جامعة نيويورك ابوظبي
NYU ABU DHABI

ICML
International Conference On Machine Learning

Fractional is Better

Learnable Derivative Orders in Neural Operator Learning

Optimal derivative orders in ML are sub-physical: $\beta^\star < m$

Fares B. Mehouachi[1] · Saif Eddin Jabari[1,2]

[1]New York University Abu Dhabi, UAE

[2]New York University, NY, USA

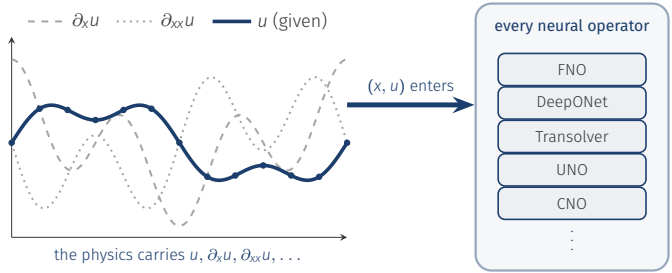
ICML 2026 - Main Track

A Universal Blind Spot

Neural operators are **fast PDE surrogates** $\mathcal{G}_\theta : u_0 \mapsto u(\cdot, T)$.

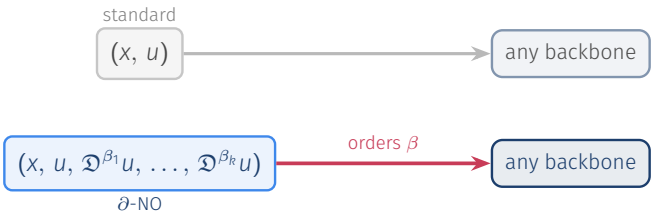
Architectures **vary widely**; the **input never does**.

Typically $(x, u(x))$ in 1D, or $(x, y, u(x, y))$ in 2D.



Derivatives: present in the PDE physics, absent from the NO input.

Can we provide the NO with the PDE derivatives?



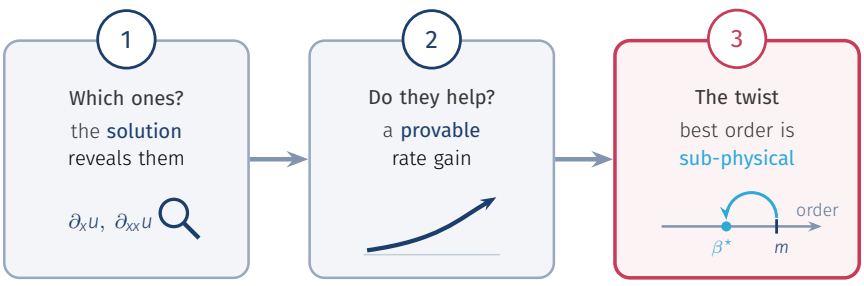
$\mathfrak{D}^{\beta}$: derivative of order β; β=1, 2 recover ∂_x, ∂_{xx}.

A backbone-agnostic plug-in: provide derivatives, make learning easier

Making It Rigorous

Three questions: does this make sense, what benefits, and which derivatives order to use?

> Example: Burgers' 1D PDE: $\partial_t u + u\,\partial_x u = \nu\,\partial_{xx} u$ depends on $\{u,\ \partial_x u,\ \partial_{xx} u,\ u\,\partial_x u\}$. The NO currently recovers *implicitly* such representation from finite, noisy samples.



Step 1 Justification: Why Derivatives?

Derivative features are not ad hoc: they appear explicitly in the solution operator.

Parabolic case: for a **semilinear evolution PDE** $\partial_t u = \mathcal{L}u + \mathcal{N}[u]$, the mild solution is $u(t) = G_t * u_0 + \int_0^t G_{t-s} * \mathcal{N}[u(s)]\, ds$.

Theorem (Derivative Emergence)

$$\mathcal{G}[u_0] = \underbrace{G_T * u_0}_{\text{smoothing of } u_0} + \underbrace{T\bar{G}_T * \mathcal{N}[u_0]}_{\text{brings derivatives of } u_0} + O(T^2)$$

where G_t: Green's function (linear semigroup), $e^{t\mathcal{L}}$ for constant coefficients · $\bar{G}_T = \frac{1}{T}\int_0^T G_s\, ds$: time-average over $[0, T]$

Beyond the parabolic case, a spectral decomposition yields the same conclusion (paper).

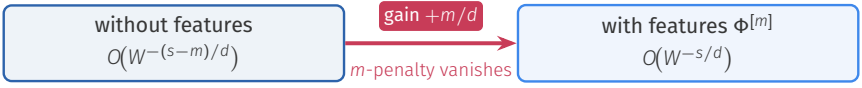
$\mathcal{N}[u_0]$ uses derivatives **up to order** $m \Longrightarrow$ **provide NO with** $\Phi^{[m]} = \{u_0,\ \partial_x u_0,\ \ldots,\ \partial_x^m u_0\}$

Step 2 Theoretical Gain: do derivatives help? Yes, provably

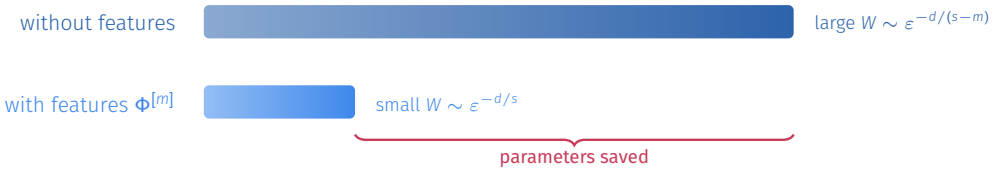
Providing ∂ to NO: improves approximation rate.

Approximation rate: How fast error decays as the network width grows, $\|\mathcal{G} - \mathcal{G}_\theta\| = O(W^{-\text{rate}})$.
A larger rate = the same accuracy with fewer parameters.

Our result: derivative features *raise* the rate



to reach the *same* accuracy ε, the width W needed:



W: width s: input regularity m: PDE order d: spatial dimension

So Far, So Expected

Two results so far, both **classic & expected**:

Result 1

Derivatives of u_0 **appear in the PDE solution**.

natural: the solution is built from them.

Result 2

Providing them **improves the approximation rate**: $W^{-(s-m)/d} \to W^{-s/d}$.

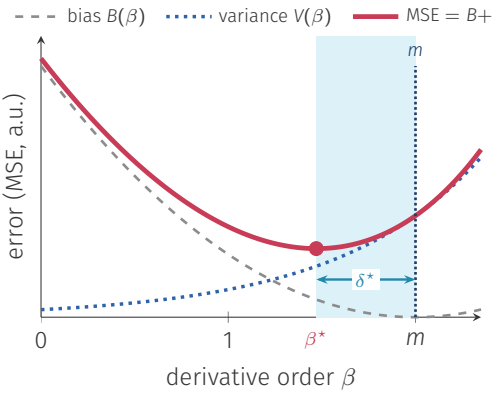
classic: aligned inputs, easier learning.

Including derivatives **makes sense**.

However, the order to use is *not* the order that appears in the PDE.

Step 3 The Twist: the Optimal Order is Sub-Physical

We expected the physics order m. The optimum is **strictly less**, from a spectral **bias / variance** tradeoff.



Theorem (Optimal Order)

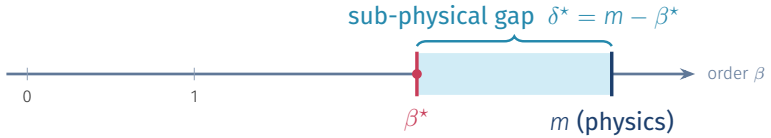
$$\beta^\star = m - \delta^\star, \quad \delta^\star > 0$$

$$\delta^\star = \frac{A_c - \ln(n/\sigma^2)}{2\ln(\xi_c\,\xi_{\max})} + O(1/\ln \xi_c)$$

m **is never optimal** for finite, noisy data: **bias** falls with β, noise **variance** ($\propto |\xi|^{2\beta}$) rises, so a **unique minimum** sits strictly below m.

n: # training samples · σ: noise level · ξ_c: spectral cutoff · $\xi_{\max}=\pi/h$: Nyquist · A_c: capacity constant

Step 3 Why the Optimum Stays Sub-Physical



Where the gap comes from. Two competing spectral effects:

- bias $B(\beta) \propto \int_{|\xi|>\xi_c} |\xi|^{2(m-\beta)} S(\xi)\, d\xi$ *falls* with β;
- variance $V(\beta) \propto \frac{\sigma^2}{n} \int_{|\xi|\le\xi_{\max}} |\xi|^{2\beta}\, d\xi$ *rises* with β.

Balancing them ($\partial_\beta \,\mathrm{MSE}{=}0$) gives $\delta^\star$, with capacity constant

$$A_c = 2s \ln \xi_c + (2m{+}d) \ln \xi_{\max}.$$

Physics is practically unreachable

$\beta^\star{=}m$ needs $n \ge n^\star = \sigma^2 e^{A_c}$.

Typical FNO: $\xi_{\max} \approx 3200$, $\xi_c \approx 100 \Rightarrow A_c \approx 68$; even at $\sigma{=}10^{-3}$, $n^\star \approx 10^{23}$ samples.

$\beta^\star$ represents the physical order m modulated with statistical effects.

From Integer to Fractional: A Continuous Knob on Order

The optimum order is not PDE order: we need **fractional** derivatives.

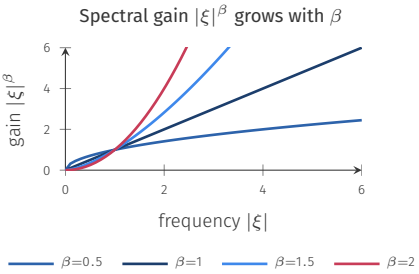
Weyl (in frequency):

$$D^{\beta}u = \mathcal{F}^{-1}\left[(i\xi)^{\beta}\,\hat{u}(\xi)\right]$$

Grünwald–Letnikov (on the grid):

$$\mathfrak{D}^{\beta}u = h^{-\beta}\sum_{j}(-1)^{j}\binom{\beta}{j}u(x-jh)$$

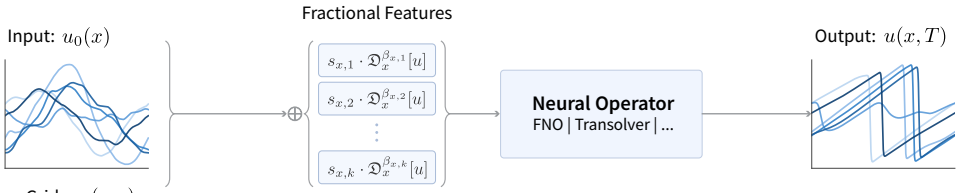
Both **differentiable in β** $\Rightarrow$ gradient-learnable;
β=1, 2 recover $\partial_x, \partial_{xx}$.



high β amplifies high frequencies $\rightarrow$ noise

Any order is reachable and gradient-learnable: we let the NO *learn* $\beta^{\star}$.

∂-NO: Derivative-Augmented Neural Operators



Replace the input (x, u) with

$$\big(x,\ u,\ s_1\mathfrak{D}^{\beta_1}u,\ \ldots,\ s_k\mathfrak{D}^{\beta_k}u\big),$$

orders β_j and scales s_j **learned jointly** with the backbone.

Architecture-agnostic:

∂-FNO, ∂-TFNO, ∂-LocalNO, ∂-Transolver, ∂-CNO.

Grünwald–Letnikov stencil; $h^{-\beta}$ dropped for stability; overhead negligible.

A drop-in augmentation: wraps any backbone, no architectural change.

∂ Improves Every Backbone, Dataset, and Noise Level

	Burgers (ν=0.1)		Burgers (ν=0.001)		KdV		Darcy		NS2D	
	smooth		*near-shock*		*dispersive*		*elliptic*		*turbulence*	
	σ=0	.05	0	.05	0	.05	0	.05	0	.05
FNO	0.065	0.250	1.138	1.632	1.598	2.255	5.643	5.590	3.479	3.575
∂-FNO	**0.055**	**0.190**	**0.837**	**1.366**	**1.549**	**2.195**	**5.426**	**5.415**	**3.428**	**3.511**
Improv.	15.1%	24.1%	26.5%	16.3%	3.0%	2.7%	3.8%	3.1%	1.5%	1.8%
TFNO	0.071	0.209	0.702	1.032	2.312	2.745	5.223	5.206	3.428	3.534
∂-TFNO	**0.062**	**0.178**	**0.676**	**0.917**	**2.227**	**2.660**	**4.843**	**4.906**	**3.228**	**3.368**
Improv.	11.9%	14.7%	3.7%	11.2%	3.7%	3.1%	7.3%	5.8%	5.8%	4.7%
LocalNO	0.063	0.236	0.768	1.272	2.587	3.625	5.511	5.534	1.994	2.620
∂-LocalNO	**0.060**	**0.199**	**0.732**	**1.203**	**2.352**	**3.441**	**5.430**	**5.487**	**1.984**	**2.533**
Improv.	6.1%	15.8%	4.7%	5.4%	9.1%	5.1%	1.5%	0.9%	0.5%	3.3%
Transolver	0.990	0.765	18.988	7.521	22.427	16.334	14.645	15.332	4.301	4.821
∂-Transolver	**0.709**	**0.736**	**4.633**	**4.049**	**17.957**	**9.588**	**13.125**	**11.212**	**4.043**	**4.341**
Improv.	28.4%	3.8%	75.6%	46.2%	19.9%	41.3%	10.4%	26.9%	6.0%	9.9%

Rel. L^2 error (%), 5 seeds. **∂ wins all 40 cells**, peak +75.6% (∂-Transolver, shock Burgers).

The Closed Form, in Experimental Terms

Substituting $\xi_c{=}2\pi k$ turns the gap into three **measurable knobs**:

$$\delta^\star = m - \beta^\star \;\propto\; \frac{2s\ln k + 2\ln\sigma - \ln n + C}{2\ln k + C'}$$

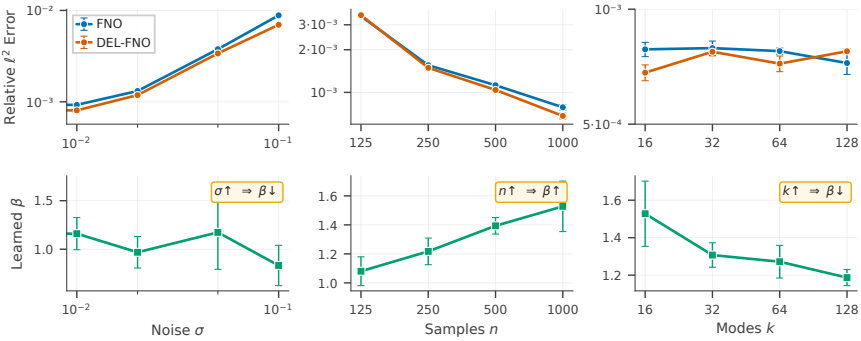
k: Fourier modes ($\xi_c{=}2\pi k$) · σ: noise · n: samples · s: input regularity · C, C': constants

noise $\sigma\uparrow \Rightarrow\ \beta^\star\downarrow$ | samples $n\uparrow \Rightarrow\ \beta^\star\uparrow$ | modes $k\uparrow \Rightarrow\ \beta^\star\downarrow$

The first two are **unconditional**. The *k*-effect depends on smoothness s: clean when $s > m + \frac{d}{2}$, and it *attenuates* near the shock boundary (low s).

Three predictions. We test them on smooth and near-shock Burgers next.

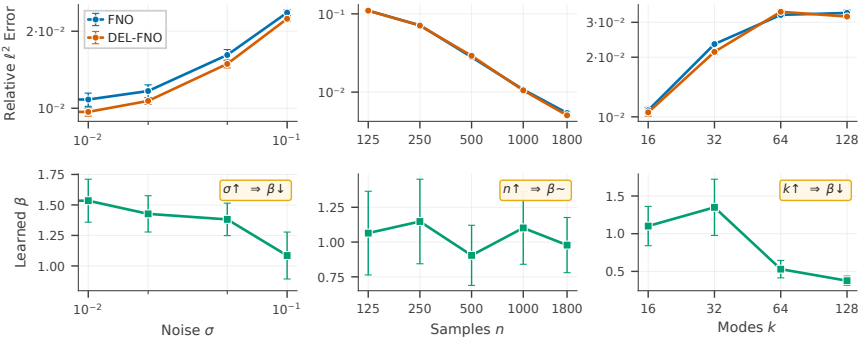
Theory Validation: $\beta^\star$ Moves as Predicted



Setup: smooth Burgers (ν=0.1, smoothness $\tau \approx 2.2$); a *single* learnable β; vary one factor at a time; 5-seed mean±std. *Top:* rel. L^2 error; *bottom:* learned $\beta^\star$.

All three predictions hold: $\sigma\uparrow \Rightarrow \beta^\star\downarrow$ · $n\uparrow \Rightarrow \beta^\star\uparrow$ · $k\uparrow \Rightarrow \beta^\star\downarrow$.

Theory Validation: Robust into the Near-Shock Regime



Setup: near-shock Burgers (ν=0.001, smoothness $\tau \approx 0.8$, near the $s > m + d/2$ boundary); same protocol as Fig. 2. *Top:* rel. L^2 error; *bottom:* learned $\beta^\star$.

The noise law $\sigma\uparrow \Rightarrow \beta^\star\downarrow$ still holds clearly; the samples/modes effects *attenuate* (low s regime).

Is the Gain Real, or Just More Capacity?

Two natural objections, before we credit the **derivative inductive bias of ∂-NO**:

"It's just more parameters."

Derivative features add inputs, hence weights.

Control: a *matched-parameter* baseline still leaves ∂ ahead (App.).

"GL is just a convolution."

A Grünwald–Letnikov derivative *is* a fixed convolution stencil, and CNO is built *entirely* from learnable convolutions.

So CNO could already learn it, if capacity were the whole story.

The **sharp test**: add ∂ to **CNO**. If it *still* helps, the gain is the **derivative inductive bias**, not capacity or generic convolution.

The Verdict: ∂-CNO Wins Anyway

Across the seven CNO benchmarks, ∂ helps *every* time, although a plain CNO could in principle learn the same stencil.

> **14/14 wins** (both L^2 & L^1): the gain is the derivative inductive bias, not added capacity.

Gains track *global differential structure*: largest on elliptic Poisson, shear NS, wave; smallest on data-dominated Darcy, Airfoil.

CNO benchmarks: rel. L^2 (%), clean

	CNO	∂-CNO	Improv.
Poisson	1.788	**1.053**	41.2%
NS Shear	9.686	**7.104**	26.7%
Wave	0.590	**0.520**	11.8%
Allen–Cahn	2.618	**1.632**	37.7%
Transp.	0.416	**0.353**	15.2%
Darcy	0.763	**0.752**	1.4%
Airfoil	1.184	**1.137**	4.0%

peak on elliptic Poisson.

Derivative features carry information convolutions don't reliably learn.

Summary

Recap

- **Gap:** operators ignore derivatives.
- **Theory:** features help, but $\beta^\star < m$.
- **Method:** ∂-NO, a backbone-agnostic plug-in.
- **Evidence:** improves every backbone tested.

Contributions

- Rates: $W^{-(s-m)/d} \to W^{-s/d}$.
- Optimal order is sub-physical (closed form): automatic spectral regularization.
- ∂-NO is $\sim$free and backbone-agnostic.
- A proven differential inductive bias, beyond convolution or added parameters.

Key takeaway: when dealing with real data (*noisy, band-limited, finite*), optimal order is sub-physical.

Thank You!

Questions?

Fares B. Mehouachi
New York University Abu Dhabi
fares.mehouachi@nyu.edu

Saif Eddin Jabari
NYU Abu Dhabi & NYU Tandon
sej7@nyu.edu

Acknowledgments: NYUAD CITIES, Tamkeen grant CG001.

∂-NO code & experiments

`github.com/FaresBMehouachi/delNO`

open source

Key References

Neural operators & backbones

- Li et al. (2021). *Fourier Neural Operator for Parametric PDEs.* ICLR.
- Lu et al. (2021). *DeepONet.* Nat. Mach. Intell.
- Wu et al. (2024). *Transolver.* ICML. Raonic et al. (2023). *Convolutional Neural Operator.* NeurIPS.
- Liu-Schiaffini et al. (2024). *Localized Integral & Differential Kernels (LocalNO).* ICML.

Derivative / physics-aware & fractional features

- Zhu et al. (2025). *Generalizing equation-aware emulators.*
- Li et al. (2024). *Physics-Informed Neural Operator (PINO).*
- Yarotsky (2017). *Error bounds for deep ReLU networks.*
- Kovachki et al. (2023). *Neural operator theory.*

Full bibliography in the paper.