LEARNING WHERE TO ADAPT: TARGET-AWARE PARAMETER-EFFICIENT FINE-TUNING FOR NEURAL OPERATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Parameter-efficient fine-tuning (PEFT) adapts pretrained neural operators to new physical regimes with a small trainable budget, but performance depends on where that budget is placed. We show that neural-operator PEFT is governed by two distinct constraints: accessibility, whether the trainable parameterization can reach useful target updates, and capacity, how much of those updates it can represent. Across controlled physical shifts, useful updates concentrate in different architectural parameter groups—the spectral kernels of a Fourier Neural Operator and the branch network of a DeepONet. Once capacity is adequate, restricting adaptation away from these updates imposes an error floor that additional rank cannot remove. We turn this observation into a cheap target-aware placement rule: a short multi-batch warm-up on target training data, scored by groupwise functional and task-loss improvement, identifies useful parameter groups and detects diffuse regimes where concentrated placement is unwarranted. On FNO, placing a standard adapter where the probe points reaches 0.93–$1.07\times$ the full-fine-tuning error using 3–12% of the trainable degrees of freedom and outperforms the tested fixed-placement and adaptive-rank PEFT baselines at matched budget on all six shifts. On DeepONet, the probe identifies branch-localized adaptation while assigning low concentration to diffuse regimes. Increasing PEFT capacity cannot compensate for useful target updates that its trainable parameterization cannot access.

1 INTRODUCTION

Neural operators are the default surrogate for partial differential equations (PDEs) in scientific machine learning (Li et al., 2021; Lu et al., 2021; Kovachki et al., 2023; Azizzadenesheli et al., 2024). The deployment pattern increasingly mirrors that of large language models: pretrain a foundation operator on abundant source data, then adapt it cheaply to each new physical regime. Parameter-efficient fine-tuning (PEFT) is the tool of choice, and within it LoRA (Hu et al., 2022) is the default, adapting a frozen weight matrix W by a low-rank update $W + BA$.

PEFT methods are usually compared on *how* they update a weight—low-rank (Hu et al., 2022), weight-decomposed (Liu et al., 2024), adaptive-rank (Zhang et al., 2023), rescaling (Liu et al., 2022)—but they share a stronger and rarely examined assumption: they fix *where* the update happens. LoRA is applied to a standard set of modules chosen once for the architecture, and that choice is carried over unchanged to every downstream task. For a language model this is reasonable, because the relevant computation is distributed. For a neural operator facing a new physical regime it need not be: a change in viscosity, in the initial-condition spectrum, or in a forcing term perturbs the operator in a specific place, and that place depends on both the architecture and the shift.

We ask a placement question: *given a pretrained neural operator and a new physical regime, where should a limited adaptation budget be spent?* We treat placement as an adaptation constraint: before spending additional PEFT capacity, the trainable parameterization must expose the parameters needed by the target update. Across controlled physical shifts on two architectures, useful updates concentrate in distinct architectural parameter groups—the spectral convolution kernels of a Fourier Neural Operator (FNO) and the branch network of a DeepONet. At small budgets, insufficient capacity dominates; once capacity is adequate, excluding target-relevant parameters creates an error

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

floor that additional rank elsewhere cannot remove. We formalize these two constraints as *accessibility*, whether the trainable parameterization can reach useful target updates, and *capacity*, how much of those updates it can represent.

This diagnosis motivates aligning PEFT with the parameter groups that carry useful target updates. We make this practical with a cheap target-aware probe. A short multi-batch warm-up on target training data produces an early update direction; scoring each parameter group by the functional and task-loss improvement attributable to that group's share of the update identifies where useful adaptation occurs. The concentration of these scores further distinguishes localized regimes, where targeted placement is beneficial, from diffuse regimes in which broad adaptation already provides access to the useful update. The probe uses only target training data and a validation split, never test labels or the full-FT solution, and costs a few percent of one fine-tuning run.

Placing a standard adapter where the probe points is effective. On FNO it selects the spectral kernels and reaches 0.93–$1.07\times$ of full-FT error at 3–12% of the trainable parameters, beating LoRA, DoRA, AdaLoRA, and $(IA)^3$ at matched budget on every tested shift. On a localized DeepONet shift it selects the branch network, and branch placement beats trunk placement at matched budget. On a diffuse DeepONet shift the probe reports low confidence and the method falls back to broad adaptation, which is correct there. The mechanism transfers to a second, chaotic PDE with a different, physics-specific signature.

Contributions.

1. **Mechanism.** We identify accessibility and capacity as distinct bottlenecks in neural-operator PEFT. At small budgets capacity dominates; once capacity is adequate, excluding target-relevant parameters imposes an error floor that additional capacity elsewhere cannot remove.
2. **Method.** We introduce a cheap target-aware placement rule based on a short multi-batch warm-up and groupwise functional and task-loss attribution. The probe identifies where useful adaptation occurs and distinguishes localized regimes that benefit from concentrated placement from diffuse regimes where broad access is sufficient.
3. **Evidence.** Across multiple FNO shifts and a second PDE, accessibility-aligned PEFT reaches 0.93–$1.07\times$ full-FT error with 3–12% trainable DOF and outperforms matched-budget LoRA, DoRA, AdaLoRA, and $(IA)^3$; DeepONet provides localized and broad-access controls.

2 Related Work

Parameter-efficient fine-tuning. LoRA (Hu et al., 2022) adapts a frozen weight with a low-rank additive update and dominates applied PEFT. Refinements change the update's structure: DoRA (Liu et al., 2024) decomposes weight into magnitude and direction, AdaLoRA (Zhang et al., 2023) allocates rank adaptively by sensitivity, VeRA (Kopiczko et al., 2024) freezes a random basis and learns only scaling, and $(IA)^3$ (Liu et al., 2022) rescales activations. Adapters (Houlsby et al., 2019) and prompt tuning (Lester et al., 2021; Li & Liang, 2021; Jia et al., 2022) modify activations instead, a family unified by He et al. (2022). These methods differ in how the budget is spent within the modules they touch, but the choice of *which* modules to touch is fixed in advance and inherited across tasks. Our work targets that choice.

PEFT for scientific operators. PEFT is only beginning to be studied for pretrained operator models. F-Adapter (Zhang et al., 2025) adapts Large Operator Models built on Fourier Neural Operators and allocates adapter capacity across frequency bands, assigning more to energy-dominant low-frequency modes; it reports state-of-the-art results on 3D Navier–Stokes. This is a *within-subspace* question: how to distribute capacity across frequencies inside the spectral layers. We study the prior *across-group* question: which parameter group should be adapted at all for a given target shift. The two are complementary, and we compare them empirically in §6.5.

Neural operators. FNO (Li et al., 2021) learns a kernel integral operator through a Fourier multiplier; DeepONet (Lu et al., 2021) uses a branch/trunk factorization; message-passing and multi-scale

variants extend the family (Brandstetter et al., 2022; Gupta & Brandstetter, 2023), and large pretrained operator models are emerging (Hao et al., 2024). Surveys (Kovachki et al., 2023; Azizzadenesheli et al., 2024) cover applications from weather (Pathak et al., 2022) to multiphase flow (Wen et al., 2022), and PDEBench (Takamoto et al., 2022) and The Well (Ohana et al., 2024) provide benchmarks. Most transfer work evaluates zero-shot or full fine-tuning; placement-resolved adaptation under controlled physical shift has not been studied.

Adaptive placement and selection in PEFT. Choosing where to adapt is not new. In language models, the Layer Card (Xu et al., 2026) places LoRA on layers selected by a projected-residual diagnostic, and PAGE (Zhang et al., 2026) places a single adapter using gradient-energy signals. Surgical fine-tuning (Lee et al., 2023) selects layer subsets for distribution shifts in vision, and PhysGuard (Zhou et al., 2026) protects Fisher-critical directions when adapting neural operators to real data. A separate family adapts *within* matrices: AdaLoRA (Zhang et al., 2023) allocates rank, EVA (Paischer et al., 2025) selects low-rank directions from activation variance, and AdaMSS (Zheng et al., 2025) selects subspaces inside each weight. We study a complementary question in neural operators: how limited adaptation capacity interacts with access to target-relevant architectural parameter groups under physical shift. Our diagnostic scores these groups by their early functional and task-loss contribution on the target data, and the resulting intervention tests whether restricted parameter access is the bottleneck at matched trainable budget.

Intrinsic dimension and adaptation subspaces. Fine-tuning many NLP tasks succeeds in a low-dimensional subspace (Li et al., 2018; Aghajanyan et al., 2021), the conceptual basis of LoRA. This work is about a different property: not the *dimension* of the useful update but its *location*. We find the update is low-rank yet standard LoRA still fails, because the low-rank structure lives in a parameter group LoRA does not touch. Intrinsic dimension and accessibility are distinct, and both constrain PEFT.

3 Problem and Mechanism

Setting. We are given a neural operator with pretrained parameters θ_0, a target regime represented by a training set D_t of input–solution pairs from a shifted PDE, and a budget B of trainable parameters. The operator's weights are partitioned into m parameter groups $G_1, \ldots, G_m$ defined by the architecture (for FNO: the spectral kernels, the pointwise convolutions, the lifting, and the projection; for DeepONet: the branch and trunk networks). A PEFT method chooses a group (or subset) and an adapter, and trains at most B parameters. The question is which group to choose.

Two constraints. We separate two properties that PEFT methods usually conflate. *Accessibility* is whether the adapter's reachable set intersects the parameters whose update most reduces target error. *Capacity* is how many of those parameters it can update. A method can fail through either: too little capacity underfits even the right group, while the wrong group is unreachable at any capacity. The two are not interchangeable, and the regime determines which binds.

Why capacity cannot repair inaccessible updates. Increasing adapter rank enlarges the set of updates representable within the parameterization exposed by PEFT, but it does not expose parameters that the adapter cannot modify. Capacity can therefore reduce approximation error inside an accessible update space, whereas a useful update component outside that space induces a floor that rank alone cannot remove. At small budgets capacity can bind first; once the accessible space is sufficiently expressive, accessibility becomes the limiting factor.

The useful update localizes, and where depends on the shift. We measure where adaptation actually happens by computing the full-FT displacement $\Delta W = \theta^*_{\mathrm{FT}} - \theta_0$ and localizing its energy $\|\Delta W\|_F^2$ over the parameter groups. On FNO under a viscosity or initial-condition shift, 98.6–98.8% of the energy lies in the spectral convolution kernels, which standard LoRA placement—lifting, pointwise convolutions, projection—never touches. This is not a parameter-volume artifact: the spectral kernels hold 95.4% of the model's real degrees of freedom, so their per-parameter update density is $1.03\times$ the model average against 0.25–$0.30\times$ for the groups LoRA does adapt, a 3.4–$4.1\times$ gap. The displacement is also low-rank (per-mode effective rank $r_{90} = 1$–9), so the failure of

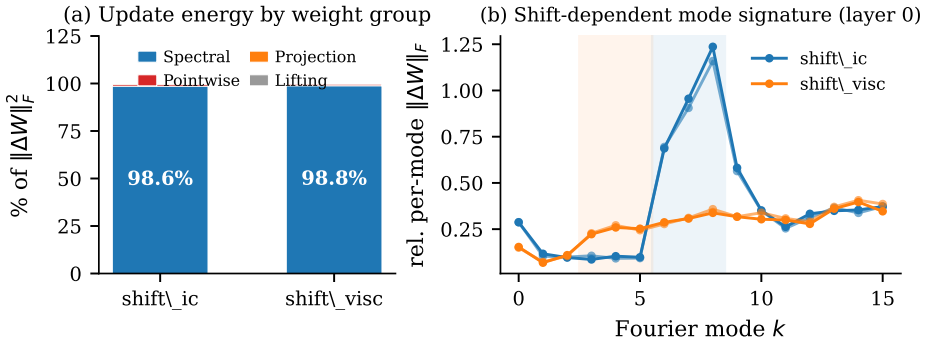

Figure 1: **Where adaptation lives on FNO.** (a) $> 98\%$ of the full-FT update energy $\|\Delta W\|_F^2$ localizes in the spectral convolution kernels for both Burgers shifts; standard LoRA placement (pointwise, projection, lifting) has no access to this group. (b) The per-Fourier-mode displacement is shift-dependent: the initial-condition shift peaks at modes 6–8 (the new high-frequency band), the viscosity shift at modes 3–5. The mode signature changes with the physical shift even when the dominant architectural group remains spectral, motivating a target-data diagnostic rather than a fixed frequency heuristic.

standard LoRA is not that the update is high-rank; it is that the update is somewhere LoRA cannot reach. On a DeepONet under an amplitude shift, the displacement instead localizes in the branch network (70% of energy), and under a stiffening shift it is diffuse. The dominant group therefore reflects how the architecture represents the target update; it is not universally spectral.

Accessibility imposes a floor; capacity sets how close. The consequence is visible in the error–budget curve. A misaligned adapter confined to the pointwise group saturates at about $2\times$ the full-FT error no matter how much rank it is given, because the group it can reach carries only $\sim 1.4\%$ of the update energy. An aligned adapter in the spectral group reaches 1.0–$1.1\times$ of full FT. The ordering reverses at tiny budgets: below roughly 1% of the parameters, the aligned adapter cannot cover the spectral subspace and the misaligned one, though capped, is cheaper to fit. Accessibility sets the floor and capacity sets how close a method gets to it; which one binds depends on the budget. The error–budget curve therefore reflects a transition from capacity-limited to accessibility-limited adaptation.

4 Method: Target-Aware Placement

The mechanism suggests measuring accessibility on the target task before committing to a restrictive PEFT placement. We need a signal that (i) uses only target training data, (ii) is cheap relative to fine-tuning, and (iii) reports not just a preferred group but whether any group should be preferred. A single gradient at initialization fails the third requirement: its energy is dominated by the output layer and picks the wrong group on five of six shifts. The early adaptation *trajectory* is far more informative.

Step 1: warm-up probe. From θ_0, run K steps of unrestricted AdamW on the target training split, cycling through minibatches, to reach θ_K ($K = 30$, about 4–8% of one fine-tuning run). The displacement $\Delta\theta = \theta_K - \theta_0$ is a cheap, noisy estimate of the direction adaptation wants to move.

Step 2: groupwise attribution. A raw parameter norm $\|\Delta\theta_g\|^2$ is miscalibrated: it is confident even on diffuse shifts. We instead score each group by the effect of *its* share of the update on the target function and loss. For group g, form $\theta^{(g)} = \theta_0 + P_g\Delta\theta$, which keeps only group g's warm-up update, and evaluate on a held-out probe split D_p:

$$s_g^{\text{func}} = \frac{\mathbb{E}_{x\sim D_p}\|f_{\theta^{(g)}}(x) - f_{\theta_0}(x)\|^2}{\mathbb{E}_{x\sim D_p}\|f_{\theta_0}(x)\|^2}, \qquad s_g^{\text{loss}} = L(D_p;\theta_0) - L(D_p;\theta^{(g)}). \tag{1}$$

Algorithm 1 Target-aware placement

Require: pretrained θ_0, target train split D_t, probe split D_p, groups $\{G_g\}$, budget B

$\theta_K \leftarrow K$ AdamW steps on D_t from θ_0; $\quad \Delta\theta \leftarrow \theta_K - \theta_0$

for each group g **do**

  $\theta^{(g)} \leftarrow \theta_0 + P_g \Delta\theta$ ▷ keep only group g's update

  $s_g \leftarrow s_g^{\text{func}}$ and s_g^{loss} on D_p ▷ Eq. (1)

end for

$C \leftarrow 1 - H(p)/\log m$ for $p_g \propto s_g$

if $C \geq \tau$ **then** $g^* \leftarrow \arg\max_g s_g$; train adapter on G_{g^*} at budget B from θ_0

else spread budget B across groups; train from θ_0

end if

s_g^{func} measures how much of the functional change group g accounts for; s_g^{loss} measures how much of the achievable loss reduction it delivers. Both are computed with a few forward passes per group and no further training.

Step 3: localized or diffuse. Normalize the scores to $p_g \propto s_g$ and compute the concentration $C = 1 - H(p)/\log m$, where H is the entropy. High C means one group dominates and placement should be concentrated; low C means the update is diffuse and no group should be preferred. This is the decision the raw-norm probe gets wrong and the functional/loss scores get right.

Step 4: adapt. Restart from θ_0 and train a standard adapter on the selected group at budget B (we use per-mode spectral LoRA on FNO and group LoRA on DeepONet, with a validation-selected learning rate). If C is low, spread the budget across groups instead of concentrating. The probe's cost—K warm-up steps plus $2m$ forward evaluations—is a few percent of the subsequent fine-tuning run and is counted in our cost accounting. No test labels and no full-FT solution influence the placement; full FT appears only as a diagnostic ceiling.

5 Experimental Setup

Backbones. We use a one-dimensional FNO (Li et al., 2021) (L=4 layers, width 64, 16 Fourier modes; 549,569 real degrees of freedom, of which the spectral kernels hold 95.4%) and a DeepONet (Lu et al., 2021) (branch and trunk networks, width 256). Parameter groups are the architectural blocks above.

PDEs and shifts. We study 1D periodic Burgers ($u_t + uu_x = \nu u_{xx}$) and Kuramoto–Sivashinsky (KS) (Kuramoto, 1978; Sivashinsky, 1977), integrated with a dealiased Fourier pseudospectral ETDRK4 solver. On FNO we use six controlled Burgers shifts spanning the dissipation and nonlinearity axes: an initial-condition (IC) band shift, a viscosity shift, and a viscosity ladder $\nu \times \{1.25, 2, 8\}$ plus an amplitude $\times 2$ shift. On DeepONet we use a localized amplitude shift and a diffuse stiffening shift. Each regime has 512 samples split $0.7/0.15/0.15$; splits and shift definitions are frozen before any test evaluation.

Methods and baselines. All methods start from the same source-pretrained checkpoint and share one adaptation recipe (AdamW, cosine schedule, early stopping on validation; learning rate selected on validation per configuration). We compare frozen, full FT, standard LoRA on the default (pointwise/lifting/projection) modules, and our probe-guided placement, against DoRA (Liu et al., 2024), AdaLoRA (Zhang et al., 2023), and $(IA)^3$ (Liu et al., 2022) at matched trainable real degrees of freedom (DOF). We count *real* DOF, so a complex spectral entry contributes 2. Full FT provides the full-capacity reference.

Metric and protocol. We report relative L_2 error on the held-out test split, evaluated once per run after the run matrix is frozen; validation is used only for early stopping and learning-rate selection. Comparative claims use 3–5 seeds. The probe uses only the target training split and a validation probe split. Probe overhead is reported separately.

Table 1: Matched-budget placement (Burgers, test rel-L_2, mean over 3 seeds, validation-selected). Same trainable real DOF and recipe; only the placement differs. Aligned = spectral, misaligned = pointwise. The aligned arm wins once capacity is adequate and loses below the $\sim 1\%$ coverage floor.

DOF (% of FT)	aligned	ic	visc	misaligned	ic / visc win
4,096 (0.75%)	spectral r1, 1 layer	0.0166	0.0180	pointwise r8	0.78× / 0.63× (loses)
8,192 (1.49%)	spectral r1, 2 layer	0.0084	0.0105	pointwise r16	**1.53×** / **1.10×**
16,384 (2.98%)	spectral r1	0.0059	0.0068	pointwise r32	**2.28×** / **1.67×**
32,768 (5.96%)	spectral r2	0.0054	0.0060	pointwise r64	**2.65×** / **1.88×**

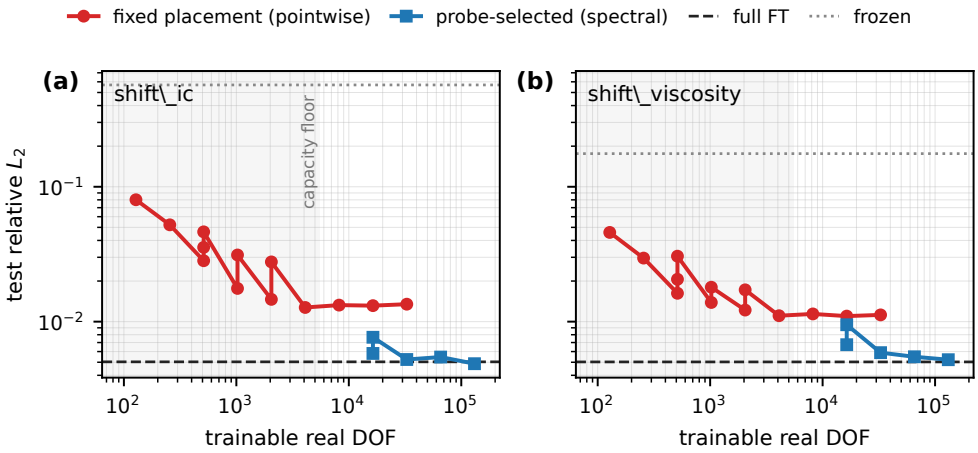

Figure 2: **Accessibility sets the floor; capacity sets how close.** Test relative L_2 against trainable real DOF (log–log) on the two Burgers shifts. A fixed-placement adapter confined to the pointwise group saturates at $\sim 2\times$ the full-FT error no matter how much rank it is given, while the probe-selected spectral adapter reaches 1.0–1.1× of full FT at 3–12% of the trainable parameters. Below the $\sim 1\%$ coverage floor (shaded) capacity dominates and the ordering reverses. Both frontiers use the same validation-selected recipe.

6 Results

6.1 Capacity helps until accessibility becomes limiting

Table 1 exposes a crossover between the two PEFT bottlenecks. At equal trainable real DOF, the only difference between the two arms is *where* the adapter sits. Below roughly 1% of the parameters, capacity dominates and the cheaper misaligned adapter can perform better. Once capacity is adequate, the ordering reverses: on both Burgers shifts the aligned spectral adapter beats the misaligned pointwise adapter by 1.7–2.7×, with the margin increasing with budget. The misaligned family saturates at about 2× full-FT error, whereas the aligned family reaches 1.0–1.1×. Increasing capacity therefore helps only until access to the target-relevant parameter group becomes limiting.

The advantage is not bought with capacity or compute. At 16,384 DOF the aligned arm uses the same budget as the misaligned one and comparable wall-clock, and against the standard LoRA placement at its own matched budget (14,384 DOF) it wins by 2.43× on the IC shift. The effect also grows with the size of the shift: along the viscosity ladder the aligned advantage over the matched misaligned arm rises monotonically from 1.05× ($\nu \times 1.25$) to 1.48× ($\nu \times 2$) to 2.91× ($\nu \times 8$). Harder shifts localize more, and placement matters more.

6.2 Target-aware placement beats strong baselines

Table 2 compares probe-guided placement against four PEFT families at matched budget across all six FNO shifts. The probe selects the spectral group on every shift, and the resulting adapter beats

Table 2: Strong-baseline comparison (FNO, test rel-L_2, mean over 3 seeds, validation-selected). LoRA, DoRA, AdaLoRA, and Ours use approximately 16K trainable DOF; DoRA-spectral is a within-subspace control with 12,288 DOF, and (IA)^3 uses its native smaller rescaling budget. Ours uses probe-guided spectral placement on all six shifts.

shift	LoRA	DoRA	AdaLoRA	(IA)^3	**Ours**	DoRA-spec	full FT
ic	0.0135	0.0137	0.0371	0.0586	**0.0059**	0.0064	0.0056
viscosity	0.0114	0.0115	0.0157	0.0204	**0.0068**	0.0057	0.0052
$\nu\times1.25$	0.0213	0.0214	0.0252	0.0240	**0.0151**	0.0150	0.0126
$\nu\times2$	0.0149	0.0146	0.0177	0.0178	**0.0095**	0.0092	0.0082
$\nu\times8$	0.0097	0.0097	0.0157	0.0295	**0.0054**	0.0041	0.0039
amp$\times2$	0.0877	0.0978	0.2280	0.2366	**0.0767**	0.0717	0.0579

Table 3: Probe scores and ground-truth placement. Top-1/top-2 confidence ratio (higher = more confident); the argmax group is shown where it matches the measured best placement. The functional and loss probes are confident exactly where the update localizes and quiet where it is diffuse; the raw parameter norm is miscalibrated (more confident on the diffuse case).

setting	true best	functional	loss	param-norm
FNO, viscosity	spectral	spectral (5.21)	spectral (1.92)	spectral (2.72)
FNO, IC	spectral	spectral (2.14)	spectral (1.55)	spectral (1.79)
DeepONet, amp$\times2$ (localized)	branch	branch (3.98)	branch (2.29)	branch (1.14)
DeepONet, $\nu\times8$ (diffuse)	—	(1.28)	(1.01)	(1.60)

LoRA, DoRA, AdaLoRA, and (IA)^3 on every shift, in most cases by a wide margin. The strongest fixed-placement baseline, DoRA on the pointwise group, is still 1.9–2.3$\times$ worse than probe-guided placement on the two original shifts. DoRA applied to the *spectral* group (last column before FT) is competitive with our LoRA-based adapter—it is a better within-subspace adapter—which confirms that the gain comes from placement, not from the particular adapter we happen to drop in.

6.3 The probe identifies where to adapt, and when not to

The probe's value is that it makes the placement decision from target data alone. Table 3 reports the groupwise scores. On both FNO shifts the functional and loss probes rank the spectral group first with a clear margin, matching the known placement. On the localized DeepONet amplitude shift they rank the branch network first, again matching the measured placement. On the diffuse DeepONet shift they report low confidence, declining to concentrate the adaptation budget. The raw parameter-norm score, by contrast, is *more* confident on the diffuse DeepONet shift (1.60) than on the localized one (1.14). The placement rule uses the functional score, with the loss score as a corroborating signal and the parameter norm as an ablation.

The concentration statistic C turns this into a decision. Across the nine (architecture, shift) cells, C predicts the benefit of concentrating the budget on the probe's top group with Spearman correlation 0.62 (Figure 4): on FNO ($C = 0.10$–0.28) concentrating always helps, while on the diffuse DeepONet shift ($C = 0.013$) concentrating *hurts* and spreading is correct. The probe thus does double duty: it picks a group when one should be picked and refuses when none should.

6.4 Cross-architecture and cross-PDE evidence

DeepONet. The strongest objection to an FNO-only result is anatomical: the spectral kernels are 95.4% of the FNO's parameters, so "spectral dominates" could be a tautology. DeepONet has no Fourier structure, and there the picture is shift-dependent. Under the amplitude shift the full-FT displacement localizes in the branch network (70% of energy), and at matched budget branch placement beats trunk placement (0.410 vs 0.422 at 8,192 DOF; 0.428 vs 0.433 at 4,096 DOF). The probe recovers this from target data. Under the stiffening shift the displacement is diffuse, the probe reports low confidence, and broad adaptation is correct. Placement matters on DeepONet too, but the useful group is the branch, not a spectral kernel—exactly what a target-aware rule should discover.

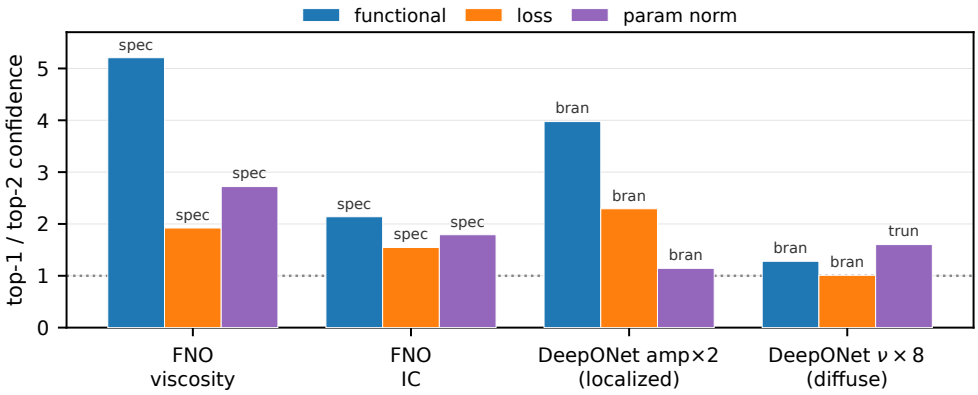

Figure 3: **The probe identifies where to adapt and when not to.** Top-1/top-2 confidence of three probe scores on four settings; the bar label is the selected group. The functional and loss probes are confident where the update localizes (spectral on FNO, branch on the localized DeepONet shift) and report low confidence on the diffuse DeepONet shift. The raw parameter-norm score is miscalibrated: it is more confident on the diffuse case than on the localized one.

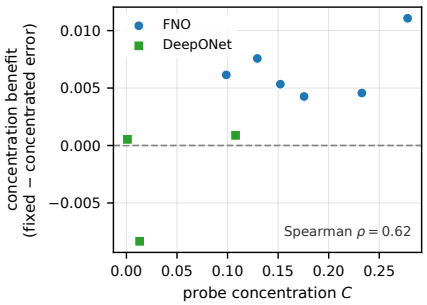

Figure 4: **Concentration predicts when to concentrate.** Probe concentration C against the benefit of concentrating the budget on the probe's top group, over nine (architecture, shift) cells. FNO shifts (high C) all benefit; the diffuse DeepONet shift (low C) is hurt by concentration. The probe's confidence is a usable prior for the placement decision.

Two boundaries temper the DeepONet evidence. Our DeepONet source model is less accurate than the FNO at source, which may inflate how diffuse its displacement appears. And because the trunk's fixed Fourier basis needs little task-specific adaptation, spreading the budget across branch and trunk is at least as effective as concentrating on the branch (matched-budget broad ties or beats branch on all three shifts); concentration pays where a group is dominant enough that excluding it imposes a floor, as the spectral group does on FNO.

Kuramoto–Sivashinsky. The mechanism also transfers across PDEs. On the chaotic KS equation under a dissipation-halving shift, the displacement again localizes in the FNO spectral kernels (97.1% of energy) but peaks at *low* modes $k = 1$–3, a physics-specific signature distinct from Burgers. Aligned spectral adaptation closes 82% of the full-FT gap while misaligned LoRA closes $\approx 0\%$, over five seeds. The principle transfers; the signature does not, which is the point of measuring it per target.

6.5 Within-subspace allocation is a separate problem

Placement selects a group; a separate question is how to allocate capacity *within* it. Allocating per-mode rank by the measured displacement is the *worst* policy we tested—worse than uniform and worse than F-Adapter's fixed low-frequency prior (mean test error over the six FNO shifts: F-

Adapter 0.0172, uniform 0.0178, displacement-guided 0.0237, full-FT-oracle-guided 0.0270). The spectral displacement is nearly rank-1 per mode, making coverage across modes more useful here than assigning additional rank to modes with large displacement. Our method addresses the group-level placement decision; within-group allocation is a separate design choice, for which the fixed low-frequency prior performs best among the policies tested here.

6.6 Controls and boundaries

Three controls isolate the mechanism. *Necessity and sufficiency at full rank:* training only the spectral kernels at full rank matches full FT (1.02–$1.27\times$), while training everything *except* them at full rank stalls at 1.8–$3.1\times$ despite using more parameters than either LoRA arm—so the spectral subspace is sufficient and access to it is necessary. *Wrong and random placement:* a low-score group is 1.7–$2.3\times$ worse than the probe-selected group at matched budget, and random placement is 3–$7\times$ worse. *Trajectory vs. gradient:* the multi-step warm-up picks the right group on six of six FNO shifts, while a single gradient at initialization picks the wrong group on five of six, which is why the probe uses a trajectory rather than a one-shot estimate. The main boundary is the coverage floor: below $\sim 1\%$ of the parameters, capacity dominates and the aligned adapter loses to the cheaper misaligned one.

7 Conclusion

PEFT for neural operators is governed by two distinct constraints: capacity and accessibility. Increasing capacity improves adaptation until useful target updates fall outside the trainable parameterization; beyond that point, additional rank cannot remove the resulting error floor. We use a short multi-batch target-data probe with groupwise functional and task-loss attribution to identify where useful adaptation occurs and align PEFT accordingly. Across the tested FNO shifts, this alignment approaches full fine-tuning at 3–12% of the trainable parameters and outperforms strong matched-budget PEFT baselines, while the DeepONet experiments show that broad adaptation remains competitive when it already provides access to the useful update. These results establish parameter accessibility as a design axis alongside capacity in parameter-efficient neural-operator adaptation.

Reproducibility

(This section is **recommended** and does not count toward the page limit.)

The train/validation/test split is fixed by a frozen seed; validation is used only for early stopping and learning-rate selection, and the test split is read once per run after the matrix is frozen. The probe uses only the target training split and a validation probe split. The probe threshold τ and all hyperparameters were fixed before the corresponding test evaluation.

The supplementary material contains the target-aware placement implementation, neural-operator and adaptation baselines, frozen experiment configurations, probe scores and analyses, per-run evaluation records, and the analysis code used for the reported results. The supplied scripts reproduce the principal matched-budget comparisons, placement analyses, mechanism controls, and reported aggregates from these records.

AI use statement

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for providing feedback on research methodology and experiments, providing feedback on mathematical arguments, and interpreting results. We have not used generative AI tools for generating synthetic data sets, formulating mathematical claims or implementing methods, and the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for literature discovery, brainstorming, drafting and editing parts of the manuscript, editing software code and figures, and simulated peer-review feedback. We have reviewed all AI-assisted work and AI-assisted code was re-run with its outputs

checked against the experimental artifacts reported in this paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 7319–7328, 2021.

Kamyar Azizzadenesheli, Nikola Kovachki, Zongyi Li, Miguel Liu-Schiaffini, Jean Kossaifi, and Anima Anandkumar. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, 6:320–328, 2024.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations (ICLR)*, 2022.

Jayesh K. Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized PDE modeling. *Transactions on Machine Learning Research (TMLR)*, 2023.

Zhongkai Hao, Chang Su, Songming Liu, Chengyang Ying, Yinpeng Dong, et al. DPOT: Auto-regressive denoising operator transformer for large-scale PDE pre-training. In *International Conference on Machine Learning (ICML)*, 2024.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning (ICML)*, pp. 2790–2799, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision (ECCV)*, pp. 709–727, 2022.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. VeRA: Vector-based random matrix adaptation. In *International Conference on Learning Representations (ICLR)*, 2024.

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

Yoshiki Kuramoto. Diffusion-induced chaos in reaction systems. *Progress of Theoretical Physics Supplement*, 64:346–367, 1978.

Yoonho Lee et al. Surgical fine-tuning improves adaptation to distribution shifts. In *International Conference on Learning Representations (ICLR)*, 2023.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3045–3059, 2021.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations (ICLR)*, 2018.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4582–4597, 2021.

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

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *International Conference on Machine Learning (ICML)*, 2024.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, 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.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, et al. The well: a large-scale collection of diverse physics simulations for machine learning. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks*, 2024.

Fabian Paischer et al. Parameter efficient fine-tuning via explained variance adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. FourCastNet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

Gregory I. Sivashinsky. Nonlinear analysis of hydrodynamic instability in laminar flames. *Acta Astronautica*, 4(11-12):1177–1206, 1977.

Makoto Takamoto, Francesco Alesiani, Mathias Niepert, et al. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks*, 2022.

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.

Yichen Xu, Yuyang Liang, Shan Dai, Tianyang Hu, Tsz Nam Chan, and Chenhao Ma. Understanding and guiding layer placement in parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2602.04019*, 2026.

Hangwei Zhang, Chun Kang, Yan Wang, and Difan Zou. F-Adapter: Frequency-adaptive parameter-efficient fine-tuning in scientific machine learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2023.

Suoxin Zhang et al. Rethinking adapter placement: A dominant adaptation module perspective. *arXiv preprint arXiv:2605.06183*, 2026.

Rui Zheng et al. AdaMSS: Adaptive multi-subspace approach for parameter-efficient fine-tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Changjian Zhou et al. PhysGuard: Fisher-guided gradient projection for sim-to-real neural PDE surrogates. *arXiv preprint arXiv:2606.16602*, 2026.

A METHOD DETAILS

A.1 PARAMETER GROUPS

The parameter groups are the architectural blocks of each backbone, chosen so that a group is the smallest unit a practitioner would independently decide to adapt.

FNO1d. Four groups: the *spectral* kernels $R \in \mathbb{C}^{64\times64\times16}$ (one per layer, the complex Fourier multipliers), the *pointwise* 1×1 convolutions W (one per layer), the *lifting* f_{c0}, and the *projection* (f_{c1}, f_{c2}). The spectral kernels hold $524{,}288$ of the model's $549{,}569$ real degrees of freedom (95.4%); the pointwise, projection, and lifting groups together hold the remaining 4.6%.

DeepONet. Two groups: the *branch* network (which encodes the input function at sensor points) and the *trunk* network (which encodes the query location), plus a final bias. The branch and trunk are comparable in size, so DeepONet has no dominant group by volume.

A.2 THE PROBE

Algorithm 1 in the main text gives the pipeline; here we record the exact configuration. The warm-up runs $K = 30$ AdamW steps (learning rate 5×10^{-4}, batch size 32, no weight decay) on the target *train* split, cycling through minibatches so that no single batch dominates the trajectory. The probe split D_p is the validation split; no test data is used. For each group g we form $\theta^{(g)} = \theta_0 + P_g\Delta\theta$ and evaluate the two scores of Eq. (1) with a single pass over D_p. The concentration statistic is $C = 1 - H(p)/\log m$ with $p_g \propto s_g$ over the m groups; we use the functional score for C and report the loss score as a corroborating signal. The placement threshold τ is fixed at 0.05 across all experiments (chosen once on the FNO viscosity shift and not tuned per setting).

Probe cost. The warm-up is $K = 30$ steps and the attribution is $2m$ forward passes over the probe split. On an RTX 3090 this is ≈ 0.3 s, about 2–3% of one 10–16 s PEFT run and 4–8% of the cost of a single full fine-tuning run. The probe is run once per (architecture, shift); its cost is included in the cost accounting of §B.3.

Why a trajectory, not a gradient. A single squared-gradient / empirical-Fisher estimate at initialization is dominated by the output projection layer (74–81% of its energy) and selects the wrong group on five of six FNO shifts. The K-step warm-up trajectory selects the right group on six of six. The trajectory averages over the early loss landscape rather than evaluating it at a single point, which is what makes it informative about where sustained adaptation will occur.

A.3 PLACEMENT RULE

If $C \geq \tau$ the budget is concentrated on $g^* = \arg\max_g s_g$; if $C < \tau$ the budget is spread across groups. On FNO the probe always selects the spectral group and C is high, so the adapter is a per-Fourier-mode spectral LoRA. On the localized DeepONet shift the probe selects the branch and C is high; on the diffuse DeepONet shift C is low and the budget is spread across branch and trunk. Within the selected group, capacity is allocated uniformly or with a fixed low-frequency prior, following the within-subspace comparison in §H.

A.4 REAL-DOF ACCOUNTING

We count *real* degrees of freedom, so a complex entry contributes 2. For FNO1d: each spectral kernel is $64 \cdot 64 \cdot 16 = 65{,}536$ complex $= 131{,}072$ real; four layers give $524{,}288$ real. A per-mode spectral LoRA of rank r learns $A_m \in \mathbb{C}^{r\times64}$ and $B_m \in \mathbb{C}^{64\times r}$ per mode, i.e. $16 \cdot (64r + 64r) = 2048r$ complex $= 4096r$ real per layer, $16{,}384r$ real over four layers. Standard LoRA of rank r on the pointwise/lifting/projection modules is real-valued; rank 16 gives $14{,}384$ real DOF. Full FT is $549{,}569$ real. The matched-budget comparisons in the main text pair arms at equal real DOF (e.g. spectral rank-1 $= 16{,}384$ vs pointwise rank-32 $= 16{,}384$).

B Experimental Setup Details

B.1 PDEs, solvers, and data

Both PDEs are integrated on $[0, 2\pi)$ with $n_x = 256$ grid points to $T = 1$ using a dealiased Fourier pseudospectral ETDRK4 scheme with the $2/3$ dealiasing rule on the quadratic term. Burgers is $u_t + uu_x = \nu u_{xx}$ with source $\nu = 0.01$ and low-frequency initial conditions drawn from a Gaussian process over Fourier band $k \in \{1..5\}$. KS is $u_t = -u_{xx} - \nu u_{xxxx} - uu_x$ with source $\nu = 0.25$ and the same initial-condition family. The solver was validated against the viscous Burgers Cole–Hopf reference and against energy-balance diagnostics for KS. Each regime has 512 samples split $0.7/0.15/0.15$ into train/validation/test with a frozen split seed.

Shifts. On FNO we use six controlled Burgers shifts: an initial-condition band shift ($k \to \{6..15\}$), a viscosity shift ($\nu \to 0.04$), a viscosity ladder $\nu \times \{1.25, 2, 8\}$, and an amplitude $\times 2$ shift. On DeepONet we use a localized amplitude $\times 2$ shift and a diffuse stiffening shift ($\nu \times 8$). On KS we use a dissipation-halving shift ($\nu \to 0.125$); the Burgers-direction shifts are degenerate for KS (increasing ν collapses the dynamics, and high-band initial conditions are hyperdissipated), which we verified numerically before freezing the shift.

B.2 Adaptation recipe

Every adaptation arm starts from the same source-pretrained checkpoint and shares one recipe: AdamW, cosine schedule, early stopping on validation relative L_2. The learning rate is selected on validation from $\{2 \times 10^{-3}, 8 \times 10^{-3}, 1.6 \times 10^{-2}\}$ per configuration. LoRA uses $\alpha = r$, giving $\alpha/r = 1$ across ranks. Source pretraining uses AdamW at learning rate 10^{-3} for 60 epochs. The metric is test relative L_2, read once per run after the run matrix is frozen.

B.3 Seeds, hardware, and runtime

Comparative claims use 3–5 seeds; the dose-response ladder uses 2. Dataset, initialization, and adaptation seeds are recorded separately. Experiments ran on RTX 3090 GPUs. A single PEFT run is 10–16 s; the probe adds ≈ 0.3 s. The full experimental program behind this paper is ≈ 21 GPU-hours across all runs.

C Baseline Details

All baselines share the source checkpoint, target data and split, validation-based early stopping, and the adaptation recipe above, and are compared at matched trainable real DOF.

- **LoRA** (Hu et al., 2022): low-rank additive update $W + BA$. Default placement is the pointwise/lifting/projection modules (misaligned on FNO); the aligned variant places it on the spectral kernels.
- **DoRA** (Liu et al., 2024): weight-decomposed LoRA, $W = m\,(W_0 + BA)/\|\cdot\|$ with the magnitude vector initialized to the row norms of W_0. Applied to the same targets as LoRA, and to the spectral kernels as a within-subspace variant.
- **AdaLoRA** (Zhang et al., 2023): SVD-form update $\Delta W = P\Lambda Q$ with an orthogonality regularizer and sensitivity-based rank allocation. The allocation is determined after a short warm-up and held fixed during the remaining adaptation. Rank is reallocated within the selected modules.
- **(IA)^3** (Liu et al., 2022): elementwise output rescaling, initialized to 1. Very small budget (output dimension only), so it is capacity-limited.
- **F-Adapter** (Zhang et al., 2025): see §H.
- **Full FT**: adaptation of all parameters, used as the full-capacity reference.

D COMPLETE FNO RESULTS

Matched-budget placement. Table 1 in the main text reports the mean over 3 seeds; the aligned arm wins at and above 8,192 DOF and loses at 4,096 DOF (the coverage floor). The misaligned family saturates at 0.0129 (IC) / 0.0111 (viscosity) regardless of rank up to $r = 64$, while the aligned family reaches 0.0054 (IC) / 0.0056 (viscosity) at 5.96% DOF.

Dose–response. Along the viscosity ladder at matched 16,384 DOF, the aligned advantage over the misaligned arm is $1.05\times$ ($\nu \times 1.25$), $1.48\times$ ($\nu \times 2$), and $2.91\times$ ($\nu \times 8$), and $1.58\times$ on the amplitude shift. The advantage grows monotonically with shift magnitude.

Strong baselines. Table 2 in the main text reports the mean over 3 seeds at approximately matched budget. Probe-guided placement beats LoRA, DoRA, AdaLoRA, and $(IA)^3$ on all six shifts; the sign of the effect is consistent across seeds.

E DEEPONET RESULTS

Table 4 reports the matched-budget DeepONet comparison. Branch placement outperforms trunk placement on the localized amplitude and IC shifts, while broad placement ties or improves on branch placement across all three shifts. The localized and diffuse cases therefore favor different degrees of concentration.

Table 4: DeepONet matched-budget comparison (test rel-L_2, 8,192 real DOF, mean over 3 seeds, validation-selected). Branch beats trunk on both localized shifts, while broad (branch+trunk) ties or beats branch on all three shifts.

shift	placement	test rel-L_2
amp$\times$2 (localized)	branch r8	**0.4101**
	trunk r8	0.4222
	broad (all r4)	0.4106
IC (localized)	branch r8	0.2159
	trunk r8	0.2636
	broad (all r4)	**0.2096**
$\nu \times 8$ (diffuse)	branch r8	0.1036
	trunk r8	0.0963
	broad (all r4)	**0.0953**

F PROBE ANALYSIS

Three scores. Table 3 in the main text reports the top-1/top-2 confidence ratio for the functional, loss, and parameter-norm scores. The functional and loss scores are confident where the update localizes and quiet where it is diffuse; the parameter-norm score reports higher confidence on the diffuse DeepONet shift (1.60) than on the localized one (1.14). The placement rule uses the functional score, with the loss score reported as a corroborating signal and the parameter norm as an ablation.

Probe-budget sensitivity. The correct group is selected at $K = 15$ warm-up steps (4% of full-FT cost) on all six FNO shifts; $K = 4$ (1%) selects correctly on five of six, and confidence grows monotonically with K. All reported experiments use $K = 30$.

Concentration and the placement decision. The concentration statistic C separates the regimes: FNO shifts have $C = 0.10$–0.28 and all benefit from concentration; the diffuse DeepONet shift has $C = 0.013$ and is hurt by it. Across the nine (architecture, shift) cells the Spearman correlation between C and the concentration benefit is 0.62 (Figure 4).

G MECHANISM CONTROLS

DOF-normalized localization. The spectral kernels hold 95.4% of the FNO's real DOF, so a raw 98.6% energy share could be mostly anatomy. Normalizing by parameter share, the spectral group is updated at 1.03– $1.04\times$ the model-average density against 0.25–$0.30\times$ for the groups LoRA adapts, a 3.4–$4.1\times$ per-parameter gap. The localization is seed-stable (per-tensor relative-norm Pearson 0.9993–0.9998).

Effective rank. The spectral displacement is low-rank: per-mode 64×64 complex blocks have effective rank $r_{90} = 1$–9, and the out-major unfolding has $r_{90} = 5$–23.

Necessity and sufficiency at full rank. Training only the spectral kernels at full rank reaches 1.02–$1.27\times$ the full-FT error, whereas training all non-spectral parameters at full rank remains at 1.8–$3.1\times$ full-FT error despite using more parameters than either LoRA arm. This establishes the accessibility boundary at full rank.

Wrong and random placement. A low-score group is 1.7–$2.3\times$ worse than the probe-selected group at matched budget, and random placement is 3–$7\times$ worse.

Capacity floor. Below $\sim 1\%$ of the parameters the aligned adapter cannot cover the spectral subspace and loses to the cheaper misaligned one. The two factors trade off: capacity binds at tiny budgets, accessibility binds once capacity is adequate.

H WITHIN-SUBSPACE ALLOCATION AND F-ADAPTER

Placement selects a group; a separate question is how to allocate capacity within it. We compared six per-mode rank-allocation policies on the spectral group at matched total DOF across the six FNO shifts: a fixed low-frequency prior (F-Adapter's rule), uniform, random, gradient-guided, probe-guided (early-displacement), and oracle-guided (full-FT displacement, diagnostic only). Table 5 reports the mean test error over the six shifts.

Table 5: Within-subspace rank allocation on the spectral group (mean test rel-L_2 over the six FNO shifts, $\sim$32K DOF, 3 seeds, validation-selected). Displacement-guided allocation (probe and oracle) is the worst; a fixed low-frequency prior (F-Adapter) or uniform is best. The oracle uses the full-FT solution and is diagnostic only.

allocation policy	mean test rel-L_2
F-Adapter (fixed low-freq prior)	$\mathbf{0.0172}$
uniform	0.0178
random	0.0228
probe (early displacement)	0.0237
gradient	0.0242
oracle (full-FT displacement, diagnostic)	0.0270

Displacement-guided allocation performs worst on this testbed, while uniform allocation and the fixed low-frequency prior perform best. This separates architectural-group placement from capacity allocation within the selected spectral subspace. F-Adapter (Zhang et al., 2025) provides the low-frequency allocation used in this comparison.

I ADDITIONAL FIGURES

Per-seed values, training curves, and the per-mode displacement signatures for all shifts are available in the released artifact. Figure 1b shows the shift-dependent mode signature for the two Burgers shifts; the KS signature peaks at low modes $k = 1$–3.

J LIMITATIONS

The experiments cover two neural-operator architectures and two PDE families under controlled one-dimensional shifts, with candidate adaptation groups defined by the architecture. The probe adds 2–3% of the cost of one PEFT run. Extending target-aware placement to higher-dimensional systems, larger operators, and learned rather than predefined parameter groups remains open.