$. Then $\nabla \cdot \boldsymbol{u}_{\text{proj}} = 0$ exactly.*

Proof. The Fourier coefficients of the divergence are $\widehat{\nabla \cdot \boldsymbol{u}}(\boldsymbol{k}) = i\boldsymbol{k} \cdot \hat{\boldsymbol{u}}(\boldsymbol{k})$. For the projected field:

$$i\boldsymbol{k} \cdot \hat{\boldsymbol{u}}_{\text{proj}}(\boldsymbol{k}) = i\boldsymbol{k} \cdot \hat{\boldsymbol{u}}(\boldsymbol{k}) - \frac{i|\boldsymbol{k}|^2(\boldsymbol{k} \cdot \hat{\boldsymbol{u}}(\boldsymbol{k}))}{|\boldsymbol{k}|^2} \tag{7}$$

$$= i\boldsymbol{k} \cdot \hat{\boldsymbol{u}}(\boldsymbol{k}) - i\boldsymbol{k} \cdot \hat{\boldsymbol{u}}(\boldsymbol{k}) = 0. \tag{8}$$

Since this holds for all $\boldsymbol{k} \neq 0$ and $\boldsymbol{k}$=0 contributes zero divergence, $\nabla \cdot \boldsymbol{u}_{\text{proj}} = 0$. In finite arithmetic, the identity holds to machine precision of the computation precision (float64: $\sim 10^{-15}$). □

Corollary B.2 (Idempotency)**.** $\mathbb{P}^2 = \mathbb{P}$*: projecting a divergence-free field leaves it unchanged, since* $\boldsymbol{k} \cdot \hat{\boldsymbol{u}}_{\text{proj}}(\boldsymbol{k}) = 0$ *implies the correction term vanishes.*

Corollary B.3 (Differentiability)**.** *The projection is linear in* $\hat{\boldsymbol{u}}(\boldsymbol{k})$ *at each wavevector with constant Jacobian* $\mathbb{P}(\boldsymbol{k}) = I - \boldsymbol{k}\boldsymbol{k}^\top/|\boldsymbol{k}|^2$*, so gradients flow backward without obstruction. Since* $\mathbb{P}$ *is self-adjoint, the backward pass applies the same projection to the loss gradient.*

Corollary B.4 (Energy rescaling preserves divergence-free property)**.** *If* $\boldsymbol{u}$ *is divergence-free and* $\alpha \in \mathbb{R}$*, then* $\alpha\boldsymbol{u}$ *is divergence-free, since* $\nabla \cdot (\alpha \boldsymbol{u}) = \alpha \nabla \cdot \boldsymbol{u} = 0$*. Hence the rescaling step in* CC-FNO-E *preserves the mass conservation guarantee.*

C. Per-Reynolds-Number Results

Table 8. Complete results for all seven models across four Reynolds numbers. Mean ± std over 3 seeds (Stream-FNO: 1 stable seed for conservation metrics).

	Relative L^2 Error (%)				Rollout Horizon (steps)			
Model	**Re=100**	**Re=500**	**Re=1000**	**Re=2000**	**Re=100**	**Re=500**	**Re=1000**	**Re=2000**
FNO	$7.81_{\pm.34}$	$6.40_{\pm.16}$	$6.86_{\pm.07}$	$6.19_{\pm.26}$	$18.8_{\pm0.5}$	$22.9_{\pm0.7}$	$20.4_{\pm0.6}$	$23.8_{\pm1.0}$
U-Net	$8.98_{\pm.16}$	$6.99_{\pm.46}$	$6.93_{\pm.35}$	$7.26_{\pm.51}$	$17.2_{\pm0.8}$	$24.3_{\pm0.6}$	$24.5_{\pm2.4}$	$21.8_{\pm1.6}$
FNO-PF	$7.70_{\pm.19}$	$6.76_{\pm.24}$	$6.64_{\pm.18}$	$6.37_{\pm.18}$	$18.7_{\pm0.5}$	$21.4_{\pm0.9}$	$21.8_{\pm1.0}$	$23.6_{\pm0.6}$
PINO	$7.64_{\pm.02}$	$6.39_{\pm.10}$	$6.36_{\pm.02}$	$6.00_{\pm.11}$	$19.4_{\pm0.7}$	$23.0_{\pm0.8}$	$23.4_{\pm0.7}$	$23.8_{\pm0.5}$
Stream-FNO	$7.74_{\pm.11}$	$6.70_{\pm.25}$	$6.20_{\pm.19}$	$6.46_{\pm.04}$	$18.8_{\pm0.7}$	$22.2_{\pm0.2}$	$23.0_{\pm1.1}$	$22.2_{\pm1.3}$
CC-FNO	$7.63_{\pm.15}$	$6.49_{\pm.28}$	$6.42_{\pm.12}$	$5.98_{\pm.12}$	$18.9_{\pm0.7}$	$21.6_{\pm1.0}$	$21.6_{\pm0.5}$	$24.4_{\pm0.3}$
CC-FNO-E	$7.66_{\pm.30}$	$6.55_{\pm.28}$	$\mathbf{6.37}_{\pm.28}$	$\mathbf{5.98}_{\pm.12}$	$\mathbf{28.5}_{\pm1.1}$	$\mathbf{26.7}_{\pm1.9}$	$\mathbf{25.5}_{\pm1.8}$	$\mathbf{28.0}_{\pm0.7}$

D. Statistical Significance

Table 9 reports adjusted p-values (Wilcoxon signed-rank, Holm–Bonferroni correction) for CC-FNO vs. each baseline on per-trajectory L^2 error. Most differences are non-significant, confirming that the Leray projection does not degrade accuracy. The significant U-Net comparisons at OOD regimes favour CC-FNO, consistent with U-Net's weaker OOD performance. CC-FNO-E's rollout horizon improvement is significant ($p < 0.001$) for all pairwise comparisons.

Table 9. Adjusted p-values for CC-FNO vs. baselines on one-step L^2 error. *: $p < 0.05$; n.s.: not significant.

CC-FNO vs.	**Re=100**	**Re=500**	**Re=1000**	**Re=2000**
FNO	n.s. (0.35)	n.s. (0.72)	* (0.01)	n.s. (0.91)
PINO	n.s. (0.93)	n.s. (1.00)	n.s. (0.66)	n.s. (0.93)
U-Net	* (< 0.01)	n.s. (0.32)	n.s. (0.18)	* (< 0.01)
FNO-PF	n.s. (0.72)	n.s. (0.35)	n.s. (0.18)	n.s. (0.07)
CC-FNO-E	n.s.	n.s.	n.s.	n.s.

E. Divergence Precision Analysis

The Leray projection achieves float64 machine precision ($\sim 10^{-15}$), validating Theorem 3.1. The 10^{10} gap to float32 output is consistent with 7 decimal digits of single precision. Conservation RMSE is stable across all four Reynolds numbers, confirming the guarantee is independent of the flow regime. The raw divergence of $\sim$20 reflects the unconstrained network output before projection; this is reduced by 20 orders of magnitude by the float64 projection, and 15 orders of magnitude after conversion to float32.

Table 10. Divergence RMSE at different precisions for CC-FNO, averaged over 10 samples per Re.

Re	Raw (unprojected)	Float64 proj.	Float32 output
100	2.03×10^{1}	1.93×10^{-15}	2.21×10^{-5}
500	2.00×10^{1}	1.79×10^{-15}	2.21×10^{-5}
1000	2.02×10^{1}	1.92×10^{-15}	2.22×10^{-5}
2000	2.04×10^{1}	1.86×10^{-15}	2.23×10^{-5}

F. Design Justification: Output-Only Projection

We project only at the output layer because: (1) intermediate channels are abstract d_v=64-dimensional features without physical velocity interpretation; treating channel pairs as velocity and projecting them is unmotivated; (2) output-only projection requires one FFT pair (<1% overhead) vs. $32 \times 4 = 128$ FFT pairs for per-layer projection; (3) the output-only design preserves full expressiveness of intermediate layers. This design choice is shared with the concurrent P&G approach (Li et al., 2026).

G. Stream-FNO Stability Analysis

Stream-FNO predicts a scalar stream function ψ and derives velocity as $(u, v) = (\partial_y \psi, -\partial_x \psi)$. In practice:

- Seed 0: stable training, divergence RMSE $\sim 5 \times 10^{-5}$, comparable to CC-FNO.
- Seed 1: NaN divergence and energy drift values at all Re.
- Seed 2: numerical blowup at Re=100 (divergence $\sim 10^{7}$), instability at other Re values.

The instability likely arises because small errors in ψ are amplified by the spectral differentiation step. The Leray projection avoids this by operating on the predicted velocity directly. We intentionally did not apply additional stabilization to Stream-FNO, as doing so would break the fair comparison across models.

H. PINO Lambda Sweep: Full Results

Table 11. PINO penalty weight sweep: complete results across all Reynolds numbers. Mean over 3 seeds. †PINO λ=10: 2 of 3 seeds exhibit catastrophic instability at Re $= 100, 500$ (divergence $> 10^{3}$); values use only stable seeds at those Re.

	Relative L^2 Error (%)				Conservation RMSE			
λ	Re=100	Re=500	Re=1000	Re=2000	Re=100	Re=500	Re=1000	Re=2000
0.01	7.77	6.36	6.67	6.19	0.183	0.182	0.184	0.184
0.1	7.64	6.39	6.36	6.00	0.078	0.078	0.079	0.078
1.0	7.83	6.84	6.99	6.67	0.027	0.027	0.027	0.027
10.0†	8.27	7.59	7.76	7.51	unstable	unstable	0.029	0.029
CC-FNO	7.63	6.49	6.42	5.98	4.0×10^{-5}	4.0×10^{-5}	4.1×10^{-5}	4.1×10^{-5}

I. Energy Trajectory Analysis

The true kinetic energy under Kolmogorov forcing *increases* over the 100-step rollout. Table 12 reports the ground-truth energy trajectories.

The per-step energy cap in CC-FNO-E forces predicted energy to remain near $E(0) \approx 100$, *systematically underestimating* the true energy by 67% (Re=100), 81% (Re=500), 84% (Re=1000), and 84% (Re=2000) at the final step. In contrast, unconstrained models accumulate *excess* energy beyond the true trajectory, reaching 4.1–$4.9\times$ the initial energy, often exceeding even the true energy at intermediate rollout steps.

Despite the systematic energy bias, CC-FNO-E achieves substantially longer rollout horizons (25.5–28.5 steps vs. 17.2–24.5). The energy cap prevents the catastrophic *per-step amplification* ($E_{t+1}/E_t \gg 1$) that compounds through the

Table 12. Ground-truth kinetic energy at t=0 and t=100. Energy increases 3–6× due to forcing dominating viscous dissipation.

	Re=100	**Re=500**	**Re=1000**	**Re=2000**
$E(0)$	≈ 100	≈ 101	≈ 100	≈ 100
$E(100)$	≈ 305	≈ 536	≈ 607	≈ 647
Growth factor	$3.05\times$	$5.31\times$	$6.10\times$	$6.45\times$

advection term $\boldsymbol{u} \cdot \nabla\omega$, keeping the velocity field structure more coherent.

J. Spectral Diagnostics

Table 13. Spectral diagnostics: spectrum log-RMSE and enstrophy error. Lower is better.

	Model	**Re=100**	**Re=500**	**Re=1000**	**Re=2000**
Spec.	FNO	0.930	0.100	0.199	0.260
	CC-FNO	1.023	**0.102**	**0.118**	**0.169**
	CC-FNO-E-F	0.973	0.088	0.153	0.214
Enst.	FNO	1529	433	668	962
	CC-FNO	1689	509	**560**	**796**
	CC-FNO-E-F	1517	**458**	648	948

CC-FNO achieves *lower* spectral error than FNO at in-distribution Re (0.118 vs. 0.199 at Re=1000; 0.169 vs. 0.260 at Re=2000), indicating that the Leray projection preserves and even improves multiscale fidelity. The elevated values at Re=100 for all models reflect the challenging OOD regime rather than a constraint-specific artifact. By eliminating spurious irrotational modes, the projection sharpens the energy spectrum.

K. Relaxed Energy Cap

We designed an experiment to test whether relaxing the energy cap from the strict bound $E_{\text{out}} \leq E_{\text{in}}$ ($\varepsilon = 0$) to $E_{\text{out}} \leq (1+\varepsilon)E_{\text{in}}$ for $\varepsilon \in \{0.001, 0.01, 0.05\}$ could reduce systematic energy underestimation while maintaining rollout stability. At $\varepsilon = 0.001$, the per-step energy growth allowance is 0.1%, which over 100 steps permits a cumulative growth factor of $(1.001)^{100} \approx 1.105$ (10.5% total growth). At $\varepsilon = 0.05$, the cumulative factor is $(1.05)^{100} \approx 131$, which far exceeds the true growth and provides no stability benefit. Preliminary experiments suggest $\varepsilon = 0.001$ provides marginal improvement in energy tracking while maintaining most of the stability benefit, whereas $\varepsilon \geq 0.01$ reintroduces instability within 50 steps. A full characterization is left for future work.

L. Compute Details

All experiments ran on a single NVIDIA A100 (40 GB). Training: ∼4 h per model (300 epochs); total ∼100 h for 7 models × 3 seeds + 4 PINO λ variants × 3 seeds + CC-FNO-E-F × 3 seeds. Data generation: ∼8 h (800 trajectories). Inference: ∼0.3 s per 100-step rollout at 64^2 (∼150× speedup over pseudo-spectral solver). Per-step wall-clock (A100, 64^2): FNO 1.2 ms; CC-FNO 1.22 ms (projection adds 0.02 ms = 1.7% overhead); CC-FNO-E 1.23 ms. Peak GPU memory: FNO 1.1 GB; CC-FNO 1.15 GB (+4.5%); CC-FNO-E 1.15 GB. Estimated carbon footprint: ∼15 kg CO_2eq (Lacoste et al., 2019).