When predict can also explain: few-shot prediction to select better neural latents

Kabir Dabholkar
Department of Mathematics
Technion - Israel Instite of Technology
kabir@campus.technion.ac.il

Omri Barak
Rappaport Faculty of Medicine and Network Biology Research Laboratory
Technion - Israel Instite of Technology
omri.barak@gmail.com

Abstract

Latent variable models serve as powerful tools to infer underlying dynamics from observed neural activity. However, due to the absence of ground truth data, prediction benchmarks are often employed as proxies. In this study, we reveal the limitations of the widely-used 'co-smoothing' prediction framework and propose an improved few-shot prediction approach that encourages more accurate latent dynamics. Utilizing a student-teacher setup with Hidden Markov Models, we demonstrate that the high co-smoothing model space can encompass models with arbitrary extraneous dynamics within their latent representations. To address this, we introduce a secondary metric – a few-shot version of co-smoothing. This involves performing regression from the latent variables to held-out channels in the data using fewer trials. Our results indicate that among models with near-optimal co-smoothing, those with extraneous dynamics underperform in the few-shot co-smoothing compared to 'minimal' models devoid of such dynamics. We also provide analytical insights into the origin of this phenomenon. We further validate our findings on real neural data using two state-of-the-art methods: LFADS and STNDT. In the absence of ground truth, we suggest a proxy measure to quantify extraneous dynamics. By cross-decoding the latent variables of all model pairs with high co-smoothing, we identify models with minimal extraneous dynamics. We find a correlation between few-shot co-smoothing performance and this new measure. In summary, we present a novel prediction metric designed to yield latent variables that more accurately reflect the ground truth, offering a significant improvement for latent dynamics inference.

1 Introduction

In neuroscience, we often have access to simultaneously recorded neurons during certain behaviors. These observations, denoted x, are believed to be a window onto the actual hidden (or latent) dynamics of the relevant brain circuit, denoted z [32]. To understand the underlying dynamics, we need to infer z given x. Finding the z variables is also known as latent variable modeling, which is part of the larger field of system identification with applications in many areas outside of neuroscience, such as fluid dynamics [31] and finance [2].

Because we don't have ground truth for z, prediction metrics on held-out parts of x are commonly used as a proxy [21]. However, it has been noted that prediction and explanation are often distinct

Preprint. Under review.

endeavors [27]. For instance, [30] use an example where ground truth is available to show how different models that all achieve good prediction nevertheless have varied latents that can differ from the ground truth. Such behavior might be expected when using highly expressive models with large latent spaces. Bad prediction with good latents is demonstrated by [12] for the case of chaotic dynamics.

Various regularization methods on the latents have been suggested to improve the similarity of z to the ground truth, such as recurrence and priors on external inputs [20], low-dimensionality of trajectories [25], low-rank connectivity [29], injectivity constraints from latent to predictions [30], low-tangling [22], piecewise-linear dynamics [12]. However, the field lacks a quantitative, prediction-based metric that credits the simplicity of the latent representation—an aspect essential for interpretability and ultimately scientific discovery, while still enabling comparisons across a wide range of LVM architectures.

Here, we characterize the diversity of model latents achieving high *co-smoothing*, a standard prediction-based framework for Neural LVMs, and demonstrate potential pitfalls of this framework. We propose a few-shot variant of co-smoothing which, when used in conjunction with co-smoothing, differentiates varying latents. We verify this approach both on synthetic toy problems and state-of-the-art methods on neural data, providing an analytical explanation of why it works in a simple setting.

2 Related Work

Our work builds on recent developments in Neural LVMs for the discovery of latent structure in noisy neural data on single trials. We refer the reader to [21] supplementary table 3 for a comprehensive list of Neural LVMs published from 2008-2021. Central to our work is the co-smoothing procedure for evaluating models based on the prediction of activity from held-out neurons provided held-in neuron activity from the same trial. Co-smoothing was first introduced in [37] and [14] for the validation of GPFA as a Neural LVM.

Pei et al. [21] curated four datasets of neural activity recorded from behaving monkeys and established a framework to evaluate co-smoothing among other prediction-based metrics on several models in the form of a standardized benchmark and competition. This led to significant advances in prediction methods. Some advances came in the form of harnessing ideas and tools from machine learning such as ensembling populations of models [33] and Bayesian hyperparameter optimization. Other advances involved incorporating statistical priors on the geometry of neural trajectories, yielding simpler and more interpretable methods [22].

In contrast to prediction approaches, a parallel line of work focuses on explaining and validating Neural LVMs on synthetic data, enabling direct comparison with the ground truth [3, 6, 25]. Versteeg et al. [30] validated their method with both ground truth and neural data, demonstrating similar predictive performance with low-dimensional latents.

Our work straddles the interface of prediction and explanation approaches. We propose a predictive framework that attempts to evaluate using unsupervised prediction metrics that can also reveal the similarity of the LVM to the ground truth without direct access to it. To verify our predictive method, we map ground truth states and latents with synthetic data to correlate with the predictive framework.

Central to our work is the concept of few-shot learning a decoder from a frozen intermediate representation. Sorscher et al. [28] developed a theory of geometric properties of representations that enables few-shot generalization to novel classes. They identified the geometric properties that determine a signal-to-noise ratio for classification, which dictates few-shot performance. While this setting differs from ours, links between our works are a topic for future research.

Another concept we introduce is cross-decoding across a population of models to find the most parsimonious representation. Several works compare representations of large model populations [15, 17]. They apply Canonical Correlation Analysis (CCA), a symmetric measure of representational similarity, whereas we use regression, which is not symmetric. The application to Neural LVMs may be novel.

To our knowledge, the use of few-shot generalization as a means to identify interpretable latent representations, particularly for Neural LVMs, is a novel idea.

3 Co-smoothing: a cross-validation framework

Let $\boldsymbol{x} \in \mathbb{Z}_{\geq 0}^{T \times N}$ be spiking neural activity of N channels recorded over a finite window of time, i.e a *trial*, and subsequently quantised into T time-bins. $x_{t,n}$ representing the number of spikes in channel n during time-bin t. The data set $X := \{\boldsymbol{x}^{(i)}\}_{i=1}^{S}$ consists of S trials of the experiment.

The latent-variable model (LVM) approach posits that each time-point in the data $\boldsymbol{x}_t^{(i)}$ is a noisy measurement of a latent state $\boldsymbol{z}_t^{(i)}$.

To infer the latent trajectory $\boldsymbol{z}$ is to learn a mapping $f : \boldsymbol{x} \mapsto \boldsymbol{z}$. On what basis do we decide the inferred $\boldsymbol{z}$? We have no ground-truth on z, so instead we test the ability of z to predict unseen or held-out data. Data may be held-out in time, e.g forward-prediction or in space, co-smoothing.[21] We focus here on co-smoothing.

The set of N available channels is partitioned into two: N^{in} held-in channels and N^{out} held-out channels. The S trials are partitioned into train and test. During training, both channel partitions are available to the model and during test, only the held-in partition is available. During evaluation the model must generate rate-predictions r^{out} for the held-out partition. This framework is visualised in figure 1A.

Importantly, the encoding-step or inference of the latents is done using a full time-window i.e analogous to *smoothing* in control-theoretic literature, whereas the decoding step, mapping the latents to predictions of the data is done on individual time-steps:

$$z_t = f(x^{\text{in}}_{1:T}; t) \tag{1}$$

$$r^{\text{out}}_t = g(z_t), \tag{2}$$

where the superscript of x denotes subsets of neurons, and the subscript denotes subsets of time bins. During evaluation the held-out data from test trials $\boldsymbol{x}^{\text{out}}$ is compared to the rate-predictions r^{out} from the model using the co-smoothing metric Q defined as the normalised log-likelihood given by:

$$Q(r_{t,n}, x_{t,n}) = \frac{1}{\mu_n \log 2}\Big(\mathcal{L}(r_{t,n}; x_{t,n}) - \mathcal{L}(\bar{r}_n; x_{t,n})\Big) \tag{3}$$

$$\mathcal{Q}(R^{\text{out}}_{\text{test}}, X^{\text{out}}_{\text{test}}) = \sum_{i \in \text{test}} \sum_{t=1}^{T} \sum_{n \in \text{held-out}} Q(\boldsymbol{r}^{(i)}_{t,n}, \boldsymbol{x}^{(i)}_{t,n}) \tag{4}$$

where $\mathcal{L}$ is poisson log-likelihood $\bar{r}_n = \frac{1}{TS}\sum_i \sum_t x^{(i)}_{t,n}$ is a the mean rate for channel n and $\mu_n := \sum_i \sum_t x^{(i)}_{t,n}$ is the total number of spikes, following [21]. We use the notation $X^{\circ}_{\star} := \{x^{(i)}_{1:T,n}\}_{i \in \star, n \in \circ}$, similarly for R.

Thus, inference of z is done by the optimization:

$$f^*, g^* = \text{argmax}_{f,g}\, \mathcal{Q}(R^{\text{out}}_{\text{test}}, X^{\text{out}}_{\text{test}}) \tag{5}$$

without direct access to $X^{\text{out}}_{\text{test}}$.

4 Good co-smoothing does not guarantee correct latents

It is reasonable to assume that being able to predict x on held-out data will guarantee that the inferred latent is the true one ([7, 9, 10, 13, 14, 16, 18, 21–23, 26, 34, 35, 37, 38]). We hypothesize, however, that good prediction guarantees that the true latents are contained within the inferred ones (assuming observability of the true latents), but not vice versa (Figure 1B).

To verify this hypothesis, we simulate a scenario where we know the ground truth. Specifically, we use a student-teacher setting, where both student and teacher are described by a discrete-space, discrete-time Hidden Markov Model (HMM). The HMM has a state space $z \in \{1, 2, \ldots M\}$, and produces observations (emissions in HMM notation) along neurons $\boldsymbol{x}$, with a state transition matrix A, emission model B and initial state distribution π. More explicitly:

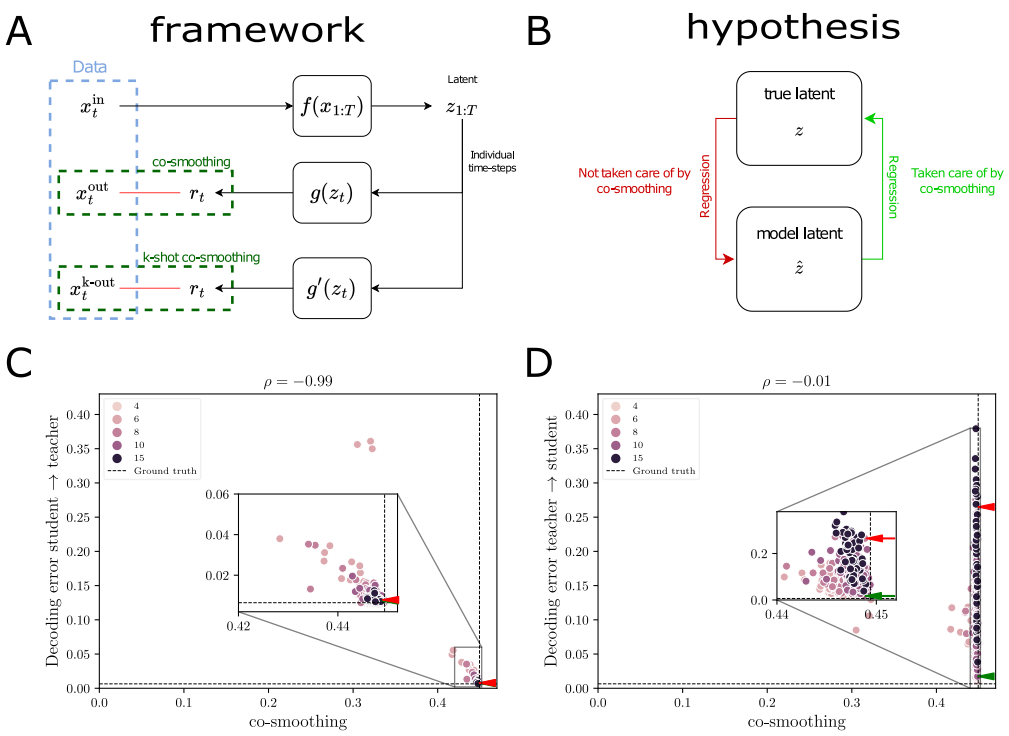

Figure 1: **A** Overview of the co-smoothing cross-validation framework for Neural LVMs. Part of the data is used to infer latents, while the rest serves as a target for evaluation. The functions f, g are optimized to improve co-smoothing. The lower arm is our few-shot proposal. where a function g' is optimized using the existing latents and a few observations of new neurons (see also Fig. 3).**B** Hypothesis: high co-smoothing ensures the model representation contains the ground truth, but not vice-versa. This may be revealed by the unequal performance of regression on the states in the two directions. **C** Several student HMMs are trained on a dataset generated by a single teacher HMM. The student→teacher decoding error is low and tightly related to the co-smoothing score. **D** The teacher→student decoding error is more varied and uncorrelated to co-smoothing. Dashed lines represent the ground truth, evaluating the teacher as a candidate model. Green and red arrows represent "Good" and "Bad" models respectively, presented in Fig. 2.

$$A_{m,l} = p(z_{t+1} = l | z_t = m) \qquad \forall\, m, l \tag{6}$$
$$B_{m,n} = p(x_{n,t} = 1 | z_t = m) \qquad \forall\, i, n \tag{7}$$
$$\pi_m = p(z_0 = m) \qquad \forall\, m \tag{8}$$

The same HMM can serve two roles: a) data-generation by sampling from (6), (7), (8) and b) inference of the latents from data on a trial-by-trial basis:

$$\xi_{t,m}^{(i)} = f_m((x_{1:T}^{\text{in}})^{(i)}) = p(z_t^{(i)} = m | (x_{1:T}^{\text{in}})^{(i)}), \tag{9}$$

i.e *smoothing*, computed exactly with the well known forward-backward algorithm [1]. Note that although z is the latent state of the HMM, we find it convenient to use its posterior probability mass function $\boldsymbol{\xi}_t$ as the relevant intermediate representation. To make predictions of the rates of held-out neurons for co-smoothing we compute:

$$r_{n,t}^{(i)} = g_n(\boldsymbol{\xi}_t^{(i)}) = \sum_m B_{m,n} \xi_{t,m}^{(i)} \qquad \forall\, n \in \text{out}, 1 \le t \le T, i \in \text{test} \tag{10}$$

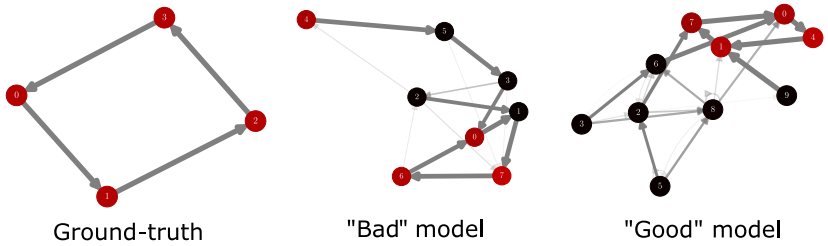

Figure 2: Graph visualisations of HMMs: the ground-truth or teacher model along with two representative extreme student models. All three models have high co-smoothing (low student→teacher decoding error). The students differ in their teacher→student decoding error (Fig. 1C,D). Node-colors represent initial state probabilities π_m (bright is high probability) and edge-width and opacity denote transition probabilities $A_{m,l}$. Edges with values below 0.02 are removed for visualisation. Note the $(1 \to 7 \to 0 \to 4)$ cycle of the good student, and the $(6 \to 0 \to 1 \to 7)$ cycle in the bad student. They differ in π, and the latter has an outgoing edge $(6 \to 2)$, with $A_{6,2} = 0.08$, $A_{6,0} = 0.89$.

As a teacher, we constructed a 4-state model of a noisy chain $A_{m,l} \propto \mathbb{I}[l = (m+1) \mod M] + \epsilon$, with $\epsilon = 1e-2$, $\pi = \frac{1}{M}$, and $B_{m,n} \sim \text{Unif}(0,1)$ sampled once and frozen (fig. 2). We trained 400 students with $6-15$ states on the same teacher using gradient-based methods (see supplement A). All students had high co-smoothing scores, with some variance. Consistent with our hypothesis, the ability to decode the teacher from the student varied little, and was highly correlated to the co-smoothing score (FIG 1C). In contrast, the ability to decode the student from the teacher displayed a large variability, and little correlation to the co-smoothing score (FIG 1D). See supplement B for details of the regression.

What is it about a student model, that produces good co-smoothing with the wrong latents? We consider the HMM transition matrix for the teacher and two exemplar students – named "Good" and "bad" – and visualise their states and transition probabilities using graphs in figure 2 (and marked in FIG 1CD). The teacher is a cycle of 4 steps. The good student has such a cycle $(1 \to 7 \to 0 \to 4)$, and the initial distribution π is only on that cycle, rendering the other states irrelevant. In contrast, the *bad* student also has this cycle $(6 \to 0 \to 1 \to 7)$, but the π distribution is not consistent with the cycle, and there is an outgoing edge from the cycle $(6 \to 2)$. Note that this does not interfere with co-smoothing, because the teacher itself is noisy. Thus, occasionally, there will be trials where the teacher will not have an exact period of 4 states. In such trials, the bad model will infer the irrelevant states instead of jumping to another relevant state, as in the teacher model.

5 Few-shot prediction selects better models

Because our objective is to obtain latent models that are close to the ground truth, the co-smoothing prediction scores described above are not satisfactory. Can we devise a new prediction score that will be correlated with ground truth similarity? The advantage of prediction benchmarks is that they can be optimized, and serve as a common language for the community as a whole to produce better algorithms [5].

We suggest **few-shot co-smoothing** as a prediction score. Few-shot co-smoothing is a modification of the standard co-smoothing procedure (Fig. 1A, bottom arm): the data for held-out channels is only made available for a limited number of k trials (see Fig. 3, right). Given these k trials, a function g' is estimated to map the latents to the held-out neurons. We highlight the following conceptual difference: standard co-smoothing does not require a defined encoding (f), latent (z), and decoding stage (g). It evaluates $g \circ f$ in an end-to-end manner. In contrast, our few-shot variation demands this distinction and explores the effects of estimating g' with lower statistical power than g, to uncover other properties of the latent z. After the estimation 'training' of g', we make rate predictions $R^{\text{out}}_{\text{test}}$ for held-out neurons on the test set inputs $X^{\text{in}}_{\text{test}}$ using $g' \circ f$ instead of $g \circ f$ and compute the few-shot co-smoothing score $\mathcal{Q}(R^{\text{out}}_{\text{test}}, X^{\text{out}}_{\text{test}})$ with (3).

In order to evaluate both co-smoothing and few-shot co-smoothing simultaneously for each model we partition the set of channels into three sets of size N^{in}, N^{out}, $N^{\text{really-out}}$ each. The really-held-out

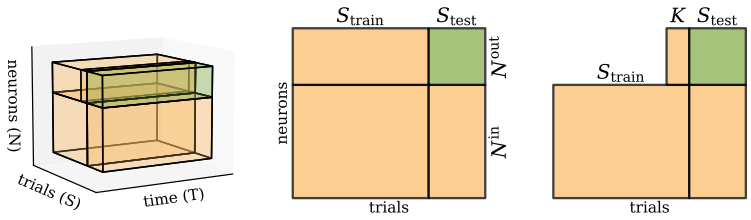

Figure 3: Neural data X is partitioned for cross-validating Latent Variable Models. Data accessible during training is in Orange, and data used for prediction is in green. Co-smoothing (center) and few-shot co-smoothing (right).

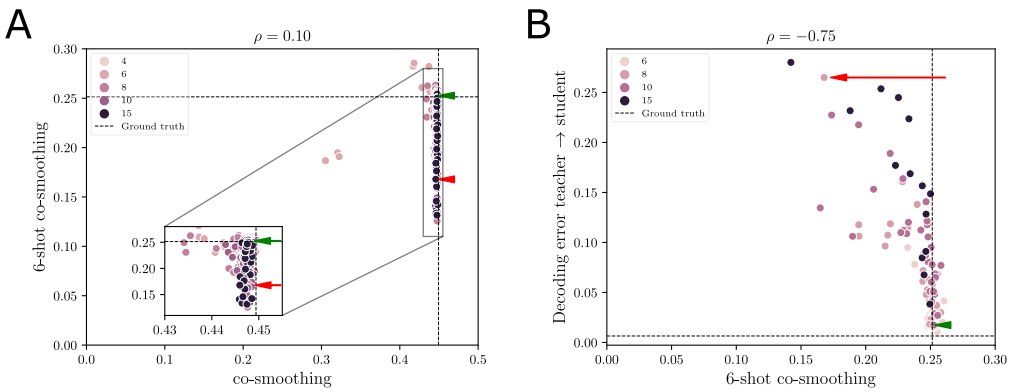

Figure 4: Few-shot prediction selects better models. **A.** Models with high co-smoothing have highly variable 6-shot co-smoothing and uncorrelated to co-smoothing. **B.** For the set of student with high co-smoothing $Q_{\text{student}} > Q_{\text{teacher}} - 10^{-3}$, 6-shot co-smoothing to held-out neurons is negatively correlated with decoding error from teacher-to-student. Following fig. 4C,D dashed lines represent the ground truth, green and red arrows represent "Good" and "Bad" models (Fig. 2).

neurons kept aside until after the training of f is complete, and used instead of the held-out neurons for training and testing g'.

This procedure may be repeated several times independently on i.i.d sets of k trials, giving several i.i.d estimates of g'. For small k, $\mathcal{Q}$ tends to be highly variable across instances of k-shot estimates. Thus we compute and report the average score across several instances.

In the case of bernoulli HMMs the maximum likelihood estimate of g' given a fixed f and k trials has a closed form:

$$\hat{B}_{m,n} = \frac{\sum_{i \in k\text{-shot}} \sum_{t=1}^{T} \mathbb{I}[x_{t,n}^{(i)} = 1] \xi_{t,m}^{(i)}}{\sum_{i' \in k\text{-shot}} \sum_{t'=1}^{T} \xi_{t',m}^{(i')}} \qquad \forall\, 1 \leq m \leq M \text{ and } n \in \text{really-out} \qquad (11)$$

To show the utility of the newly-proposed prediction score, we return to the same HMM students from figure 1. For each student, we evaluate the average k-shot smoothing. First, we see that it provides new information on the models, as it is not correlated with co-smoothing (Fig. 4A). We are only interested in models that have good co-smoothing, and thus select the students with the highest co-smoothing, those satisfying $Q_{\text{student}} > Q_{\text{teacher}} - \epsilon$, choosing $\epsilon = 10^{-3}$. For these students, we see that despite having very similar co-smoothing scores, their k-shot scores are highly correlated with their distance from the ground truth (as measured by decoding of student states from teacher) Fig. 4B.

6 Why does few-shot work?

The example HMM students of fig. 2 can help us understand why few-shot prediction identifies good models. The students differ in that the *bad* student has more than one state corresponding to the same teacher state. Because these states provide the same output, this feature does not hurt co-smoothing. In the few-shot setting, however, the output of all states needs to be estimated using a limited amount of data. Thus the information from the same amount of observations has to be distributed across more states. This data efficiency argument can be made more precise.

Consider a student-teacher scenario as in section 4. We let $T = 2$ and a stationary teacher$z_1^{(i)} = z_2^{(i)} = m$. Now consider two examples of inferred students. To ensure a fair comparison, both must have two latent states. In the *good* student, ξ, these two states statistically do not depend on time, and therefore it does not have extraneous dynamics. In contrast, the *bad* student, μ, uses one state for the first time step, and the other for the second time step. A particular example of such students is:

$$\xi_t = [0.5 \quad 0.5]^T \;\; t \in \{1, 2\} \tag{12}$$

$$\mu_{t=1} = [1 \quad 0]^T \qquad \mu_{t=2} = [0 \quad 1]^T \tag{13}$$

where each vector corresponds to the two states, and we only consider two time steps $t = 1, 2$.

We can now evaluate the maximum likelihood estimator of the emission matrix from K trials for both students, using equation (11). We consider a single neuron, and thus omit n. Because both states play the same role, we write the $m = 1$ case:

$$\hat{B}_1(\xi) = \tfrac{0.5(C_1+C_2)}{0.5KT} \qquad \hat{B}_1(\mu) = \tfrac{C_1}{KT} \tag{14}$$

where C_t is the number of times x occurs at time t in K trials.

The average of both quantities is the same, but the good student averages over more Bernoulli variables, and hence has a smaller variance. We show in the supplement that this larger variability translates to lower performance. Overall we that every time that a student has an extra state instead of reusing existing states, this costs the estimator more variance. In the appendix, we show how a more general setting can be approximated.

7 SOTA LVMs on neural data

In section 4 we showed that models with near perfect co-smoothing may possess latents with extraneous dynamics. We established this in a synthetic student-teacher setting with simple HMM models.

To show the applicability in more realistic scenarios, we trained several models from two SOTA architectures on `mc_maze_20` consisting of neural activity recorded from monkeys performing a maze solving task [4], curated by [21]. The `20` indicates that spikes were binned into $20ms$ time bins. We evaluate co-smoothing on a validation set of trials and define the set of models with the best co-smoothing (D).

An integral part of lfads-torch and STNDT frameworks is the random hyperparameter sweep which generates several candidate solutions to the optimization problem (5).

With each model f_u, we infer latents evaluated over a fixed set of test trials $X^{\text{test}} = \{\boldsymbol{x}^{(i)}\}_{i \in \text{test}}$ $Z_u := \{(z_t^{(i)})_u\}_{i \in \text{test}, 1 \leq t \leq T}$ by (1).

In the HMM case, we had ground truth that enabled us to directly compare the student latent to that of the teacher. With real neural data we do not have this privilege. To nevertheless reveal the presence or absence of extraneous dynamics, we instead compare the models to each other. The key idea is that all models contain the teacher latent, because they have good co-smoothing. One can then imagine that each student contains a selection of several extraneous features. The best student is the one containing the least such features, which would imply that all other students can decode its latents,

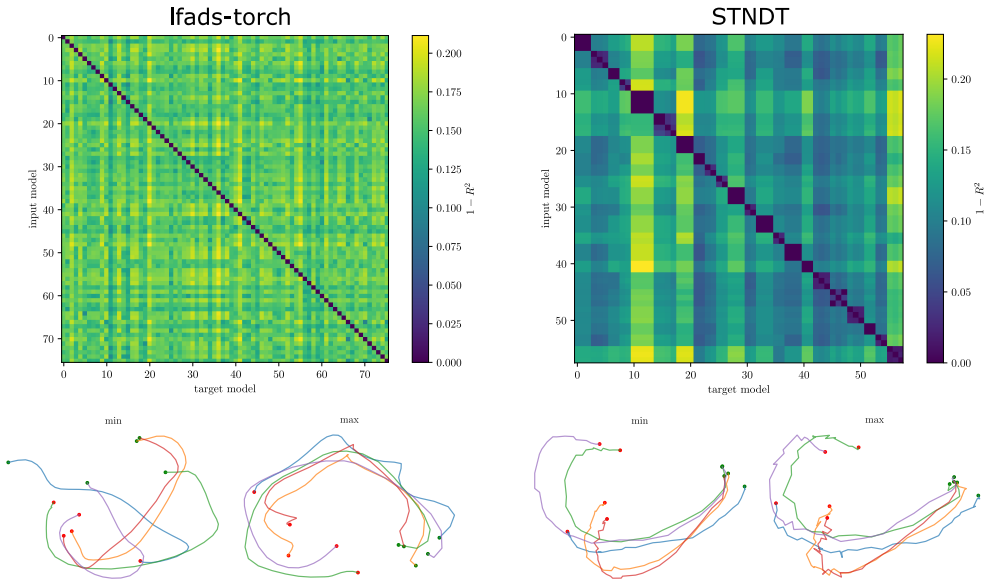

Figure 5: Cross-decoding as a proxy for distance to the ground truth in near-SOTA models. 200 LFADS models (left) and 120 STNDT models (right) were trained on the mc-maze data set, yielding The latents of each pair of models were decoded from one another, and the decoding error is shown in the matrices. Good models are expected to be decoded from all other models, and hence have low values in their corresponding columns. **Bottom left:** Trajectories of two LFADS models, with the lowest (left, best model) and highest (right) column averaged $1 - R^2$ projected onto their leading two principal components. Each trace is the trajectory for a single trial, starting at a green dot and ending at a red dot. **Bottom right** Same for STNDT

while it cannot decode theirs. We therefore use cross-decoding among student models as a proxy to the ground truth.

Instead of decoding from each student to the teacher as in section 4 we perform a cross-decoding from latents of model u to model v for every pair of models u and v using a linear regression (supp. D). We then evaluate a R^2 score for each mapping. In Fig 5 the results are visualised by a $U \times U$ matrix with entries $(R^2)_{u,v}$.

We hypothesize that the latents z_u contain the information necessary to output good rate predictions r that match the outputs plus the arbitrary extraneous dynamics. This former components must be shared across all models, whereas the latter could be unique in each model – or less likely to be consistent in the population. The ideal model v^* would have no extraneous dynamics therefore, all the other models should be able to decode to it, i.e $(R^2)_{u,v^*} = 1 \; \forall \; u$. Provided a large and diverse population $\mathcal{F}$ only the 'pure' ground truth would satisfy this condition. To evaluate how close is a model v to the ideal v^* we propose a simple metric: the column average $\mu_v = \frac{1}{U} \sum_{u=1}^{U} 1 - (R^2)_{u,v}$. μ_v will serve as proxy for the distance to ground-truth, analogous to the teacher→student decoding error in figure 4.

Having developed a proxy for the ground-truth we can now correlate it with the few-shot regression to held-out neurons. Fig 6 shows a negative correlation for both architectures, similar to the HMM examples above. As an illustration of the latents of different models, fig 6 shows the PCA projection of several trials from two different models. Both have high co-smoothing scores (SCORE1SCORE2), but differ in their cross-decoding column average μ. Note the somewhat smoother trajectories in the model with higher few-shot score.

8 Discussion

Latent variable models aim to infer the underlying latents using observations of a target system. We showed that co-smoothing, a common prediction measure of the goodness of such models cannot

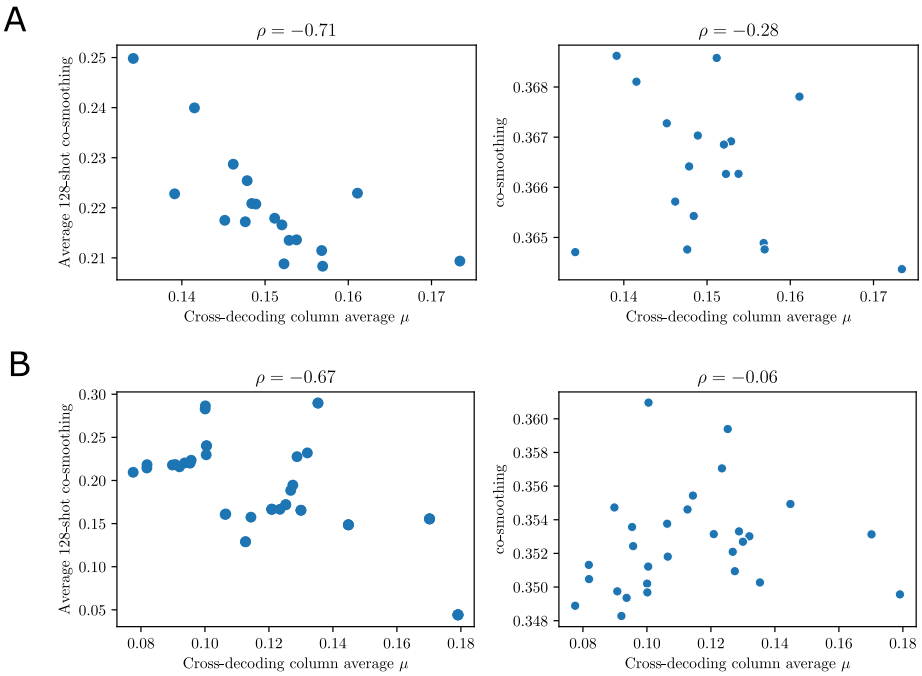

Figure 6: Few-shot scores correlate with the proxy of distance to the ground truth. Two models (LFADS top, STNDT bottom) were trained on neural recordings from monkeys performing a maze task, the `mc_maze_20` benchmark[4, 21]. Distance to ground truth was approximated by the cross-decoding column average μ (Fig. 5). Few-shot ($K = 128$) co-smoothing scores (left) negatively correlate with μ, while regular co-smoothing (right) does not.

discriminate between certain classes of latents. In particular, extraneous dynamics can be invisible to such a measure.

We suggest a new prediction measure: few-shot co-smoothing. Instead of directly regressing from held-in to held-out neurons as is done to evaluate co-smoothing, we distinguish the encoder and the decoder. To evaluate the trained model we substitute the decoder with a new decoder estimated using only a 'few' (k) number of trials. The rate predictions provided by the k-shot decoder may be evaluated the same as in standard co-smoothing. Using synthetic datasets and HMM models, we show numerically and analytically that this measure correlated with the distance of model latents to the ground truth.

We demonstrate the applicability of this measure to real world neural data sets, with SOTA architectures. This required developing a new proxy to ground truth – cross decoding. For each pair of SOTA models we obtained, we performed a linear regression across model latents, provided identical input data. Models with extraneous dynamics showed up as a bad target latent on average, and vice versa. Finally we show that these two characterisations of extraneous dynamics are correlated.

While we believe the combination of results in the toy setting and SOTA put forth a strong arguement, we address here a few limitations of our work. Firstly, our SOTA results use only one of the datasets in the benchmark suite [21]. With regard to the few-shot regression, while the bernoulli HMM scenario has a closed form solution: the maximum likelihood estimate, the poisson GLM regression is sensitive to the l2 hyperparameter `alpha`.

Overall, our work advances latent dynamics inference in general and prediction frameworks in particular. By exposing a failure mode of standard prediction metrics, we can guide the design of inference algorithms that take this into account. Furthermore, the few-shot prediction can be incorporated into existing benchmarks and help guide the community to build models that are closer to the desired goal of uncovering latent dynamics in the brain.

References

[1] David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[2] Luc Bauwens and David Veredas. The stochastic conditional duration model: a latent variable model for the analysis of financial durations. *Journal of econometrics*, 119(2):381–412, 2004.

[3] Manuel Brenner, Georgia Koppe, and Daniel Durstewitz. Multimodal teacher forcing for reconstructing nonlinear dynamical systems. *arXiv preprint arXiv:2212.07892*, 2022.

[4] Mark M Churchland, John P Cunningham, Matthew T Kaufman, Stephen I Ryu, and Krishna V Shenoy. Cortical preparatory activity: representation of movement or first cog in a dynamical machine? *Neuron*, 68(3):387–400, 2010.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[6] Daniel Durstewitz, Georgia Koppe, and Max Ingo Thurm. Reconstructing computational system dynamics from neural data with recurrent neural networks. *Nature Reviews Neuroscience*, 24 (11):693–710, 2023.

[7] Evren Gokcen, Anna I Jasper, João D Semedo, Amin Zandvakili, Adam Kohn, Christian K Machens, and Byron M Yu. Disentangling the flow of signals between populations of neurons. *Nature Computational Science*, 2(8):512–525, 2022.

[8] Xiao Shi Huang, Felipe Perez, Jimmy Ba, and Maksims Volkovs. Improving transformer optimization through better initialization. In *International Conference on Machine Learning*, pages 4475–4483. PMLR, 2020.

[9] Stephen Keeley, Mikio Aoi, Yiyi Yu, Spencer Smith, and Jonathan W Pillow. Identifying signal and noise structure in neural population activity with gaussian process factor models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13795–13805. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/9eed867b73ab1eab60583c9d4a789b1b-Paper.pdf`.

[10] Mohammad Reza Keshtkaran, Andrew R Sedler, Raeed H Chowdhury, Raghav Tandon, Diya Basrai, Sarah L Nguyen, Hansem Sohn, Mehrdad Jazayeri, Lee E Miller, and Chethan Pandarinath. A large-scale neural network training framework for generalized estimation of single-trial population dynamics. *Nature Methods*, 19(12):1572–1577, 2022.

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] Georgia Koppe, Hazem Toutounji, Peter Kirsch, Stefanie Lis, and Daniel Durstewitz. Identifying nonlinear dynamical systems via generative recurrent neural networks with applications to fmri. *PLoS computational biology*, 15(8):e1007263, 2019.

[13] Trung Le and Eli Shlizerman. Stndt: Modeling neural population activity with spatiotemporal transformers. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 17926–17939. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/72163d1c3c1726f1c29157d06e9e93c1-Paper-Conference.pdf`.

[14] Jakob H Macke, Lars Buesing, John P Cunningham, Byron M Yu, Krishna V Shenoy, and Maneesh Sahani. Empirical models of spiking in neural populations. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL `https://proceedings.neurips.cc/paper_files/paper/2011/file/7143d7fbadfa4693b9eec507d9d37443-Paper.pdf`.

[15] Niru Maheswaranathan, Alex Williams, Matthew Golub, Surya Ganguli, and David Sussillo. Universality and individuality in neural dynamics across large populations of recurrent networks. *Advances in neural information processing systems*, 32, 2019.

[16] Ganga Meghanath, Bryan Jimenez, and Joseph G Makin. Inferring population dynamics in macaque cortex. *Journal of Neural Engineering*, 20(5):056041, nov 2023. doi: 10.1088/1741-2552/ad0651. URL https://dx.doi.org/10.1088/1741-2552/ad0651.

[17] Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/a7a3d70c6d17a73140918996d03c014f-Paper.pdf.

[18] Thomas Soares Mullen, Marine Schimel, Guillaume Hennequin, Christian K. Machens, Michael Orger, and Adrien Jouary. Learning interpretable control inputs and dynamics underlying animal locomotion. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=MFCjgEOLJT.

[19] Toan Q Nguyen and Julian Salazar. Transformers without tears: Improving the normalization of self-attention. In *Proceedings of the 16th International Conference on Spoken Language Translation*, 2019.

[20] Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

[21] Felix Pei, Joel Ye, David Zoltowski, Anqi Wu, Raeed Chowdhury, Hansem Sohn, Joseph O' Doherty, Krishna V Shenoy, Matthew Kaufman, Mark Churchland, Mehrdad Jazayeri, Lee Miller, Jonathan Pillow, Il Memming Park, Eva Dyer, and Chethan Pandarinath. Neural latents benchmark '21: Evaluating latent variable models of neural population activity. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/979d472a84804b9f647bc185a877a8b5-Paper-round2.pdf.

[22] Sean M Perkins, John P Cunningham, Qi Wang, and Mark M Churchland. Simple decoding of behavior from a complicated neural manifold. *BioRxiv*, pages 2023–04, 2023.

[23] Marine Schimel, Ta-Chu Kao, Kristopher T Jensen, and Guillaume Hennequin. iLQR-VAE : control-based learning of input-driven dynamics with applications to neural data. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=wRODLDHaAiW.

[24] Andrew R Sedler and Chethan Pandarinath. lfads-torch: A modular and extensible implementation of latent factor analysis via dynamical systems. *arXiv preprint arXiv:2309.01230*, 2023.

[25] Andrew R Sedler, Christopher Versteeg, and Chethan Pandarinath. Expressive architectures enhance interpretability of dynamics-based neural population models. *arXiv preprint arXiv:2212.03771*, 2022.

[26] Qi She and Anqi Wu. Neural dynamics discovery via gaussian process recurrent neural networks. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 454–464. PMLR, 22–25 Jul 2020. URL https://proceedings.mlr.press/v115/she20a.html.

[27] Galit Shmueli. To explain or to predict? 2010.

[28] Ben Sorscher, Surya Ganguli, and Haim Sompolinsky. Neural representational geometry underlies few-shot concept learning. *Proceedings of the National Academy of Sciences*, 119 (43):e2200800119, 2022. doi: 10.1073/pnas.2200800119. URL https://www.pnas.org/doi/abs/10.1073/pnas.2200800119.

[29] Adrian Valente, Jonathan W. Pillow, and Srdjan Ostojic. Extracting computational mechanisms from neural data using low-rank RNNs. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=M12autRxeeS.

[30] Christopher Versteeg, Andrew R Sedler, Jonathan D McCart, and Chethan Pandarinath. Expressive dynamics models with nonlinear injective readouts enable reliable recovery of latent features from neural activity. *arXiv preprint arXiv:2309.06402*, 2023.

[31] Ricardo Vinuesa and Steven L Brunton. Enhancing computational fluid dynamics with machine learning. *Nature Computational Science*, 2(6):358–366, 2022.

[32] Saurabh Vyas, Matthew D Golub, David Sussillo, and Krishna V Shenoy. Computation through neural population dynamics. *Annual review of neuroscience*, 43:249–275, 2020.

[33] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[34] Anqi Wu, Nicholas A. Roy, Stephen Keeley, and Jonathan W Pillow. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/b3b4d2dbedc99fe843fd3dedb02f086f-Paper.pdf.

[35] Anqi Wu, Stan Pashkovski, Sandeep R Datta, and Jonathan W Pillow. Learning a latent manifold of odor representations from neural responses in piriform cortex. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/17b3c7061788dbe82de5abe9f6fe22b3-Paper.pdf.

[36] Joel Ye and Chethan Pandarinath. Representation learning for neural population activity with neural data transformers. *arXiv preprint arXiv:2108.01210*, 2021.

[37] Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper_files/paper/2008/file/ad972f10e0800b49d76fed33a21f6698-Paper.pdf.

[38] Yuan Zhao and Il Memming Park. Variational Latent Gaussian Process for Recovering Single-Trial Dynamics from Population Spike Trains. *Neural Computation*, 29(5):1293–1316, 05 2017. ISSN 0899-7667. doi: 10.1162/NECO_a_00953. URL https://doi.org/10.1162/NECO_a_00953.

A Hidden Markov Model training

HMMs are traditionally trained with expectation maximisation (EM), but they can also be trained using gradient-based methods. We focus here on the latter, using an implementation of HMMs in dynamax – a library of differentiable state-space models built with jax.

We seek HMM parameters $\theta := (A, B^{[\text{in,out}]}, \pi)$ that minimise the negative log-likelihood loss L of the held-in and held-out neurons in the train trials:

$$L(\theta; X_{\text{train}}^{[\text{in,out}]}) = -\log p(X_{\text{train}}^{[\text{in,out}]}; \theta) \tag{15}$$

$$= \sum_{i \in \text{train}} -\log p\left(\left(x_{1:T}^{[\text{in,out}]}\right)^{(i)}; \theta\right) \tag{16}$$

To find the minimum we do full-batch gradient descent on L, using dynamax together with the Adam optimiser.[11]

B Decoding across HMM latents: fitting and evaluation

Consider $\boldsymbol{\xi}^{(i)}$ and $\hat{\boldsymbol{\xi}}^{(i)}$ the latents inferred by the M-state teacher and $\hat{M}$-state student HMMs respectively from the data $(x^{\text{in}}_{1:T})^{(i)}$ using (9).

We now perform a multinomial regression from individual time-steps $\boldsymbol{\xi}_t^{(i)}$ to $\hat{\boldsymbol{\xi}}_t^{(i)}$ (and vice versa by reversing the hat notation) using `sklearn.linear_model.LogisticRegression`.

$$\hat{\boldsymbol{p}}_{\boldsymbol{t}}^{(\boldsymbol{i})} = \sigma\left(h\left(\boldsymbol{\xi}_t^{(i)}\right)W + \boldsymbol{b}\right) \tag{17}$$

where $W \in \mathbb{R}^{\hat{M}\times M}$, $\boldsymbol{b} \in \mathbb{R}^{\hat{M}}$ σ is the softmax. During training we sample states from the target PMFs $\hat{z}_t^{(i)} \sim \hat{\boldsymbol{\xi}}_t^{(i)}$ to a more well know problem scenario: classification. We optimize W and b to minimise a cross-entropy loss to the target $\hat{z}_t^{(i)}$.

To evaluate decoding error, we evaluate the average Kullback-Leibler divergence D_{KL} between target and predicted distributions:

$$\frac{1}{S^{\text{test}}T}\sum_{i\in\text{test}}\sum_{t=1}^{T} D_{KL}(\hat{\boldsymbol{p}}_t^{(i)}, \hat{\boldsymbol{\xi}}_t^{(i)}) \tag{18}$$

where D_{KL} computed with a `scipy.special.rel_entr`.

C Cramer Rao approximation for score

In the main text, we showed how a good student outperforms a bad one on few-shot learning. This was done in a specific case of two time points, a single teacher state and two student states. To link few-shot score and representations in more general settings, we look at the drop in average few-shot score for finite k: $\mathbb{E}\mathcal{L}(\hat{B}^k) - \mathcal{L}(\hat{B}^\infty)$. We use $\hat{B}^\infty$, the limit when$k \to \infty$, as a proxy for regular co-smoothing. $\hat{B}^k$ is the maximum likelihood estimator (MLE) of:

$$L_K(B) = \sum_{t=1}^{T}\sum_{i=1}^{k} \log f(\tilde{x}_t^{(i)}; r_t^{(i)}(B)) \tag{19}$$

$$= \sum_{t=1}^{T}\sum_{i=1}^{k} \log \begin{cases} r_t^{(i)}(B) & \text{if } \tilde{x}_t^{(i)} = 1 \\ 1 - r_t^{(i)}(B) & \text{if } \tilde{x}_t^{(i)} = 0 \end{cases} \tag{20}$$

$$= \sum_{t=1}^{T}\sum_{i=1}^{k} \log \begin{cases} \sum_m B_m \xi_{t,m}^{(i)} & \text{if } \tilde{x}_t^{(i)} = 1 \\ 1 - \sum_m B_m \xi_{t,m}^{(i)} & \text{if } \tilde{x}_t^{(i)} = 0 \end{cases} \qquad \text{using (10)} \tag{21}$$

In the main text, we showed that the good student has a less variable $\hat{B}$. To link this with few-shot performance, we compute the sensitivity of the log-likelihood score to variations in B^k.

$$\mathbb{E}_{\hat{B}_K} L(\hat{B}_K) = \mathbb{E}_{\hat{B}_K}\left[L(B_\infty) + \frac{1}{2}(\hat{B}_K - B_\infty)^T \frac{\partial^2 L}{\partial B^2}\bigg|_{B_\infty}(\hat{B}_K - B_\infty) + \dots\right] \tag{22}$$

$$= L(B_\infty) + \mathbb{E}_{\hat{B}_K}\frac{1}{2}(\hat{B}_K - B_\infty)^T H(B_\infty)(\hat{B}_K - B_\infty) + \dots \tag{23}$$

$$= L(B_\infty) + \frac{1}{2}\text{Tr}[\Sigma H] + \dots \tag{24}$$

where $\Sigma := \text{Cov}[\hat{B}_k]$. Using the Cramer-Rao bound, we can estimate this covariance:

$$\text{Cov}(\hat{B}^k) \geq I^{-1}(\hat{B}^k)|_{\hat{B}^k = B^*} \tag{25}$$

where I is the Fisher Information matrix:

$$I_{mm} := \mathbb{E}\left[\frac{\partial L_K}{\partial B_m}\frac{\partial L_K}{\partial B_m}\right] \tag{26}$$

$$= \mathbb{E}\sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{s=1}^{T}\sum_{t=1}^{T}\frac{(2\tilde{x}_s^{(i)}-1)(2\tilde{x}_t^{(j)}-1)}{f(\tilde{x}_s^{(i)};r_s^{(i)})f(\tilde{x}_t^{(j)};r_t^{(j)})}\xi_{s,m}^{(i)}\xi_{t,m}^{(j)} \tag{27}$$

using the tilde on $\tilde{x}$ to denote that these are the target of estimate.

From this, we obtain

$$\mathbb{E}_{\hat{B}_K} L(\hat{B}_K) \geq L(B_\infty) + \frac{1}{2}\text{Tr}[I(B_\infty)^{-1}H] + \dots \tag{28}$$

Finally we have

$$\mathbb{E}\mathcal{L}(\hat{B}^k) - \mathcal{L}(\hat{B}^\infty) \approx \frac{1}{2}\sum_m \frac{H_{mm}}{I_{mm}} \tag{29}$$

where

$$H_{mm} := \mathbb{E}\frac{\partial^2 L}{\partial B_m \partial B_m}\Big|_{B=\hat{B}^\infty} = -\sum_i\sum_{t=1}^{T}\frac{\left(\hat{\xi}_{t,m}^{(i)}\right)^2}{f(x_t^{(i)};r_t^{(i)})^2} \tag{30}$$

are the diagonal terms of the Hessian of the log-likelihood with respect to $\hat{B}$.

D Analysis of SOTA models

We denote the set of high co-smoothing models as those satisfying $Q_{\text{model}} > Q_{\text{best model}} - \epsilon$, choosing $\epsilon = 5 \times 10^{-3}$ for lfads-torch and $\epsilon = 1.3 \times 10^{-2}$ for STNDT. $\mathcal{F} := \{(f_u, g_u)\}_{u=1}^U$, the encoders and decoders respectively. Note that both architectures are deep neural networks given by the composition $g \circ f$, and the choice of intermediate layer whose activity is deemed the 'latent' z is arbitrary. Here we consider g the last 'read-out' layer and f to represent all the layers up-to g. g takees the form of Poisson Generalised Linear Model, a natural and simple choice for the few-shot version g'.

This choice of f, suggests a natural choice g' to be a Poisson Generalised Linear Model (GLM). We use `sklearn.linear_model.PoissonRegressor`. The poisson regressor has a hyperparameter `alpha`, the amount of l2 regularisation. For the results in the main text we select $\alpha = 10^{-3}$.

To perform few-shot co-smoothing, we partition the train data into several subsets of k trials. To implement this in a standarised way, we build upon the `nlb_tools` library (supp. E). This way we ensure that all models are trained and tested on identitical partitions.

We perform a cross-decoding from Z_u to Z_v for every pair of models u and v using a linear mapping $h(\boldsymbol{z}) := W\boldsymbol{z} + \boldsymbol{b}$ implemented with `sklearn.linear_model.LinearRegression`:

$$\left(\hat{z}_t^{(i)}\right)_v = h_{v\leftarrow u}\left(\left(z_t^{(i)}\right)_u\right) \tag{31}$$

minimising a mean squared error loss. We then evaluate a R^2 score (`sklearn.metrics.r2_score`) of z_v and $\hat{z}_v$ for each mapping. Results are accumulated into a matrix $(R^2)_{u,v}$.

E Code repositories

Table 1: Summary of key repositories used in this paper

Repository	Forked from	Citations
https://github.com/KabirDabholkar/nlb_tools_fewshot	https://github.com/neurallatents/nlb_tools	[21]
https://github.com/KabirDabholkar/STNDT_fewshot	https://github.com/trungle93/STNDT	[8, 13, 19, 21, 36]
https://github.com/KabirDabholkar/lfads-torch	https://github.com/arsedler9/lfads-torch	[10, 20, 24]
https://github.com/KabirDabholkar/hmm_analysis		
https://github.com/probml/dynamax		

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes] ,

   Justification: We provide examples in a range of settings, from simple HMMs to SOTA: RNN and transformer based architectures. We use the HMMs to provide analytical insight as why the result occurs.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We address limitations in the discussion. The main one being: only one SOTA benchmark.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.

- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   We prove increased variance in a simple case, and do not claim to rigorously prove the general case. Rather, we use math to provide intuition.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have no new training methods only analysis which is simpler. We provide the code for this.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case

of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Detailed in methods and in code

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Figure 6 has error bars (SEM) that are smaller than the symbols used.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper confirms with all items of the code

   Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Advances basic science, and does not have a direct societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: not relevant

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: existing assets (NLB, datasets mentioned there, and algorithms) are all credited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: few-shot code is provided

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: no human subjects

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: no human subjects

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.