A Generalist Intracortical Motor Decoder

Joel Ye[1*] **Fabio Rizzoglio**[3] **Adam Smoulder**[1] **Hongwei Mao**[2] **Xuan Ma**[3]
Patrick Marino[2] **Raeed Chowdhury**[2] **Dalton Moore**[5] **Gary Blumenthal**[2]
William Hockeimer[2] **Nicolas G. Kunigk**[2] **J. Patrick Mayo**[2] **Aaron P. Batista**[2]
Steven Chase[1] **Adam Rouse**[4] **Michael L. Boninger**[2] **Charles Greenspon**[5]
Andrew B. Schwartz[2] **Nicholas G. Hatsopoulos**[4] **Lee E. Miller**[3]
Kristofer E. Bouchard[6] **Jennifer L. Collinger**[2] **Leila Wehbe**[1] **Robert Gaunt**[2†]

[1]Carnegie Mellon University [2]University of Pittsburgh [3]Northwestern University
[4]University of Chicago [5]University of Kansas Medical Center
[6]Lawrence Berkeley National Laboratory

Abstract

Mapping the relationship between neural activity and motor behavior is a central aim of sensorimotor neuroscience and neurotechnology. While most progress to this end has relied on restricting complexity, the advent of foundation models instead proposes integrating a breadth of data as an alternate avenue for broadly advancing downstream modeling. We quantify this premise for motor decoding from intracortical microelectrode data, pretraining an autoregressive Transformer on 2000 hours of neural population spiking activity paired with diverse motor covariates from over 30 monkeys and humans. The resulting model is broadly useful, benefiting decoding on 8 downstream decoding tasks and generalizing to a variety of neural distribution shifts. However, we also highlight that scaling autoregressive Transformers seems unlikely to resolve limitations stemming from sensor variability and output stereotypy in neural datasets. Code: `https://github.com/joel99/ndt3`

1 Introduction

Intracortical neural data collection is growing rapidly. This growth comprises not only larger individual datasets with more neurons and higher behavioral complexity [1, 2], but also an increase in the collective number of datasets. This wealth of data presents an opportunity to develop insights and applications that span datasets, provided we can reconcile their inherent diversity. Large deep networks appear very suitable for this task, so much so that the generic creation of deep networks operating on broad domain data has been termed foundation modeling [3]. Efforts to create foundation models are now proliferating beyond their origins in natural language processing (NLP) and computer vision (CV) into many domains of engineering and science [4]. Here, we evaluate the efficacy of foundation modeling for motor decoding from intracortical spiking activity.

Motor decoding is a valuable domain for characterizing neural data foundation models. Academic and industrial efforts to create iBCIs for neuroprosthetics provide a path for scaling data collection from hundreds to thousands of subject-hours, and also fuel a need for pretrained models that generalize quickly and perform robustly for new users and settings. Behavior prediction metrics for BCI performance are also more intuitive for benchmarking progress than neural data prediction or the abstract goal of providing scientific insight (e.g. with latent variable models or in silico models) [5, 6]. Finally, recent work has shown that deep networks are able to transfer learn across motor cortical

*Correspondence: `joelye9@gmail.com`
†Correspondence: `rag53@pitt.edu`

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

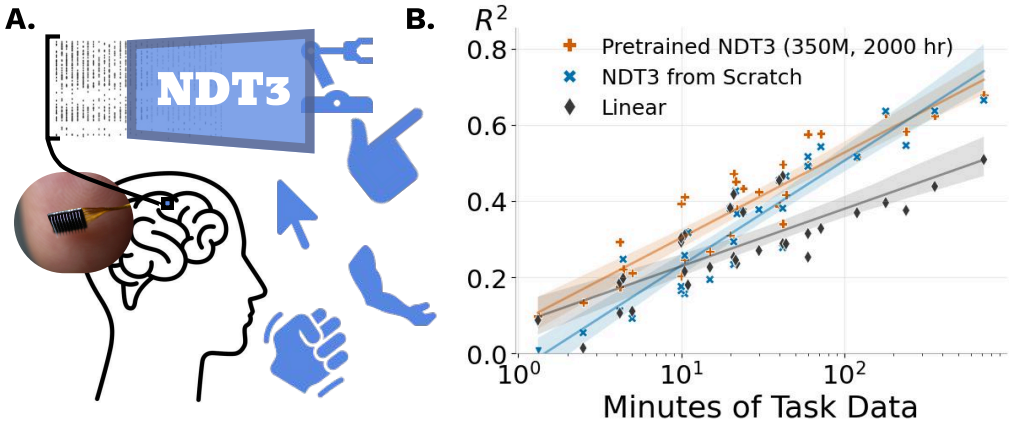

Figure 1. A. NDT3 decodes intracortical spiking activity into low-dimensional time series for various motor effectors[3]. **B.** We aggregate decoding performance on tasks with variable amounts of data (from Fig. 11). Pretrained NDT3 models reliably outperform from-scratch models and linear baselines, up to 1.5 hrs of data.

datasets collected at different timepoints, subjects, or tasks [7–9]. These ingredients provide the motivation and means for scaling neural data modeling.

However, scaling may be constrained by the unique variability present in each neural dataset. This unique variability induces irreducible error, not from the neural data signal to noise, but from what the same neural data means across contexts. To be precise, we distinguish two sources of unique variability introduced by comparing across subjects. The first difference stems from the physical difference in electrodes and recorded neurons, which we term sensor variability. For example, suppose we have toy 2-neuron subjects, where one neuron fires on leftward motion and the other fires on rightward motion. No amount of pretraining will reduce the data needed from *new* subjects to infer each neuron's preferred direction. The second difference is the residual variability in subject behavior and neural dynamics. This is also significant as shared structure between neural datasets, as probed by linear methods, fade quickly beyond the data's top principal components [10, 11]. Since pretraining cannot address the unique variability of each neural dataset, pretraining gains overall may be limited. Neural scaling efforts must therefore be alert to whether performance limits are due to fundamental data barriers or methodological shortcomings.

We combined simple tokenization and a standard autoregressive Transformer to enable pretraining over diverse data and fine-tuning to new tasks without any task-specific parameters (Fig. 1A). We pretrained this model, which we call Neural Data Transformer 3 (NDT3), using up to 2000 hours of neural and behavioral data from motor neuroscience experiments with monkeys and clinical intracortical BCI (iBCI) trials with humans. We then evaluated NDT3's decoding performance on eight motor datasets (Section 3.1), finding that tuning NDT3 yields models that either improve over task-specific models trained from scratch when task data is under 1.5 hours, and match them beyond this limit (Fig. 1B). Further, these gains persist under real-world distribution shifts (Section 3.3). These benefits may enable more complex experimental design and decrease the burden of decoder training for people using iBCIs. However, beyond the 1.5 hour limit, we observe that increasing data scale without increasing model capacity degrades performance, which provides two indicators that data heterogeneity will challenge the productivity of scaling on neural data. To guide future efforts to improve scaling, we analyze two measures of generalization where NDT3 currently fails, in robustness to input sensor variability and output stereotypy (Section 3.2).

2 Approach

2.1 Data

NDT3 models datasets of paired neural spiking activity and behavior (Fig. 2). Given our focus on motor decoding, most of the data comes from devices implanted in motor cortex of various monkeys

[3]Photo courtesy of Nicho Hatsopoulos and The Chicago Tribune.

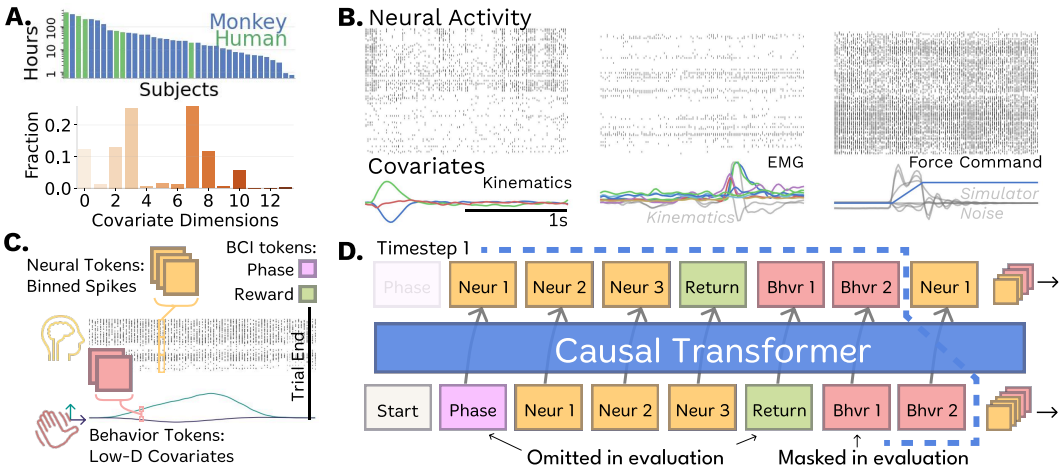

Figure 2. NDT3 Data and Model Design: A. NDT3 models paired neural spiking activity and behavioral covariate timeseries. We plot the distribution of 2000 hours of pretraining data by subjects (top) and covariate dimensionality (bottom). **B.** Examples of the neural and behavioral data for each of the three types of behavioral covariates in pretraining: kinematics, EMG (electromyography), or forces. Not all modeled dimensions in data are meaningfully task-related (right, grey behavior). **C.** Neural spiking activity is tokenized in time by binning the number of spikes every 20 ms, and in "space" using patches of channels (usually 32), as in NDT2 [8]. Behavior is low-dimensional in our data, so we use 1 token per behavior dimension, also per 20 ms timestep. NDT3 also pretrains on data from BCI control, which we annotate with two additional tokens. The phase token indicates whether the user is controlling or observing the behavior and the reward token indicates if the BCI task was completed. **D.** NDT3 models tokens in a single flat stream with linear readins and readouts. Every real-world timestep (shown by the blue cutout) yields several tokens, which we order to allow causal decoding at inference-time. At inference, we omit return and phase tokens and zero-mask behavior tokens.

and humans (Fig. 2A). These devices are intracortical multielectrode arrays or probes that record 30 kHz extracellular potentials. After bandpass-filtering, spikes are extracted from these potentials when they exceed preset thresholds determined by the researcher. The resulting neural data in our pretraining are diverse (Fig. 2B top). Data can have markedly different profiles across electrodes due to physical separation (being on different electrode arrays in the same subject (left)), and the interaction of recording noise and threshold preprocessing can yield either many inactive channels (middle) or high and noisy firing (right).

The typical behaviors in the pretraining data are reaching and grasping, nearly all from experimental paradigms that consist of short, repeated trials. While neural data were always recorded from microelectrodes, motor covariate signals came from various sensors. In monkey datasets, these sensors measure actual limb activity (e.g., Fig. 2B, left: limb kinematics from optical tracking, middle: electromyography (EMG)). In human datasets, physical movements are typically not possible, so the behavioral signals are programmatically generated. These signals are "paired" with the neural data in that they are cued or otherwise instructed to the person, who will attempt or imagine the corresponding behavior, such as grasping at a specified force level (Fig. 2B right). This force panel also shows that in pretraining, we cannot always automatically discern the primary task covariates (e.g., blue line, force, in the panel) from other recorded behavioral variables (grey). Thus, some behavior variables may be unpredictable. Finally, we include closed loop iBCI data, where some behavior is generated by an iBCI decoder (not NDT3, see modeling strategy in Section 2.2). The data used for pretraining NDT3 is detailed in Section A.3.4.

We minimize preprocessing of these data to maximize the applicability of our generalist model. Kinematic signals are typically converted to velocities, and all behavior (kinematics, EMG, force) is normalized per dataset such that the maximum absolute value of each variable is one. Data are cut or concatenated into fixed length sequences, without additional annotation of data discontinuity. This strategy, common in language modeling [12], homogenizes the data for improved GPU utilization while maintaining throughput of real data. We used a length of two seconds as it is roughly the timescale of the behavior in our data (Fig. 2A). Sequences with no spikes or covariate variability are

discarded. In total, this yielded about 3 million sequences, 1 billion neural tokens, or 1750 hours of recorded data. We round this to 2 khrs in text for simplicity.

2.2 Model

NDT3 is an autoregressive Transformer with linear readin and readout layers per modality, similar to GATO or TDMs [13–15]. This requires data tokenization (Fig. 2C). We tokenize neural data by patching spike counts [8]; each token is a vector of the binned spikes at a fixed temporal resolution (20 ms) and spatial dimension (32 channels). For example, spikes sampled from an electrode array with 100 channels would patch into $4 = \lceil 100/32 \rceil$ 32D neural tokens per 20 ms timestep. As the behavioral variables are already low-dimensional, we simply assign 1 token per dimension at the same temporal resolution as neural data. Finally, we add tokens marking whether the behavior are generated by a BCI system or by physical limb movement. While measured kinematics, EMG, or force will reflect a natural relationship with neural activity, behavioral data from BCI tasks are controlled by a program or learned decoder. We frame BCI-driven behavior as suboptimal control [16] and adopt a scheme from Decision Transformers [17, 18]. In this scheme, we use a Phase token to denote timesteps where behavior is driven by neural activity (i.e. native body movement or BCI control) versus programmatic, open loop control. We also use a Return token reflecting controller quality based on future task completion. Note that these signals are only considered for pretraining, and are ablated entirely from the model at evaluation. Similarly, input behavior tokens are masked out in inference, so that the model input only indicates how many behavioral dimensions must be predicted. NDT3 is trained with mean-squared error for prediction of behavioral variables, and categorical cross-entropy losses for prediction of neural spike count and return.

All modalities are flattened into a single token stream, with the multiple tokens in each real-world timestep sorted to respect a canonical, causal order required for control (Fig. 2D). We use rotary embeddings [19] to track real-world timesteps, and learned, modality-specific position embeddings to distinguish tokens within a timestep. Note that NDT3's autoregressive objective will condition some tokens on other tokens from the same real-world timestep.

Pretraining and Fine-Tuning We pretrain NDT3 models over many scales of pretraining data (1.5 hrs to 2 khrs) and in sizes of 45M and 350M parameters to assess the impact of data and model scaling. Pretraining is stopped early according to validation loss or terminated at a maximum of 400 epochs. The 200 hour, 45M model trains for 480 A100-hours while the 2000 hour (2 khr) 350M model takes 20K A100-hours. Fine-tuning maintains the pretraining objectives and updates all parameters.

2.3 Evaluation Strategy

Evaluation datasets and tuning Our main evaluation (Section 3.1) uses four human and four monkey datasets sampling varied upper limb movements, which we detail in Section A.3.4. All downstream monkeys are held-out of pretraining, and all humans are held-out of < 2 khr models. That is, we evaluate **scaling of cross-subject generalization**, as it is the most general use case for BCI-oriented foundation models. Note the human data leakage was allowed as significant data came from humans included in pre-determined evaluation datasets, and this leakage did not advantage the 2 khr model (Section A.2.5). Each dataset contains multiple sessions of data, typically from a single monkey or human. We will refer to each such setting as a "task," distinguished from the behavioral procedure performed in each dataset. Each session has unique variability, so one fine-tuning strategy, used in prior work [7, 8, 20], is to focus evaluations by tuning separate models for each evaluation session. We instead fine-tune NDT3 jointly over data combined from multiple evaluation sessions, which helps manage compute and storage demands, and better reflects that real-world deep network based BCIs typically jointly tune over multiple sessions. Fig. 12 confirms this joint tuning is comparable to per-session tuning for multi-session data.

Baselines We compare with Wiener filters (WF) and NDT2 [8]. WFs are a conventional linear method for motor iBCI [21]. We implement them as ridge regression with multi-timestep history. NDT2 is a Transformer that uses MAE self-supervision [22] to learn across neural datasets. NDT2 currently leads on FALCON [23], a BCI decoding benchmark that notably includes RNN baselines; we thus omit RNNs in our evaluation. We compare to NDT2 both prepared from scratch on each evaluation dataset (as in FALCON), or tuned from the public pretrained weights from pretraining on 100 hours of human data. We are also aware of two other proposals for scaled Transformers on

spiking activity, POYO [7] and NDT-MTM [20]. We omit full comparisons with these models, as our reproduction of POYO underperforms NDT3 on FALCON when both models are trained from scratch, and NDT-MTM is qualitatively similar to NDT (see Section A.3.3).

Downstream Hyperparameters We tune all deep networks (NDT2 and NDT3) over 3 learning rates. This hyperparameter sweep is limited for computational tractability. Importantly, the same sweep is used for all tasks; we list it and show its sufficiency relative to wider sweeps in Section A.3.2. The best learning rate is chosen based on the average validation score over three random seeds, and we report the average score on a separate test split.

3 Results

NDT3's pretraining advances intracortical models by an order of magnitude in data and model scale, from 200 to 2000 hours and 10M to 100M+ parameters. In Section 3.1, we show that this joint scaling is productive, but only when downstream datasets are < 1.5 hours. To analyze this limit, Section 3.2 shows that despite outperforming baselines, NDT3's transfer remains pathologically sensitive to shifts in data input dimensions. Section 3.2 also shows that scaling does not improve generalization to new behaviors. Section 3.3 concludes by showing that NDT3's pretraining gains, despite their limits, do generalize to novel settings to provide a practical foundation for motor decoding.

3.1 Multi-scale evaluation across motor decoding tasks

To set expectations for how data scale and model size will impact performance, we first measure behavior prediction R^2 computed during pretraining. We report on a test split comprising multiple sessions of 2D reaching mainly from one monkey. Models can learn this as 1.5 hours of data sampled from this task are included in all pretraining datasets. The first three bars in Fig. 3A compare a model pretrained on solely these 1.5 hours, or one of two 200 hour datasets: The "Subject" model uses data from 10 other monkeys, while the "Session" model uses 200 hours from the test monkey (from a separate set of experiments with similar behavior). Only the session model provides an improvement, suggesting cross-subject data is too distant to benefit a model that already uses 1.5 hours of task-specific data. The right three bars compare how further scaling to 2 khrs can actually degrade performance, suggesting dissimilar pretraining data can interfere with learning of the test task. This interference is mitigated by increasing model size to 350M parameters, consistent with recommendations to scale model size and dataset size in tandem [24–26]. However, rescued performance still remains near the 200 hour reference.

In light of previous work showing cross-subject transfer at low downstream data scales [7, 8], our upstream saturation suggests downstream data scale might influence pretraining gain. We thus evaluate downstream at multiple data scales, illustrated in Fig. 3B. These two datasets are from a human performing open loop iBCI calibration for bimanual cursor use [27], and a monkey performing self-paced reach to random targets [28]. Both individuals are held-out from pretraining entirely, so the task-specific data are only seen in tuning. In the bimanual task, NDT3 performance improves with increased pretraining data at all downstream data scales. The self-paced reach dataset has more ambiguous results. For example, from-scratch NDT3 is competitive at all but the smallest data scale. These examples raise the concern that pretraining efficacy will depend on not only data scale but also its quality, like the subject or behavior studied. However, such variable efficacy across downstream settings is not uncommon in pretraining [29, 30], and does not preclude an initial aggregate evaluation.

Tuning over 2000 models in 31 evaluation settings, we find that pretraining scale provides overall benefits (Fig. 3C). To begin, from-scratch NDT3 outperforms the WF and NDT2 (pretrained or not). We discuss NDT2's relatively poor performance in Section A.2.7. From-scratch NDT3 performance can be improved with minimal pretraining (1.5 hrs), consistent with findings in computer vision [31]. Performance continues improving up to 2000 hours of pretraining data, but only when paired with increased model size to 350M parameters, as in pretraining. The gain of the 350M 2 khr model over other models is significant under nonparametric Wilcoxon signed rank tests, for all comparisons except the 350M 200 hr model (Fig. 3D).

We now return to analyzing the impact of downstream dataset scale and quality on pretraining efficacy. Fig. 3E organizes downstream results by data scale, revealing that said scale is the primary factor in determining pretraining efficacy. Specifically, after normalizing by 200-hr model performance to

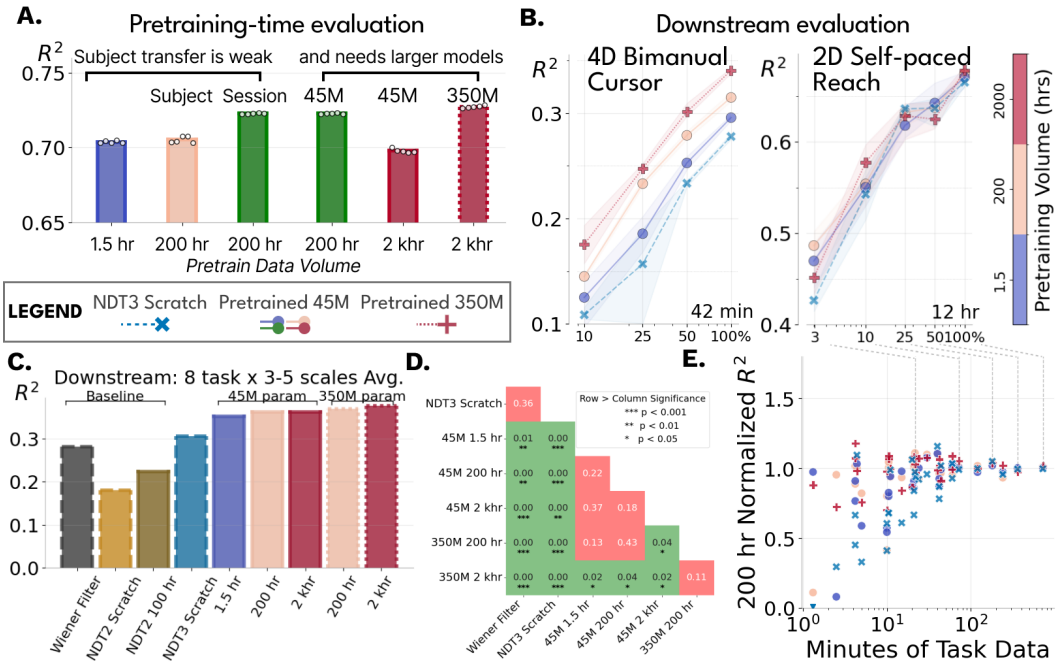

Figure 3. Evaluation on diverse motor tasks: A single legend is used throughout. **A.** Pretraining evaluation for 5 models, the center two bars are repeated for clarity. The top 5 evaluations across pretraining checkpoints are averaged and scattered individually. All models' pretraining data include 1.5 hours of calibration data for the test dataset. We compare a model with just this data (1.5 hr) to using 200 hours of additional data from over 10 other monkeys (Subject) vs 200 hours from the test monkey (Session). Only the additional test monkey data improves over the calibration model. The final two models pretrain on our full pretraining dataset (2 khr), which includes the 200 hours from the test monkey. Using 2 khrs underperforms the 200 hr model at 45M parameters, with performance restored by increasing capacity to 350M. **B.** Examples of good and bad data-scaling in downstream multiscale evaluation on two datasets. The bottom right text shows total time used in downstream fine-tuning. The x-axis scales this full dataset down by random subsampling. Shading shows standard deviation on 3 tuning seeds. Increasing pretraining data yields performance gains at all downstream scales in the 4D task, but effects are unclear in the self-paced reach task. **C.** Downstream performance averaged for 31 settings comprised of different downstream datasets and scales, for different NDT3s and baselines. 45M NDT3s improve with data from 1.5 hrs to 200 hrs but saturate at 2 khrs. Increasing model size to 350M parameters enables further gain. **D.** p-values computed from FDR-corrected pairwise t-tests for each pair of models. The 350M 2 khr NDT3 significantly outperforms other pretrained NDT3s, except the 350M 200 hr NDT3, and is the only model to do so. NDT2s omitted for brevity, see Fig. 13. **E.** Per-task performance, normalized by the 350M 200 hr NDT3 performance, is shown against task time for different NDT3 models. Each vertical band shows models trained on the same evaluation setting, e.g. dashed lines show the evaluations from the self-paced reaching dataset. Model variability vanishes by 1.5 hours, implying scaling will only help below this limit.

account for dataset variability in absolute performance, the distinction between pretrained models and from-scratch models vanishes by 1.5 hours. This convergence also explains the earlier observation of upstream saturation, as it also used 1.5 hours of calibration. The remaining variability across models at any fixed downstream dataset scale should be attributed to dataset quality. The effect is significant but secondary to data scale, so we detail individual dataset results in Section A.2.5. Moreover, a precise analysis on the impact of dataset quality is beyond the scope of this work, as our evaluation datasets have unique subjects and behaviors, while a controlled analysis would require multi-subject, multi-task datasets. Thus, we conclude provisionally that further scaling of NDT-style neural decoders will mainly be productive at downstream data scales <1.5 hours.

3.2 NDT3 does not generalize to novel input order and output conditions

Section 3.1 showed that pretraining gains saturate at data scales that, while larger than previously measured for neural data models, are practically low for BCI applications, as 1.5 hours can be reached after a few sessions of data collection. Unique variability across neural datasets does imply an

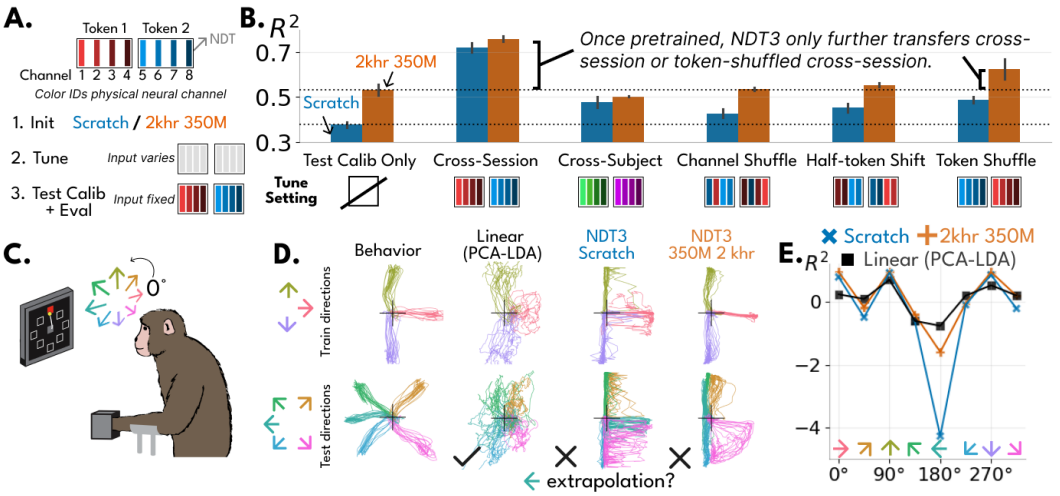

Figure 4. Pathological sensitivity to input order and stereotyped outputs. A. Setup for fine-tuning for input analysis. Colored bar marks a constant physical electrode channel of neural data, boxes show how they are tokenized for NDT. Schematic uses only 8 channels and 2 tokens, data has 96 channels and 3 tokens. Using models initialized from scratch or from pretraining (1), we tune and evaluate performance on a self-paced reaching dataset (3). We precede evaluation by training on an intermediate setting (2) with altered neural inputs relative to (3). (3) only uses a minute of data, while (2) uses several hours. **B.** The first three settings are no intermediate training (Test Calib Only), tuning with data Cross-Session data or Cross-Subject data with the same behavior. The latter settings alter cross-session data to emulate how cross-subject data might fail to transfer. Shuffle channel randomly permutes inputs, half-token shift rolls channels so that each channel i uses data from $i + 16$, and shuffle token permutes data tokenwise, while preserving the channels grouped in each token. Gains above the Test Calib Only baseline are only preserved under token-shuffling. **C.** We study angular extrapolation in an isometric monkey dataset where exerted forces are mapped to cursor positions in 8 different angles. We use three held-in and five held-out angles. **D.** Predictions are made either by fitting a Wiener Filter to neural data after dimensionality reduction (Linear), or from NDT3 (Scratch, 350M 2khr). While the linear model extrapolates to held-out angles, NDT3 predictions strictly follow held-in angles. **E.** Pretraining improves over from-scratch in held-in angles, but far underperforms in held-out angles.

eventual ceiling on pretraining gains, but this low threshold raises the concern that our modeling decisions, like tokenization, pretraining hyperparameters, or a lack of post-training, may be at fault. We next begin to dissociate these two possibilities, by highlighting NDT3's sensitivity to the specific neural inputs and behavioral outputs seen in tuning.

Input order sensitivity limits cross-subject transfer. Transfer learning is greatly reduced when using cross-subject neural data relative to cross-session neural data, even when the data are collected in identical experiments and thus controlled for other variables [8]. Limited cross-subject transfer may thus be a more precise cause of limited scaling in pretraining. This hypothesis predicts that pretraining may already account for all feasible cross-subject transfer. We show this by measuring transfer *after* large-scale pretraining, as schematized in Fig. 4A. Specifically, we tune NDT3 using calibration data from a single session of monkey self-paced reach [28], optionally preceded by intermediate tuning on additional cross-session or cross-subject data. Fig. 4B shows that cross-session intermediate tuning remains beneficial even after pretraining. In contrast, pretrained models do not benefit from cross-subject tuning. Further, cross-subject data does not benefit from-scratch models beyond the pretrained baseline (Test Calib Only). This shows **pretraining occludes further cross-subject transfer, but not cross-session transfer**, and supports the view that scaling is limited by cross-subject variability.

Recall that cross-subject differences can in turn be decomposed into sensor variability and changes in the neural dynamics underlying subject behavior. We can isolate how NDT3 resolves sensor variability while controlling for changes in aggregate neural statistics by performing intermediate tuning on cross-session data with altered input structure. We consider three alterations of decreasing strength: a complete channel shuffle, a half-token shift that preserves channel adjacencies but disrupts the channels in each (multi-channel) token, and a token shuffling (Fig. 4B Right). In all settings, intermediate tuning improves the from-scratch model, consistent with NDT's ability to transfer nontrivially across varied datasets. However, as in the cross-subject setting, no gains outperform

the pretrained baseline (Test Calib Only). Moreover, the pretrained model only achieves positive transfer from intermediate tuning on token shuffle cross-session data. These results show NDT3's cross-session transfer, and equivalently its downstream scaling, depends on the consistency of neural data dimensions, at least at the granularity of tokens. These conclusions are also supported in extended analysis (Section A.2.6), where we vary intermediate data scales and tuning strategy.

Pretraining does not enable angular extrapolation. Increased data efficiency after pretraining suggests that NDT3 could decode a new subject's behavior without sampling its full dynamic range. For example, center-out tasks require the subject to make stereotyped ballistic reaches and allow us to hold out reach angles to test decoding generalization. Center-out specifically provides a useful litmus test as the neural activity underlying center-out reach can be projected to a planar subspace [32], guaranteeing that some generalization is possible by construction. It is further known that non-pretrained deep networks fail at such held-out angle generalization [33]. Thus achieving this generalization would be a qualitative milestone for pretraining.

We analyze one session of an isometric monkey center-out dataset from [33], where the monkey exerts forces in one of eight angles and its force level is mapped to cursor position (Fig. 4C). We separate this data into 3 held-in and 5 held-out angles, and plot predictions from a Wiener Filter, a from-scratch NDT3, and a pretrained NDT3 (Fig. 4D). The WF, when fit to latents constructed through dimensionality reduction (PCA and LDA, see Section A.2.1 for methods), yields predictions that generalize to the held-out angle of leftward reach. Further as expected, from scratch predictions fail to generalize, and qualitatively are constrained to their nearest held-in angle. Pretrained NDT3, however, still fails to generalize, and appears to replicate from-scratch behavior of making predictions constrained to held-in angles. We quantify accuracy in Fig. 4E, showing that while pretrained NDT3 improves over from-scratch NDT3 overall, both models far underperform the PCA-LDA-fit WF on held-out generalization. This attractor strategy is undesirable for continuous BCI control [27], but is viable when the data are stereotyped, as is common in neural datasets. We replicate this result in two more settings in Section A.2.1. Importantly, this analysis does not confirm whether pretraining has failed to learn that simple dimensionality reduction can recover planar structure underlying ballistic reaches; plausibly, angular generalization could be achieved with a lightweight post-training protocol that discourages attractors in favor of continuous decoding.

Minimal benchmarks for neural data scaling Both sensor variability and output stereotypy pose fundamental challenges in neural pretraining, rather than trivial shortcomings specific to NDT3. For example, while our analysis raises concerns with NDT's patch-wise tokenization, the alternative of neuron-level tokenization [7] still requires input-channel identification and performs poorly in our hands (Fig. 19). Moreover, despite enthusiasm for neural foundation models, cross-subject transfer issues persist across neural modalities [34, 35]. Given NDT3's current state-of-the-art performance, these distillations of its current limits serve as concise benchmarks for generalization that future neural foundation models will likely need to overcome.

3.3 NDT3 does generalize to varied real world distribution shifts.

Having understood NDT's limitations under systematic synthetic shifts, we next provide examples where its pretraining usefully generalizes under real-world distribution shifts.

Neural distribution shifts Neural data are nonstationary, with shifts rising from a mix of controlled experimental variables to more speculative factors. For example, the firing rate of different channels will evolve over the course of an hour, implying a distribution shift associated with change in time (Fig. 5A top left). Shifts also occur between activity in two arm postures or whether finger motion occurs under spring load or not (Fig. 5A top middle and right). Since these shifts are common in neural data, pretraining gains should ideally be robust to their effect. We thus tune models on data from one setting (in-distribution, ID) and measure the performance of models in that same setting and the shifted setting (out-of-distribution, OOD). Positive correlation of performance in all cases implies the ID gains conferred by pretraining persist OOD. More practically, these examples suggest that pretraining benefits are not dependent on narrow features specific to the choice of tuning dataset.

Trial structure DNNs have been observed to overfit the temporal structure of experimental data, challenging their use in iBCI control [27, 36]. For example, a DNN might learn there is always no motion before the start of a behavioral trial, independent of the neural activity. To date, these claims have been studied exclusively on un-pretrained deep networks. We next assess how pretraining affects

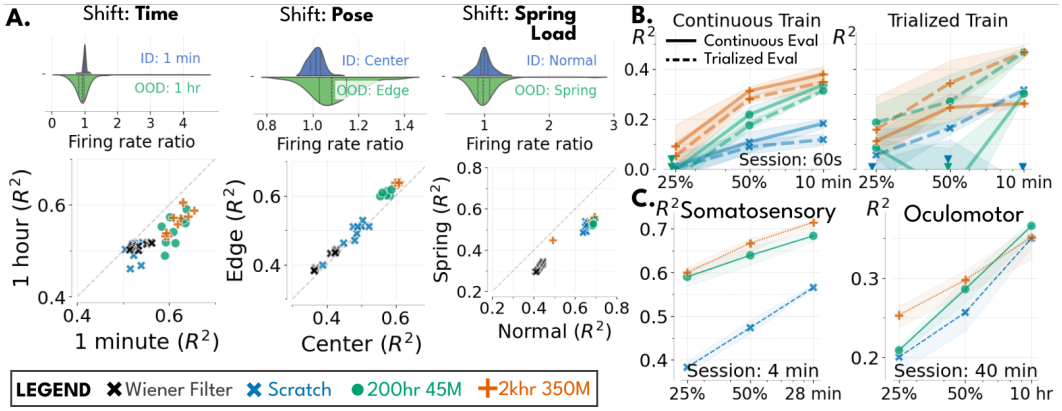

Figure 5. Generalization of pretraining gains. A. Models fine-tuned in one distribution of data are evaluated in-distribution (ID) and out-of-distribution (OOD). Top plots show the distribution across channels of neural firing rates from OOD and ID trials, normalized by average ID firing rates. Lower plots scatter OOD vs ID performance, with each point being a single model with different hyperparameters. The **time** shift uses two human cursor datasets collected one hour apart. Models were tuned in each block and were evaluated in the second block. **Pose** shift uses a monkey center-out reach task which was performed with the hand starting in different locations in the workspace. **Spring Load** uses a dataset of monkey 1D finger motion with or without spring force feedback. **B.** Models are evaluated on a human open-loop cursor dataset prepared in two ways. Trialized training receives inputs according to trial boundaries, varying from 2-4 seconds in length. Continuous training receives random 1 second snippets (that can cross trial boundaries). Trialized evaluation matches trialized training, and continuous evaluation is done by streaming up to 1 second of history. ▼ indicates points below 0.0. Continuously trained models perform well in both evaluation settings, while models trained on trialized data fail in continuous evaluation. **C.** Multiscale fine-tuning performance of NDT3 on datasets recorded outside motor cortex, namely S1 (Somatosensory) and FEF/MT (Oculomotor).

this overfitting in open loop human cursor control data by comparing a continuous and trialized setting. In the continuous setting, we cut random one second intervals of data in training and continuously stream up to one second of data in evaluation. The trialized setting formats data to respect trial boundaries, so the model always sees data aligned to the start of behavior.

In Fig. 5B, we show that models using both trialized training and evaluation outperform models with both continuous training and evaluation. This implies NDT3 will learn to exploit clear trial structure in data. However, while trialized from-scratch models become subtrivial under continuous evaluation (solid blue line is off-panel), pretrained models degrade more gracefully. For example, the 350M 2 khr model evaluated continuously only performs slightly worse with trialized tuning than with continuous tuning. Pretraining NDT3 thus reduces its dependence on trial structure, which should benefit both data analysis and iBCI control. Note, however, the contrast in these results with Fig. 4D, which show that DNNs clearly do overfit to tuning data in some cases. These nuances underscore the importance of rigorously evaluating model generalization in future work.

New brain areas In Fig. 5C, we return to multiscale fine-tuning to test how NDT3, pretrained on motor cortex, performs in somatosensory cortex (S1) and oculomotor areas (FEF and MT). The S1 data shares the same self-paced reaching behavior of Fig. 3B, while the Oculomotor data is an analogous visual pursuit task. The gain from pretraining over from-scratch models is high in S1, but also nontrivial in the Oculomotor dataset. While the former can be attributed to the close interaction of sensorimotor areas, the latter implies NDT3 has learned a broader prior. Although these results are encouraging, our previous results suggest these gains could reflect common experimental artifacts like trial structure, rather than neurophysiological priors (e.g. declining subject focus over time [37]). For example, this Oculomotor dataset contains 4 behavioral conditions, which may benefit from the tendency to learn classifiers shown in Fig. 4D rather than a prior on neural dynamics.

4 Discussion

Broader Impacts NDT3 is built as a generic pretrained model to accelerate BCI motor decoding workflows. To the extent it can, and with the cost and complexity of pretrained deep networks

over simpler decoders considered, such BCI devices are intended to restore function to individuals with disabilities or impairments resulting from brain injury or disease. Human BCIs do however pose risks for data privacy and personal agency [38], which are accentuated by the adoption of pretraining paradigms. NDT3 may also be used in non-human primate BCI research. We believe that the increased data efficiency of pretrained models in those settings can help reduce the need for redundant data collection.

Limitations This work provides an initial assessment of scaling up neural data decoding with a Transformer. Beyond the explicitly analyzed limitations in Section 3.2, this work can be expanded on both architectural and data design axes. Architectural choices include the use of different neural features (spike times, bandpower) or metadata features (per-day embeddings). Data design axes are harder to explore, as it is constrained by the expense of creating intracortical datasets (public or not). They are nonetheless vital to understand, as NDT3's results may poorly predict scaling trends on datasets with more diverse neural activity or behavior, with unconstrained (naturalistic) experimental paradigms, or with many more (1000s) of subjects.

Many fields are now pursuing large-scale deep learning as "a tide that lifts all boats" [39], hoping that improvements in effective pretraining will yield field-wide, downstream improvements. Such a unifying abstraction would be timely for neuroscience, given the increasing volume, diversity, and complexity of modern neural data. Joining other pretraining efforts on varied modalities of neural data (Section A.1), we trained NDT3 on 2000 hours of paired neural population activity from motor cortex and behavior, and then conducted a broad downstream decoding evaluation. Consistent with other foundation models, we found the best aggregate performance comes from increasing data scale and model size jointly (Section 3.1). We release the 350M 2 khr NDT3 as a strong baseline foundation model for intracortical decoding from spiking activity, with the disclaimer that benefits will likely be minimal by 1.5 hours of downstream data. To move forward, we highlight input order sensitivity and output attractors as issues future foundation models should address.

Acknowledgments

JY is supported by the DOE CSGF (i.e. U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Department of Energy Computational Science Graduate Fellowship under Award Number DE-SC0023112). This research was supported in part by the University of Pittsburgh Center for Research Computing, RRID:SCR_022735, through the resources provided. Specifically, this work used the H2P cluster, which is supported by NSF award number OAC-2117681. This research also used resources of the NERSC, a Department of Energy Office of Science User Facility, granted as part of JY's fellowship. Research reported in this publication was supported by the National Institute of Neurological Disorders and Stroke of the National Institutes of Health under Award Numbers R01NS121079 and UH3NS107714. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

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: Our abstract and introduction describe the two core results of the paper, that scaling pretraining on BCI data is productive but mainly in limited regimes.

   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: Main results in Section 3.1 emphasize qualifications around performance gains, and Section 3.2 analyze model shortcomings.

   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: [NA]

Justification:

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: Our codebase for pretraining and evaluation is released, and an order of 100 hours of public data can be used to pretrain smaller models. Our largest scale model depends on private pretraining data, but the model weights are released. We provide a fine-tuning notebook to reproduce our evaluations of these models on public datasets. Code is also available to reproduce limitation analyses in Section 3.2.

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: [No]

   Justification: Code is available and supports the scraping and preprocessing of public data used in this work. The majority of pretraining data is private due to several factors, including the overall sensitivity of human neural data, the necessity of not releasing private collaborator data being used in ongoing research, and the fact that the majority of data is unlabeled and not suitable for public release. Configurations for each experiment in the main result are documented in the codebase.

   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: Main details are described in text, and code for all configurations are released.

   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: Our main scaling performance is tested with paired statistical comparisons. Variability across random seeds in subsequent analysis is shown either as raw data or as standard error.

   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: [Yes]

   Justification: We provide reference compute costs in GPU-hours for the 45M 200 hour and 350M 2000 hour runs, which dominated compute costs for the project.

   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: I have reviewed the Code of Ethics. Human and animal data in this study were collected for previous studies approved by respective home institutions and federal regulatory bodies.

   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: [Yes]

Justification: A broader impact statement is available. It is tentatively in the supplement as the negative impact is fairly distant from the main focus of this work, but it can be moved to main text with extra page.

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: We do not believe the released pretrained models have high risk for misuse as their utility is limited to data from restricted clinical devices.

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: Data used from public and private sources are appropriately cited. Scraping scripts with URLs are available in the codebase. All other assets are original.

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: Pretrained models are made available with model cards located in the codebase. (Redacted in publication for anonymity).

    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 asseot 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 new instructions were provided to human participants for the purpose of this study.

    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: [Yes]

Justification: No new potential risks were incurred by this study as it was conducted with data collected for other studies. These other studies were performed with the appropriate regulatory approvals. Animal datasets were collected with approval by Institutional Animal Care and Use Committees. Human datasets were collected with Institutional Review Board approval, as part of clinical trials conducted under FDA Investigational Device Exemptions.

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.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs were not used in original components of this research.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

For data collection, the animal datasets used in this work were collected for other studies that were approved by Institutional Animal Care and Use Committees. Human datasets were also collected for other studies. These studies were performed under an approved Investigational Device Exemption from the FDA and were approved by the Institutional Review Board at the University of Pittsburgh. The clinical trial is registered at clinicaltrials.gov (ID: NCT01894802) and informed consent was obtained before any experimental procedures were conducted. Details on the implants and clinical trial are described in [40, 41]. We discuss the potential benefit of NDT3 to reduce user burden for iBCI-based neuroprosthetics, though the dissemination of pretrained models on these data raise the privacy concern that the original human data may be recoverable from model weights. Since this seems technically challenging at this point, and since the source data are restricted to binned spiking activity to begin with, we deem the risk low enough to justify the potential scientific benefit of sharing our pretrained models.

References

[1] Anne E. Urai, Brent Doiron, Andrew M. Leifer, and Anne K. Churchland. Large-scale neural recordings call for new insights to link brain and behavior. *Nature Neuroscience*, 25:11–19, 2022. doi: 10.1038/s41593-021-00980-9.

[2] Ian H. Stevenson. Tracking advances in neural recording. Statistical Neuroscience Lab, University of Connecticut, 2023. URL https://stevenson.lab.uconn.edu/scaling/. Accessed September 6, 2024.

[3] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022. URL https://arxiv.org/abs/2108.07258.

[4] Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

[5] Felix C Pei, Joel Ye, David M. Zoltowski, Anqi Wu, Raeed Hasan Chowdhury, Hansem Sohn, Joseph E O'Doherty, Krishna V. Shenoy, Matthew Kaufman, Mark M Churchland, Mehrdad Jazayeri, Lee E. Miller, Jonathan W. Pillow, Il Memming Park, Eva L Dyer, and Chethan Pandarinath. Neural latents benchmark '21: Evaluating latent variable models of neural population activity. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=KVMS3fl4Rsv.

[6] Eric Y Wang, Paul G Fahey, Kayla Ponder, Zhuokun Ding, Andersen Chang, Taliah Muhammad, Saumil Patel, Zhiwei Ding, Dat Tran, Jiakun Fu, et al. Towards a foundation model of the mouse visual cortex. *bioRxiv*, 2023.

[7] Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael Mendelson, Blake Richards, Matthew Perich, Guillaume Lajoie, and Eva Dyer. A unified, scalable framework for neural population decoding. *Advances in Neural Information Processing Systems*, 36, 2024.

[8] Joel Ye, Jennifer L Collinger, Leila Wehbe, and Robert Gaunt. Neural data transformer 2: Multi-context pretraining for neural spiking activity. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=CBBtMnlTGq.

[9] Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, May 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06031-6. URL https://doi.org/10.1038/s41586-023-06031-6.

[10] Juan A. Gallego, Matthew G. Perich, Stephanie N. Naufel, Christian Ethier, Sara A. Solla, and Lee E. Miller. Cortical population activity within a preserved neural manifold underlies multiple motor behaviors. *Nature Communications*, 9(1):4233, Oct 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-06560-z. URL https://www.nature.com/articles/s41467-018-06560-z.

[11] Mostafa Safaie, Joanna C Chang, Junchol Park, Lee E Miller, Joshua T Dudman, Matthew G Perich, and Juan A Gallego. Preserved neural dynamics across animals performing similar behaviour. *Nature*, 623 (7988):765–771, 2023.

[12] Jonas Geiping and Tom Goldstein. Cramming: Training a language model on a single gpu in one day, 2022. URL https://arxiv.org/abs/2212.14034.

[13] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent, 2022. URL https://arxiv.org/abs/2205.06175.

[14] Ingmar Schubert, Jingwei Zhang, Jake Bruce, Sarah Bechtle, Emilio Parisotto, Martin Riedmiller, Jost Tobias Springenberg, Arunkumar Byravan, Leonard Hasenclever, and Nicolas Heess. A generalist dynamics model for control, 2023. URL https://arxiv.org/abs/2305.10912.

[15] Team Chameleon. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

[16] Josh Merel, David Carlson, Liam Paninski, and John P Cunningham. Neuroprosthetic decoder training as imitation learning. *PLoS computational biology*, 12(5):e1004948, 2016.

[17] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling, 2021. URL https://arxiv.org/abs/2106.01345.

[18] Kuang-Huei Lee, Ofir Nachum, Mengjiao Yang, Lisa Lee, Daniel Freeman, Winnie Xu, Sergio Guadarrama, Ian Fischer, Eric Jang, Henryk Michalewski, and Igor Mordatch. Multi-game decision transformers, 2022. URL https://arxiv.org/abs/2205.15241.

[19] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

[20] Yizi Zhang, Yanchen Wang, Donato Jimenez-Beneto, Zixuan Wang, Mehdi Azabou, Blake Richards, Olivier Winter, The International Brain Laboratory, Eva Dyer, Liam Paninski, et al. Towards a" universal translator" for neural dynamics at single-cell, single-spike resolution. *arXiv preprint arXiv:2407.14668*, 2024.

[21] Chethan Pandarinath and Sliman J Bensmaia. The science and engineering behind sensitized brain-controlled bionic hands. *Physiological Reviews*, 102(2):551–604, 2022.

[22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. URL https://arxiv.org/abs/2111.06377.

[23] Brianna M Karpowicz, Joel Ye, Chaofei Fan, Pablo Tostado-Marcos, Fabio Rizzoglio, Clay Washington, Thiago Scodeler, Diogo de Lucena, Samuel R Nason-Tomaszewski, Matthew J Mender, Xuan Ma, Ezequiel Matias Arneodo, Leigh R Hochberg, Cynthia A Chestek, Jaimie M Henderson, Timothy Q Gentner, Vikash Gilja, Lee E Miller, Adam G Rouse, Robert A Gaunt, Jennifer L Collinger, and Chethan Pandarinath. Few-shot algorithms for consistent neural decoding (falcon) benchmark. *bioRxiv*, 2024. doi: 10.1101/2024.09.15.613126. URL https://www.biorxiv.org/content/early/2024/09/16/2024.09.15.613126.

[24] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[25] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning, 2020. URL https://arxiv.org/abs/1912.11370.

[26] Armen Aghajanyan, Lili Yu, Alexis Conneau, Wei-Ning Hsu, Karen Hambardzumyan, Susan Zhang, Stephen Roller, Naman Goyal, Omer Levy, and Luke Zettlemoyer. Scaling laws for generative mixed-modal language models. In *International Conference on Machine Learning*, pages 265–279. PMLR, 2023.

[27] Darrel R Deo, Francis R Willett, Donald T Avansino, Leigh R Hochberg, Jaimie M Henderson, and Krishna V Shenoy. Brain control of bimanual movement enabled by recurrent neural networks. *Scientific Reports*, 14(1):1598, 2024.

[28] Joseph E. O'Doherty, Mariana M. B. Cardoso, Joseph G. Makin, and Philip N. Sabes. Nonhuman primate reaching with multichannel sensorimotor cortex electrophysiology, May 2017. URL https://doi.org/10.5281/zenodo.788569.

[29] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024. URL https://arxiv.org/abs/2407.07726.

[30] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

[31] Rahim Entezari, Mitchell Wortsman, Olga Saukh, M Moein Shariatnia, Hanie Sedghi, and Ludwig Schmidt. The role of pre-training data in transfer learning. *arXiv preprint arXiv:2302.13602*, 2023.

[32] Mark M Churchland, John P Cunningham, Matthew T Kaufman, Justin D Foster, Paul Nuyujukian, Stephen I Ryu, and Krishna V Shenoy. Neural population dynamics during reaching. *Nature*, 487(7405): 51–56, 2012.

[33] Fabio Rizzoglio, Ege Altan, Xuan Ma, Kevin L. Bodkin, Brian M. Dekleva, Sara A. Solla, Ann Kennedy, and Lee E. Miller. Monkey-to-human transfer of brain-computer interface decoders. *bioRxiv*, 2022. doi: 10.1101/2022.11.12.515040. URL https://www.biorxiv.org/content/early/2022/11/13/2022.11.12.515040.

[34] Hubert Banville, Yohann Benchetrit, Stéphane d'Ascoli, Jérémy Rapin, and Jean-Rémi King. Scaling laws for decoding images from brain activity. *arXiv preprint arXiv:2501.15322*, 2025.

[35] Ann-Kathrin Kiessner, Robin T Schirrmeister, Joschka Boedecker, and Tonio Ball. Reaching the ceiling? empirical scaling behaviour for deep eeg pathology classification. *Computers in Biology and Medicine*, 178:108681, 2024.

[36] Joseph T Costello, Hisham Temmar, Luis H Cubillos, Matthew J Mender, Dylan M Wallace, Matthew S Willsey, Parag G Patil, and Cynthia A Chestek. Balancing memorization and generalization in rnns for high performance brain-machine interfaces. *bioRxiv*, pages 2023–05, 2023.

[37] Nicholas A Steinmetz, Peter Zatka-Haas, Matteo Carandini, and Kenneth D Harris. Distributed coding of choice, action and engagement across the mouse brain. *Nature*, 576(7786):266–273, 2019.

[38] Eran Klein, Tim Brown, Matthew Sample, Anjali R Truitt, and Sara Goering. Engineering the brain: ethical issues and the introduction of neural devices. *Hastings Center Report*, 45(6):26–35, 2015.

[39] Samira Abnar, Mostafa Dehghani, Behnam Neyshabur, and Hanie Sedghi. Exploring the limits of large scale pre-training, 2021. URL https://arxiv.org/abs/2110.02095.

[40] Jennifer L Collinger, Brian Wodlinger, John E Downey, Wei Wang, Elizabeth C Tyler-Kabara, Douglas J Weber, Angus JC McMorland, Meel Velliste, Michael L Boninger, and Andrew B Schwartz. High-performance neuroprosthetic control by an individual with tetraplegia. *The Lancet*, 381(9866):557–564, 2013.

[41] B Wodlinger, J E Downey, E C Tyler-Kabara, A B Schwartz, M L Boninger, and J L Collinger. Ten-dimensional anthropomorphic arm control in a human brain-machine interface: difficulties, solutions, and limitations. *Journal of Neural Engineering*, 12(1):016011, Feb 2015. ISSN 1741-2560, 1741-2552. doi: 10.1088/1741-2560/12/1/016011. URL https://iopscience.iop.org/article/10.1088/1741-2560/12/1/016011.

[42] Weibang Jiang, Liming Zhao, and Bao liang Lu. Large brain model for learning generic representations with tremendous EEG data in BCI. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=QzTpTRVtrP.

[43] Zhizhang Yuan, Fanqi Shen, Meng Li, Yuguo Yu, Chenhao Tan, and Yang Yang. Brainwave: A brain signal foundation model for clinical applications, 2024. URL https://arxiv.org/abs/2402.10251.

[44] Chaoqi Yang, M. Brandon Westover, and Jimeng Sun. Biot: Cross-data biosignal learning in the wild, 2023. URL https://arxiv.org/abs/2305.10351.

[45] Rahul Thapa, Bryan He, Magnus Ruud Kjaer, Hyatt Moore, Gauri Ganjoo, Emmanuel Mignot, and James Zou. Sleepfm: Multi-modal representation learning for sleep across brain activity, ecg and respiratory signals, 2024. URL https://arxiv.org/abs/2405.17766.

[46] Armin W. Thomas, Christopher Ré, and Russell A. Poldrack. Self-supervised learning of brain dynamics from broad neuroimaging data, 2023.

[47] Josue Ortega Caro, Antonio Henrique de Oliveira Fonseca, Syed A Rizvi, Matteo Rosati, Christopher Averill, James L Cross, Prateek Mittal, Emanuele Zappala, Rahul Madhav Dhodapkar, Chadi Abdallah, and David van Dijk. BrainLM: A foundation model for brain activity recordings. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=RwI7ZEfR27.

[48] Christopher Wang, Vighnesh Subramaniam, Adam Uri Yaari, Gabriel Kreiman, Boris Katz, Ignacio Cases, and Andrei Barbu. BrainBERT: Self-supervised representation learning for intracranial recordings. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=xmcYx_reUn6.

[49] Geeling Chau, Christopher Wang, Sabera Talukder, Vighnesh Subramaniam, Saraswati Soedarmadji, Yisong Yue, Boris Katz, and Andrei Barbu. Population transformer: Learning population-level representations of intracranial activity, 2024. URL https://arxiv.org/abs/2406.03044.

[50] Sabera J Talukder, Jennifer J. Sun, Matthew K Leonard, Bingni W Brunton, and Yisong Yue. Deep neural imputation: A framework for recovering incomplete brain recordings. In *NeurIPS 2022 Workshop on Learning from Time Series for Health*, 2022. URL https://openreview.net/forum?id=c9qFg8UrIcn.

[51] Samuel M Peterson, Shiva H Singh, Benjamin Dichter, Kelvin Tan, Craig DiBartolomeo, Devapratim Theogarajan, Peter Fisher, and Josef Parvizi. Ajile12: Long-term naturalistic human intracranial neural recordings and pose. *Scientific Data*, 9(1):184, 2022. ISSN 2052-4463. doi: 10.1038/s41597-022-01280-y. URL https://doi.org/10.1038/s41597-022-01280-y.

[52] Adrien Doerig, Rowan Sommers, Katja Seeliger, Blake Richards, Jenann Ismael, Grace Lindsay, Konrad Kording, Talia Konkle, Marcel A. J. Van Gerven, Nikolaus Kriegeskorte, and Tim C. Kietzmann. The neuroconnectionist research programme, 2022. URL https://arxiv.org/abs/2209.03718.

[53] Richard Antonello, Aditya Vaidya, and Alexander G. Huth. Scaling laws for language encoding models in fmri, 2024. URL https://arxiv.org/abs/2305.11863.

[54] Tyler Benster, Guy Wilson, Reshef Elisha, Francis R Willett, and Shaul Druckmann. A cross-modal approach to silent speech with llm-enhanced recognition. *arXiv preprint arXiv:2403.05583*, 2024.

[55] Weihao Xia, Raoul de Charette, Cengiz Öztireli, and Jing-Hao Xue. Umbrae: Unified multimodal brain decoding. In *European Conference on Computer Vision (ECCV)*, 2024.

[56] Quilee Simeon, Leandro Venâncio, Michael A Skuhersky, Aran Nayebi, Edward S Boyden, and Guangyu Robert Yang. Scaling properties for artificial neural network models of a small nervous system. In *SoutheastCon 2024*, pages 516–524. IEEE, 2024.

[57] Motoshige Sato, Kenichi Tomeoka, Ilya Horiguchi, Kai Arulkumaran, Ryota Kanai, and Shuntaro Sasai. Scaling law in neural data: Non-invasive speech decoding with 175 hours of eeg data, 2024. URL https://arxiv.org/abs/2407.07595.

[58] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19. ACM, January 2019. doi: 10.1145/3287560.3287596. URL http://dx.doi.org/10.1145/3287560.3287596.

[59] Edward H Adelson, James R Bergen, et al. *The plenoptic function and the elements of early vision*, volume 2. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of . . . , 1991.

[60] Gregor Bachmann and Vaishnavh Nagarajan. The pitfalls of next-token prediction, 2024. URL https://arxiv.org/abs/2403.06963.

[61] Colin Lea, Rene Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks: A unified approach to action segmentation, 2016. URL https://arxiv.org/abs/1608.08242.

[62] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting, 2024. URL https://arxiv.org/abs/2310.10688.

[63] Demetres Kostas, Stéphane Aroca-Ouellette, and Frank Rudzicz. BENDR: Using transformers and a contrastive self-supervised learning task to learn from massive amounts of EEG data. *Frontiers in Human Neuroscience*, 15, 2021. ISSN 1662-5161. doi: 10.3389/fnhum.2021.653659. URL https://www.frontiersin.org/articles/10.3389/fnhum.2021.653659.

[64] Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. What language model architecture and pretraining objective work best for zero-shot generalization?, 2022. URL https://arxiv.org/abs/2204.05832.

[65] Nur Muhammad Mahi Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone, 2022. URL https://arxiv.org/abs/2206.11251.

[66] Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taïga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, Aviral Kumar, and Rishabh Agarwal. Stop regressing: Training value functions via classification for scalable deep rl, 2024. URL https://arxiv.org/abs/2403.03950.

[67] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. URL https://arxiv.org/abs/2104.14294.

[68] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[69] Francis R. Willett, Donald T. Avansino, Leigh R. Hochberg, Jaimie M. Henderson, and Krishna V. Shenoy. High-performance brain-to-text communication via handwriting. *Nature*, 593(7858):249–254, May 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03506-2. URL https://www.nature.com/articles/s41586-021-03506-2.

[70] Chen Yang, Junzhuo Li, Xinyao Niu, Xinrun Du, Songyang Gao, Haoran Zhang, Zhaoliang Chen, Xingwei Qu, Ruibin Yuan, Yizhi Li, Jiaheng Liu, Stephen W. Huang, Shawn Yue, and Ge Zhang. The fine line: Navigating large language model pretraining with down-streaming capability analysis, 2024. URL https://arxiv.org/abs/2404.01204.

[71] Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models, 2022. URL https://arxiv.org/abs/2212.00638.

[72] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

[73] Linxing Preston Jiang, Shirui Chen, Emmanuel Tanumihardja, Xiaochuang Han, Weijia Shi, Eric Shea-Brown, and Rajesh P. N. Rao. Data heterogeneity limits the scaling effect of pretraining neural data transformers. *bioRxiv*, 2025. doi: 10.1101/2025.05.12.653551. URL https://www.biorxiv.org/content/early/2025/05/15/2025.05.12.653551.

[74] Robert D Flint, Eric W Lindberg, Luke R Jordan, Lee E Miller, and Marc W Slutzky. Accurate decoding of reaching movements from field potentials in the absence of spikes. *Journal of neural engineering*, 9(4): 046006, 2012.

[75] Chethan Pandarinath, Paul Nuyujukian, Christine H Blabe, Brittany L Sorice, Jad Saab, Francis R Willett, Leigh R Hochberg, Krishna V Shenoy, and Jaimie M Henderson. High performance communication by people with paralysis using an intracortical brain-computer interface. *elife*, 6:e18554, 2017.

[76] Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. Brain–computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791, 2002.

[77] Beata Jarosiewicz, Anish A Sarma, Daniel Bacher, Nicolas Y Masse, John D Simeral, Brittany Sorice, Erin M Oakley, Christine Blabe, Chethan Pandarinath, Vikash Gilja, et al. Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface. *Science translational medicine*, 7 (313):313ra179–313ra179, 2015.

[78] Mariana P. Branco, Lisanne M. De Boer, Nick F. Ramsey, and Mariska J. Vansteensel. Encoding of kinetic and kinematic movement parameters in the sensorimotor cortex: A brain-computer interface perspective. *European Journal of Neuroscience*, 50(5):2755–2772, September 2019. ISSN 0953-816X, 1460-9568. doi: 10.1111/ejn.14342. URL https://onlinelibrary.wiley.com/doi/10.1111/ejn.14342.

[79] Kristin M Quick, Jessica L Mischel, Patrick J Loughlin, and Aaron P Batista. The critical stability task: quantifying sensory-motor control during ongoing movement in nonhuman primates. *Journal of Neurophysiology*, 120(5):2164–2181, 2018.

[80] Patrick J Marino, Lindsay Bahureksa, Carmen Fernández Fisac, Emily R Oby, Adam L Smoulder, Asma Motiwala, Alan D Degenhart, Erinn M Grigsby, Wilsaan M Joiner, Steven M Chase, et al. A posture subspace in primary motor cortex. *bioRxiv*, pages 2024–08, 2024.

[81] Matthew J Mender, Samuel R Nason-Tomaszewski, Hisham Temmar, Joseph T Costello, Dylan M Wallace, Matthew S Willsey, Nishant Ganesh Kumar, Theodore A Kung, Parag Patil, and Cynthia A Chestek. The impact of task context on predicting finger movements in a brain-machine interface. *eLife*, 12:e82598, jun 2023. ISSN 2050-084X. doi: 10.7554/eLife.82598. URL https://doi.org/10.7554/eLife.82598.

[82] Xuan Ma, Fabio Rizzoglio, Eric J. Perreault, Lee E. Miller, and Ann Kennedy. Using adversarial networks to extend brain computer interface decoding accuracy over time. Aug 2022. doi: 10.1101/2022.08.26.504777. URL https://www.biorxiv.org/content/10.1101/2022.08.26.504777v1.

[83] Kendra K. Noneman and J. Patrick Mayo. Decoding continuous tracking eye movements from cortical spiking activity. *International Journal of Neural Systems*, page S0129065724500709, October 2024. ISSN 0129-0657, 1793-6462. doi: 10.1142/S0129065724500709. URL https://www.worldscientific.com/doi/10.1142/S0129065724500709.

[84] Chaofei Fan, Nick Hahn, Foram Kamdar, Donald Avansino, Guy H Wilson, Leigh Hochberg, Krishna V. Shenoy, Jaimie M. Henderson, and Francis R Willett. Plug-and-play stability for intracortical brain-computer interfaces: A one-year demonstration of seamless brain-to-text communication. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=STqaMqhtDi.

[85] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning, 2023. URL https://arxiv.org/abs/2307.08691.

[86] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin F. Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Patrick Collier, Alexey Gritsenko, Vighnesh Birodkar, Cristina Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetić, Dustin Tran, Thomas Kipf, Mario Lučić, Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters, 2023. URL https://arxiv.org/abs/2302.05442.

[87] Mitchell Wortsman, Peter J Liu, Lechao Xiao, Katie E Everett, Alexander A Alemi, Ben Adlam, John D Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, Jeffrey Pennington, Jascha Sohl-Dickstein, Kelvin Xu, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Small-scale proxies for large-scale transformer training instabilities. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=d8w0pmvXbZ.

[88] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories, 2021. URL https://arxiv.org/abs/2012.14913.

A Appendix

Contents

A.1 Related works and a proposed taxonomy 24
A.2 Supplementary Results 25
A.2.1 Further tests of held-out angle generalization. 25
A.2.2 Ablations 26
A.2.3 Isolated scaling in neural data 27
A.2.4 Pretraining does not benefit FALCON H2 (Handwriting) 28
A.2.5 Multiscale decoding on individual motor tasks 28
A.2.6 Sequential tuning is similar to joint tuning 29
A.2.7 Aggregate performance on all NDT3 models with significance tests 31
A.2.8 Neural vs Behavioral Objectives 32
A.2.9 Downstream probe through pretraining 32
A.2.10 Evaluation on the Neural Latents Benchmark 33
A.2.11 Neural objectives are needed in fine-tuning 34
A.3 Methods 34
A.3.1 Metrics and Evaluation 34
A.3.2 Training 34
A.3.3 Baselines and Other Transformers 35
A.3.4 Pretraining and Evaluation datasets 38
A.3.5 Generalization analyses and Further evaluations 38
A.3.6 Architectural Details 38

A.1 Related works and a proposed taxonomy

Neural data is sufficiently diverse so as to support many distinct efforts to train large neural data models. The scale of pretraining is somewhat larger in the non-implanted modalities, where data is more abundant. The largest EEG models have reached a scale of 2.5K [42] to 40K hours of data [43], or higher volumes if also considering non-brain biosignals (EKG) [44, 45]. Current fMRI models operate in the 1K [46] to 7K [47] hour range. The largest models in these studies are in the 0.1B-1B parameter range. Intracranial modalities, including sEEG [48, 49], ECoG [50, 51], and spiking activity [6], have thus far been studied at an order of magnitude smaller scales of data and model size (20-1000 hours, <0.1B parameters).

Direct scaling on neural data modeling should be distinguished from NeuroAI efforts [52] to measure how models of the human sensorimotor experience (e.g. language, vision, audio models) predict neural data [53]. However, as multimodal efforts begin to blur this distinction [54, 55], care will be required to distinguish advances in modeling neural data, embodied data, or their interaction.

Comparing neural data models Current efforts to understand scaling in neural data Simeon et al. [56], Sato et al. [57] will have their reach limited by the specificity of every neural dataset. A meta-challenge for the field is understanding how different parameters (species, brain area, modality, task) impact scaling properties. This would be greatly aided by development of reporting practices for different neural data models. To facilitate comparison, we create a model card in the NDT3 codebase [58]. In addition to the standard model card, we propose reporting an additional taxonomy to aid comparisons across neural data models, using two concepts.

First: neural data models can be conceptualized as modeling slices of the *plenneural function*, inspired by the plenoptic function in vision [59]. The plenoptic function is a model of an idealized eye which

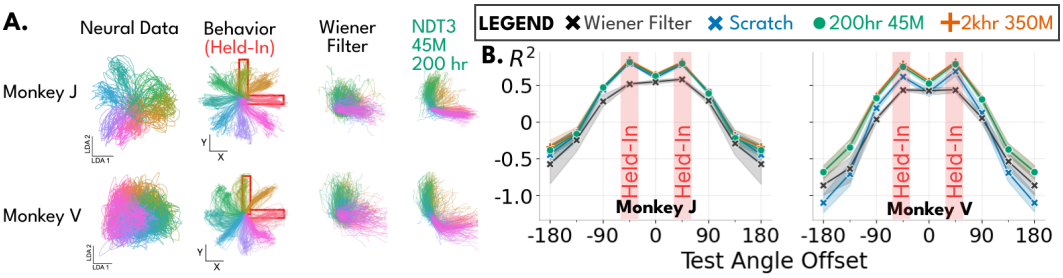

Figure 6. A. Two monkey 2D center-out datasets with 8 angular conditions. LDA projections show monkey J's data is distinctly more separable than monkey V's. We then test generalization of behavioral decoding to held-out angles after training on 2 of 8 angles (boxed in red). All NDT3s and the WF produce predictions constrained between the held-out angles. The Wiener Filter here, unlike in the main text, is a direct fit on neural data without dimensionality reduction. **B.** We quantify performance for decoding on each angle with respect to distance from the central angle between the held-in angles. We average performance over all 8 central angles.

parameterizes all possible images with 7 dimensions: 4D to describe the global spacetime of the view, 2D to describe viewing angle (spherical) or coordinate (Cartesian) of the image, and 1D for wavelength. Since neural data models are primarily interested in circumscribed systems rather than the physical world, a similar global coordinate system (e.g. 4D for all possible electric potentials) would be uninformative. We thus propose reporting more qualitative coordinates:

1. Identity: The network or individual being recorded.
2. Task: The behavior, stimuli, or other activity the network is reflecting.
3. Spacetime: Coordinates specified in a network-local coordinate frame (e.g. brain area).

Second: The modeled extent of this plenneural function is conveniently discretized in three resolutions in a Transformer-like sequence modeling framework: the token, the sequence, and the full training data. The token is the most granular unit of data being modeled; NDT3 models neural populations 32 neurons at a time, in 20ms bins. At the sequence input level, NDT3 models inputs from single humans or monkeys, across 128-256 neurons in 2 second snippets, while performing effectively one "movement." Finally, NDT3's pretraining spans dozens of individuals, records motor and premotor areas over 2.5K hours, over a variety of arm and hand movements.

A.2 Supplementary Results

A.2.1 Further tests of held-out angle generalization.

To further support our single-dataset illustration of attractor structure in pretrained NDT3, we evaluate reach angle generalization across sessions in the isometric, force-based dataset (Monkey J) setting and a second, manipulandum-based (Monkey J) setting with monkey movement. To begin, we visualize the separability of these neural data with LDA (Fig. 6A Neural Data). We next train decoders on every pair of angles separated by 90 degrees (one shown) and plot predictions on held-out trials from all angles. NDT3 and WFs, here directly fit to high-D data instead of after PCA-LDA, both fail to extrapolate to held-out angles, consistent with Rizzoglio et al. [33], illustrating the necessity of the dimensionality reduction for generalization in the main text. We quantify prediction performance in Fig. 4D. These plots again show that held-out generalization is subtrivial, while being entirely consistent with pretraining's overall narrative of improved performance in all conditions.

Methods for PCA-LDA. Our PCA-LDA preparation used in Fig. 4C-E adheres closely to our standard data preparation and is directly comparable to the inputs received by the deep networks. To begin, we smooth all spike counts with an exponential kernel, as in our Wiener Filter baseline (Section A.3.3). We then z-score these smoothed spikes, and fit PCA to the high-dimensional (96D) z-scored neural activity at each timestep to extract the top 10 PCs. Finally, we fit linear discriminant analysis (LDA) to reduce this 10D PC space to neural activity to 2D. Example neural data projections are shown in Fig. 6A, showing clear radial structure that can be exploited for generalization. The training data for PCA and LDA are both restricted to only the held-in angles. Note that LDA uses categorical labels to separate the 3 held-in reach directions but is applied without class labels after fitting. While this train-time labeling is technically not given to NDT3, this does not influence our

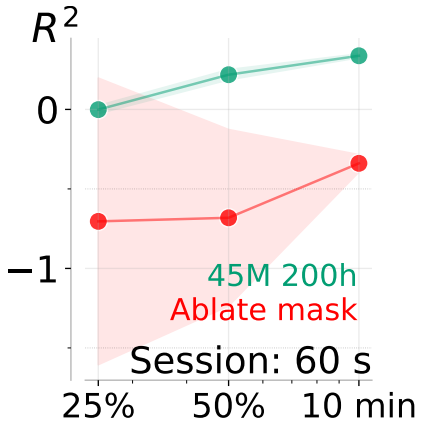

Figure 7. Ablation of covariate masking on an open 2D Cursor + Click dataset. Covariate inputs are completely masked in inference for the default NDT3, and autoregressively generated in the ablation.

argument that NDT3 pretraining should be capable of angular generalization, as NDT3 does not confuse the reach angle of held-in predictions. The final Wiener Filter is fit to the 2D data and uses 1 second of history.

A.2.2 Ablations

We ablate the major design decisions made to enable NDT3's large scale pretraining. These ablations give us confidence that NDT3 overcomes the basic challenges we encountered in development, but compute restrictions prevent more exhaustive comparisons or exploration of model design space. We encourage further work exploring the influence of different hyperparameters. In these plots, we distinguish validation split performance and evaluation split performance, which is computed by batch-mode prediction (not the costly streaming evaluation used throughout main experiments).

Covariate dropout We find the default next-step prediction objective fails for learning decoding of highly autocorrelated covariate timeseries, perhaps because simply relying on teacher-forced behavioral inputs provides a severe shortcut that prevents learning of a proper neural to behavior decoding map [60]. Different time-series models have addressed this by adopting convolutional input-output layers [61], tokenizing along temporal dimensions [62], or learning with contrastive objectives [49, 63]. We avoid introducing architectural modifications and instead adopt a simple dropout procedure that masks a portion of covariate inputs some fraction of the time. Specifically, on every training batch, two random numbers are drawn. The first, $M \sim U[0, 1]$, determines what fraction of covariate inputs should be masked. On 90% of batches, we also sample $T \sim U[0, 2]$ seconds, such that the mask is only applied after timestep T. That is, on 90% of batches, the model is provided a prefix-prompt. We do not block losses on this prefix as in prefix-LMs [64]. Pretraining metrics for validation and evaluation are always computed with a prefix and full masking of non-prefix timesteps. In Fig. 7, we ablate covariate masking (which also removes the prefix logic), and tune on a 2D Cursor + Click task. The ablated model performs subtrivially with student-forced predictions provided as input at test time. Note that the ablated model performs trivially with masked inputs (not shown).

BCI-phase and return conditioning NDT3's pretraining includes several hundred hours of BCI control data, where the covariates were set by another decoder. We introduced phase and return conditioning tokens to differentiate the several types of BCI control data from recorded behavior. Specifically, in BCI data, NDT3 receives input tokens specifying what fraction of the behavior reflects neural input (BCI control is on) vs programmatic input (BCI control is off, as in open loop BCI calibration). Further, we provide inputs encoding reward (trial success) when trials change, and return (future reward over a 10 second horizon, which crosses data boundaries). This design is intended to evaluate the potential for a Decision-Transformer like offline learning strategy for improved online control, but we do not discuss this in this work. In Fig. 8A, B, we focus on whether these inputs improves pretraining loss and R^2 in validation splits, which has contains BCI data, and the held-out evaluation split containing only monkey behavior. The figures show that the ablation significantly

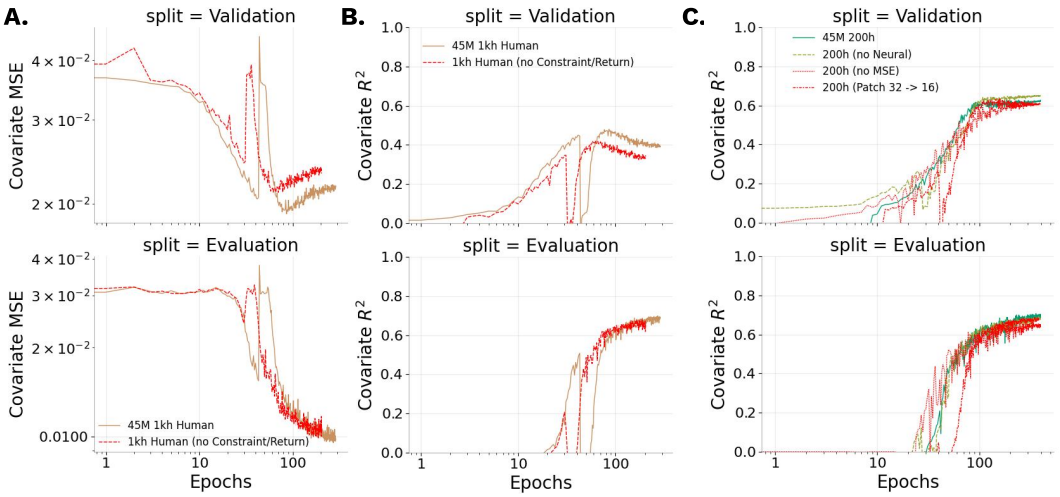

Figure 8. Ablations evaluated based on upstream evaluation split. **A., B.** Ablation of BCI control tokens **C.** Ablation of neural objective and covariate MSE objective in favor of classification over quantized covariates.

decreases validation split performance, and causes a slightly earlier stopping point leading to worse evaluation performance. Note both models early before the full training budget of 400 epochs.

Neural reconstruction objective All main NDT3 models used a neural reconstruction objective inherited from the self-supervised learning pretraining from NDT2. We ablate this choice post-hoc and see it may actually minorly harm pretraining (validation split), though the neural objective doesn't harm evaluation split decoding (Fig. 8C). Note the scalar weighting of neural vs covariate objectives were set to be roughly balanced in pretraining. Section A.2.3 provides a downstream analysis on the standalone value of the neural reconstruction objective.

MSE over classification In robotics and certain generalist models [14], continuous action spaces are sometimes better decoded and controlled when quantized [65]. This is because MSE is an insufficient objective when the output distribution is multimodal (e.g. one of two possible paths in robotics). While it seems unlikely that the close relationship between movement behavior and motor cortex is multimodal, multimodal behavior may be appropriate when pretrained on heterogeneous data, i.e. when similar neural activity corresponds to different behavior in two datasets. We attempted such a quantization, including HL-Gauss smoothing [66] which we found to help; but this does not recover the performance of the default MSE objective (Fig. 8C) on the evaluation split. We found this performance gap persisted under fine-tuning (not shown). This suggests that NDT3 is differentiates neural data inputs from different datasets.

Patch size NDT2 and NDT3 both tokenize neural data by patching them into fixed size clusters. It is unclear whether transfer learning might occur for sub-token features, which motivates the use of smaller tokens in larger datasets that might afford it [67]. We change patch size to 16 and show this performs slightly worse in the 45M 200h model (Fig. 8)C. Smaller patches (and subsequent increased neural tokens) may be more beneficial in the larger scale models, but their benefit must be weighed against their increased compute burden.

A.2.3 Isolated scaling in neural data

Due to NDT3's joint modeling of behavior and neural data, it is difficult to dissociate whether scaling gains in behavior come from improved behavioral or neural priors. To assess whether NDT3 can scale solely from neural data modeling, we pretrain a new set of 45M parameter models up to 200 hours with only causal neural data modeling objective. As before, we then tune to a downstream decoding task, in this case, a Critical Stability Task dataset (CST). From a representation probing perspective, improved downstream performance implies higher quality neural representations. We use the standard single-stream autoregressive modeling objective as in the rest of this work in the downstream setting, we find direct linear probing of neural representations perform worse. Fig. 9 compares the scaling on downstream neural and behavioral metrics after the standard fine-tuning procedure.

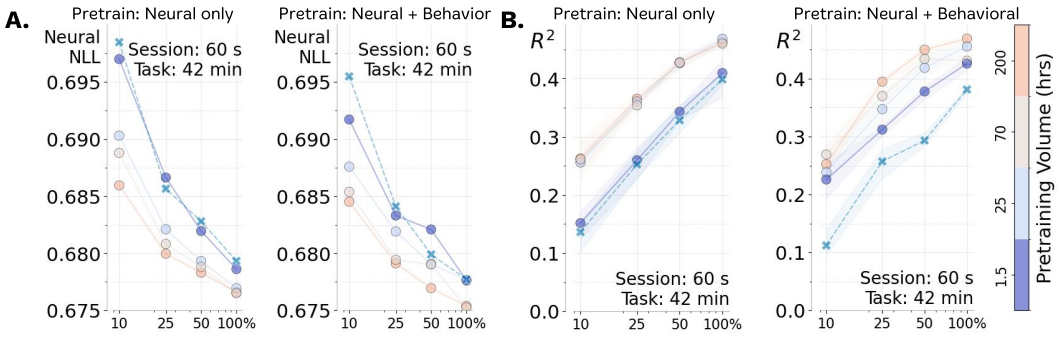

Figure 9. Scaling of unsupervised pretraining After pretraining models up to 200 hours with only the neural reconstruction objective, we fine-tune models in a similar multiscale evaluation as in Section 3.1, with a newly initialized behavioral readout. We use the CST task here. **A.** left shows neural loss scales with neural-only pretraining. The right panel plots the neural loss, also present in the standard evaluation, also scales with joint pretraining. **B.** left shows that with neural pretraining, downstream decoding performance saturates at a flat performance after 25 hours. This is compared against the nonsaturated scaling from joint pretraining. Colorbar is common for all plots, and Xs are from-scratch NDT3 models.

Fig. 9A shows that downstream neural reconstruction improves with increased pretraining data either when using only the neural objective or both neural and behavioral objectives (as in the standard setting). The joint pretraining achieves advances neural metrics in all settings, illustrating that decoding behavior is a complementary objective to neural data reconstruction even for representation learning.

Fig. 9B contrasts decoding curves in the two pretraining settings, in that neural pretraining has saturated decoding after just 25 hours of pretraining. This is consistent with the interpretation that the behavioral readout reflects only one aspect of the neural data. Together with the neural metric plots, this analysis shows scaling over solely neural data is possible, but also that decoding behavior is a complementary pretraining objective for improving neural representation learning and decoding.

A.2.4 Pretraining does not benefit FALCON H2 (Handwriting)

We also evaluated NDT3 for decoding of letters in a human-open loop handwriting task (FALCON H2). Although this is also a motor cortical decoding task, we excluded H2 from NDT3's aggregate evaluation since it is a sequence-to-sequence as opposed to continuous task. To apply NDT3 to this task, we pool neural tokens at each timestep and add a linear projection and optimize with a CTC loss [68]. We maintain the default neural reconstruction loss and causal attention mask, and do not apply data augmentation.

Note that RNNs are the current standard architecture for communication tasks like H2 [23, 69]. Training and tuning was less stable than for our continuous decoding tasks and required more extensive hyperparameter tuning, perhaps because the overall dataset size remains small (<1k samples), specific parameters are listed in the codebase. We observe three regimes in both training and fine-tuning. First, the model can fail to achieve an initial learning period. Second, the model can achieve reasonable nontrivial solutions, comparable to expected performance for unaugmented RNNs (though we do not quantify this). Third, some models will exhibit learning instabilities that resolve in significantly improved performance. We illustrate these regimes in example validation curves below. Overall, the third regime is rarely achieved. More relevant to the main narrative of this work, fine-tuning appears to degrade both final solution quality and reduces the range of nontrivial hyperparameters (not shown). Investigating a sequence to sequence objective over CTC loss would be valuable future work.

A.2.5 Multiscale decoding on individual motor tasks

Fig. 11A plots model performance for each of the 31 evaluation settings we study in the eight primary evaluation datasets we use. Studying any individual dataset will yield variable conclusions on whether pretraining structure is helpful, underscoring the need for proposed foundation models to be evaluated across many different datasets. We see the most clear scaling with pretraining data (color gradient

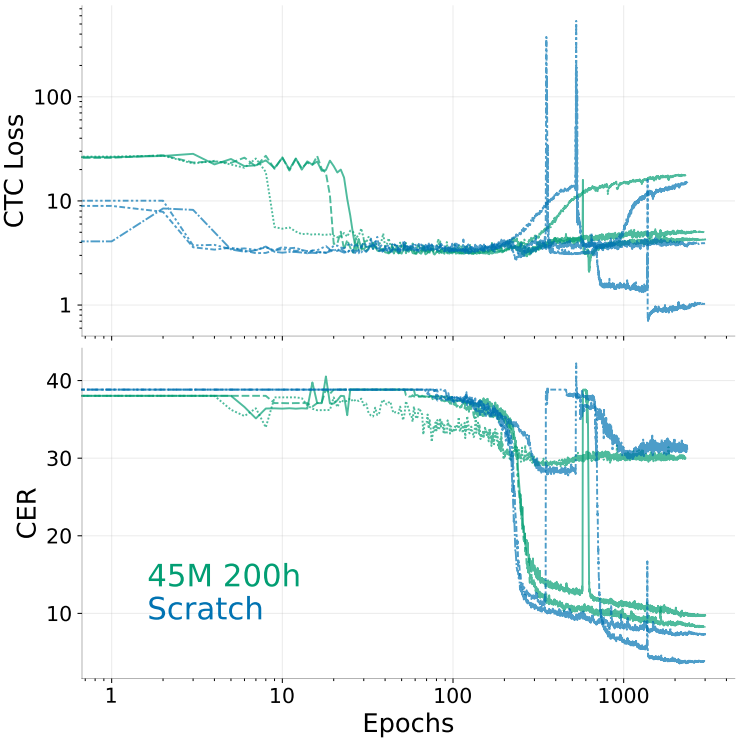

Figure 10. Three regimes of NDT3 training for handwriting decoding. We show validation loss and character error rates for example runs of from-scratch and fine-tuned NDT3s.

with red on top) in the Critical Stability Task and Bimanual Task. FALCON tasks and Self-paced Reach appear minimally affected by scaling pretraining data, in that either pretrained models are generally slightly above a from scratch model at all data scales with no particular best pretrained model. The 2D + Click and Grasp datasets uniquely show high variability in model performance and strong degradation of the 350M 2 khr model at low data scales. Grasp instability was so high that we trained 9 seeds instead of the standard 3 to better estimate model performance. We propose this degradation is due to the instability of full fine-tuning of large models at the extremely low data scales these datasets present (e.g. 2.5 minutes at the 25% scaling). Finally, we remind that the 2D + Click, FALCON H1, and 1D Grasp Force tasks are datasets from human participants that are included in the 2 khr pretraining. Surprisingly, this leakage did not provide clear benefit to the 2 khr model.

These scaling plots also provide more precise context for baseline performance. NDT2 performs particularly poorly in the low data regime, while Wiener Filters perform poorly in the high data regimes.

In Fig. 11B, we illustrate qualitative predictions on private datasets. These visualizations show a diversity in covariate timescales and structure. They also illustrate that the summary R^2 obscure several features of model predictions. For example, pretrained models in Cursor Y tend have false positive deflections in movement. R^2 also is not easily comparable in tasks with continuous dynamics (CST) vs. transient dynamics (Cursor G1).

A.2.6 Sequential tuning is similar to joint tuning

In Section 3.2, we showed that channel shuffling and half-token shifts were sufficient to reduce cross-session transfer to the extent of cross-subject transfer, in the context of a pretrained 350M 2khr NDT3 model and a from-scratch NDT3 model. Here we add 1) joint tuning results in addition to the sequential tuning in the main text, 2) extensions of this analysis to multiple intermediate tuning data scales, 3) an additional replication of this analysis for the 45M 200hr NDT3 model.

The comparison of sequential fine-tuning against with joint fine-tuning is done for completeness. Given cross-context data and a test session, we can either jointly tune on all data (as we do in Section

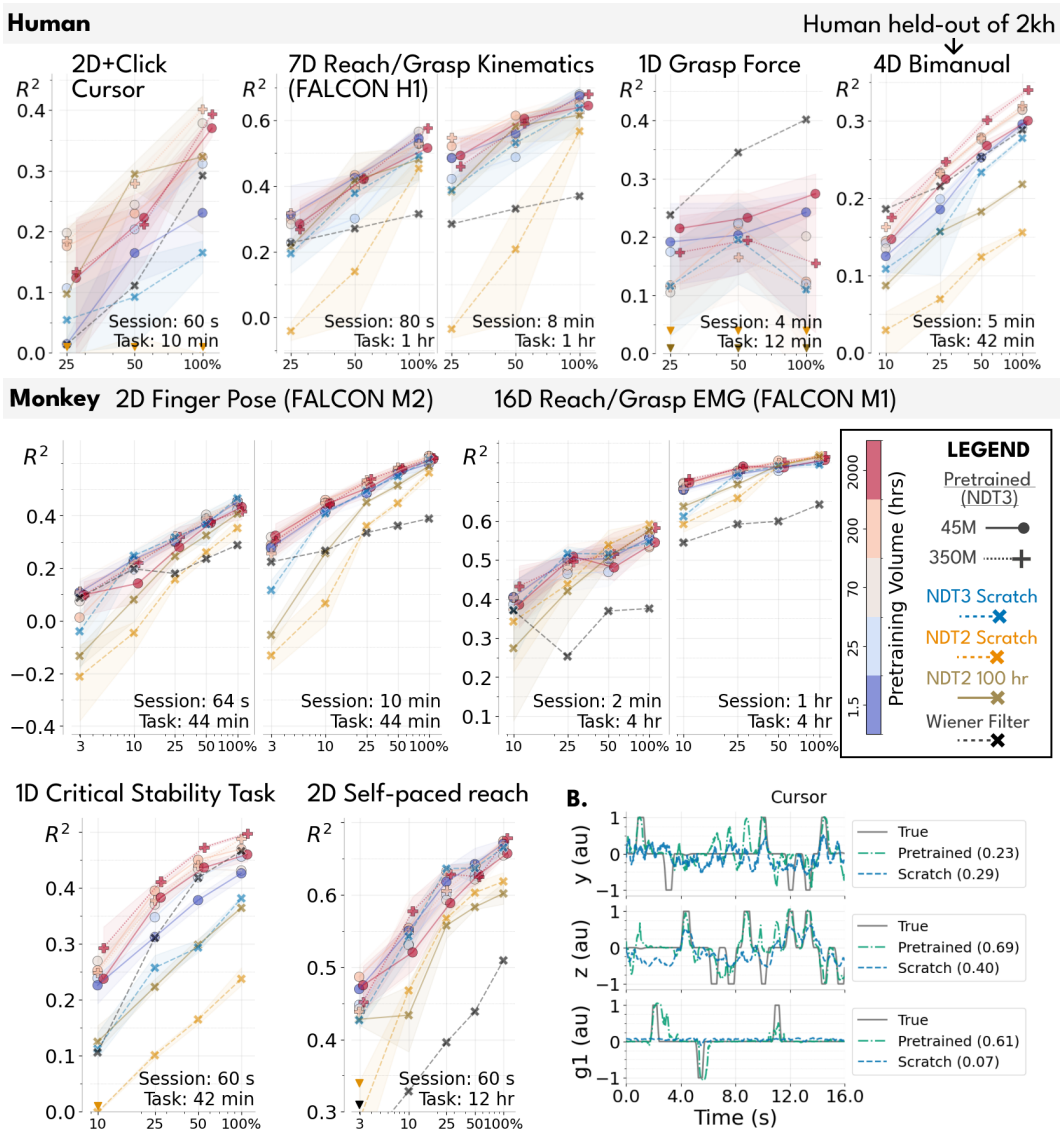

Figure 11. A. Fine-tuning evaluations for individual datasets. Performance on both held-out (left) and held-in (right) splits are shown side by side by FALCON datasets. We shade the standard deviation of 3 model seeds in fine-tuning. Different tasks show substantial variability in benefit from pretraining. B. We show example predictions of a pretrained (45M 200h) and from-scratch NDT3 for the 2D + Click Cursor task to give a sense of what different prediction performances mean in terms of open loop data prediction. Numbers in legend are the R^2 for that model's predictions in the shown snippet.

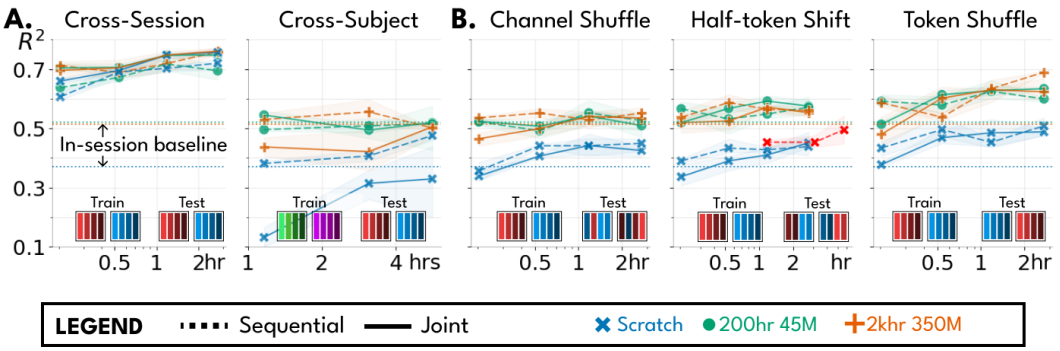

Figure 12. A replication of Fig. 4A/B, but showing results for both sequential and joint fine-tuning in each setting, and on multiple fine-tuning data scales. As before, each model here tunes with some data from other settings (e.g. cross-subject data for the cross-subject panel), and a fixed amount of data on the test session. Sequential tuning first tunes on other-setting data, and then tunes on test session data. Joint tuning uses all data at once. In Fig. 4A/B, we only showed the better choice for each panel, i.e. we showed joint tuning for cross-session data, and sequential tuning for the other settings. In addition, we overlay how the cross-subject tuned models perform when applied to half-token shifted test data, confirming that subject and session data transfer similarly to shifted test data. Shading shows the standard deviation on 3 model seeds. In-session baseline refers to the no-intermediate tuning setting in Fig. 4B.

3.1), or sequentially tune first on the cross-context data and then on the test session. The latter is the protocol used in most other works and what we adopt in the main text for this analysis. We find for pretrained models that the results are similar, but that for from-scratch models, sequential tuning is essential for cross-subject transfer and slightly harmful for cross-session models (Fig. 12A). Seeing that sequential tuning is mainly advantageous for from-scratch models, we speculate that sequential tuning is particularly helpful for filtering learning signals in highly heterogeneous data, but pretrained models develop a capacity for doing such filtering during fine-tuning.

For the multiple intermediate data scale, only the largest scale was reported in the main text. Performance for most models decay towards the no intermediate tuning baseline. The exception here is cross-session tuning, where even minor amounts of data provide substantial gain, and cross-subject transfer in the joint tuning setting for from-scratch models, where, as mentioned, the model performs very poorly.

The 45M 200hr model performs differently than the 350M 2khr at individual settings but without any changes in overall trends.

A.2.7 Aggregate performance on all NDT3 models with significance tests

We additionally report the average performance of an NDT2 model pretrained with 100 hours of human data and two NDT3 models pretrained with 25 hours and 70 hours. These models are placed in context with the models from Fig. 3D, in Fig. 13A. Note that for NDT3 models each successively larger data scale uses a strict superset of data from smaller scales. We also provide the p-values computed for the significance of the difference between each pair of models in Fig. 13B. P-values are computed as FDR-corrected pairwise t-tests. The 350M 2 khr model has $p < 0.06$ improvements over all but the 350M 200 hr model. Interestingly, all other pretrained models, except the 25 hr model, appear equivalent, at least statistically. We presume this is due to the fact that our evaluation of 31 task settings may be insufficiently large.

NDT2 performs relatively poorly in our evaluation. This is true even when tuning from the public checkpoint trained on 100 hours of neural data from humans, though tuning does in general improve over the from-scratch NDT2 training. We believe pretrained NDT2's performance gap with NDT3 is partially due to NDT2's need to newly initialize decoding layers in each downstream task, which increases NDT2's dependence on thorough hyperparameter tuning. This makes NDT2 a poor candidate for a foundation model. Section A.3.3 and Section A.3.6 describe a number of methodological innovations that likely each contribute to the remaining performance differences between NDT2 and NDT3.

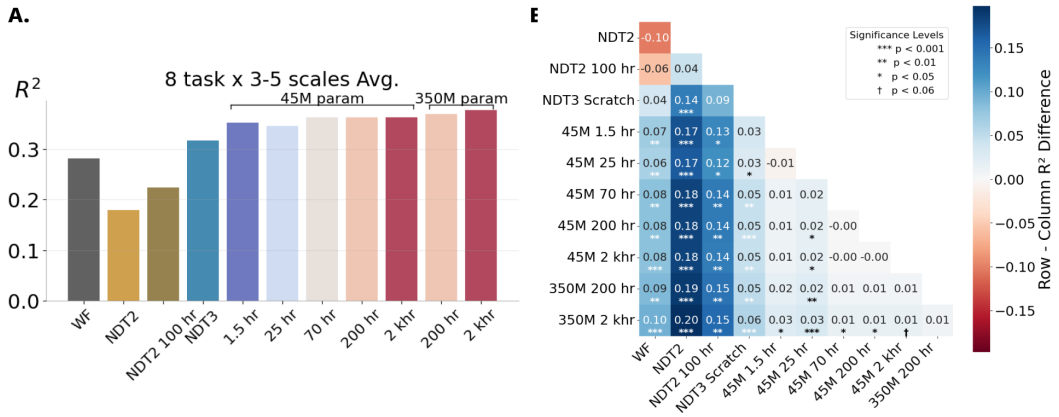

Figure 13. A. A replication of Fig. 3D including an additional 25 hr and 70 hr model. The two additional models show the precise performance we measure may be noisily related to to pretraining data scale. B. Heatmap of differences between performances of pairs of models with significance tested with FDR-corrected pairwise t-tests. Note that coloring is used here to indicate differences, not significance. Positive numbers with significance indicate the row model outperforms the column model.

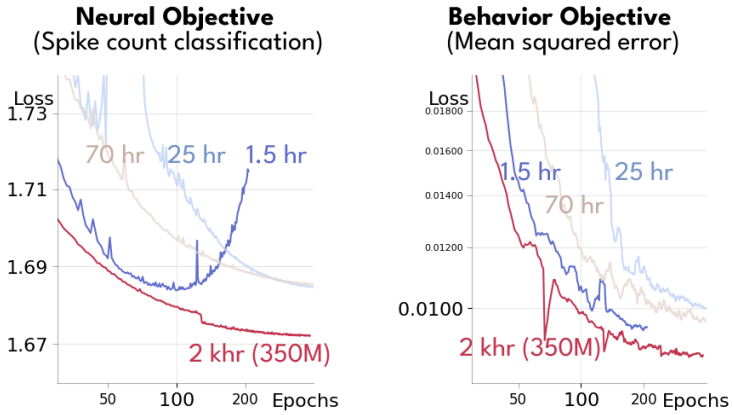

Figure 14. Pretraining curves shown for 45M parameter NDT3s at 1.5 hour, 25 hour, and 70 hours, in addition to the curves for the 2 khr 350M parameter NDT3. We separately show metrics on neural and behavioral objectives through training. Since early stopping is used in model selection, we verify here that neither objective is overfits significantly, except for the 1.5 hour model's neural objective.

A.2.8 Neural vs Behavioral Objectives

In this work, the neural objective is present mainly as an auxiliary objective to improve downstream decoding. We see that neural and behavioral objectives are complementary in Section A.2.3.

In Fig. 14, we provide some additional context, showing that neural objectives also improve through pretraining, i.e. that we are not overfitting the neural objective and thus degrading decoding.

A.2.9 Downstream probe through pretraining

Upstream and downstream performance are generally assumed to be correlated in pretraining studies. This assumption supports the use of a single evaluation at the end of pretraining, rather than throughout pretraining, particularly if pretraining is not overfit. It is possible, however, that the narrower tasks used in evaluation may be learned or even overfit earlier than the general pretraining task. To assess this, we conduct a small downstream probe of checkpoints every 20 epochs of pretraining for a 45M 25 hour NDT3. This probe measures the performance specifically at the 25% scaling of the bimanual task. In Fig. 15, we show this progression and compare the variability of checkpoints in pretraining with the differences across different task scales and pretraining models. The downstream probe for this task shows quick benefit from pretraining, providing the full gain over from-scratch models by the first checkpoint we evaluate at epoch 20, and then plateaus. This contrasts with the

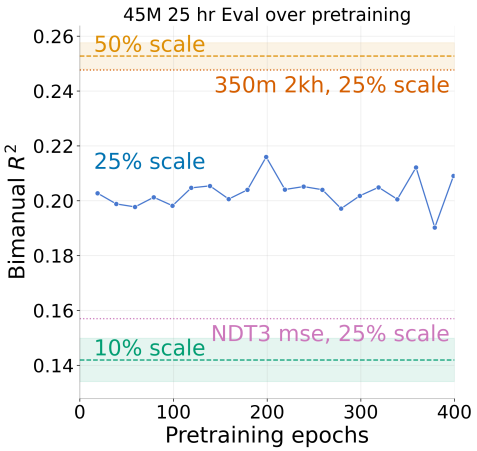

Figure 15. Downstream evaluation conducted through pretraining of a 45M 25 hour NDT3, for the 25% scaling of the bimanual task, provided in context of the final evaluation achieved by the 10% and 50% bimanual task scale settings for the 45M 25 hour models, and the 25% scalings for the from-scratch NDT3 and best NDT3 (350M 2 khr). The benefits from pretraining are attained by epoch 20, and plateaus without overfitting for the rest of pretraining.

Dataset	RTT	Maze	Maze Large	Maze Medium	Maze Small
NDT1	0.1643	0.3597	0.3739	0.3081	0.2788
SoTA	0.2010	0.3650	0.3831	0.3329	0.3458
NDT3 Scratch	0.1533	0.3093	0.2892	0.1859	0.1853
NDT3 350M 2kh	0.1695	0.2775	0.2781	0.1937	0.2116

Table 1. NDT1 vs NDT3 co-bps (higher is better) on NLB's MC Maze and MC RTT datasets in the 20ms split. The NDT3 results were produced were the standard sweeps used in this work, e.g. three random learning rates and three random seeds. SoTA results come from LangevinFlow for RTT and Maze and MINT for the Maze scaling datasets.

pretraining plots in Fig. 14. The mismatch between upstream and downstream progress here differs from correlated upstream-downstream progress in single-epoch language model studies [70], which may be due to differences in domain or pretraining scale.

A.2.10 Evaluation on the Neural Latents Benchmark

The Neural Latents Benchmark [5] is a benchmark for evaluating latent variable modeling of neural activity. This evaluation differs from NDT3's direct decoding evaluations in that all of NLB's metrics are derived from acausally inferred neural firing rates, akin to a pixel-level objective in computer vision. Note specifically that the decoding metric officially reported in the NLB is derived from submitted firing rates on the evaluation server; report decoding performance on NLB datasets without modeling neural activity is not an expected use of the benchmark.

Despite its distinct setting, the NLB provides two well-established datasets on which to evaluate models of motor cortical activity, providing context for NDT3's performance outside of decoding. To apply NDT3 to the NLB, we tuned NDT3 in a new downstream task where insert new a token at each timestep, from which we linearly decode firing rates of held out neurons. We performed this tuning separately for each of the maze and random target tasks (RTT), reporting the resulting co-bps scores in Table 1. NDT3 performs poorly. From-scratch training on the single-session benchmark datasets underperform NDT1 in all tasks. Tuning from pretrained models improved performance, sample efficiency, and robustness to hyperparameters (only performance is reported here), but did not dramatically change model competitiveness. Zhang et al. [20] reported the value of more diverse neuron-level objectives for neural activity prediction, though their submission also dramatically underperforms NLB SoTA, warranting critical examination of whether pretraining benefits low-level modeling of neural activity.

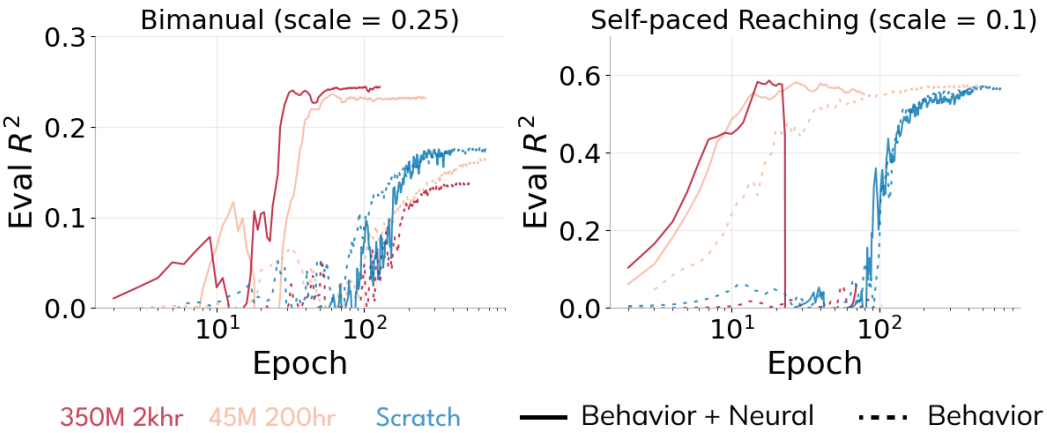

Figure 16. We compare fine-tuning on bimanual and self-paced reach tasks (RTT), for variants with the neural objective and behavior objective to those with only the behavior objective. Curves from runs with the best validation R^2 of a standard 3 seed x 3 LR fine-tuning sweep are shown.

A.2.11 Neural objectives are needed in fine-tuning

Our recommended fine-tuning for NDT3 is to match pretraining in most hyperparameters (except learning rate). This includes the use of the neural reconstruction objective, even though the downstream settings we used only evaluate behavior reconstruction. We show that including this neural objective is important in fine-tuning in Fig. 16. Specifically, for pretrained models, ablating the neural objective can cause a substantial drop in performance. Interestingly, the drop is more significant for the larger scale 350M 2khr NDT3. This dependence of NDT3 on the neural objective is not desirable, but consistent with findings in other pretrained models that fine-tuning performs best when consistent with pretraining settings [71]. For from-scratch models, the neural objective does not cause significant impact.

A.3 Methods

A.3.1 Metrics and Evaluation

Throughout this work we evaluate offline decoding of continuous covariates timeseries. The metric we specifically use is the coefficient of determination, R^2, as computed by scikit-learn's *r2_score* function. R^2 is a useful metric over MSE as 1 represents perfect prediction and 0 is the score achieved by best-guess baseline, the mean of the data. In pretraining, R^2 is computed over the flat average of all covariate dimensions, since each datapoint has differing covariate dimensionalities. In evaluation, R^2 is computed as a variance-weighted average of R^2s in each covariate dimension. Another difference between training and evaluation metrics is that training predictions are made over batched data, while evaluation predictions are mostly computed in a *streaming* fashion. Streaming requires continuous neural data across different behavioral epochs, and so cannot be performed for the Oculomotor and CST datasets. We also omit it for the motor cortex self-paced reach dataset, which has a very large evaluation split. Streaming allows timesteps at the beginning of each sequence to leverage neural context from the preceding sequence, which raises performance slightly, as shown in the continuous vs trialized analysis (Section 3.3). We limit history in streaming evaluations to the max history seen in tuning (1 second).

A.3.2 Training

Pretraining hyperparameters were manually tuned in preliminary experiments at the 45M parameter models on small datasets. 350M models diverged at the chosen $4e-4$ peak LR, so we lowered peak LR to $1e-4$. For tuning, the explored LRs are $1e-4, 3e-4, 5e-4$ for training from scratch and $3e-5, 1e-4, 4e-4$ for fine-tuning. While this is far from an exhaustive search, we show in Fig. 17 that other regularization hyperparameters are set to reasonable defaults such that this sweep finds near optimal results for both a from scratch model and fine-tuning the 45M model. Fine-tuning, like pretraining, is early stopped with a patience of 100 epochs. Batch size is uniformly

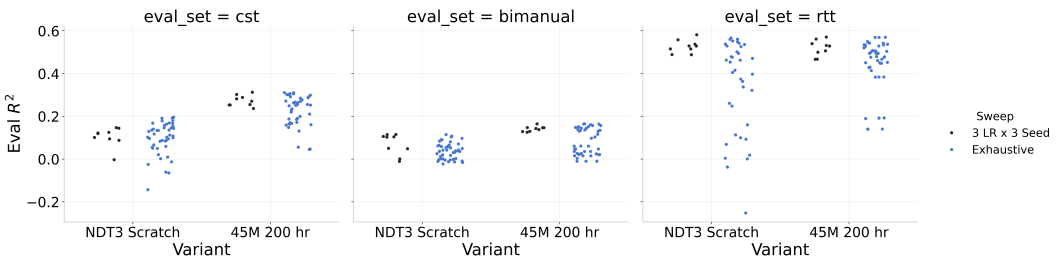

Figure 17. For 3 monkeys datasets at 10% scale, we extend a HP sweep to 5 LRs and dropout in $[0.0, 0.1, 0.3]$ (vs default 0.1). For fine-tuning, we also sweep weight decay in $[0.0001, 0.01, 0.1]$ (vs default 0.1), while for from-scratch models we also sweep Transformer width ($[256, 512, 1024]$) vs default 512. This yields a 45-model sweep on 1 seed. We compare the range of scores achieved by this larger sweep against the standard 3 LR x 3 seed sweep.

set to 16K in pretraining, and scaled to be roughly 10-20% of dataset size in fine-tuning. NDT3 from-scratch models were trained at the 11M parameter range. Exact model configurations for different experiments are documented in the codebase.

NDT3's simple architectural design allows us to train on batches from different tasks and dimensionalities. To avoid excess padding in training, we concatenate pretraining data that is otherwise discontinuous (trialized) into 2 second data. We do not add any separator tokens, as this does not appear to have a performance impact for language models [12]. With mixed-precision training, the 350M parameter NDT3 can fit the 4-8K tokens in each input context in the memory of 40G NVIDIA A100 GPUs. Thus we can restrict NDT3's pretraining parallelism to data-parallelism.

Using Kaplan et al. [72]'s equation for FLOP computation, $C_{\text{forward}} = 2N + 2n_{\text{layer}}n_{\text{ctx}}d_{\text{attn}}$, we compute the footprint of the 350M 2kh model. We use about 0.9B FLOPs per token in the forward pass, and about 0.9T neural tokens processed over training, which yields a pretraining footprint of about 2.4e21 FLOPs.

A.3.3 Baselines and Other Transformers

Many models have been proposed for neural data decoding. While we believe the selected Wiener Filter and NDT2 baselines are representative of standard approaches, it may seem insufficient relative to the number of available approaches. To this end, we want emphasize that this work's primary aim is to evaluate the scaling of a base Transformer on neural data, rather than establish uniform superiority of a model over all others. Such claims can only be established in a field with many benchmarks and external evaluations, which BCI decoding currently lacks.

Wiener Filter The Wiener Filter baseline was cross-validated over regularization strength. We also swept history of neural input up to the max length provided to NDT, and reported the R^2 of the best WF according to test data in primary evaluation (slightly advantaging the WF). Generalization plots in Section 3.3 report the performance of WF models at these different histories. For evaluating angular generalization, WFs were only swept up to 1s history due to memory limits; performance was not varying substantially with history so we do not expect this to have impacted conclusions. The WF was for simplicity directly fit on the concatenated trial data, which may have slightly negatively impacted its performance in trialized datasets (Oculomotor, CST, Generalization analyses).

In the primary evaluations in Section 3.1, we considered WFs fit either independently per session in a dataset or jointly on all sessions, which is helpful for sessions in very low data regimes. We report the better of the 2. In generalization analyses, for simplicity, we only report joint fits, which may cause a slight downward bias in performance.

NDT2. NDT2 is a Transformer that has shown effective transfer learning across BCI datasets. It currently leads the FALCON BCI decoding benchmark and outperforms RNN decoders. NDT2 baselines were prepared with its public codebase. We trained NDT2 models both from-scratch and from the public checkpoint pretrained on 100 hours of human data. Max context length and patience were held constant across the models. This restriction to a patience of 100 epochs accounts for some difference with the reported FALCON benchmark results in Karpowicz et al. [23], as we note in Table 2. The only departure we take from NDT2's default design is that we train NDT2 with both

Dataset	Patience	Held-In R^2	Held-Out R^2
H1	100	$0.567_{\pm 0.034}$	$0.453_{\pm 0.030}$
H1 (reproduction)	250	$0.628_{\pm 0.011}$	$0.517_{\pm 0.016}$
H1 ([23])	250	0.62	0.52
M2	100	$0.563_{\pm 0.015}$	$0.352_{\pm 0.028}$
M2 (reproduction)	250	$0.582_{\pm 0.002}$	$0.391_{\pm 0.009}$
M2 ([23])	250	0.63	0.43

Table 2. NDT2 H1 and M2 results when trained with 100 epochs of patience (this work) in fine-tuning vs 250 as in Karpowicz et al. [23]. We report mean and standard deviation of 3 model seeds on the FALCON evaluation (which is in turn a cross-session mean).

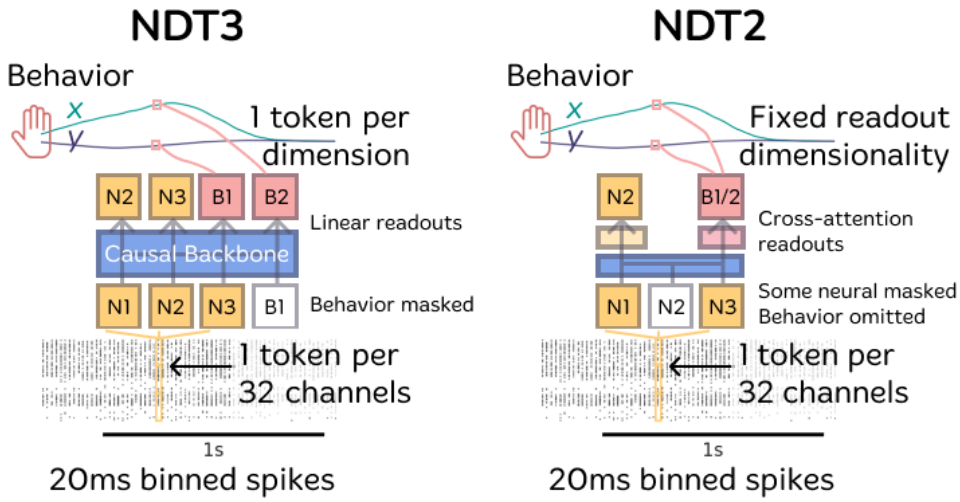

Figure 18. NDT2 and NDT3 architectures differ mainly in the conversion of NDT2's representation learning backbone to NDT3's single multimodal stream. NDT3 directly intakes neural and behavioral tokens and predicts with a next-step objective. NDT2 employs explicit masking of input neural tokens and extracts neural and behavioral predictions at each timestep with cross-attention layers.

a neural reconstruction loss (25% masking) and a supervised decoding loss. This joint objective was introduced late in the NDT2 paper ([8] Fig 5) and performs comparably to multi-stage training while simplifying large scale evaluation. This is true for all eight evaluation tasks except CST. In the CST task, we used only the supervised decoding loss, as the token dropout used in reconstruction can dropout all neural input.

For hyperparameter tuning, we matched NDT3's tuning budget for the pretrained NDT2 checkpoint by only exploring 3 learning rates. Given mediocre NDT2 from-scratch performance, we swept NDT2 over 2 model sizes in addition to the standard 3 learning rates. We set the NDT2 from-scratch model sizes to 20M and 72M to be comparable with NDT3 45M, but note that the NDT2 pretrained checkpoint is only 6M parameters.

NDT2 vs NDT3. NDT3 builds off of NDT2 but departs in several manners to enable more streamlined scaling and analysis over heterogeneous decoding tasks, which we overview in Fig. 18. Lower level technical changes are described in Section A.3.6. Both models relate neural data to behavior and train with both neural data and behavior prediction objectives. Both models use both these objectives in fine-tuning, but NDT2 only uses the neural objective in pretraining. NDT2's neural data objective is based on MAE [22], such that some fraction of input neural data tokens are masked and reconstructed in a decoder separate from the main backbone. However, this explicit masking was originally developed to study representation learning on images, not timeseries decoding. Causal domains like BCI control can also learn nontrivial representations simply through next-step prediction. NDT2 employed both explicit MAE masking and a causal attention mask in its backbone, which is redundant computationally and also reduces the context available to make predictions. NDT3 thus dispenses with the masking mechanism and uses next-step prediction alone. NDT2 also differs from NDT3 in its readout of behavior. Again, since NDT2 studied representation learning, it used additional

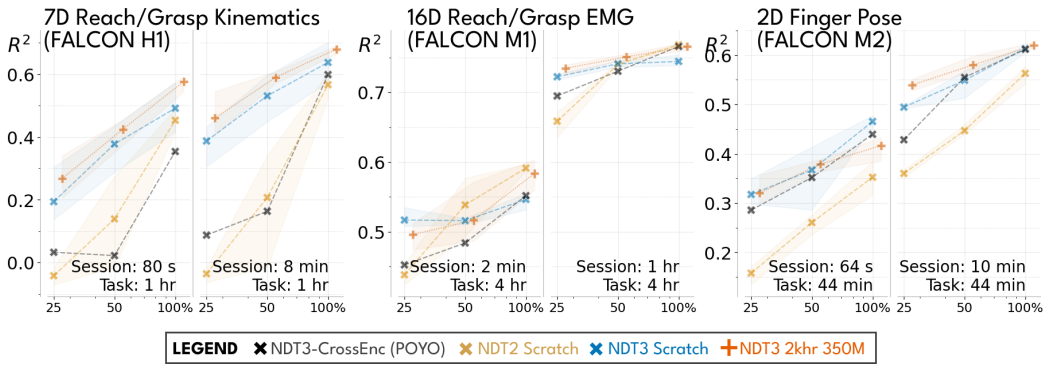

Figure 19. NDT3-CrossEnc (POYO reproduction) compared with NDT3 and NDT2 on FALCON movement datasets. We find that the cross-attention does not provide benefits over patch-wise tokenization in NDT3. For NDT3-CrossEnc, we report the best model from a 9x larger hyperparameter sweep (over model size and dropout) than the other models.

cross-attention readout layers to "probe" behavior predictions at each timestep. NDT3 simplifies this two-part encoder-decoder design to a decoder-only design. In this flow, masked behavior tokens are provided at the input and filled in through the backbone, and varying the input tokens allows us to make predictions of the corresponding behavior dimensionality.

NDT-MTM vs NDT Zhang et al. [20] proposed a Multi-task Masking method (MtM) as a means of enabling improved scaling for spiking activity. MtM increases the sophistication of the masks applied in the MAE strategy used to train NDT1 and NDT2, and finds improvements on varied neural data metrics. We view this as analogous to introducing more granular objectives to improve "pixel" level metrics, and believe the results are consistent with the original image MAE [22] not achieving particularly good pixel metrics. However, decoding metrics, the focus of this work, are similar to those achieved by the baseline NDT models.

Of note, Jiang et al. [73] also shows that NDT-MTM fails to scale across subjects in the mouse neural prediction setting in which NDT-MTM was developed, replicating our core claim on the difficulty of cross-subject transfer. Given these similarities and that NDT-MTM makes no claim on advancing decoding, we did not attempt comparisons.

POYO vs NDT POYO [7] is a Transformer that can pretrain on spiking activity, proposed at a similar time as NDT2. POYO uses a cross-attention encoder, which is a qualitatively large departures from the more basic Transformer formula used in NDT. Specifically, rather than directly operating on patched neural data, as in NDT, POYO first "tokenizes" individual spikes on all channels and performs cross-attention to reduce the large set of resulting tokens to a smaller number of latent tokens. In practice, this yields a pre-specified constant number of tokens per timestep, as opposed to NDT's choice of scaling the number of tokens depending on input data dimensionality.

The official POYO codebase was not supported as of writing this work, and neither were its pretrained weights available, so rather than attempt an exhaustive comparison, we sought a basic test of whether POYO's cross-attention encoder would enable improved performance and thus scaling in NDT3. A priori, we deem this unlikely as the Perceiver-based cross-attention encoder has not been widely adopted in other domains. Our re-implemented cross-attention encoder, holding all other elements constant within NDT3, performs worse than direct patching tokenization Fig. 19. For clarity, we enumerate the differences between NDT3/NDT3-CrossEnc and POYO:

- NDT3 uses joint neural / behavior prediction losses. POYO is trained only with behavior prediction losses.
- NDT3 bins neural spike counts, while POYO encodes individual spike times. While the higher fidelity is likely useful, most BCI pipelines do not typically use individual spike times, and not all data releases include these more granular features.
- Dataset-specific embeddings: Both NDT2 and POYO use learned dataset embeddings to account for dataset-specific variability. NDT3 discards these learned tokens for simplicity, and accepts the small performance hit this introduces.

- Unit-specific embeddings: POYO additionally learns unit-specific embeddings for each neuron or channel, which varies across datasets. NDT's analogous spatial embedding, learned for each patch (or unit, when patch size is 1), is fixed across datasets.

While our reproduction may understate POYO's overall performance, it does suggest that the main design choice that could potentially improve scaling does not outperform simple patching. POYO does outperform NDT2 in M2, suggesting common choices with NDT3 such as rotary positional embeddings, might account for its reported benefits.

A.3.4 Pretraining and Evaluation datasets

Pretraining datasets were comprised of historical data from several labs, the rough composition of which is shown in Fig. 2B. The evaluation behavior used during pretraining was reaching in 2 monkeys. The first monkey dataset came from a public release [74], and the second from a private dataset (Chase lab). The latter had center-out reach in standard conditions and under visual feedback perturbations. The monkey in the second dataset is also present in the 1khr monkey and 2kh and up model dataset sizes, though performing in a different set of experiments.

Inherent to the process of large-scale scraping is a loss of detail on what precise tasks were used, so we only have a qualitative description of tasks we believe are well represented. NDT3 trains on a wide variety of reaching behaviors from relatively constrained (2D center-out reaching to fixed number of targets) to relatively unconstrained (self-paced, more targets, potentially 3D) and under experimental manipulations (delayed onset, multiple targets, different error thresholds requiring more precision). These reaching behaviors are described in both endpoint kinematics and as EMG. A smaller fraction of pretraining data are isometric and force related (force exerted against manipulandums) for wrist and arm motion. Human datasets contain a variety of iBCI tasks, with closed loop datasets reflecting both high and low quality control. These tasks include reach and grasp behavior from 1-10 degrees of freedom, as well as some individuated finger tasks for clicking.

We detail evaluation datasets in Table 3. Three datasets come from the FALCON benchmark [23], two are based on public datasets ([28, 27]), and three are private. Note we avoid the Neural Latents Benchmark [5] as it does not directly measure decoding performance. For each evaluation dataset, we specify a tuning split and an evaluation split. Only tuning split data is changed when varying data scale. Tuning and evaluation splits are block-contiguous, i.e. trials are not interleaved, for better downstream applicability.

A.3.5 Generalization analyses and Further evaluations

Intra-session generalization Posture, spring, and angular generalization evaluate OOD performance in the standard setup of comparing in-distribution and out-of-distribution performance directly (with changes in the underlying evaluation dataset) The intra-session temporal shift analysis is evaluated in an inverted, slightly more rigorous setting. Specifically, we trained two sets of models on the two different temporal blocks, and evaluated on an evaluation split in the later block, rather than only training on the early block and evaluating on both blocks. This way, the OOD shift is measured with respect to the same evaluation dataset.

A.3.6 Architectural Details

NDT3 adopts several architectural innovations used in recent Transformer models. These were compared against baselines in preliminary experiments, but formal ablations in the final experimental setting were not conducted. We defer full description of the Transformer dimensions to the public codebase.

- FlashAttention 2 [85] is used to increase training and inference speeds. On the NERSC Perlmutter cluster, with FA2, 45M NDT3 trained at about 270M neural tokens per 40G A100 hour, 350M NDT3 trained at about 70M neural tokens per A100 hour. Note, FA2 also enables use of the 350M model for real-time (<20ms) inference latency. On an NVIDIA 4090, we see mean inference times of 4ms for the 45M parameter NDT3 and 9ms for the 350M parameter NDT3.
- Positional Embeddings [19]: Rotary embeddings are applied to indicate the real-world timestep of every input token. Additionally, 48 categorical learned embeddings are reserved

to distinguish token modality and position within a timestep (10 for neural, 16 for covariates, 16 for covariate constraints, 1 for reward/return, 1 for dummy tokens, remainder unused).

- QK Normalization [86, 87]: An additional layer norm is applied to the query and key embeddings, before the rotary embeddings, which helped stabilize training of the 350M parameter models.
- No context embeddings [8]: Differing from NDT2, no learned embeddings for disambiguating input datasets were prepended to each input. This was removed for simplicity. Per GATO [13] and language modeling practices, we instead leave task / dataset disambiguation to the modeling process: In pretraining, the covariate maskout strategy allows for many tasks to be specified in-context (as later behavior can be inferred on the basis of earlier neural-behavioral token relationships). In fine-tuning, the tuning dataset already uniquely specifies the function to be learned.
- Cross entropy loss for spiking data prediction: We used the standard cross entropy loss to classify spike count over the Poisson loss common in many neural data architectures. Since the overall ablation of neural objective shows no large impact in this work, it is likely that this decision should be evaluated with neural data related tasks rather than decoding.

We document the Transformer model shapes considered in our work in Table 4. This shape is not systematically explored in our work, and is by historical artifact, slightly different than the shapes used in NLP/CV. Embedding parameters are negligible. One possible area of interest is that the feedforward expansion factor is 1 in our model, i.e. the MLP dimension is low. If MLPs do serve as memory stores in Transformers [88], increasing this shape may yield more performant model size scaling, given the heterogeneity of our datasets.

Table 3. Evaluation datasets used for multiscale decoding and generalization analyses. The references provide extended description of the behavioral task. Dashed line separates datasets for Section 3.1 and for analysis. Datasets use unsorted multi-unit activity and are processed in 1s chops unless otherwise mentioned.

Dataset	Description
FALCON H1, M1, M2 [23]	3 separate single-subject multi-session datasets for different iBCI tasks. Data comes in a high data split (held-in), and a low-data split (held-out), with the intention on identifying methods that can achieve parity in the two settings. H1 is an open loop human dataset for calibrating 7D reach-and-grasp in a robot arm. M1 is a monkey reach-and-grasp task to different objects with EMG recordings. M2 is a monkey 2D finger movement task with manipulandum-measured kinematics. Scaling scores are reported on the test set.
Self-paced reach (RTT) [28]	Monkeys reach for random targets one at a time in a small planar workspace. We decode 2D arm velocity in monkey Indy. Has neural data from M1 and S1, we use M1 in Section 3.1 and Section 3.2 and S1 in Section 3.3.
Bimanual Cursor Control [27]	A human open loop dataset where the participant attempts movement of one or both hands to control two cursors.
2D Cursor + Click (private)	Cursor control is a classic iBCI endpoint [75–77]. Two human participants attempt movement according to visually cued cursor movement and audiovisual click cues. We also use this dataset for trial structure analysis in Section 3.3.
Grasp force (private)	A open-loop dataset with two human participants attempting isometric power grasps. Specifically, participants were asked to match force output according to visual cues in a Mujoco environment. Grasps cued were both static (instant onset, hold, and offset) or dynamic (gradually increasing force). This dataset is valuable for human iBCI study because force modulation is required in many motor behaviors, and grasp force has primarily only been characterized in monkeys until now [78]. Uses 2 second intervals due to long behavior timescale. We expect this dataset can be released by end of 2024.
Critical Stability Task [79] (private, trialized, sorted)	A monkey dataset collected to study continuous control relative to ballistic movement. The monkey balances a virtual cursor on a 1D workspace for up to 6 seconds.
Posture-varied Center-Out [80] (private, trialized, sorted)	A monkey center-out task, but the monkey's hand is adjusted to one of 6 different starting positions. We use the central position as center and the rest as edge.
Spring-load [81]	A monkey moves fingers, clamped together in a manipulandum for effective 1DoF, is neutral or under spring load.
Center-out, Monkey J [82] (trialized)	Used in Section 3.2. A monkey performs an isometric center out task. Forces are measured by the manipulandum and converted to cursor velocity signals.
Center-out, Monkey V (private, trialized)	Used in Section 3.2. A monkey reaches to one of 8 radially arranged targets by moving a manipulandum (Kinarm).
Oculomotor pursuit [83] (private, trialized, sorted)	A monkey visually tracks (via smooth pursuit) a target that moves from center of workspace to one of four directions. A few dozen neurons are recorded on probes in each of frontal eye field (FEF) and area MT. We decode pupil velocity. The small number of neurons in this dataset required resetting NDT3 neural readin/readout layers.
FALCON H2	Human open loop dataset where a participant attempts movement to write letters cued on a screen [69, 84]. The large number of timesteps in this dataset required resetting NDT3 neural readin/readout layers (to use fewer neural tokens).

Model	Layers	Width	MLP Size	Heads	Parameters (M)
NDT2 PT [8]	4	256	256	4	6
NDT3 Base	6	1024	1024	8	45
NDT3 Big	12	2048	2048	16	350

Table 4. Transformer Model Shapes used in this work.