LEADS: Learning Dynamical Systems that Generalize Across Environments
Supplemental Material

Yuan Yin[1], Ibrahim Ayed[1,2], Emmanuel de Bézenac[1], Nicolas Baskiotis[1], Patrick Gallinari[1,3]
[1]Sorbonne Université, Paris, France
[2]ThereSIS Lab, Thales, Paris, France [3]Criteo AI Lab, Paris, France
{yuan.yin,ibrahim.ayed,emmanuel.de-bezenac,
nicolas.baskiotis,patrick.gallinari}@sorbonne-universite.fr

A Proof of Proposition 1

Proposition 1 (Existence and Uniqueness)**.** *Assume Ω is convex, then the existence of a minimal decomposition $f^\star, \{g_e^\star\}_{e\in E} \in \mathcal{F}$ of Eq. 3 is guaranteed. Furthermore, if Ω is strictly convex, this decomposition is unique.*

Proof. The optimization problem is:

$$\min_{f, g_e \in \mathcal{F}} \quad \sum_{e\in E} \Omega(g_e) \quad \text{subject to} \quad \forall x^{e,i} \in \hat{\mathfrak{T}}, \forall t, \frac{\mathrm{d}x_t^{e,i}}{\mathrm{d}t} = (f + g_e)(x_t^{e,i}) \tag{3}$$

The idea is to first reconstruct the full functional from the trajectories of $\hat{\mathfrak{T}}$. By definition, $\mathcal{A}^e$ is the set of points reached by trajectories in $\hat{\mathfrak{T}}$ from environment e so that:

$$\mathcal{A}^e = \{x \in \mathbb{R}^d \mid \exists x_\cdot^e \in \hat{\mathfrak{T}}, \exists t,\ x_t^e = x\}$$

Then let us define a function f_e^{data} in the following way, $\forall e \in E$, take $a \in \mathcal{A}^e$, we can find $x_\cdot^e \in \hat{\mathfrak{T}}$ and t_0 such that $x_{t_0}^e = a$. Differentiating $x_\cdot^e$ at t_0, which is possible by definition of $\hat{\mathfrak{T}}$, we take:

$$f_e^{\text{data}}(a) = \left.\frac{\mathrm{d}x_t^e}{\mathrm{d}t}\right|_{t=t_0}$$

For any (f, g_e) satisfying the constraint in Eq. 3, we then have $(f + g_e)(a) = \left.\frac{\mathrm{d}x_t}{\mathrm{d}t}\right|_{t_0} = f_e^{\text{data}}(a)$ for all $a \in \mathcal{A}^e$. Conversely, any pair such that $(f, g_e) \in \mathcal{F}\times\mathcal{F}$ and $f + g_e = f_e^{\text{data}}$, verifies the constraint.

Thus we have the equivalence between Eq. 3 and the following objective:

$$\min_{f\in\mathcal{F}} \quad \sum_e \Omega(f_e^{\text{data}} - f) \tag{S1}$$

The result directly follows from the fact that the objective is a sum of (strictly) convex functions in f and is thus (strictly) convex in f. □

B Further details on the generalization with *LEADS*

In this section, we will give more details on the link between our framework and its generalization performance. After introducing the necessary definitions in Sec. B.1, we show the proofs of the results for the general case in Sec. 3. Then in Sec. B.3 we provide the instantiation for linear approximators. Finally, we show how we derived our heuristic instantiation for neural networks in Eq. 8 in Sec. 3.3 from the existing capacity bound for neural networks.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

Table S1: Capacity definitions of different sets by covering number with associated metric or pseudo-metric.

Capacity	Metric or pseudo-metric	Mentioned in
$\mathcal{C}(\varepsilon, \mathbb{H}^m) := \sup_{\mathcal{P}} \mathcal{N}(\varepsilon, \mathbb{H}^m, d_{\mathcal{P}})$	$d_{\mathcal{P}}((f+g_1,\dots,f+g_m),(f'+g'_1,\dots,f'+g'_m)) = \int_{(\mathcal{A}\times T\mathcal{A})^m} \frac{1}{m}\lvert\sum_{e\in E}\lVert(f+g_e)(x^e)-y^e\rVert^2 - \sum_{e\in E}\lVert(f'+g'_e)(x^e)-y^e\rVert^2\rvert \mathrm{d}\boldsymbol{\mathcal{P}}(\mathbf{x},\mathbf{y})$	Theorem S1; Prop. S1
$\mathcal{C}_{\hat{\mathcal{G}}}(\varepsilon, \hat{\mathcal{F}}) := \sup_{\mathcal{P}} \mathcal{N}(\varepsilon, \hat{\mathcal{F}}, d_{[\mathcal{P},\hat{\mathcal{G}}]})$	$d_{[\mathcal{P},\hat{\mathcal{G}}]}(f,f') = \int_{\mathcal{A}\times T\mathcal{A}} \sup_{g\in\hat{\mathcal{G}}} \lvert\lVert(f+g)(x)-y\rVert^2 - \lVert(f'+g)(x)-y\rVert^2\rvert \mathrm{d}\mathcal{P}(x,y)$	Prop. 2, S1, S3; Cor. S1
$\mathcal{C}_{\hat{\mathcal{F}}}(\varepsilon, \hat{\mathcal{G}}) := \sup_{\mathcal{P}} \mathcal{N}(\varepsilon, \hat{\mathcal{G}}, d_{[\mathcal{P},\hat{\mathcal{F}}]})$	$d_{[\mathcal{P},\hat{\mathcal{F}}]}(g,g') = \int_{\mathcal{A}\times T\mathcal{A}} \sup_{f\in\hat{\mathcal{F}}} \lvert\lVert(f+g)(x)-y\rVert^2 - \lVert(f+g')(x)-y\rVert^2\rvert \mathrm{d}\mathcal{P}(x,y)$	Prop. 2, S1, S2
$\mathcal{C}(\varepsilon, f+\hat{\mathcal{G}}) := \sup_{\mathcal{P}} \mathcal{N}(\varepsilon, f+\hat{\mathcal{G}}, d_{\mathcal{P}})$	$d_{\mathcal{P}}(f+g,f+g') = \int_{\mathcal{A}\times T\mathcal{A}} \lvert\lVert(f+g)(x)-y\rVert^2 - \lVert(f+g')(x)-y\rVert^2\rvert \mathrm{d}\mathcal{P}(x,y)$	Prop. 3
$\mathcal{C}(\varepsilon, \hat{\mathcal{G}}, L^1) := \sup_{\mathcal{P}} \mathcal{N}(\varepsilon, \hat{\mathcal{G}}, d_{L^1(\mathcal{P})})$	$d_{L^1(\mathcal{P})}(g,g') = \int_{\mathbb{R}^d} \lVert(g-g')(x)\rVert_1 \mathrm{d}\mathcal{P}(x)$	Prop. S2; Theorem S3
$\mathcal{C}(\varepsilon, \hat{\mathcal{G}}, L^2) := \sup_{\mathcal{P}} \mathcal{N}(\varepsilon, \hat{\mathcal{G}}, d_{L^2(\mathcal{P})})$	$d_{L^2(\mathcal{P})}(g,g') = \sqrt{\int_{\mathbb{R}^d} \lVert(g-g')(x)\rVert_2^2 \mathrm{d}\mathcal{P}(x)}$	Prop. 4; Lemma S1

B.1 Preliminaries

Table S1 gives the definition of the different capacity instances considered in the paper for each hypothesis space, and the associated distances. We say that a space $\mathcal{H}$ is ε-covered by a set H, with respect to a metric or pseudo-metric $d(\cdot,\cdot)$, if for all $h \in \mathcal{H}$ there exists $h' \in H$ with $d(h,h') \leq \varepsilon$. We define by $\mathcal{N}(\varepsilon, \mathcal{H}, d)$ the cardinality of the smallest H that ε-covers $\mathcal{H}$, also called covering number [S7]. The capacity of each hypothesis space is then defined by the maximum covering number over all distributions. Note that the loss function is involved in every metric in Table S1. For simplicity, we therefore omit the notation of loss function for the hypothesis spaces.

As in [S2], covering numbers are based on pseudo-metrics. We can verify that all distances in Table S1 are pseudo-metrics:

Proof. This is trivially verified. For example, for the distance $d_{\mathcal{P}}(f+g, f+g')$ given in Table S1, which is the distance between $f+g, f+g' \in f+\hat{\mathcal{G}}$, it is easy to check that the following properties do hold:

- $d_{\mathcal{P}}(f+g, f+g') = 0$ (subtraction of same functions evaluated on same x and y)
- $d_{\mathcal{P}}(f+g, f+g') = d_{\mathcal{P}}(f+g', f+g)$ (evenness of absolute value)
- $d_{\mathcal{P}}(f+g, f+g') \leq d_{\mathcal{P}}(f+g, f+g'') + d_{\mathcal{P}}(f+g'', f+g')$ (triangular inequality of absolute value)

Other distances in Table S1 can be proven to be pseudo-metrics in the same way. □

B.2 General Case

B.2.1 Proof of Proposition 2

Proposition 2. *Given m environments, let $\varepsilon_1, \varepsilon_2, \delta > 0, \varepsilon = \varepsilon_1 + \varepsilon_2$. Assume the number of examples n per environment satisfies*

$$n \geq \max\left\{\frac{64}{\varepsilon^2}\left(\frac{1}{m}\left(\log\frac{4}{\delta} + \log \mathcal{C}_{\hat{\mathcal{G}}}\left(\frac{\varepsilon_1}{16}, \hat{\mathcal{F}}\right)\right) + \log \mathcal{C}_{\hat{\mathcal{F}}}\left(\frac{\varepsilon_2}{16}, \hat{\mathcal{G}}\right)\right), \frac{16}{\varepsilon^2}\right\} \tag{5}$$

Then with probability at least $1-\delta$ (over the choice of training sets $\{\hat{\mathcal{P}}_e\}$), any learner $(f+g_1,\dots,f+g_m)$ will satisfy $\frac{1}{m}\sum_{e\in E} \mathrm{er}_{\mathcal{P}_e}(f+g_e) \leq \frac{1}{m}\sum_{e\in E} \hat{\mathrm{er}}_{\hat{\mathcal{P}}_e}(f+g_e) + \varepsilon$.

Proof. We introduce some extra definitions that are necessary for proving the proposition. Let $\mathcal{H} = f + \hat{\mathcal{G}}$ defined for each $f \in \hat{\mathcal{F}}$, and let us define the product space $\mathcal{H}^m = \{(f+g_1, \dots, f+g_m) : f + g_e \in \mathcal{H}\}$. Functions in this hypothesis space all have the same f, but not necessarily the same g_e. Let $\mathbb{H}$ be the collection of all hypothesis spaces $\mathcal{H} = f + \hat{\mathcal{G}}, \forall f \in \hat{\mathcal{F}}$. The hypothesis space associated to multiple environments is then defined as $\mathbb{H}^m := \bigcup_{\mathcal{H}\in\mathbb{H}} \mathcal{H}^m$.

Our proof makes use of two intermediary results addressed in Theorem S1 and Prop. S1.

Theorem S1 ([S2], Theorem 4, adapted to our setting)**.** *Assuming $\mathbb{H}$ is a permissible hypothesis space family. For all $\varepsilon > 0$, if the number of examples n of each environment satisfies:*

$$n \geq \max\left\{ \frac{64}{m\varepsilon^2} \log \frac{4\mathcal{C}(\frac{\varepsilon}{16}, \mathbb{H}^m)}{\delta}, \frac{16}{\varepsilon^2} \right\}$$

Then with probability at least $1-\delta$ (over the choice of $\{\hat{\mathcal{P}}_e\}$), any $(f+g_1, \dots, f+g_m)$ will satisfy

$$\frac{1}{m}\sum_{e\in E} \mathrm{er}_{\mathcal{P}_e}(f+g_e) \leq \frac{1}{m}\sum_{e\in E} \hat{\mathrm{er}}_{\hat{\mathcal{P}}_e}(f+g_e) + \varepsilon$$

Note that permissibility (as defined in [S2]) is a weak measure-theoretic condition satisfied by many real world hypothesis space families [S2]. We will now express the capacity of $\mathbb{H}^m$ in terms of the capacities of its two constituent component-spaces $\hat{\mathcal{F}}$ and $\hat{\mathcal{G}}$, thus leading to the main result.

Proposition S1. *For all $\varepsilon, \varepsilon_1, \varepsilon_2 > 0$ such that $\varepsilon = \varepsilon_1 + \varepsilon_2$,*

$$\log \mathcal{C}(\varepsilon, \mathbb{H}^m) \leq \log \mathcal{C}_{\hat{\mathcal{G}}}(\varepsilon_1, \hat{\mathcal{F}}) + m \log \mathcal{C}_{\hat{\mathcal{F}}}(\varepsilon_2, \hat{\mathcal{G}}) \tag{S2}$$

Proof of Proposition S1. To prove the proposition it is sufficient to show the property of covering sets for any joint distribution defined on all environments $\boldsymbol{\mathcal{P}}$ on the space $(\mathcal{A} \times T\mathcal{A})^m$. Let us then fix such a distribution $\boldsymbol{\mathcal{P}}$. and let $\bar{\mathcal{P}} = \frac{1}{m}\sum_{e\in E} \mathcal{P}_e$ be the average distribution.

Suppose that F is an ε_1-cover of $(\hat{\mathcal{F}}, d_{[\bar{\mathcal{P}},\hat{\mathcal{G}}]})$ and $\{G_e\}_{e\in E}$ are ε_2-covers of $(\hat{\mathcal{G}}, d_{[\mathcal{P}_e,\hat{\mathcal{F}}]})$. Let $H = \{(x_1, \dots, x_m) \mapsto ((f+g_1)(x_1), \dots, (f+g_m)(x_m)),\ f \in F,\ g_e \in G_e\}$, be a set built from the covering sets aforementioned. Note that by definition $|H| = |F| \cdot \prod_{e\in E} |G_e| \leq \mathcal{C}_{\hat{\mathcal{G}}}(\varepsilon_1, \hat{\mathcal{F}}) \mathcal{C}_{\hat{\mathcal{F}}}(\varepsilon_2, \hat{\mathcal{G}})^m$ as we take some distribution instances.

For each learner $(f+g_1, \dots, f+g_m) \in \mathbb{H}^m$ in the hypothesis space, we take any $f' \in F$ such that $d_{[\bar{\mathcal{P}},\hat{\mathcal{G}}]}(f, f') \leq \varepsilon_1$ and $g'_e \in G_e$ for all e such that $d_{[\mathcal{P}_e,\hat{\mathcal{F}}]}(g_e, g'_e) \leq \varepsilon_2$, and we build $(f'+g'_1, \dots, f'+g'_m)$. The distance is then:

$$\begin{aligned}
&d_{\boldsymbol{\mathcal{P}}}((f+g_1, \dots, f+g_m), (f'+g'_1, \dots, f'+g'_m)) \\
&\leq d_{\boldsymbol{\mathcal{P}}}((f+g_1, \dots, f+g_m), (f'+g_1, \dots, f'+g_m)) \\
&\quad + d_{\boldsymbol{\mathcal{P}}}((f'+g_1, \dots, f'+g_m), (f'+g'_1, \dots, f'+g'_m)) \\
&\qquad\qquad \text{(triangular inequality of pseudo-metric)} \\
&\leq \frac{1}{m}\left[\sum_{e\in E} d_{\mathcal{P}_e}(f+g_e, f'+g_e) + \sum_{e\in E} d_{\mathcal{P}_e}(f'+g_e, f'+g'_e)\right] \\
&\qquad\qquad \text{(triangular inequality of absolute value)} \\
&\leq \frac{1}{m}\sum_{e\in E} d_{[\mathcal{P}_e,\hat{\mathcal{G}}]}(f, f') + \frac{1}{m}\sum_{e\in E} d_{[\mathcal{P}_e,\hat{\mathcal{F}}]}(g_e, g'_e) \qquad \text{(by definition of } d_{[\mathcal{P}_e,\hat{\mathcal{G}}]} \text{ and } d_{[\mathcal{P}_e,\hat{\mathcal{F}}]}) \\
&= d_{[\bar{\mathcal{P}},\hat{\mathcal{G}}]}(f, f') + \frac{1}{m}\sum_{e\in E} d_{[\mathcal{P}_e,\hat{\mathcal{F}}]}(g_e, g'_e) \leq \varepsilon_1 + \varepsilon_2 \\
&\qquad\qquad \text{(mean of the distance on different } \mathcal{P}_e \text{ is the distance on } \bar{\mathcal{P}})
\end{aligned}$$

To conclude, for any distribution $\boldsymbol{\mathcal{P}}$, when F is an ε_1-cover of $\hat{\mathcal{F}}$ and $\{G_e\}$ are ε_2-covers of $\hat{\mathcal{G}}$, the set H built upon them is an $(\varepsilon_1+\varepsilon_2)$-cover of $\mathbb{H}^m$. Then if we take the maximum over all distributions we conclude that $\mathcal{C}(\varepsilon_1+\varepsilon_2, \mathbb{H}^m) \leq \mathcal{C}_{\hat{\mathcal{G}}}(\varepsilon_1, \hat{\mathcal{F}}) \mathcal{C}_{\hat{\mathcal{F}}}(\varepsilon_2, \hat{\mathcal{G}})^m$ and we have Eq. S2. ∎

We can now use the bound developed in Prop. S1 and use it together with Theorem S1, therefore concluding the proof of Prop. 2. □

B.2.2 Proof of Proposition 3

Proposition 3. *For all ε, δ with $0 < \varepsilon, \delta < 1$ if the number of samples n' satisfies*

$$n' \geq \max\left\{\frac{64}{\varepsilon^2}\log\frac{4\mathcal{C}(\frac{\varepsilon}{16}, f+\hat{\mathcal{G}})}{\delta}, \frac{16}{\varepsilon^2}\right\}, \tag{6}$$

then with probability at least $1-\delta$ (over the choice of novel training set $\hat{\mathcal{P}}_{e'}$), any learner $f + g_{e'} \in f + \hat{\mathcal{G}}$ will satisfy $\mathrm{er}_{\mathcal{P}_{e'}}(f+g_{e'}) \leq \hat{\mathrm{er}}_{\hat{\mathcal{P}}_{e'}}(f+g_{e'}) + \varepsilon$.

Proof. The proof is derived from the following theorem which can be easily adapted to our context:

Theorem S2 ([S2], Theorem 3)**.** *Let $\mathcal{H}$ a permissible hypothesis space. For all $0 < \varepsilon, \delta < 1$, if the number of examples n of each environment satisfies:*

$$n \geq \max\left\{\frac{64}{m\varepsilon^2}\log\frac{4\mathcal{C}(\frac{\varepsilon}{16}, \mathcal{H})}{\delta}, \frac{16}{\varepsilon^2}\right\}$$

Then with probability at least $1-\delta$ (over the choice of dataset $\hat{\mathcal{P}}$ sampled from $\mathcal{P}$), any $h \in \mathcal{H}$ will satisfy

$$\mathrm{er}_{\mathcal{P}}(h) \leq \hat{\mathrm{er}}_{\hat{\mathcal{P}}}(h) + \varepsilon$$

Given that $\hat{\mathcal{P}}_{e'}$ is sampled from the same environment distribution Q, then by fixing the pre-trained f, we fix the space of hypothesis to $f + \hat{\mathcal{G}}$, and we apply the Theorem S2 to obtain the proposition. □

B.3 Linear case

We provide here the proofs of theoretical bounds given in Sec. 3.2. See the description in Sup. D for the detailed information on the example linear ODE dataset and the training with varying number of environments.

B.3.1 Proof of Proposition 4

Proposition 4. *If for all linear maps $\mathrm{L}_{\boldsymbol{G}_e} \in \hat{\mathcal{G}}$, $\|\boldsymbol{G}\|_F^2 \leq r$, if the input space is bounded s.t. $\|x\|_2 \leq b$, and the MSE loss function is bounded by c, then*

$$\log \mathcal{C}_{\hat{\mathcal{F}}}(\varepsilon, \hat{\mathcal{G}}) \leq \lceil rcd(2b)^2/\varepsilon^2 \rceil \log 2d^2 =: \omega(r, \varepsilon)$$

Proof. Let us take G an $\frac{\varepsilon}{2\sqrt{c}}$-cover of $\hat{\mathcal{G}}$ with L^2-distance: $d_{L^2(\mathcal{P})}$ (see definition in Table S1). Therefore, for each $\mathrm{L}_{\boldsymbol{G}} \in \hat{\mathcal{G}}$ take $g' \in G$ such that $d_{L^2}(\mathrm{L}_{\boldsymbol{G}}, \mathrm{L}_{\boldsymbol{G}'}) \leq \frac{\varepsilon}{2\sqrt{c}}$, then

$$\begin{aligned}
&d_{[\mathcal{P},\hat{\mathcal{F}}]}(\mathrm{L}_{\boldsymbol{G}}, \mathrm{L}_{\boldsymbol{G}'}) \\
&= \int_{\mathcal{A}\times\mathcal{A}'} \sup_{\mathrm{L}_{\boldsymbol{F}}\in\hat{\mathcal{F}}} |\|(\boldsymbol{F}+\boldsymbol{G})x - y\|_2^2 - \|(\boldsymbol{F}+\boldsymbol{G}')x - y\|_2^2| \mathrm{d}\mathcal{P}(x,y) \\
&\leq \int_{\mathcal{A}\times T\mathcal{A}} \sup_{\mathrm{L}_{\boldsymbol{F}}\in\hat{\mathcal{F}}} \|(\boldsymbol{G}-\boldsymbol{G}')x\|_2 (\|(\boldsymbol{F}+\boldsymbol{G})x - y\|_2 + \|(\boldsymbol{F}+\boldsymbol{G}')(x) - y\|_2) \mathrm{d}\mathcal{P}(x,y) \\
&\leq \sqrt{\int_{\mathcal{A}} \|(\boldsymbol{G}-\boldsymbol{G}')x\|_2 \mathrm{d}\mathcal{P}(x)} \sqrt{\int_{\mathcal{A}\times T\mathcal{A}} \sup_{\mathrm{L}_{\boldsymbol{F}}\in\hat{\mathcal{F}}} (\|(\boldsymbol{F}+\boldsymbol{G})x - y\|_2 + \|(\boldsymbol{F}+\boldsymbol{G}')x - y\|_2)^2 \mathrm{d}\mathcal{P}(x,y)} \\
&\leq 2\sqrt{c}\sqrt{\int_{\mathbb{R}^d} \|(\boldsymbol{G}-\boldsymbol{G}')x\|_2 \mathrm{d}\mathcal{P}(x)} \leq \varepsilon
\end{aligned}$$

We have the $\mathcal{C}_{\mathcal{F}}(\varepsilon, \hat{\mathcal{G}}) \leq \mathcal{C}(\frac{\varepsilon}{2\sqrt{c}}, \hat{\mathcal{G}}, L^2)$. According to the following lemma:

Lemma S1 ([S1], Lemma 3.2, Adapted)**.** *Given positive reals (a, b, ε) and positive integer d. Let vector $x \in \mathbb{R}^d$ be given with $\|x\|_p \leq b$, $\hat{\mathcal{G}} = \{\mathrm{L}_{\boldsymbol{G}} : \boldsymbol{G} \in \mathbb{R}^{d\times d}, \|\boldsymbol{G}\|_F^2 \leq r\}$ where $\|\cdot\|_F$ is the*

Frobenius norm. Then

$$\log \mathcal{C}(\varepsilon, \hat{\mathcal{G}}, L^2) \leq \left\lceil \frac{rdb^2}{\varepsilon^2} \right\rceil \log 2d^2$$

And we obtain that

$$\log \mathcal{C}_{\hat{\mathcal{F}}}(\varepsilon, \hat{\mathcal{G}}) \leq \left\lceil \frac{rcd(2b)^2}{\varepsilon^2} \right\rceil \log 2d^2 =: \omega(r, \varepsilon)$$

where $\omega(r, \varepsilon)$ is a strictly increasing function w.r.t. r. □

B.3.2 Proof of Proposition 5

Proposition 5. *If for linear maps* $\mathrm{L}_{\boldsymbol{F}} \in \hat{\mathcal{F}}$, $\|\boldsymbol{F}\|_F^2 \leq r'$, $\mathrm{L}_{\boldsymbol{G}} \in \hat{\mathcal{G}}$, $\|\boldsymbol{G}\|_F^2 \leq r$, $\|x\|_2 \leq b$, *and if the MSE loss function is bounded by* c, *given* m *environments and* n *samples per environment, with the probability* $1-\delta$, *the generalization error upper bound is* $\varepsilon = \max\{\sqrt{(p+\sqrt{p^2+4q})/2}, \sqrt{16/n}\}$ *where* $p = \frac{64}{mn}\log\frac{4}{\delta}$ *and* $q = \frac{64}{n}\lceil(\frac{r'}{mz^2} + \frac{r}{(1-z)^2})cd(32b)^2\rceil \log 2d^2$ *for any* $0 < z < 1$.

Proof. This can be derived from Prop. 2 with the help of Prop. 4 for linear maps. If we take the lower bounds of two capacities $\log \mathcal{C}_{\hat{\mathcal{F}}}(\frac{\varepsilon_1}{16}, \hat{\mathcal{G}})$ and $\log \mathcal{C}_{\hat{\mathcal{G}}}(\frac{\varepsilon_2}{16}, \hat{\mathcal{F}})$ for the linear maps hypothesis spaces $\hat{\mathcal{F}}, \hat{\mathcal{G}}$, then the number of required samples per environment n now can be expressed as follows:

$$n = \max\left\{\frac{64}{\varepsilon^2}\left(\frac{1}{m}\log\frac{4}{\delta} + \frac{1}{m}\left\lceil\frac{r'cd(32b)^2}{\varepsilon_1^2}\right\rceil \log 2d^2 + \left\lceil\frac{rcd(32b)^2}{\varepsilon_2^2}\right\rceil \log 2d^2\right), \frac{16}{\varepsilon^2}\right\}$$

To simplify the resolution of the equation above, we take $\varepsilon_1 = z\varepsilon$ for any $0 < z < 1$, then $\varepsilon_2 = \varepsilon - \varepsilon_1 = (1-z)\varepsilon$. Then by resolving the equation, the generalization margin is then upper bounded by ε with:

$$\varepsilon = \max\left\{\sqrt{\frac{p+\sqrt{p^2+4q}}{2}}, \sqrt{\frac{16}{n}}\right\}$$

where $p = \frac{64}{mn}\log\frac{4}{\delta}$ and $q = \frac{64}{n}\left\lceil\left(\frac{r}{mz^2} + \frac{r'}{(1-z)^2}\right)cd(32b)^2\right\rceil \log 2d^2$. □

B.4 Nonlinear case: instantiation for neural networks

We show in this section how we design a concrete model for nonlinear dynamics following the general guidelines given in Sec. 3.1. This is mainly composed of the following two parts: (a) choosing an appropriate approximation space and (b) choosing a penalization function Ω for this space. It is important to note that, even if the bounds given in the following sections may be loose in general, it could provide useful intuitions on the design of the algorithms which can be validated by experiments in our case.

B.4.1 Choosing approximation space $\hat{\mathcal{F}}$

We choose the space of feed-forward neural networks with a fixed architecture. Given the universal approximation properties of neural networks [S5], and the existence of efficient optimization algorithms [S3], this is a reasonable choice, but other families of approximating functions could be used as well.

We then consider the function space of neural networks with D-layers with inputs and outputs in $\mathbb{R}^d$: $\hat{\mathcal{F}}_{\text{NN}} = \{\nu : x \mapsto \sigma_D(W_D \cdots \sigma_1(W_1 x))) : x, \nu(x) \in \mathbb{R}^d\}$, D is the depth of the network, σ_j is a Lipschitz activation function at layer j, and W_j weight matrix from layer $j-1$ to j. The number of adjustable parameters is fixed to W for the architecture. This definition covers fully connected NNs and convolutional NNs. Note that the Fourier Neural Operator [S6] used in the experiments for NS can be also covered by the definition above, as it performs alternatively the convolution in the Fourier space.

B.4.2 Choosing penalization Ω

Now we choose an Ω for the space above. Let us first introduce a practical way to bound the capacity of $\hat{\mathcal{G}} \in \hat{\mathcal{F}}_{\text{NN}}$. Proposition S2 tells us that for a fixed NN architecture (implying constant parameter

number W and depth D), we can control the capacity through the maximum output norm R and Lipschitz norm L defined in the proposition.

Proposition S2. *If for all neural network $g \in \hat{\mathcal{G}}$, $\|g\|_\infty = \operatorname{ess\,sup}|g| \leq R$ and $\|g\|_{\text{Lip}} \leq L$, with $\|\cdot\|_{\text{Lip}}$ the Lipschitz semi-norm, then:*

$$\log \mathcal{C}_{\hat{\mathcal{F}}}(\varepsilon, \hat{\mathcal{G}}) \leq \omega(R, L, \varepsilon) \tag{S3}$$

where $\omega(R, L, \varepsilon) = c_1 \log \frac{RL}{\varepsilon} + c_2$ for $c_1 = 2W$ and $c_2 = 2W \log 8e\sqrt{c}D$, with c the bound of MSE loss. $\omega(R, L, \varepsilon)$ is a strictly increasing function w.r.t. R and L.

Proof. To link the capacity to some quantity that can be optimized for neural networks, we need to apply the following theorem:

Theorem S3 ([S4], Theorem 11, Adapted)**.** *With the neural network function space $\hat{\mathcal{F}}_{NN}$, let W be the total number of adjustable parameters, D the depth of the architecture. Let $\hat{\mathcal{G}} \subseteq \hat{\mathcal{F}}_{NN}$ be all functions into $[-R, R]^d$ representable on the architecture, and all these functions are at most L-Lipschitz. Then for all $0 < \varepsilon < 2R$,*

$$\mathcal{C}(\varepsilon, \hat{\mathcal{G}}, L^1) \leq \left(\frac{2e \cdot 2R \cdot DL}{\varepsilon}\right)^{2W}$$

Here, we need to prove firstly that the $\hat{\mathcal{F}}$-dependent capacity of $\hat{\mathcal{G}}$ is bounded by a scaled independent capacity on L^1 of itself. We suppose that the MSE loss function (used in the definitions in Table S1) is bounded by some constant c. This is a reasonable assumption given that the input and output of neural networks are bounded in a compact set. Let us take G an $\frac{\varepsilon}{2\sqrt{c}}$-cover of $\hat{\mathcal{G}}$ with L^1-distance: $d_{L^1(\mathcal{P})}$ (see definition in Table S1). Therefore, for each $g \in \hat{\mathcal{G}}$ take $g' \in G$ such that $d_{L^1}(g, g') \leq \frac{\varepsilon}{2\sqrt{c}}$, then

$$\begin{aligned}
d_{[\mathcal{P}, \hat{\mathcal{F}}]}(g, g') &= \int_{\mathcal{A} \times \mathcal{A}'} \sup_{f \in \hat{\mathcal{F}}} |\|(f+g)(x) - y\|_2^2 - \|(f+g')(x) - y\|_2^2| \mathrm{d}\mathcal{P}(x, y) \\
&\leq \int_{\mathcal{A} \times T\mathcal{A}} \sup_{f \in \hat{\mathcal{F}}} \|(g - g')(x)\|_2 (\|(f+g)(x) - y\|_2 + \|(f+g')(x) - y\|_2) \mathrm{d}\mathcal{P}(x, y) \\
&\leq 2\sqrt{c} \int_{\mathbb{R}^d} \|(g - g')(x)\|_1 \mathrm{d}\mathcal{P}(x) \leq \varepsilon
\end{aligned}$$

Then we have the first inequality $\mathcal{C}_{\mathcal{F}}(\varepsilon, \hat{\mathcal{G}}) \leq \mathcal{C}(\frac{\varepsilon}{2c}, \hat{\mathcal{G}}, L^1)$. As we suppose that $\|g\|_\infty \leq R$ for all $g \in \hat{\mathcal{G}}$, then for all $g \in \hat{\mathcal{G}}$, we have $g(x) \in [-R, R]^d$. We now apply the Theorem S3 on $\hat{\mathcal{G}}$, we then have the following inequality

$$\log \mathcal{C}\left(\frac{\varepsilon}{2\sqrt{c}}, \hat{\mathcal{G}}, L^1\right) \leq 2W \log \frac{8e\sqrt{c}DRL}{\varepsilon} \tag{S4}$$

where e is the base of the natural logarithm, W is the number of parameters of the architecture, D is the depth of the architecture. Then if we consider W, c, D as constants, the bound becomes:

$$\log \mathcal{C}\left(\frac{\varepsilon}{2\sqrt{c}}, \hat{\mathcal{G}}, L^1\right) \leq c_1 \log \frac{RL}{\varepsilon} + c_2 = \omega(R, L, \varepsilon) \tag{S5}$$

for $c_1 = 2W$ and $c_2 = 2W \log 8e\sqrt{c}D$. □

This leads us to choose for Ω a strictly increasing function that bounds $\omega(R, L, \varepsilon)$. Given the inequality (Eq. S3), this choice for Ω will allow us to bound practically the capacity of $\hat{\mathcal{G}}$.

Minimizing Ω will then reduce the effective capacity of the parametric set used to learn g_e. Concretely, we choose for Ω:

$$\Omega(g_e) = \|g_e\|_\infty^2 + \alpha \|g_e\|_{\text{Lip}}^2 \tag{7}$$

where $\alpha > 0$ is a hyper-parameter. This function is strictly convex and attains its unique minimum at the null function.

With this choice, let us instantiate Prop. 2 for our familly of NNs. Let $r = \sup_{g \in \hat{\mathcal{G}}} \Omega(g)$, and $\omega(r, \varepsilon) = c_1 \log \frac{r}{\varepsilon\sqrt{\alpha}} + c_2$ (strictly increasing w.r.t. the r) for given parameters $c_1, c_2 > 0$. We have:

Table S2: Details for the results of evaluation error in test on linear systems in Fig. 1 .

Samples/env.	Method	$m = 1$	$m = 2$	$m = 4$	$m = 8$
$n = 2 \cdot K$	*LEADS no min.*	8.13±5.56 e-2	6.81±4.44 e-2	4.92±4.26 e-2	4.50±3.10 e-2
	LEADS (Ours)		**5.11±3.20 e-2**	**3.93±2.88 e-2**	**2.10±0.96 e-2**
$n = 4 \cdot K$	*LEADS no min.*	4.08±2.57 e-2	3.96±2.56 e-2	3.10±2.08 e-2	2.23±1.44 e-2
	LEADS (Ours)		**2.74±1.96 e-2**	**1.61±1.24 e-2**	**1.02±0.74 e-2**

Proposition S3. *If $r = \sup_{g \in \hat{\mathcal{G}}} \Omega(g)$ is finite, the number of samples n in Eq. 5, required to satisfy the error bound in Proposition 2 with the same $\delta, \varepsilon, \varepsilon_1$ and ε_2 becomes:*

$$n \geq \max\left\{\frac{64}{\varepsilon^2}\left(\frac{1}{m}\log\frac{4\mathcal{C}_{\hat{\mathcal{G}}}(\frac{\varepsilon_1}{16}, \hat{\mathcal{F}})}{\delta} + \omega\left(r, \frac{\varepsilon_2}{16}\right)\right), \frac{16}{\varepsilon^2}\right\} \tag{S6}$$

Proof. If $\Omega(g_e) \leq r$, we have $2\log R \leq \log r$ and $2\log L + \log\alpha \leq \log r$, then

$$\log RL \leq \log \frac{r}{\sqrt{\alpha}}$$

We can therefore bound $\omega(R, L, \varepsilon)$ by

$$\omega(R, L, \varepsilon) = c_1 \log\frac{RL}{\varepsilon} + c_2 \leq c_1 \log\frac{r}{\varepsilon\sqrt{\alpha}} + c_2 = \omega(r, \varepsilon)$$

The result follows from Proposition S2. □

This means that the number of required samples will decrease with the size the largest possible $\Omega(g) = r$. The optimization process will reduce $\Omega(g_e)$ until a minimum is reached. The maximum size of the effective hypothesis space is then bounded and decreases throughout training. In particular, the following result follows:

Corollary S1. *Optimizing Eq. 4 for a given λ, we have that the number of samples n in Eq. 5 required to satisfy the error bound in Proposition 2 with the same $\delta, \varepsilon, \varepsilon_1$ and ε_2 is:*

$$n \geq \max\left\{\frac{64}{\varepsilon^2}\left(\frac{1}{m}\log\frac{4\mathcal{C}_{\hat{\mathcal{G}}}(\frac{\varepsilon_1}{16}, \hat{\mathcal{F}})}{\delta} + \omega\left(\lambda\kappa, \frac{\varepsilon_2}{16}\right)\right), \frac{16}{\varepsilon^2}\right\} \tag{S7}$$

where $\kappa = \sum_{e\in E}\sum_{i=1}^{l}\int_0^T \left\|\frac{\mathrm{d}x_s^{e,i}}{\mathrm{d}t}\right\|^2 \mathrm{d}s$.

Proof. Denote $\mathcal{L}_\lambda(f, \{g_e\})$ the loss function defining Eq. 4. Consider a minimizer $(f^\star, \{g_e^\star\})$ of $\mathcal{L}_\lambda$. Then:

$$\mathcal{L}_\lambda(f^\star, \{g_e^\star\}) \leq \mathcal{L}_\lambda(0, \{0\}) = \kappa$$

which gives:

$$\forall e,\ \Omega(g_e^\star) \leq \sum_e \Omega(g_e^\star) \leq \lambda\kappa$$

Defining $\hat{\mathcal{G}} = \{g \in \hat{\mathcal{F}} \mid \Omega(g) \leq \lambda\kappa\}$, we then have that Eq. 4 is equivalent to:

$$\min_{f\in\hat{\mathcal{F}}, \{g_e\}_{e\in E}\in\hat{\mathcal{G}}} \sum_{e\in E}\left(\frac{\Omega(g_e)}{\lambda} + \sum_{i=1}^{l}\int_0^T \left\|\frac{\mathrm{d}x_s^{e,i}}{\mathrm{d}t} - (f + g_e)(x_s^{e,i})\right\|^2 \mathrm{d}s\right) \tag{S8}$$

and the result follows from Proposition S3. □

We can then decrease the sample complexity in the chosen NN family by: (a) increasing the number of training environments engaged in the framework, and (b) decreasing $\Omega(g_e)$ for all g_e, with $\Omega(g_e)$ instantiated as in Sec. 3.1. Ω provides a bound based on the largest output norm and the Lipschitz constant for a family of NNs. The experiments (Sec. 4) confirm that this is indeed an effective way to control the capacity of the approximating function family. Note that in our experiments, the number of samples needed in practice is much smaller than suggested by the theoretical bound.

Table S3: Detailed results of evaluation error in test on LV systems for Fig. 4. For the case of $m = 1$, all baselines except FT-RNN are equivalent to *One-Per-Env.*. The arrows indicate that the table cells share the same value.

Samples/env.	Method	$m = 1$	$m = 2$	$m = 4$	$m = 8$
$n = 1 \cdot K$	*One-For-All*	7.87±7.54 e-3	0.22±0.06	0.33±0.06	0.47±0.04
	One-Per-Env.	7.87±7.54 e-3	→	→	→
	FT-RNN	4.02±3.17 e-2	1.62±1.14 e-2	1.62±1.40 e-2	1.08±1.03 e-2
	FT-NODE	7.87±7.54 e-3	7.63±5.84 e-3	4.18±3.77 e-3	4.92±4.19 e-3
	GBML-like	7.87±7.54 e-3	6.32±5.72 e-2	1.44±0.66 e-1	9.85±8.84 e-3
	LEADS (Ours)	7.87±7.54 e-3	**3.65±2.99 e-3**	**2.39±1.83 e-3**	**1.37±1.14 e-3**
$n = 2 \cdot K$	*One-For-All*	1.38±1.61 e-3	0.22±0.04	0.36±0.07	0.60±0.11
	One-Per-Env.	1.38±1.61 e-3	→	→	→
	FT-RNN	7.20±7.12 e-2	2.72±4.00 e-2	1.69±1.57 e-2	1.38±1.25 e-2
	FT-NODE	1.38±1.61 e-3	9.02±8.81 e-3	1.11±1.05 e-3	1.00±0.95 e-3
	GBML-like	1.38±1.61 e-3	9.26±8.27 e-3	1.17±1.09 e-2	1.96±1.95 e-2
	LEADS (Ours)	1.38±1.61 e-3	**8.65±9.61 e-4**	**8.40±9.76 e-4**	**6.02±6.12 e-4**
$n = 4 \cdot K$	*One-For-All*	1.36±1.25 e-4	0.19±0.02	0.31±0.04	0.50±0.04
	One-Per-Env.	1.36±1.25 e-4	→	→	→
	FT-RNN	8.69±8.36 e-4	3.39±3.38 e-4	3.02±1.50 e-4	2.26±1.45 e-4
	FT-NODE	1.36±1.25 e-4	1.74±1.65 e-4	1.78±1.71 e-4	1.39±1.20 e-4
	GBML-like	1.36±1.25 e-4	2.57±7.18 e-3	2.65±3.26 e-3	2.36±3.58 e-3
	LEADS (Ours)	1.36±1.25 e-4	**1.10±0.92 e-4**	**1.03±0.98 e-4**	**9.66±9.79 e-5**
$n = 8 \cdot K$	*One-For-All*	5.98±5.13 e-5	0.16±0.03	0.35±0.06	0.52±0.06
	One-Per-Env.	5.98±5.13 e-5	→	→	→
	FT-RNN	2.09±1.73 e-4	1.18±1.16 e-4	1.13±1.13 e-4	9.13±8.31 e-5
	FT-NODE	5.98±5.13 e-5	6.91±4.46 e-5	7.82±6.95 e-5	6.88±6.39 e-5
	GBML-like	5.98±5.13 e-5	1.02±1.68 e-4	1.41±2.68 e-4	0.99±1.53 e-4
	LEADS (Ours)	5.98±5.13 e-5	**5.47±4.63 e-5**	**4.52±3.98 e-5**	**3.94±3.49 e-5**

Table S4: Test MSE of experiments on LV ($m = 4, n = 1 \cdot K$) with different empirical norms.

Empirical Norm	$p = 1$	$p = 2$	$p = 3$	$p = 10$	$p = \infty$
Test MSE	2.30e-3	2.36e-3	2.34e-3	3.41e-3	6.12e-3

C Optimizing Ω in practice

In Sec. 3.3, we developed an instantiation of the *LEADS* framework for neural networks. We proposed to control the capacity of the g_es components through a penalization function Ω defined as $\Omega(g_e) = \|g_e\|^2_\infty + \alpha\|g_e\|^2_{\text{Lip}}$. This definition ensures the properties required to control the sample complexity.

However, in practice, both terms in $\Omega(g_e)$ are difficult to compute as they do not yield an analytical form for neural networks. For a fixed activation function, the Lipschitz-norm of a trained model only depends on the model parameters and, for our class of neural networks, can be bounded by the spectral norms of the weight matrices, as described in Sec. 4.4. This allows for a practical implementation.

The infinity norm on its side depends on the domain definition of the function and practical implementations require an empirical estimate. Since there is no trivial estimator for the infinity norm of a function, we performed tests with different proxies such as the *empirical* L^p and L^∞ norms, respectively defined as $\|g_e\|_{L^p(\hat{\mathcal{P}}_e)} = \left(\frac{1}{n}\sum_{x\in\hat{\mathcal{P}}_e}|g_e(x)|^p\right)^{1/p}$ for $1 \leq p < \infty$ and $\|g_e\|_{L^\infty(\hat{\mathcal{P}}_e)} = \max_{x\in\hat{\mathcal{P}}_e}|g_e(x)|$. Here $|\cdot|$ is an ℓ^2 vector norm. Note that on a finite set of points, these norms reduce to vector norms $\|(|g_e(x_1)|, \ldots, |g_e(x_n)|)^\top\|_p$. They are then all equivalent on the space defined by the training set. Table S4 shows the results of experiments performed on LV equation with different $1 \leq p \leq \infty$. Overall we found that L^p for small values of p worked better and chose in our experiments set $p = 2$.

Moreover, using both minimized quantities $\|g_e\|^2_{L^2(\hat{\mathcal{P}}_e)}$ and the spectral norm of the product of weight matrices, denoted $L(g_e)$ and $\Pi(g_e)$ respectively, we can give a bound on $\Omega(g_e)$. First, for any x in the compact support of $\mathcal{P}_e$, we have that, fixing some $x_0 \in \hat{\mathcal{P}}_e$:

$$|g_e(x)| \leq |g_e(x) - g_e(x_0)| + |g_e(x_0)|$$

For the first term:

$$|g_e(x) - g_e(x_0)| \leq \|g_e\|_{\text{Lip}}|x - x_0| \leq \Pi(g_e)|x - x_0|$$

and the support of $\mathcal{P}_e$ being compact by hypothesis, denoting by δ its diameter:

$$|g_e(x) - g_e(x_0)| \leq \delta\Pi(g_e)$$

Moreover, for the second term:

$$|g_e(x_0)| = \sqrt{|g_e(x_0)|^2} \leq \sqrt{L(g_e)}$$

and summing both contributions gives us the bound:

$$\|g_e\|_\infty \leq \delta\Pi(g_e) + \sqrt{L(g_e)}$$

so that:

$$\Omega(g_e) \leq (\delta + \alpha)\Pi(g_e) + \sqrt{L(g_e)}$$

Note that this estimation is a crude one and improvements can be made by considering the closest x_0 from x and taking δ to be the maximal distance between points not from the support of $\mathcal{P}_e$ and $\hat{\mathcal{P}}_e$.

Finally, we noticed that minimizing $\|\frac{g_e}{id}\|^2_{L^2(\hat{\mathcal{P}}_e)}$ in domains bounded away from zero gave better results as normalizing by the norm of the output allowed to adaptively rescale the computed norm. Formally, minimizing this quantity does not fundamentally change the optimization as we have that:

$$\forall x, \frac{1}{M^2}|g_e(x)|^2 \leq \left|\frac{g_e(x)}{x}\right|^2 \leq \frac{1}{m^2}|g_e(x)|^2$$

meaning that:

$$\frac{1}{M^2}L(g_e) \leq \left\|\frac{g_e}{id}\right\|^2_{L^2(\hat{\mathcal{P}}_e)} \leq \frac{1}{m^2}L(g_e)$$

where m, M are the lower and upper bound of $|x|$ on the support of $\mathcal{P}_e$ with $m > 0$ by hypothesis (the quantity we minimize is still higher than $L(g_e)$ even if this is not the case).

D Additional experimental details

D.1 Details on the environment dynamics

Lotka-Volterra (LV). The model dynamics follow the ODE:

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \alpha u - \beta uv, {}^{\mathrm{d}v}/_{\mathrm{d}t} = \delta uv - \gamma v$$

with u, v the number of prey and predator, $\alpha, \beta, \gamma, \delta > 0$ defining how the two species interact. The initial conditions u_0^i, v_0^i are sampled from a uniform distribution $P_0 = \text{Unif}([1, 2]^2)$. We characterize the dynamics by $\theta = ({}^{\alpha}/_{\beta}, {}^{\gamma}/_{\delta}) \in \Theta = \{0.5, 1, 1.44, 1.5, 1.86, 2\}^2$. An environment e is then defined by parameters θ_e sampled from a uniform distribution over the parameter set Θ.

Gray-Scott (GS). The governing PDE is:

$$\frac{\partial u}{\partial t} = D_u\Delta u - uv^2 + F(1 - u), {}^{\partial v}/_{\partial t} = D_v\Delta v + uv^2 - (F + k)v$$

where the u, v represent the concentrations of two chemical components in the spatial domain S with periodic boundary conditions. D_u, D_v denote the diffusion coefficients respectively for u, v, and are held constant to $D_u = 0.2097, D_v = 0.105$, and F, k are the reaction parameters depending on the environment. As for the initial conditions $(u_0, v_0) \sim P_0$, we place 3 2-by-2 squares at uniformly sampled positions in S to trigger the reactions. The values of (u_0, v_0) are fixed to $(0, 1)$ outside the squares and to $(1 - \epsilon, \epsilon)$ with a small $\epsilon > 0$ inside. An environment e is defined by its parameters $\theta_e = (F_e, k_e) \in \Theta = \{(0.037, 0.060), (0.030, 0.062), (0.039, 0.058)\}$. We consider a set of θ_e parameters uniformly sampled from the environment distribution Q on Θ.

Navier-Stokes (NS). We consider the Navier-Stokes PDE for incompressible flows:

$$\frac{\partial w}{\partial t} = -v \cdot \nabla w + \nu \Delta w + \xi \qquad \nabla \cdot v = 0$$

where v is the velocity field, $w = \nabla \times v$ is the vorticity, both v, w lie in a spatial domain S with periodic boundary conditions, ν is the viscosity and ξ is the constant forcing term in the domain S. We fix $\nu = 10^{-3}$ across the environments. We sample the initial conditions $w_0^e \sim P_0$ as in [S6]. An environment e is defined by its forcing term $\xi_e \in \Theta_\xi = \{\xi_1, \xi_2, \xi_3, \xi_4\}$ with

$$\begin{aligned}
\xi_1(x, y) &= 0.1(\sin(2\pi(x + y)) + \cos(2\pi(x + y))) \\
\xi_2(x, y) &= 0.1(\sin(2\pi(x + y)) + \cos(2\pi(x + 2y))) \\
\xi_3(x, y) &= 0.1(\sin(2\pi(x + y)) + \cos(2\pi(2x + y))) \\
\xi_4(x, y) &= 0.1(\sin(2\pi(2x + y)) + \cos(2\pi(2x + y)))
\end{aligned}$$

where $(x, y) \in S$ is the position in the domain S. We uniformly sampled a set of forcing terms from Q on Θ_ξ.

Linear ODE. We take an example of linear ODE expressed by the following formula:

$$\frac{\mathrm{d}u_t}{\mathrm{d}t} = \mathrm{L}_{\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top}(u_t) = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top u_t$$

where $u_t \in \mathbb{R}^8$ is the system state, $\boldsymbol{Q} \in M_{8,8}(\mathbb{R})$ is an orthogonal matrix such that $\boldsymbol{Q}\boldsymbol{Q}^\top = 1$, and $\boldsymbol{\Lambda} \in M_{8,8}(\mathbb{R})$ is a diagonal matrix containing eigenvalues. We sample $\boldsymbol{\Lambda}_e$ from a uniform distribution on $\Theta_{\boldsymbol{\Lambda}} = \{\boldsymbol{\Lambda}_1, \ldots, \boldsymbol{\Lambda}_8\}$, defined for each $\boldsymbol{\Lambda}_i$ by:

$$[\boldsymbol{\Lambda}_i]_{jj} = \begin{cases} 0, & \text{if } i = j \text{ for } i, j \in \{1, \ldots, 8\}, \\ -0.5, & \text{otherwise.} \end{cases}$$

which means that the i-th eigenvalue is set to 0, while others are set to a common value -0.5.

D.2 Choosing hyperparameters

As usual, the hyperparameters need to be tuned for each considered set of systems. We therefore chose the hyperparameters using standard cross-validation techniques. We did not conduct a systematic sensitivity analysis. In practice, we found that: (a) if the regularization term is too large w.r.t. the trajectory loss, the model cannot fit the trajectories, and (b) if the regularization term is too small, the performance is similar to *LEADS no min.* The candidate hyperparameters are defined on a very sparse grid, for example, for neural nets, $(10^3, 10^4, 10^5, 10^6)$ for λ and $(10^{-2}, 10^{-3}, 10^{-4}, 10^{-5})$ for α.

D.3 Details on the experiments with a varying number of environments

We conducted large-scale experiments respectively for linear ODEs (Sec. 3.2, Fig. 1) and LV (Sec. 4, Fig. 4) to compare the tendency of *LEADS* w.r.t. the theoretical bound and the baselines by varying the number of environments available for the instantiated model.

To guarantee the comparability of the test-time results, we need to use the same test set when varying the number of environments. We therefore propose to firstly generate a global set of environments, separate it into subgroups for training, then we test these separately trained models on the global test set.

We performed the experiments as follows:

- In the training phase, we consider $M = 8$ environments in total in the environment set E_{total}. We denote here the cardinality of an environment set E by $\mathrm{card}(E)$, the environments are then arranged into $b = 1, 2, 4$ or 8 disjoint groups of the same size, i.e. $\{E_1, \ldots, E_b\}$ such that $\bigcup_{i=1}^{b} E_i = E_{\text{total}}$, $\mathrm{card}(E_1) = \cdots = \mathrm{card}(E_b) = \lfloor M/b \rfloor =: m$, where m is the number of environments per group, and $E_i \cap E_j = \emptyset$ whenever $i \neq j$. For example, for $m = 1$, all the original environments are gathered into one global environment, when for $m = 8$ we keep all the original environments. The methods are then instantiated respectively for each E_i. For example, for *LEADS* with b environment groups, we instantiate *LEADS*$_1, \ldots,$ *LEADS*$_b$ respectively on $E_1, \ldots, E_b$. Other frameworks are applied in the same way.

Table S5: Results on 2 novel environments for LV, GS, and NS at different traning steps with n data points per env. The arrows indicate that the table cells share the same value.

Dataset	Training Schema	Test MSE at training step		
		50	2500	10000
LV ($n = 1 \cdot K$)	*Pre-trained-f-Only*	0.36	$\longrightarrow$	
	One-Per-Env. from scratch	0.23	8.85e-3	3.05e-3
	Pre-trained-f-Plus-Trained-g_e	0.73	**1.36e-3**	**1.11e-3**
GS ($n = 1 \cdot K$)	*Pre-trained-f-Only*	5.44e-3	$\longrightarrow$	
	One-Per-Env. from scratch	4.20e-2	5.53e-3	3.05e-3
	Pre-trained-f-Plus-Trained-g_e	2.29e-3	**1.45e-3**	**1.27e-3**
NS ($n = 8 \cdot K$)	*Pre-trained-f-Only*	1.75e-1	$\longrightarrow$	
	One-Per-Env. from scratch	6.76e-2	1.70e-2	1.18e-2
	Pre-trained-f-Plus-Trained-g_e	1.37e-2	**8.07e-3**	**7.14e-3**

Note that when $m = 1$, having $b = 8$ environment groups of one single environment, *One-For-All*, *One-Per-Env.* and *LEADS* are reduced to *One-Per-Env.* applied on all M environments. We can see in Fig. 4 that each group of plots starts from the same point.

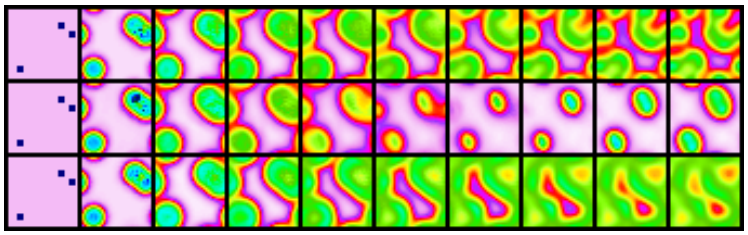
- In the test phase, the performance of the model trained with the group E_i is tested with the test samples of the corresponding group. Then we take the mean error over all b groups to obtain the results on all M environments. Note that the result at each point in figures 1 and 4 is calculated on the same total test set, which guarantees the comparability between results.

D.4 Additional experimental results

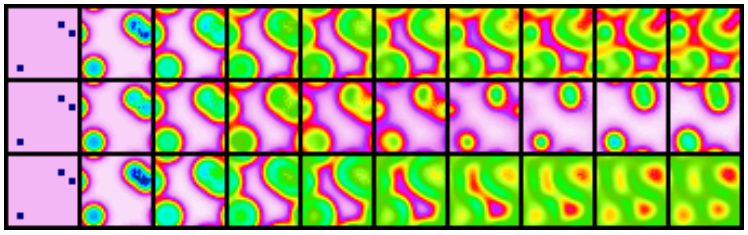
Experiments with a varying number of environments We show in tables S2 and S3 the detailed results used for the plots in figures 1 and 4, compared to baseline methods.

Learning in novel environments We conducted same experiments as in Sec. 4.3 to learn in unseen environments for GS and NS datasets. The test MSE at different training steps is shown in Table S5.

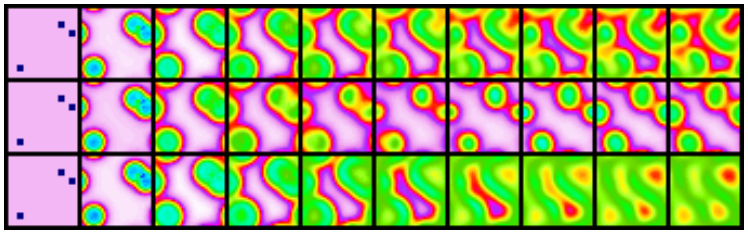
Full-length trajectories We provide in figures S1-S4 the full-length sample trajectories for GS and NS of Fig. 2.



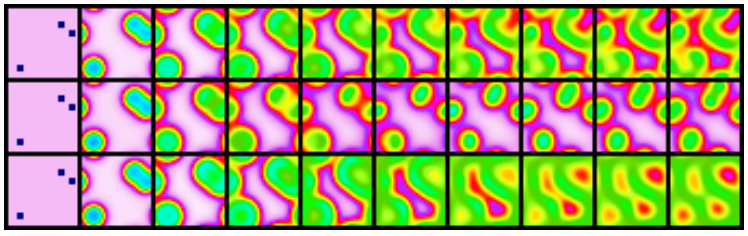
(a) *One-Per-Env.*



(b) FT-NODE

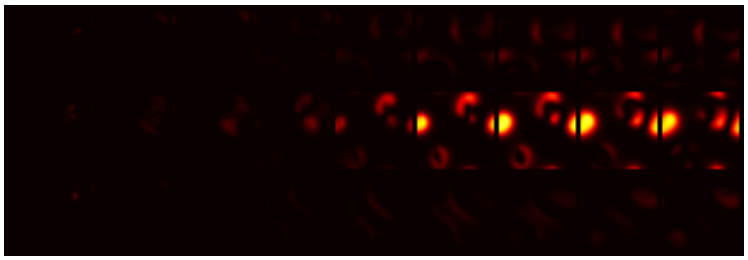


(c) *LEADS*

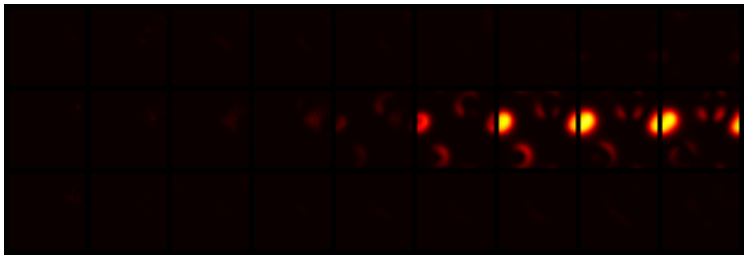


(d) Ground truth

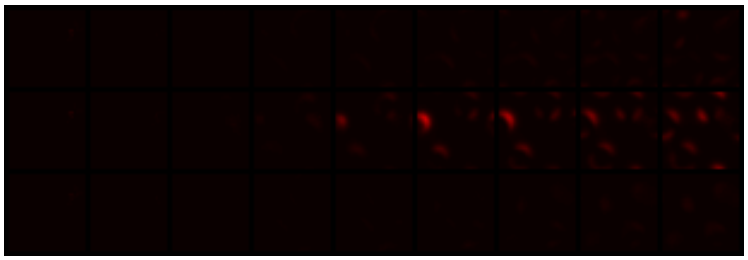
Figure S1: Full-length prediction comparison of Fig. 2 for GS. In each figure, from top to bottom, the trajectory snapshots are output respectively from 3 training environments. The temporal resolution of each sequence is $\Delta t = 40$.



(a) Difference between *One-Per-Env.* and Ground truth

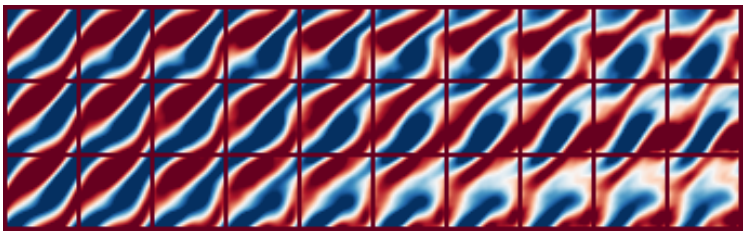


(b) Difference between FT-NODE and Ground truth

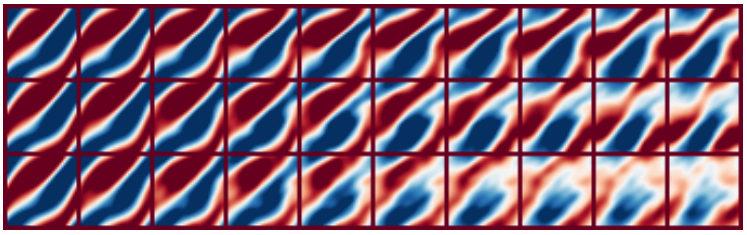


(c) Difference between *LEADS* and Ground truth

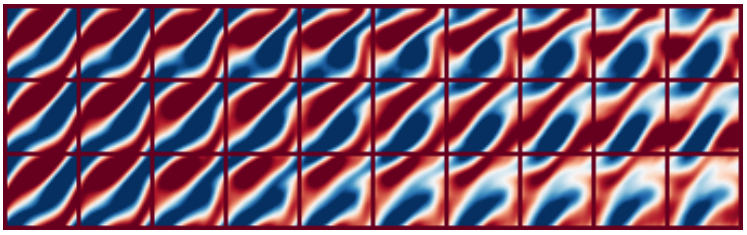
Figure S2: Full-length error maps of Fig. 2 for GS. In each figure, from top to bottom, the trajectory snapshots correspond to 3 training environments, one per row. The temporal resolution of each sequence is $\Delta t = 40$.



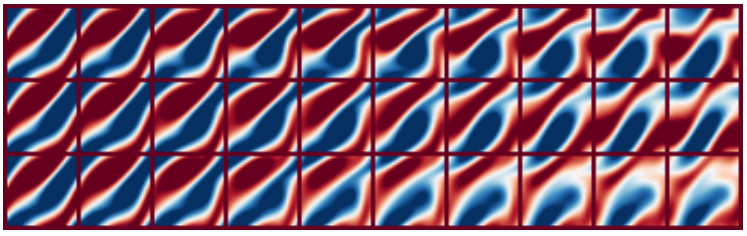
(a) *One-Per-Env.*



(b) FT-NODE

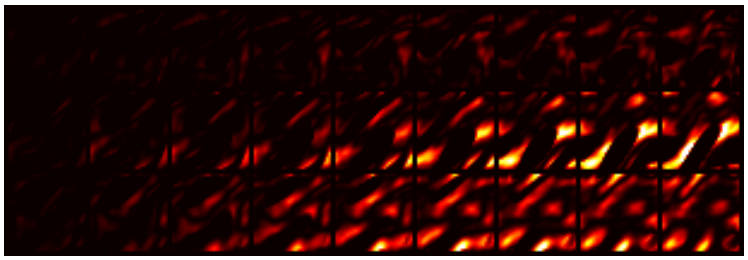


(c) *LEADS*

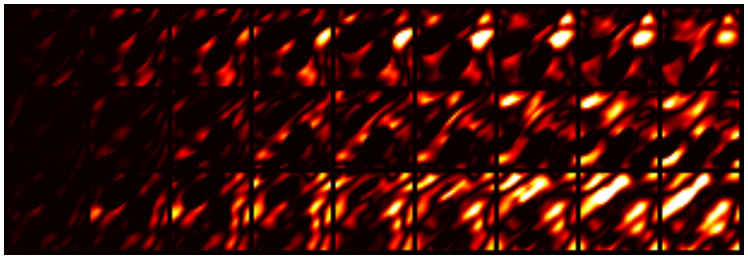


(d) Ground truth

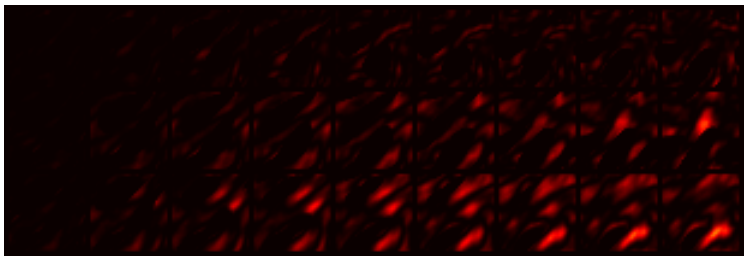
Figure S3: Full-length prediction comparison of Fig. 2 for NS. In each figure, from top to bottom, the trajectory snapshots correspond to 3 training environments. The temporal resolution of each sequence is $\Delta t = 1$.



(a) Difference between *One-Per-Env.* and Ground truth



(b) Difference between FT-NODE and Ground truth



(c) Difference between *LEADS* and Ground truth

Figure S4: Full-length error maps of Fig. 2 for NS. In each figure, from top to bottom, the trajectory snapshots correspond to from 3 training environments. The temporal resolution of each sequence is $\Delta t = 1$.

References

[S1] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky. Spectrally-normalized margin bounds for neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6240–6249. Curran Associates, Inc., 2017.

[S2] J. Baxter. A model of inductive bias learning. *J. Artif. Int. Res.*, 12(1):149–198, Mar. 2000.

[S3] L. Chizat and F. Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 3036–3046. Curran Associates, Inc., 2018.

[S4] D. Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Information and Computation*, 100(1):78 – 150, 1992.

[S5] P. Kidger and T. Lyons. Universal Approximation with Deep Narrow Networks. In J. Abernethy and S. Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 2306–2327. PMLR, 09–12 Jul 2020.

[S6] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

[S7] S. Shalev-Shwartz and S. Ben-David. *Covering Numbers*, page 337–340. Cambridge University Press, 2014.