PARAMETER-EFFICIENT FINE-TUNING OF LARGE MULTIMODAL MODELS FOR BRAIN ENCODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Brain encoding models aim to predict neural responses to stimuli. The most widely used approach fits readouts from frozen pretrained backbones. Using neural data to train large models from scratch or fully fine-tune them after task-pretraining has proved difficult because neural data are costly and limited. We show that parameter-efficient fine-tuning (PEFT) enables us to effectively adapt very large pretrained backbones for brain encoding with a small number of trainable parameters. To make it computationally practical, we exploit the backbone's causal structure to predict multiple timepoints in a single pass instead of using sliding windows, significantly reducing feature-extraction cost. This enables efficient fine-tuning of multimodal backbones with up to 104B parameters, the largest model adapted for brain encoding to our knowledge. Trained with a low-rank adapter (LoRA) and a lightweight readout, our approach consistently improves brain encoding performance across nine diverse multimodal backbones. On the same QWEN3-OMNI backbone as the prior state-of-the-art (SoTA), our method BRAINLORA achieves the best published performance on the Algonauts 2025 Challenge in- and out-of-distribution, using only about 5% as many trainable parameters. We also show the adapted multimodal backbones remain interpretable: modality attribution recovers the canonical sensory topography that prior work obtains by construction from separate unimodal encoders. To scale out the number of adapted models, we develop a multi-LoRA kernel that accelerates joint training of multiple LoRAs on a shared backbone. This enables an ensemble of 200+ adapted models that ranks first on the public Algonauts post-challenge leaderboard at the time of writing, also surpassing unpublished submissions. Together, our results highlight PEFT as an efficient and powerful paradigm for brain encoding with large pretrained models.

1 INTRODUCTION

Human brain encoding models predict neural responses to stimuli, providing computational insights into how the brain processes information. State-of-the-art (SoTA) encoding models mostly follow a goal-driven paradigm (Yamins & DiCarlo, 2016): a large pretrained model is used as a *frozen* feature extractor, and only a readout is learned from neural data (Adeli et al., 2025; Hwang et al., 2025; Gokce et al., 2026a; d'Ascoli et al., 2026). This approach leverages strong pretrained priors, but misses the opportunity to exploit the backbone's expressiveness to specialize for brain encoding.

Adapting large pretrained backbones with neural data promises to scale models to the complexity required to capture human brain computation, and to best combine the information from neural and task training. In practice, however, such adaptation is difficult, because neural data are costly and scarce relative to the size of the models. Approaches that do let neural data shape the representation, such as training smaller models from scratch (St-Yves et al., 2023; Khosla & Wehbe, 2026) or fully fine-tuning models after task-pretraining, have thus yet to yield SoTA encoding performance on human brain modeling.

We argue that **parameter-efficient fine-tuning (PEFT) is a powerful yet underexplored regime** for building strong brain encoding models. It retains the strong prior and expressive capacity of a large pretrained model *and* allows limited neural data to adapt its representations through a small adaptation (§ 3). While the framework applies to any pretrained backbone, stimulus modality, or neural recording, we evaluate it in a particularly demanding setting: the CNeuroMod/Algonauts

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

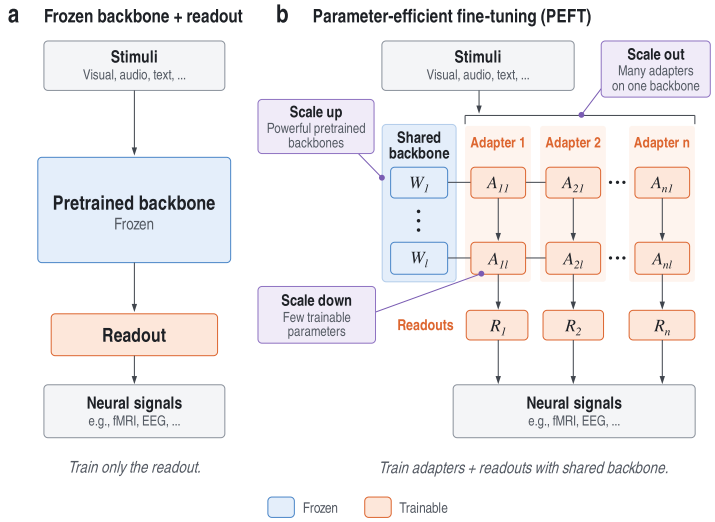

Figure 1: **From frozen backbones to PEFT, and the scaling directions it opens up.** *(a)* The dominant paradigm: a frozen pretrained backbone with readouts trained for brain encoding. The backbone prior is strong, but it is not specialized for brain encoding. *(b)* We advocate using PEFT to adapt strong pretrained backbones, retaining the powerful prior while allowing flexible neural adaptation. This suggests three complementary scaling directions: *scale up* to more powerful backbones, *scale down* the size of trainable adapters, and *scale out* to many adapters with different specializations. We treat these as a framing for future work rather than scaling laws we establish here.

2025 Challenge (Gifford et al., 2025), which requires predicting whole-brain fMRI responses to long, naturalistic audiovisual movies and tests generalization to held-out out-of-distribution stimuli. We show that PEFT is powerful at scale and sets a new SoTA.

We first make adaptation for brain encoding at scale computationally practical. Prior approaches typically rely on overlapping sliding windows for feature extraction, resulting in substantial redundant computation of shared stimulus context. By exploiting the backbone's innate causal masking, we encode each training window in a single forward pass and obtain representations for all timesteps simultaneously. This reduces feature-extraction cost by approximately $350\times$ relative to the sliding-window protocol used by the prior SoTA at little performance cost (Gokce et al., 2026a; §4).

With adaptation made practical, we show that low-rank adapter (LoRA; Hu et al., 2021) consistently improves brain encoding on *all* nine multimodal backbones we test, spanning 4.7B–104B parameters, the largest models ever adapted to brain encoding to our knowledge. On the same backbone QWEN-3-OMNI as the prior SoTA (Gokce et al., 2026a), our main method, BRAINLORA, pairs LoRA with a simple readout. With only about 5% as many trainable parameters, BRAINLORA sets a new SoTA among published methods both in- and out-of-distribution (§§5.1 and 5.2).

Moreover, we show the adapted multimodal model remains interpretable. Prior work often identifies a region's preferred modality by determining which unimodal encoder its readout relies on. With multimodal backbones, we instead use feature attribution to identify modality contribution, which recovers the canonical topography: early visual cortex is driven by video, and auditory and lateral temporal cortex by audio. Comparison with the frozen backbone shows that adaptation reweights the three modalities' contributions and increases cross-modal mixing within the backbone (§5.4).

Finally, to scale the number of adapted models, we develop a batched multi-LoRA kernel that trains many LoRAs jointly on one shared backbone, significantly reducing training time. This enables 200+ independently adapted models, whose ensemble ranks first on the public Algonauts post-challenge leaderboard, surpassing unpublished entries as well. We also show that the ensemble performance is jointly governed by ensemble size, individual model quality, and model diversity (§ 5.5).

Our contributions are as following:

- **Efficiency.** We make PEFT on large multimodal backbones computationally efficient for brain encoding (§4), reducing feature-extraction cost by approximately $350\times$ through single-pass

extraction compared to prior SoTA (Gokce et al., 2026a) and accelerating multi-adapter training significantly with a batched multi-LoRA training kernel.

- **Effectiveness.** LoRA consistently improves performance over frozen backbones on all nine multimodal backbones we test (§5.1). LoRA with a lightweight readout beats the previous published SoTA on both ID and OOD test sets using $\approx 5\%$ as many trainable parameters (§5.2).
- **Interpretability.** An adapted multimodal backbone remains interpretable: modality attribution recovers the canonical sensory topography that prior work obtains by construction from separate unimodal encoders. Comparing with the frozen backbone reveals that adaptation increases the contribution of audio inputs and the degree of cross-modal mixing in the backbone (§5.4).
- **Scalability.** We scale up backbone sizes to up to 104B, the largest backbone adapted for brain encoding to our knowledge, and scale out to 200+ independently adapted models (§5.5). The ensemble ranks first on the public Algonauts post-challenge leaderboard at the time of writing.

Our results establish PEFT as an efficient and powerful framework to leverage large pretrained models for brain encoding. While the adapted models do not yet constitute a brain foundation model, they suggest a direction towards one for future work: scaling *up* the capability of the pretrained backbone, scaling *down* the size of each adaptation, and scaling *out* the number of adaptations (§ 6).

2 Related work

We organize by where the neural signal enters the encoding model pipeline (extension in App. A).

Frozen backbone, trained readout. Most brain encoding methods use pretrained task-optimized backbones as *frozen* feature extractors and train readouts to predict neural responses (Naselaris et al., 2011; Huth et al., 2016; Yamins & DiCarlo, 2016; Schrimpf et al., 2021). Every leading entry in the Algonauts 2025 challenge (Gifford et al., 2025) followed this recipe (Gokce et al., 2026a; d'Ascoli et al., 2026; Carlström Schad et al., 2025). Compared to more classic voxel-wise ridge regression methods (Naselaris et al., 2011), recent top-performing models have been using more expressive and flexible readout (d'Ascoli et al., 2026; Adeli et al., 2025; Yu et al., 2025; Yang et al., 2023), yet the training is still confined to only the readout module.

Backbones fine-tuned by brain data. Relatively far fewer studies have instead directly adapted pretrained models using neural data. Brain-tuning of language and speech models improves not only alignment but downstream semantic tasks (Schwartz et al., 2019; Moussa et al., 2025; Moussa & Toneva, 2025), and the vision analogue gains robustness and behavioral alignment (Li et al., 2019; Federer et al., 2020). PEFT is already used within this family, e.g. BrainWavLM matches a full fine-tune with 166K LoRA parameters (Vattikonda et al., 2025). Our work follows this direction but extends it to substantially larger model scale: to our knowledge, prior models above 15B parameters have been used only as frozen feature extractors, whereas we directly adapt backbones up to 104B.

Other ways to inject neural data. Other approaches incorporate neural signals through auxiliary alignment objectives (Dapello et al., 2023; Gokce et al., 2026b), yet shuffled-target controls sometimes reproduce much of the reported benefit (Guo et al., 2025); through self-supervised pretraining on neural data alone (Ortega Caro et al., 2024; Lane et al., 2026), which captures intrinsic neural dynamics rather than stimulus-referred computation (Zahorodnii et al., 2025); through unified stimulus–brain models (Wu et al., 2026), or train the model on neural data from scratch (Lurz et al., 2021; Klindt et al., 2017; St-Yves et al., 2023; Khosla & Wehbe, 2026), none of which, however, are yet outperforming brain encoders built on large task-optimized feature backbones on public benchmarks (Schrimpf et al., 2018; Gifford et al., 2025).

3 Preliminaries

Let X denote a stimulus (e.g. video, audio, text, or a combination thereof) and Y the neural response it evokes (e.g. fMRI, EEG/MEG, or spiking activity). An encoding model consists of a pretrained backbone f_θ followed by a readout g_ϕ, and is trained by

$$\hat{Y} = g_\phi\big(f_\theta(X)\big), \qquad \min_{\theta \in \Theta,\, \phi} \mathcal{L}\big(Y, \hat{Y}\big), \tag{1}$$

where $\mathcal{L}$ is a loss appropriate for the task (e.g. squared error or negative correlation). The readout g_ϕ also performs the temporal alignment between the stimulus and the neural sampling rate.

Given pretrained weights $\theta_0 \in \mathbb{R}^d$, existing approaches differ in the set Θ that is optimized over:

$$\Theta_{\text{frozen}} = \{\theta_0\} \subset \Theta_{\text{PEFT}} = \{\theta_0 + \Delta(\psi) : \psi \in \mathbb{R}^k\} \subset \Theta_{\text{full}} = \mathbb{R}^d, \qquad k \ll d, \;\; \Delta(0) = 0. \quad (2)$$

With a frozen backbone, only the readout is learned from neural data. Full fine-tuning and training from scratch share Θ_{full} but differ in initialization (θ_0 vs. random), with the latter forgoing the task-pretrained prior entirely. Parameter-efficient fine-tuning (PEFT) retains θ_0 and lets neural data modify the representation only through a low-dimensional parameterization $\Delta(\psi)$. This formulation is architecture-agnostic: LoRA (Hu et al., 2021), adapter layers (Houlsby et al., 2019), and prompt tuning (Lester et al., 2021) are all instances of Δ.

We argue that PEFT is a powerful yet underexplored regime for building strong brain encoding models: it inherits the strong prior of pretrained models, which training from scratch on limited neural data cannot afford, while still letting neural data leverage the flexibility of the backbone to shape the representation, which freezing the backbone precludes.

4 METHOD

Task and data. We evaluate on CNeuroMod/Algonauts 2025 (Gifford et al., 2025) challenge, which contains whole-brain fMRI recordings, parcellated into 1,000 regions, from subjects viewing seven seasons of *Friends*, with synchronized video, audio, and transcript. We train on four subjects, validate on Season 6, and evaluate on held-out Season 7 (ID) and movies (OOD). Our main experiments use QWEN3-OMNI (Xu et al., 2025), the same as the best published method (Gokce et al., 2026a), for a controlled comparison. We also conduct experiments on a diverse set of nine backbones (Tab. 10, 12).

Single-pass extraction. The prior SoTA (Gokce et al., 2026a) predicts each fMRI timestep from an overlapping sliding window, requiring about 2,800 forward passes per clip (Fig. 2a). However, since the backbone is already causally masked, the representation at stimulus position t depends only on positions $\leq t$, making much of the repeated computation of a sliding window unnecessary. We therefore encode each 149 s window (100 TRs) with a single forward pass and extract representations for all timesteps simultaneously (Fig. 2b).

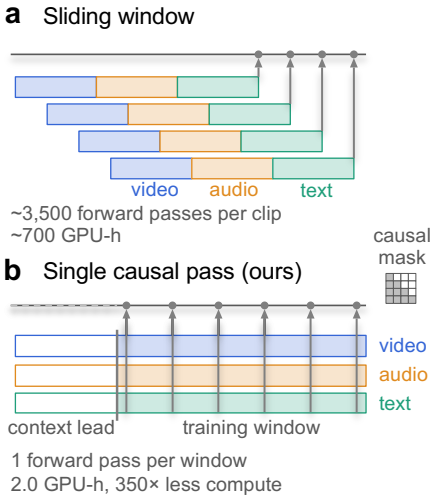

Figure 2: **One causally masked pass replaces a sliding window.** The baseline re-encodes an overlapping window per fMRI timestep; a single pass over the window gives features at every timestep.

This single-pass design reduces feature-extraction cost by $\approx 350\times$ relative to the sliding-window protocol of Gokce et al. (2026a) (from about 700 to 2 GPU-hours on A800), making fine-tuning through the backbone computationally affordable. Under a frozen readout, single-pass features incur only a small performance drop, which we attribute mainly to missing stimulus context at the start of each window (Tab. 3). We thus incorporate a 60s *context lead* that largely recovers it (Fig. 6). With LoRA fine-tuning, our configuration surpasses the original sliding-window baseline (App. B).

The modality encoders preceding the decoder are locally bidirectional, over 8s audio blocks and 1s visual patches. To assess this limited non-causality, we re-encode strict prefixes of each input and compare representations at shared timesteps (App. F). The small differences mainly result from length-dependent bf16 MoE numerics, not leakage of future stimuli.

PEFT for Brain Encoding. For our main configuration on QWEN3-OMNI, we insert rank-16 LoRA adapters (Hu et al., 2021) into the attention modules through layer 32 and truncate the decoder at this layer, introducing 8.91M trainable parameters. The adapters and readout are jointly optimized for one epoch. For hyperparameter tuning, see App. E. We also show that other PEFT methods like IA^3 (Liu et al., 2022) and adapter layers (Houlsby et al., 2019) show similar effects (App. H).

Readout. We use linear or cross-attention (Adeli et al., 2025) readouts. Backbone features are pooled into one vector per modality at each 2 Hz timestep and aggregated into eight log-spaced temporal-lag blocks aligned with the fMRI sampling rate. A final linear projection maps to 1,000 brain parcels,

with 52.3M trainable parameters. We pool the original $\sim$19.3k tokens to 894 vectors per input window. As an alternative, we replace the fixed temporal-lag aggregation with per-TR cross-attention over an expanded set of $\sim$3.2k tokens, adding only 0.2M parameters. Ablations in §5.3 attribute the improvement of cross-attention over linear readout primarily to the access to the richer token set.

Closed-form re-fit. After joint optimization with SGD, we re-fit the linear readout W in closed form. Forward passes accumulate subject-specific Gram matrices, after which W is solved using parcel-wise ridge regression following Nunez-Elizalde et al. (2019). As shown in §5.3, the improvement primarily arises from the closed-form re-estimation rather than the parcel-specific regularization.

5 Experiments

We use the mean Pearson correlation across the four subjects as the primary fMRI prediction metric. The seed variability is small: methods with a difference smaller than 0.00077 cannot be separated (LoRA+linear, 15 seeds; App. D). We therefore support each method comparison using seed variations whenever multiple runs are available, or for two single models, a paired bootstrap over parcels, which tests whether the performance difference holds across the cortex.

5.1 Adapters improve performance across backbones

We find that PEFT significantly improves brain-encoding performance across all nine multimodal backbones tested, spanning 4.7–104B parameters, compared to the frozen backbone baseline (Fig. 3). LoRA improves every backbone over its corresponding frozen backbone by 0.032 to 0.066 on linear readouts, while cross-attention yields an additional 0.003 gain on average and improves on all nine models. In general, a stronger readout extracts more from frozen features and leaves less room for adaptation improvements: the gain is +0.019 between before and after adaptation with cross-attention on Qwen3-Omni (Tab. 1). Adaptation still matters more than readout capacity: LoRA+linear readout outperforms every frozen baseline, including MIRAGE's 1.1B-parameter readout (Tab. 1). The adaptation improves prediction performance for every parcel, with the largest gains observed in parietal and visual areas (App. O). We additionally show that other PEFT methods (IA^3 and adapter layers) also improve over the frozen backbone significantly (App. H).

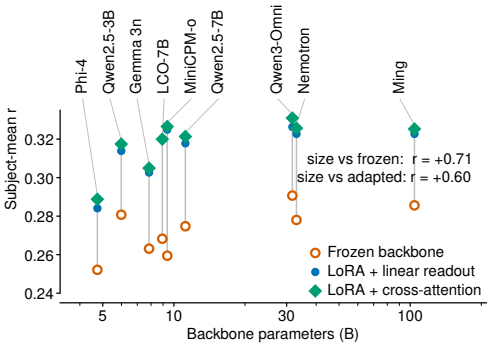

Figure 3: **Adaptation improves performance on all tested backbones.** Nine multimodal backbones share the same adapter training configuration except for the tapped layer (App. I).

The relationship between encoding performance and backbone size is far from linear (r=0.71 frozen; r=0.60 adapted), as also observed in prior work on neural predictivity in large models (Gokce et al., 2026b). We thus do not treat them as evidence for a simple scaling law for brain encoding. We use Qwen3-Omni for the remainder of the paper, as it achieves the best adapted performance and is also the backbone of the prior SoTA (Gokce et al., 2026a), enabling a controlled comparison (App. I).

5.2 A small adapter beats a large readout

As shown in Tab. 1, the previous single-model SoTA, MIRAGE (Gokce et al., 2026a), uses a frozen backbone plus a 1.1B-parameter transformer-based readout. In contrast, our LoRA+linear model trains only 61.2M parameters ($\approx$5% of MIRAGE) and already beats it on both ID and OOD test sets (Fig. 7). Replacing the linear readout with cross-attention, our best *single model* method, BrainLoRA, further improves performance to 0.327 on ID and 0.228 on OOD, exceeding the published 15-model MIRAGE ensemble on ID and matching it on OOD, setting a new SoTA on the Algonauts challenge. Ensembling a population of 200+ models further reaches 0.343 and 0.248 on ID and OOD, respectively (§ 5.5). Selecting a subset of 15 models still maintains strong performance.

Full fine-tuning of all 20.25B parameters below the tap layer reaches 0.323 (linear) and 0.330 (cross-attention) on validation, on par with or slightly below the corresponding LoRA configurations, with

Table 1: **Main results on CNeuroMod/Algonauts 2025.** Validation is on Friends season 6; ID and OOD are the challenge server's scores on held-out Friends season 7 and OOD movies. Two rows within 0.00077 are not separated by seed variability (App. D). Rows marked † are cited authors' own numbers quoted from the papers. Our method BRAINLORA achieves a new state of the art.

Model	Trainable params	Pearson r val	ID	OOD
Single model				
TRIBE† (d'Ascoli et al., 2026)	–	–	0.303	0.196
MIRAGE† (Gokce et al., 2026a)	1.1B	0.319	0.310	0.217
Linear + frozen backbone	52.3M	0.291	0.287	0.187
Linear + full finetuning	20.3B	0.323	0.320	0.217
Linear + LoRA	61.2M	0.326	0.322	0.222
Cross-attn + frozen backbone	52.5M	0.312	0.307	0.212
Cross-attn + full finetuning	20.3B	0.330	0.326	0.225
Cross-attn + LoRA	61.4M	**0.331**	**0.327**	**0.228**
Ensemble				
TRIBE†, 1000 models (d'Ascoli et al., 2026)	–	–	0.320	0.215
Vibe† (Carlström Schad et al., 2025)	20×329M	–	0.323	0.212
MIRAGE†, 15 models (Gokce et al., 2026a)	15×1.1B	0.335	0.323	0.227
Ours, best 15 models	15×61M	0.347	0.341	0.247
Ours, 261 models	261×61M	**0.348**	**0.343**	**0.248**

330× more trainable parameters. The held-out test sets show the same patterns, with a larger decrease on OOD (Tab. 1). Full fine-tuning thus does not outperform LoRA at this data scale.

The adaptation transfers to unseen subjects and a new dataset. The learned adaptation generalizes beyond both the subjects and the dataset used for training, with gains essentially matching those observed in-distribution. When trained on three subjects and evaluated on a held-out fourth subject with the LoRAs frozen and only the readout re-fitted, it retains 97% of the improvement observed for subjects included in LoRA training (App. M). The same adapter and readout also transfer *zero-shot* to the Naturalistic Neuroimaging Database (NNDb; Aliko et al., 2020), collected at a different site with a different scanner and repetition time, and with subjects viewing films. The adapter is applied without any further training on NNDb, including the readout. Even so, it yields a gain of +11.1 percentage points of the noise ceiling, on par with the +10.7-point fine-tuning gain on our validation set, and improves performance for all 20 participants (App. N). The adapter thus captures stimulus representations that generalize to other subjects or datasets.

5.3 ABLATING THE MODEL COMPONENTS AND THE INPUT DESIGN

Model components. Tab. 2 quantifies the contribution of each component in our main configuration. The LoRA adapter provides the largest improvement over the frozen-backbone baseline, increasing performance by +0.0356, none of which survives when the adapter is trained on fMRI shifted out of alignment with the stimulus (App. L). The closed-form re-fit contributes +0.0061 of the adapted model's score, an effect larger than those obtained from any of the other training-side modifications we evaluated. This improvement pattern is consistently observed across all tested configurations.

Table 2: **Ablation.** LoRA r16+linear readout, 60s context lead, closed-form re-fit. The first block builds our main configuration; the rest vary one choice from the main configuration. Rows within the 0.00077 seed spread are not separable.

Configuration	s06
Frozen backbone	0.2907
+ LoRA r16 **(ours)**	0.3263
without closed-form re-fit	0.3202
Rank 32	0.3268
Gated-MLP readout	0.3269
Cross-attention readout	0.3310
No audio	0.3026
QWEN3.8-27B backbone	0.3191
GEMMA-3-27B backbone	0.3029

Input design. Tab. 3 in App. B evaluates our input-design choices incrementally, under the MIRAGE or linear heads, frozen or adapted. Replacing the sliding windows with a single causally masked pass over the full clip reduces performance by 0.0038 under the frozen head, at two orders of magnitude less compute. Restricting the causal pass to the 149 s training window causes a further drop of 0.0046, concentrated at

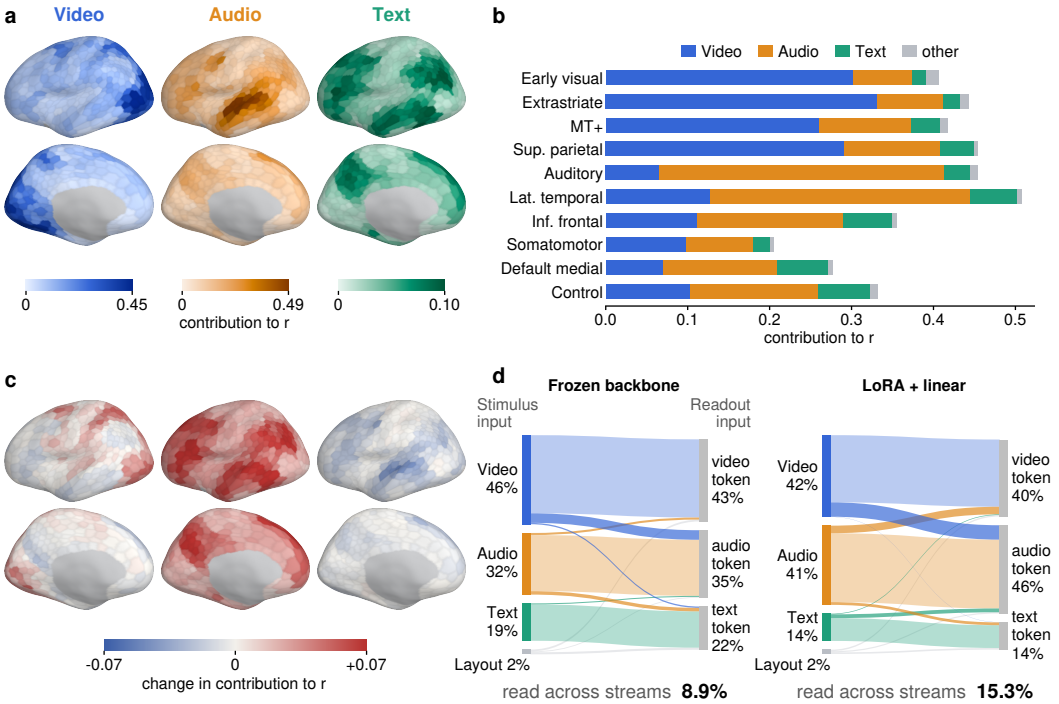

Figure 4: **Modality attribution recovers the canonical topography, and adaptation reweights and mixes the modalities.** *(a)* Contribution to encoding accuracy of the video, audio and text content, on our LoRA+linear model, left hemisphere (lateral above, medial below). *(b)* Modality contribution by region. The Shapley terms from the three modalities and a small layout residual sum to the region's r. *(c)* Change in modality contribution from the frozen to the adapted model. Audio contribution increases the most, while text falls. *(d)* Share of the prediction carried from each stimulus modality (left) to the backbone's output at each modality's tokens (right), which is the readout's input. Crossing ribbons are cross-modal mixing within the backbone, which nearly doubles after adaptation (App. P).

the start of each window (Fig. 6). Adding a context lead largely recovers this loss, with the benefit saturating at 60 s. We thus adopt 60 s for the final configuration (App. B).

The input design choices affect the frozen and adapted cases differently. With frozen readout heads, performance decreases as input context is removed, since the readout is trained for a different representation. However, once the backbone and readout are adapted to the new input configuration, we can recover the lost performance. Thus, the computational efficiency of single-pass encoding does not come at a cost to performance once the backbone is allowed to adapt.

App. K summarizes null results observed during exploration. Applying cross-attention to the same pooled feature set yields no measurable improvement, indicating that the gains from the cross-attention readout arise primarily from access to a larger token set with the same number of trainable parameters. Varying the training subsets likewise does not produce useful ensemble diversity. Moreover, several apparent improvements, including increasing the LoRA rank and replacing the linear readout with a gated MLP, disappear after the closed-form re-fit (App. E).

5.4 Interpreting the multimodal backbone

Encoding models can be used to ask *which* stimulus information is processed, and therefore drives the prediction of neural activity, in a given brain region. Prior work often answers this by construction, using separate unimodal backbones so that a region's preferred modality or semantic selectivity can be read off directly from the readout (Adeli et al., 2025; d'Ascoli et al., 2026). A single multimodal backbone lacks this explicit separation, with information from different modalities merging early in the backbone. However, we can still estimate modality contribution using feature attribution methods and examine whether the adapted model recovers the expected cortical topography.

Design. For each window, we compute exact Shapley values over the three input modalities, $\{\text{video}, \text{audio}, \text{text}\}$. A modality is ablated by replacing its input embeddings with a content-free mean embedding while preserving token count, position, and temporal alignment. Therefore, only stimulus *information* is removed. Evaluating all eight (2^3) modality subsets yields an exact additive decomposition of each modality's contribution to encoding performance. The same decomposition can also tell us through which token positions each modality's information reaches the readout. At every timestep, the readout takes three hidden states, one for each modality, at the truncated layer. After many layers of attention, however, each of these states can carry information from all three input modalities, so audio information, for example, may reach the readout through a video token. To separate these routes, we apply the Shapley decomposition to each of the three hidden states separately. This yields a 3×3 attribution matrix whose entry (i, j) is the contribution of input modality j that is read out at modality i's token position by the readout.

The attribution recovers the canonical topography. The resulting attribution maps recover the canonical cortical organization (Fig. 4a,b, App. P): Early visual cortex is predominantly driven by video (72.5%), auditory cortex by audio (72.9%), and lateral temporal cortex by audio (60.3%) rather than text (12.5%). Text contributes most strongly to predicting the activity in the default-mode (21.6%) and control regions. These patterns are highly robust: 1) all regional contrasts have bootstrap intervals excluding zero, 2) the attribution maps are consistent under a donor-window baseline (r=0.994) , and 3) two independently trained seeds yield highly consistent contribution maps (r=0.996–1.000).

Adaptation shifts modality attribution toward audio. To see how adaptation changes modality attribution, we analyze the frozen baseline as well. The spatial maps are largely unchanged: each modality's contribution map correlates at 0.963–0.990 between the frozen and adapted models. What changes is the relative weighting of the three modalities (Fig. 4c). Averaged over parcels, the audio contribution rises by 39% and video by 5%, while text falls by 17%. The dominant modality changes in 249 parcels, all toward audio, with 246 switching from video- to audio-dominated and none the other way. Consistent with this, removing audio hurts the performance more after adaptation (App. P). This shift is robust across seeds, which agree on the dominant modality for 97.2% of parcels.

Adaptation increases cross-modal mixing. Feature attribution also reveals the cross-modal contributions within the multimodal backbone: 15.3% of the prediction is attributed to information read out at another modality's token positions, with the largest cross-modal contribution coming from video information at audio-token positions (7.5%). Yet the frozen backbone reads only 8.9% across modalities. Adaptation thus greatly increases the cross-modality mixing in the backbone (App. P).

5.5 Training a population of independently adapted models

Efficient multi-adapter training. To efficiently train many adapters, we develop a specialized multi-LoRA training kernel. Since LoRA modules are small and structurally identical, the routed updates for N adapters can be fused into a single batched gather–matrix multiplication, enabling parallel training of N adapters together (Fig. 5a). The speedup arises from improved hardware utilization rather than reduced FLOPs: at N=32, the fused step costs only 1.08x a single-adapter step, yielding a 29.8x speedup over sequential training. End-to-end speedups are smaller because of longer sequences and larger batches. See App. C for details.

Large-scale ensemble performance. We train 261 independently adapted models spanning different hyperparameters, readout architectures, and backbones, and aggregate with a parcel-wise minimum-variance stack fitted on out-of-fold predictions (App. G). Ensemble accuracy reaches 0.3483 on validation, 0.3428 on ID, and 0.2483 on OOD, achieving the top score on the public post-challenge leaderboard at the time of writing, outperforming unpublished submissions as well (Tab. 1). All reported models were selected using validation performance only. A carefully selected 15-model subset reaches 0.3465 on validation, within 0.0018 of the full ensemble, at one seventeenth of the inference cost. Both the subset and its weights are chosen on the validation set (App. D).

To characterize the effect of ensemble size, we score random subsets of the 261-model pool at every size n and fit their mean accuracy with $m(n) = m_\infty - An^{-b}$, which results in $b \approx 0.94$ (R^2=0.998; Fig. 5b). An exponent near 1 is consistent with ensemble theory: if members' residuals have mean pairwise correlation ρ, the ensemble's residual variance is $\sigma^2[\rho + (1-\rho)/n]$ (Krogh & Vedelsby, 1994; Wood et al., 2023), whose second term shrinks as $1/n$. The first term does not shrink with size,

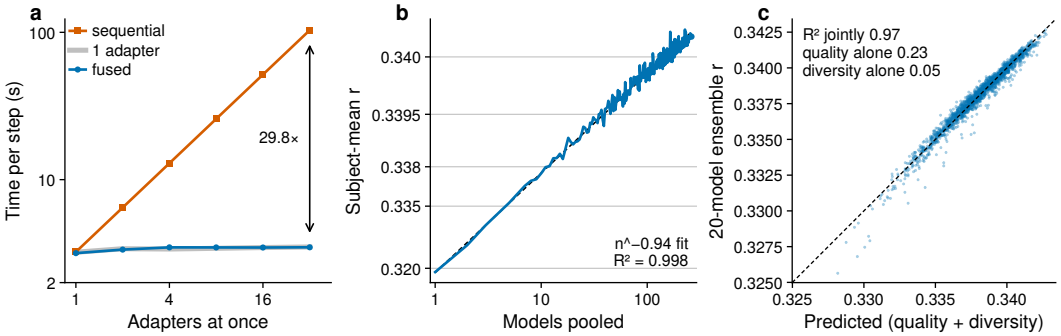

Figure 5: **Scaling out: training a population of adapted models.** *(a)* Training-step time for N rank-16 adapters on a shared backbone: one fused batched gather-matmul vs. N sequential runs. The fused step matches the cost of a single adapter on the same batch (App. C). *(b)* Ensemble performance vs. ensemble size, averaged over random subsets of the 261 adapted models, which is nearly a straight line in the log-log space (App. G). *(c)* Performance of 2,000 random 20-model ensembles vs. the performance predicted linearly on two properties of their members: mean single-model accuracy (quality) and mean pairwise residual correlation (diversity). Neither factor predicts ensemble accuracy alone (R^2 0.23 and 0.05), but together they predict the performance well (R^2=0.97).

which sets the ceiling performance m_∞. The ensemble ceiling thus depends on how correlated the members' errors are ($\sigma^2\rho$), and hence on the ensemble's composition rather than its size alone.

We next examine, at a fixed pool size, what ensemble properties influence the performance. For random 20-model subsets, mean residual correlation and mean single-model performance explain 5% and 23% of the variance in ensemble score individually, but explain 97% of the variance jointly (Fig. 5c; (Krogh & Vedelsby, 1994; Kuncheva & Whitaker, 2003; Douven et al., 2026)). These factors are highly correlated in our pool (r=0.74), as models with more diversity tend to be weaker individually. Therefore, separate coefficients of the fit should not be read as the effect of changing one property while holding the other fixed.

Useful diversity arises primarily from changes in model architecture rather than minor training variations. Models with different readout architectures or backbones show substantially lower residual correlation with the rest of the pool. In contrast, changes in seed, LoRA rank, layer choice, or training data subset provide little diversity (Fig. 9, App. J).

6 Discussion

Three axes for scaling PEFT for brain encoding. Together, our results suggest three complementary scaling directions for future brain encoding models: **1) Scale up** the pretrained prior by leveraging increasingly capable backbones; **2) Scale down** the adaptation by updating only a small number of parameters; and **3) Scale out** by training a large number of lightweight adapters to shared backbones with various inductive biases. Together, PEFT turns a single powerful pretrained model into a family of specialized encoding models without paying the cost of training each model from scratch.

Toward a brain foundation model. Our experiments do not constitute a brain foundation model, but the three directions combine naturally into one. A large pretrained backbone could carry a single shared adapter trained on neural data spanning tasks, datasets, subjects, and even species. In this work, we only scale out the number of adapters for ensemble. However, future work can train a separate adapter for each task, dataset, subject, or even ROI with a hierarchical structure. This hierarchical approach, together with the three scaling directions we discuss above, can create a brain foundation model with various efficient specializations for different downstream tasks.

Limitations. Our models are trained on one benchmark with four subjects, all watching audiovisual narrative. The adapters transfer to a held-out subject (App. M) and to a second dataset (App. N), but neither test leaves the domain of narrative film. Code will be released upon publication.

AI USE STATEMENT

We used an AI coding assistant throughout this work. We used it for engineering: implementing code and analyses. It was also used to draft and edit the text of this paper, and to search the literature for related work. The authors reviewed all generated code, ran the test suite, verified each cited work against its original publication, and checked the claims in the text against the artifacts they refer to. We take responsibility for the final content of this work, including its text, claims and code.

ETHICS STATEMENT

This work uses two public fMRI datasets and collects no new human data. The CNeuroMod/Algonauts 2025 data (Gifford et al., 2025) and the Naturalistic Neuroimaging Database (Aliko et al., 2020) were acquired by their respective teams under their own ethics approvals, from participants who consented to public release, and are distributed de-identified; we use them under their stated terms and add no identifying information. The film and television stimuli remain the property of their rights holders and are used only as model input, as the dataset licences provide.

REPRODUCIBILITY STATEMENT

The configuration needed to reproduce the single models is stated in full in App. S: backbone and revision, tapped layer, adapter rank and placement, learning rates, schedule, window geometry and context lead, the ridge penalties and how they are chosen. App. D states the selection and evaluation protocol: everything is selected on the validation set, the held-out scores were never used to choose a model, the ensemble stack is fitted out of fold by episode, and the subset search is nested. App. G lists every member of the pool with the configuration that distinguishes it. Because many effects here are small, we report what can be resolved: the seed spread of the shipped configuration, paired parcel-level bootstraps between fixed runs, and the resolution floor below which a validation difference does not survive to the held-out benchmarks (App. R). Code will be released on publication.

REFERENCES

Hossein Adeli, Sun Minni, and Nikolaus Kriegeskorte. Transformer brain encoders explain human high-level visual responses. *Advances in neural information processing systems*, 2025.

Sarah Aliko, Jiawen Huang, Florin Gheorghiu, Stefanie Meliss, and Jeremy I Skipper. A naturalistic neuroimaging database for understanding the brain using ecological stimuli. *Scientific data*, 7(1): 347, 2020.

Antonis Antoniades, Yiyi Yu, Joseph Canzano, William Wang, and Spencer LaVere Smith. Neuroformer: Multimodal and multitask generative pretraining for brain data. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2311.00136.

Daniel Carlström Schad, Shrey Dixit, Janis Keck, Viktor Studenyak, Aleksandr Shpilevoi, and Andrej Bicanski. Vibe: Video-input brain encoder for fmri response modeling. *arXiv preprint arXiv:2507.17958*, 2025.

Charlotte Caucheteux and Jean-Rémi King. Brains and algorithms partially converge in natural language processing. *Communications Biology*, 5:134, 2022. doi: 10.1038/s42003-022-03036-1.

Lequn Chen, Zihao Ye, Yongji Wu, Danyang Zhuo, Luis Ceze, and Arvind Krishnamurthy. Punica: Multi-tenant LoRA serving. In *Proceedings of Machine Learning and Systems (MLSys)*, 2024. arXiv:2310.18547.

Michael W. Cole, Takuya Ito, Danielle S. Bassett, and Douglas H. Schultz. Activity flow over resting-state networks shapes cognitive task activations. *Nature Neuroscience*, 19(12):1718–1726, 2016. doi: 10.1038/nn.4406.

Joel Dapello, Kohitij Kar, Martin Schrimpf, Robert Geary, Michael Ferguson, David D. Cox, and James J. DiCarlo. Aligning model and macaque inferior temporal cortex representations improves model-to-human behavioral alignment and adversarial robustness. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL `https://openreview.net/forum?id=SMYdcXjJh1q`.

Stéphane d'Ascoli, Jérémy Rapin, Yohann Benchetrit, Hubert Banville, and Jean-Rémi King. Tribe: Trimodal brain encoder for whole-brain fmri response prediction. In *International Conference on Learning Representations (ICLR)*, 2026.

Adrien Doerig, Rowan P. Sommers, Katja Seeliger, Blake Richards, Jenann Ismael, Grace W. Lindsay, Konrad P. Kording, Talia Konkle, Marcel A. J. van Gerven, Nikolaus Kriegeskorte, and Tim C. Kietzmann. The neuroconnectionist research programme. *Nature Reviews Neuroscience*, 24(7): 431–450, 2023. doi: 10.1038/s41583-023-00705-w.

Zijian Dong, Ruilin Li, Yilei Wu, Thuan Tinh Nguyen, Joanna Su Xian Chong, Fang Ji, Nathanael Ren Jie Tong, Christopher Li Hsian Chen, and Juan Helen Zhou. Brain-jepa: Brain dynamics foundation model with gradient positioning and spatiotemporal masking. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 86048–86073, 2024. Spotlight.

Igor Douven, Nikolaus Kriegeskorte, Patrick Stinson, and Josh Ying. Condorcet and beyond: An empirical comparison of voting rules. *Cognitive Science*, 50(7):e70242, 2026.

Callie Federer, Haoyan Xu, Alona Fyshe, and Joel Zylberberg. Improved object recognition using neural networks trained to mimic the brain's statistical properties. *Neural Networks*, 131:103–114, 2020. doi: 10.1016/j.neunet.2020.07.013.

Emily S. Finn and Peter A. Bandettini. Movie-watching outperforms rest for functional connectivity-based prediction of behavior. *NeuroImage*, 235:117963, 2021. doi: 10.1016/j.neuroimage.2021.117963.

Trevor Gale, Deepak Narayanan, Cliff Young, and Matei Zaharia. MegaBlocks: Efficient sparse training with mixture-of-experts. In *Proceedings of Machine Learning and Systems (MLSys)*, 2023. arXiv:2211.15841.

Alessandro T. Gifford et al. The algonauts project 2025 challenge: How the human brain makes sense of multimodal movies. *arXiv preprint arXiv:2501.00504*, 2025.

Abdulkadir Gokce, Badr AlKhamissi, and Martin Schrimpf. Mirage: Adaptive multimodal gating for whole-brain fmri encoding. *arXiv preprint arXiv:2605.29850*, 2026a.

Abdulkadir Gokce, Yingtian Tang, and Martin Schrimpf. Multimodal scaling laws for task & data-optimized models of visual cortex. In *International Conference on Machine Learning (ICML)*, 2026b.

Manshan Guo, Bhavin Choksi, Sari Sadiya, Alessandro T. Gifford, Martina G. Vilas, Radoslaw M. Cichy, and Gemma Roig. Limited but consistent gains in adversarial robustness by co-training object recognition models with human eeg. In *Computer Vision – ECCV 2024 Workshops (Human-Centric Computer Vision)*, volume 15636 of *LNCS*. Springer, 2025.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Alexander G. Huth, Wendy A. de Heer, Thomas L. Griffiths, Frédéric E. Theunissen, and Jack L. Gallant. Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature*, 532 (7600):453–458, 2016. doi: 10.1038/nature17637.

Ethan Hwang, Hossein Adeli, Wenxuan Guo, Andrew Luo, and Nikolaus Kriegeskorte. In silico mapping of visual categorical selectivity across the whole brain. *Advances in Neural Information Processing Systems*, 2025.

Wei-Bang Jiang, Li-Ming Zhao, and Bao-Liang Lu. Large brain model for learning generic representations with tremendous eeg data in bci. In *International Conference on Learning Representations (ICLR)*, 2024. Spotlight; arXiv:2405.18765.

Wei-Bang Jiang et al. Neurolm: A universal multi-task foundation model for bridging the gap between language and eeg signals. In *International Conference on Learning Representations (ICLR)*, 2025.

Meenakshi Khosla and Leila Wehbe. Higher visual areas act like domain-general filters with strong selectivity and functional specialization. *Nature Communications*, 17(1):7484, 2026.

David Klindt, Alexander S Ecker, Thomas Euler, and Matthias Bethge. Neural system identification for large populations separating "what" and "where". *Advances in neural information processing systems*, 30, 2017.

Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. *Advances in neural information processing systems*, 7, 1994.

Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207, 2003.

Connor Lane, Mihir Tripathy, Leema Krishna Murali, Ratna Sagari Grandhi, Shamus Sim Zi Yang, Sam Gijsen, Debojyoti Das, Manish Ram, Utkarsh Kumar Singh, Cesar Kadir Torrico Villanueva, Yuxiang Wei, Will Beddow, Gianfranco Cortés, Suin Cho, Daniel Z. Kaplan, Benjamin Warner, Tanishq Mathew Abraham, and Paul S. Scotti. Scaling vision transformers for functional mri with flat maps. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2510.13768v2; v1 appeared at NeurIPS 2025 Workshop, Foundation Models for the Brain and Body.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 3045–3059, 2021.

Zhe Li, Wieland Brendel, Edgar Y. Walker, Erick Cobos, Taliah Muhammad, Jacob Reimer, Matthias Bethge, Fabian H. Sinz, Xaq Pitkow, and Andreas S. Tolias. Learning from brains how to regularize machines. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in neural information processing systems*, 35:1950–1965, 2022.

Zitong Lu, Yile Wang, and Julie D Golomb. Achieving more human brain-like vision via human eeg representational alignment. *Communications Biology*, 9(1):463, 2026.

Konstantin-Klemens Lurz, Mohammad Bashiri, Konstantin Willeke, Akshay K. Jagadish, Eric Wang, Edgar Y. Walker, Santiago A. Cadena, Taliah Muhammad, Erick Cobos, Andreas S. Tolias, Alexander S. Ecker, and Fabian H. Sinz. Generalization in data-driven models of primary visual cortex. In *International Conference on Learning Representations (ICLR)*, 2021.

Weijian Mai, Mu Nan, Yu Zhu, Jiahang Cao, Rui Zhang, Yuqin Dai, Chunfeng Song, Andrew F. Luo, and Jiamin Wu. Neuroflow: Toward unified visual encoding and decoding from neural activity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2604.09817.

Omer Moussa and Mariya Toneva. Brain-tuning improves generalizability and efficiency of brain alignment in speech models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2510.21520.

Omer Moussa, Dietrich Klakow, and Mariya Toneva. Improving semantic understanding in speech language models via brain-tuning. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=KL8Sm4xRn7.

Thomas Naselaris, Kendrick N. Kay, Shinji Nishimoto, and Jack L. Gallant. Encoding and decoding in fmri. *NeuroImage*, 56(2):400–410, 2011. doi: 10.1016/j.neuroimage.2010.07.073.

Anwar O. Nunez-Elizalde, Alexander G. Huth, and Jack L. Gallant. Voxelwise encoding models with non-spherical multivariate normal priors. *NeuroImage*, 197:482–494, 2019. doi: 10.1016/j.neuroimage.2019.04.012.

Josue Ortega Caro, Antonio Henrique de Oliveira Fonseca, Syed Rizvi, Matteo Rosati, Christopher Averill, James Cross, Prateek Mittal, Emanuele Zappala, Rahul Dhodapkar, Chadi Abdallah, et al. Brainlm: A foundation model for brain activity recordings. In *International Conference on Learning Representations*, volume 2024, pp. 565–576, 2024.

Michael P Perrone and Leon N Cooper. When networks disagree: Ensemble methods for hybrid neural networks. In *How we learn; How we remember: Toward an understanding of brain and neural systems: Selected papers of Leon N Cooper*, pp. 342–358. World Scientific, 1995.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume*, pp. 487–503, 2021.

Cassidy Pirlot, Richard C. Gerum, Cory Efird, Joel Zylberberg, and Alona Fyshe. Improving the accuracy and robustness of cnns using a deep cca neural data regularizer. *arXiv preprint arXiv:2209.02582*, 2022.

Xuelin Qian, Yikai Wang, Xinwei Sun, Yanwei Fu, Xiangyang Xue, and Jianfeng Feng. Lea: Learning latent embedding alignment model for fmri decoding and encoding. *Transactions on Machine Learning Research (TMLR)*, 2024. arXiv:2303.14730.

Shahd Safarani, Arne Nix, Konstantin Willeke, Santiago A. Cadena, Kelli Restivo, George Denfield, Andreas S. Tolias, and Fabian H. Sinz. Towards robust vision by multi-task learning on monkey visual cortex. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J Majaj, Rishi Rajalingham, Elias B Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, et al. Brain-score: Which artificial neural network for object recognition is most brain-like? *BioRxiv*, pp. 407007, 2018.

Martin Schrimpf, Idan A. Blank, Greta Tuckute, Carina Kauf, Eghbal A. Hosseini, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. The neural architecture of language: Integrative modeling converges on predictive processing. *Proceedings of the National Academy of Sciences*, 118(45):e2105646118, 2021. doi: 10.1073/pnas.2105646118.

Dan Schwartz, Mariya Toneva, and Leila Wehbe. Inducing brain-relevant bias in natural language processing models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 14123–14133, 2019.

Paul S. Scotti and Mihir Tripathy. Insights from the algonauts 2025 winners. *arXiv preprint arXiv:2508.10784*, 2025.

Ying Sheng, Shiyi Cao, Dacheng Li, Coleman Hooper, Nicholas Lee, Shuo Yang, Christopher Chou, Banghua Zhu, Lianmin Zheng, Kurt Keutzer, Joseph E. Gonzalez, and Ion Stoica. S-LoRA: Serving thousands of concurrent LoRA adapters. In *Proceedings of Machine Learning and Systems (MLSys)*, 2024. arXiv:2311.03285.

Stephen M. Smith, Peter T. Fox, Karla L. Miller, David C. Glahn, P. Mickle Fox, Clare E. Mackay, Nicola Filippini, Kate E. Watkins, Roberto Toro, Angela R. Laird, and Christian F. Beckmann. Correspondence of the brain's functional architecture during activation and rest. *Proceedings of the National Academy of Sciences*, 106(31):13040–13045, 2009. doi: 10.1073/pnas.0905267106.

Ghislain St-Yves, Emily J Allen, Yihan Wu, Kendrick Kay, and Thomas Naselaris. Brain-optimized deep neural network models of human visual areas learn non-hierarchical representations. *Nature communications*, 14(1):3329, 2023.

Shawn Tan, Yikang Shen, Rameswar Panda, and Aaron Courville. Scattered mixture-of-experts implementation. *arXiv preprint arXiv:2403.08245*, 2024.

I. Tavor, O. Parker Jones, R. B. Mars, S. M. Smith, T. E. Behrens, and S. Jbabdi. Task-free MRI predicts individual differences in brain activity during task performance. *Science*, 352(6282): 216–220, 2016. doi: 10.1126/science.aad8127.

Mariya Toneva and Leila Wehbe. Interpreting and improving natural-language processing (in machines) with natural language-processing (in the brain). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Nishitha Vattikonda, Aditya R. Vaidya, Richard J. Antonello, and Alexander G. Huth. Brainwavlm: Fine-tuning speech representations with brain responses to language. In *ICML 2025 Workshop on Machine Learning for Audio (arXiv:2502.08866)*, 2025.

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 *International Conference on Learning Representations (ICLR)*, 2023.

Danny Wood, Tingting Mu, Andrew M Webb, Henry WJ Reeve, Mikel Lujan, and Gavin Brown. A unified theory of diversity in ensemble learning. *Journal of machine learning research*, 24(359): 1–49, 2023.

Haitao Wu, Qirui Zhang, Zhouheng Yao, Shangquan Sun, Qihao Zheng, Mianxin Liu, Chi Zhang, Wanli Ouyang, Chunfeng Song, Changqing Zhang, et al. Brainjanus: A unified model for understanding and generation across brain, vision, and language. *arXiv preprint arXiv:2606.30319*, 2026.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, Baosong Yang, Bin Zhang, Ziyang Ma, Xipin Wei, Shuai Bai, Keqin Chen, Xuejing Liu, Peng Wang, Mingkun Yang, Dayiheng Liu, Xingzhang Ren, Bo Zheng, Rui Men, Fan Zhou, Bowen Yu, Jianxin Yang, Le Yu, Jingren Zhou, and Junyang Lin. Qwen3-Omni technical report. *arXiv preprint arXiv:2509.17765*, 2025. Checkpoint `Qwen/Qwen3-Omni-30B-A3B-Thinking`.

Daniel L. K. Yamins and James J. DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19(3):356–365, 2016. doi: 10.1038/nn.4244.

Minghao Yan, Zhuang Wang, Zhen Jia, Shivaram Venkataraman, and Yida Wang. Plora: Efficient concurrent lora training for large language models. In *Forty-third International Conference on Machine Learning*, 2026.

Huzheng Yang, James Gee, and Jianbo Shi. Memory encoding model. *arXiv preprint arXiv:2308.01175*, 2023.

Joel Ye, Fabio Rizzoglio, Xuan Ma, Adam Smoulder, Hongwei Mao, Gary Blumenthal, William Hockeimer, Nicolas Kunigk, Dalton Moore, Patrick Marino, et al. A generalist intracortical motor decoder. *Advances in Neural Information Processing Systems*, 38:110547–110587, 2026.

Muquan Yu, Mu Nan, Hossein Adeli, Jacob Prince, John A Pyles, Leila Wehbe, Maggie Henderson, Michael Tarr, and Andrew Luo. Meta-learning an in-context transformer model of human higher visual cortex. *Advances in Neural Information Processing Systems*, 2025.

Andrii Zahorodnii, Christopher Wang, Bennett Stankovits, Charikleia Moraitaki, Geeling Chau, Andrei Barbu, Boris Katz, and Ila R. Fiete. Neuroprobe: Evaluating intracranial brain responses to naturalistic stimuli. *arXiv preprint arXiv:2509.21671*, 2025.

Hui Zheng, Hai-Teng Wang, Wei-Bang Jiang, Zhong-Tao Chen, Li He, Pei-Yang Lin, Peng-Hu Wei, Guo-Guang Zhao, and Yun-Zhe Liu. Du-in: Discrete units-guided mask modeling for decoding speech from intracranial neural signals. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Jingwei Zuo, Xinze Feng, Zien Liu, Kaijian Wang, Fanjiang Ye, Ye Cao, Zhuang Wang, and Yuke Wang. ALTO: Adaptive LoRA tuning and orchestration for heterogeneous LoRA training workloads. *arXiv preprint arXiv:2604.05426*, 2026.

A EXTENDED RELATED WORK

The full version of §2, by family: *where does the neural signal enter the encoding model pipeline?*

Frozen backbone, trained readout. The dominant paradigm freezes a model pretrained on a behavioral or self-supervised task and fits a readout from its hidden states to neural responses: the lineage running from voxelwise ridge regression (Naselaris et al., 2011; Huth et al., 2016; Nunez-Elizalde et al., 2019) through task-optimised network comparisons (Yamins & DiCarlo, 2016; Schrimpf et al., 2021; Caucheteux & King, 2022) to today's multimodal encoders. Every leading entry in the Algonauts 2025 challenge leaderboard (Gifford et al., 2025) followed this recipe, including MIRAGE (Gokce et al., 2026a), which we build on directly, TRIBE (d'Ascoli et al., 2026) and VIBE (Carlström Schad et al., 2025). A retrospective report argued that architectural choices mattered less than ensembling (Scotti & Tripathy, 2025). Readout design has been studied on its own terms (Adeli et al., 2025; Lurz et al., 2021), and the parameter counts there are instructive: a ROI-query transformer readout is $\sim$37M against classical ridge's $\sim$1,200M, so within this family a *larger* readout is not what improved accuracy (Adeli et al., 2025).

Backbones fine-tuned by brain data. A second family updates the stimulus model's own weights from neural recordings. Inducing brain-relevant bias in language models (Schwartz et al., 2019; Toneva & Wehbe, 2019) leads to brain-tuning of speech models, which improves not only alignment but downstream semantic tasks (Moussa et al., 2025; Moussa & Toneva, 2025). The vision analogue trains with neural regularisation and gains robustness and behavioural alignment (Li et al., 2019; Federer et al., 2020; Safarani et al., 2021; Lu et al., 2026). This is the family we belong to, and parameter-efficient adaptation is already used within it: BrainWavLM brain-tunes with 166K LoRA parameters and matches a full fine-tune (Vattikonda et al., 2025), and rank-8 LoRA at 0.625% of parameters suffices for cross-participant brain-tuning (Moussa & Toneva, 2025). What is new here is **scale**: the largest backbones whose weights have been updated by brain data are $\sim$90M–320M, and models above 15B appear in this literature only as *frozen* feature extractors. We update backbones up to 104B, and we test it on a whole-brain encoding benchmark rather than on a specific brain area for downstream task transfer.

Joint task and brain-alignment objectives. A third family trains a primary objective together with an auxiliary alignment term against neural data (Dapello et al., 2023; Pirlot et al., 2022). The binding constraint is that the stimuli with recordings are a tiny fraction of the task corpus, which is precisely the asymmetry parameter-efficient adaptation is built for. This family is also where the field's methodological warning lives: shuffled- or permuted-target controls sometimes reproduce much of the reported benefit (Federer et al., 2020; Guo et al., 2025), so a gain from neural supervision has to be shown against matched compute. We hold ourselves to the same standard throughout (§5.3), and run the shifted-target control itself (App. L).

Brain-only pretraining, aligned afterwards. A fourth family pretrains self-supervised on neural data alone across subjects, sessions and datasets: masked autoencoding for fMRI (Ortega Caro et al., 2024; Dong et al., 2024; Lane et al., 2026), EEG (Jiang et al., 2024; 2025), intracranial recordings (Wang et al., 2023; Zheng et al., 2024) and other modalities, and attaches the stimulus afterwards. This scales along the axis that is hardest to buy, but the recent evidence is that what it captures is intrinsic dynamics and connectivity rather than stimulus-referred computation: the seed work for this family reports a null result for brain-only pretraining on trait prediction (Lane et al., 2026), an intracranial benchmark finds such models losing to linear baselines on stimulus decoding (Zahorodnii et al., 2025), and pretraining gains shrink under data-matched controls (Ye et al., 2026).

Unified models over mixed tokens. A fifth family puts stimulus and brain tokens into one model so that encoding and decoding share parameters (Wu et al., 2026; Mai et al., 2026; Qian et al., 2024; Antoniades et al., 2024). It is worth noting that the most recent of these adapts a large pretrained backbone with LoRA rather than training from scratch (Wu et al., 2026), the same bet we are making. No model in this family has been shown to beat a specialised encoder on voxelwise correlation; they report decoding or semantic-level proxies instead.

Table 3: **From MIRAGE's sliding window to a single causal pass.** Friends s06 subject-mean Pearson over four subjects. The sliding-window protocol re-encodes the preceding stimulus at every timestep (∼2,800 forward passes per clip); a single causal pass encodes each clip or window once, and a context lead supplies the preceding stimulus that the first timesteps of a window would otherwise lack. Step time and training cost are for the linear readout. The sliding-window cost is the ∼700 GPU-hours reported by the baseline's authors, which we reproduced on our hardware (App. C). The frozen MIRAGE head was trained at a 30 s lead and has no longer-lead entries; with LoRA the head was trained at 30 and 60 s, and its re-fit re-solves the per-subject output layer in closed form. Both LoRA column pairs show seed 0. The linear readout has no whole-clip entries because whole-clip encoding is the limit of its lead ladder.

	Extract	MIRAGE head (1,123M)			Linear (52.3M)			Train
Stimulus design	(GPU-h)	frozen	LoRA	+ re-fit	LoRA	+ re-fit	s/step	(GPU-h)
Sliding window, full-res. media	∼700	0.3187	–	–	–	–	–	–
Sliding window, 336×224 media	∼700	0.3178	–	–	–	–	–	–
Single pass, whole clip	1.99	0.3149	–	–	–	–	–	–
Single pass, whole clip, 336×224	1.99	0.3137	–	–	–	–	–	–
+ window-local (149 s), no lead	1.99	0.3091	–	–	0.3148	0.3206	9.7	4.0
+ 30 s context lead	1.99	0.3124	0.3214	0.3236	0.3187	0.3249	10.4	4.3
+ 60 s context lead	1.99	–	0.3220	0.3247	0.3202	**0.3263**	12.0	4.8
+ 90 s context lead	1.99	–	–	–	0.3196	0.3260	12.9	5.2
+ 120 s context lead	1.99	–	–	–	0.3203	0.3263	14.4	5.6

Parameter-efficient fine-tuning. LoRA (Hu et al., 2021) was developed to specialise large models cheaply. Its usual justification is engineering; here it is statistical. The size of the update should match the size of the supervision, and a rank-16 update on a 30B backbone is a defensible amount of capacity to give a few dozen hours of fMRI from four subjects, where full fine-tuning is not.

Batched multi-adapter kernels. Routing one batch of tokens to many adapters is a solved problem in the forward direction: Punica's segmented gather matmul (Chen et al., 2024), S-LoRA's paged adapter pool (Sheng et al., 2024). Serving needs no gradient, so none of that work has one. Training-side systems are newer and batch per adapter, each receiving its own input, so every group is a full sequence (Yan et al., 2026; Zuo et al., 2026); the per-token form of the primitive comes instead from mixture-of-experts training, where MegaBlocks pads groups to block multiples (Gale et al., 2023) and ScatterMoE does not (Tan et al., 2024), and stock PyTorch now ships a differentiable grouped matmul taking a distinct matrix per group. We therefore claim no new operator, and neither idea ours is built from, gathering rather than padding and carrying the adapter index on a grid dimension, is new either. What is specific is the regime. Grouped matmuls are built for many tokens per group, and a pool with one adapter per model, subject or parcel sits at the far end of that axis, where a group is a handful of rows: there the fused kernel's cost is close to independent of pool size and the alternatives' costs are not. At this paper's operating point that is a convenience, but it is what would make an adapter per parcel trainable at all (§6).

What a brain foundation model should capture. Our reading of the fourth family reflects a position worth stating plainly: a model that predicts brain state from brain state has removed the input from the system, and the standard of explanation in this field has long been a *stimulus-computable* model (Naselaris et al., 2011; Yamins & DiCarlo, 2016; Doerig et al., 2023). Empirically, naturalistic stimulus-driven acquisition carries more behaviourally relevant signal than rest (Finn & Bandettini, 2021). We do not claim the stronger version: intrinsic activity is demonstrably informative about task-evoked responses (Tavor et al., 2016; Cole et al., 2016; Smith et al., 2009), so brain-only pretraining is not modelling noise. The claim is about explanation rather than prediction: a foundation model of the brain should capture its computations *in relation to the outside world*, which makes a strong stimulus-side prior a component of the goal rather than scaffolding to be discarded.

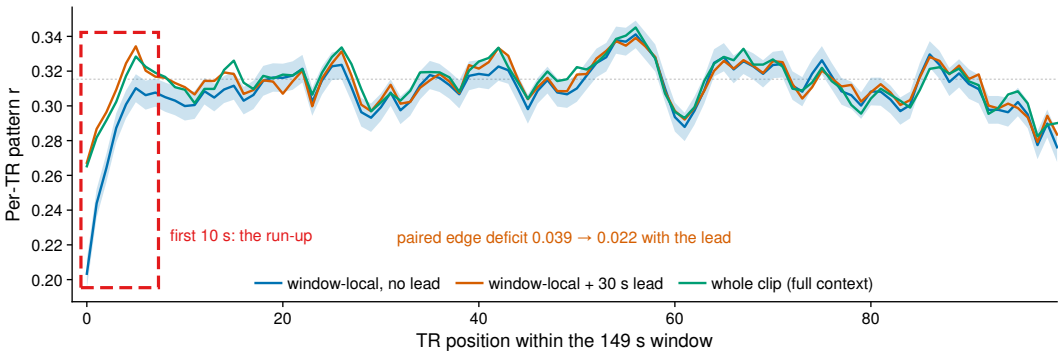

Figure 6: **Accuracy within the window.** Accuracy against TR position inside the 149 s window for three input designs, scored with the same frozen MIRAGE head on Friends s06 (606 complete interior windows; the band is the s.e.m. over windows for the no-lead arm). The dashed box marks the first 10 s, where the designs differ. Accuracy here is the per-TR spatial correlation between predicted and measured 1000-parcel patterns, which is defined at a single time point; it is not comparable to the per-parcel headline metric. Without a lead a window starts 0.110 below its plateau and needs about five TRs to reach it. With a 30 s lead it starts 0.048 below plateau, and with the whole clip 0.051. The first window of each clip is excluded because of a clip-onset transient.

B THE FULL INPUT DESIGN

Tab. 3 gives the full results behind §5.3. Lowering the stimulus resolution to 336×224 costs 0.0009, so all later rows use the low-resolution media. Replacing the sliding window with a single causal pass over the whole clip costs 0.0038 under the frozen MIRAGE head at original resolution (0.3187 against 0.3149) and 0.0041 at low resolution, at two orders of magnitude less extraction compute. Restricting the pass to the 149 s window costs a further 0.0046, and the context lead recovers it.

MIRAGE head with LoRA. With LoRA on single-pass window-local features, the MIRAGE head reaches 0.3247 at a 60 s lead, above the 0.3187 of the sliding-window protocol it was designed for. The input design therefore costs nothing once the backbone adapts, whichever readout is used. The head is trained in two stages, LoRA under the frozen head and then jointly with its per-subject output layer, so it receives twice the linear readout's 1,429 steps. At matched lead with both readouts re-fit, the linear readout is still ahead by $+0.0020$ [$+0.0008$, $+0.0033$] over four seed-matched pairs (0.3261 against 0.3241). Lead and readout do not interact: going from 30 to 60 s is worth $+0.0010$ to the head and $+0.0012$ to the linear readout. The frozen head, fitted on 30 s features, cannot use a longer lead ($+0.0003$ from 30 to 60 s before the joint stage); the joint stage recovers the gain.

Lead ladder. Step time grows from 9.7 s at no lead to 14.4 s at 120 s. Accuracy is flat past 60 s, so we use 60 s. Each rung is a single run, so we compare rungs with a paired parcel bootstrap. The first 30 s of lead is worth $+0.00396$ and the next 30 s $+0.00142$, both with intervals above zero and both well above the seed s.d. of this configuration. The linear readout encodes the span $[t_0 - \ell,\ t_1]$ for a lead ℓ, so whole-clip encoding is the $\ell \to t_0$ limit of the ladder, and the table has no separate whole-clip rows for it. The flat ladder shows that 60 s suffices at the resolution of this metric.

Within-window accuracy. Fig. 6 shows the same effect per timestep. The paired middle-minus-start deficit is 0.0393 ± 0.0043 with no lead, 0.0220 ± 0.0042 with a 30 s lead, and 0.0235 ± 0.0044 with the whole clip. The lead therefore saturates on this measure as it does on the score. The remaining deficit of about 0.022 is present even with the whole clip, so it is not caused by missing context; only the 0.0173 difference between arms is. At the end of the window, the three arms' last five TRs sit 0.025, 0.026 and 0.022 below their plateaus, so stimulus after the window boundary is worth at most 0.003, within the s.e.m. (see also App. F). The closed-form re-fit is worth $+0.0059$ to $+0.0064$ at every rung of the ladder, with no trend in the lead.

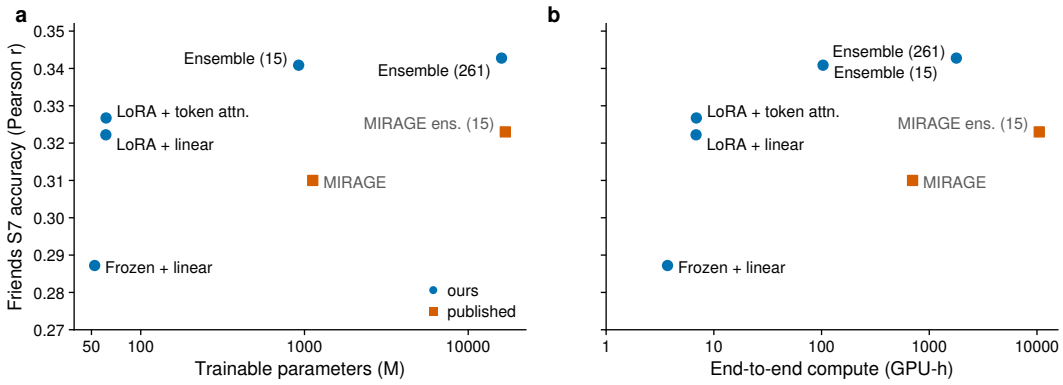

Figure 7: **Accuracy against cost.** Held-out Friends S7 accuracy against trainable parameters *(a)* and against end-to-end compute including feature extraction *(b)*, both on log scales. Published baselines are squares and our models circles; an N-model ensemble is plotted at N times a member's cost.

C Efficiency and the multi-adapter kernel

Fig. 7 places our models and the published baselines on two cost axes: trainable parameters, and end-to-end GPU-hours including feature extraction. Our single models use an order of magnitude fewer trainable parameters than the frozen-feature baselines and two orders of magnitude less compute than the sliding-window protocol, at equal or better accuracy.

Compute breakdown. Tab. 4 breaks down the cost of both configurations on one A800-80GB over the same 331-clip corpus.

Extraction. The sliding-window protocol performs 2,758 forward passes per clip, one per cutoff on the 2 Hz grid and one per word onset, against 5.1 for a single causal pass, i.e. $545\times$ fewer forwards. Our forwards are longer ($\approx$14,900 tokens each against a capped prefix), so the saving in GPU-hours is $352\times$: 700.6 against 1.99 GPU-h. We measured this on our own hardware; it is the figure quoted in §4.

End to end. Our total comprises extraction (1.99), adapter training (4.84) and the closed-form re-fit (1.67), 8.5 GPU-h against the baseline's 702.5, an $83\times$ reduction. We include the re-fit because every accuracy we report is post-re-fit. Without it the total is 6.8 GPU-h ($103\times$). The baseline is charged its head training (1.9 GPU-h, from our reproduction of the published recipe).

The sliding-window extraction was measured over the 49 validation clips and scaled to the full corpus; these clips are 1.1% longer than the corpus mean, so the figure slightly over-estimates. The two protocols also compute different representations: the sliding window re-encodes a capped prefix (60 s of audio, 4 s of video, 1,024 words) at every cutoff, whereas a single pass with a 60 s lead gives each timestep the whole window's joint context. The ratio compares the two protocols as run, not two ways of computing the same features.

Multi-adapter kernel. Adapters are small and structurally identical, so the routed low-rank update for a whole pool can be written as one batched gather-matmul instead of N separate ones. Fig. 8 benchmarks our Triton kernel for this on the backbone of §4 (the data behind Fig. 5a). Against N runs taken one at a time, the fused pass is $29.8\times$ faster at $N = 32$. This is an occupancy effect, not a saving in arithmetic: a batch-1 step on a sparse mixture-of-experts backbone leaves the GPU underused, and the fused step at $N = 32$ costs only $1.08\times$ a single-adapter step. Against the same batch of N through a single adapter, the fused pass costs $1.00\times$ (0.99–1.03 over $N \leq 32$), so using different adapters adds no cost over batching. Memory is the limit: at rank 16 the pool and its optimiser state fill an 80 GB card between 32 and 64 adapters.

Table 4: **Compute for the two single-model configurations.** Both columns use one A800-80GB, the same 331-clip corpus and the same 336×224 media cache (built once on CPU and shared). *Sliding window* re-encodes a capped prefix (60 s audio, 4 s video, 1,024 words) at every cutoff; *single causal pass* encodes each 149 s window once with a 60 s lead. Sliding-window extraction was measured over the 49 validation clips and scaled to 331 clips. Baseline training is our reproduction of the published head recipe. The closed-form re-fit is included in our column because all reported accuracies are post-re-fit. Extraction is 352× cheaper and the end-to-end cost 83× lower.

	Frozen + 1,123M readout	LoRA + 52.3M readout
Hardware and precision		
Device	1× A800-80GB	
Precision	bf16 backbone, fp32 patch-embed conv	
Stimulus encoding		
Protocol	sliding window	single causal pass
Forward passes / clip	2,758	5.1
Tokens / forward	≤2,000	14,890
Seconds / forward	2.76	4.28
Extraction, 331 clips (GPU-h)	**701**	**1.99**
Training		
Reads	stored features	live re-encode from media
Batch	16 windows	1 window (400 target rows)
Optimiser steps	5,670 (15 epochs)	1,429 (1 epoch)
Backward through backbone	no	yes, 32 of 48 layers
Seconds / step	–	12.0
Training (GPU-h)	1.88	4.84
Closed-form re-fit (GPU-h)	–	1.67
Trainable parameters	1,123M	61.2M
End-to-end (GPU-h)	**702**	**8.5**

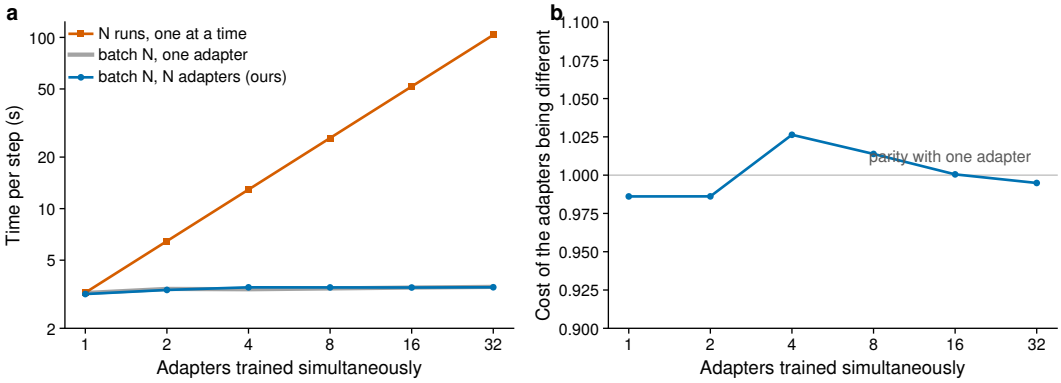

Figure 8: **Multi-adapter kernel benchmark.** `Qwen3-Omni` truncated at the tap, rank-16 adapters on $q/k/v/o$, sequence length 128, one sample per adapter, one A800, median of 5 timed steps after 2 warm-up steps. *(a)* One fused batched gather-matmul over N different adapters (blue) matches the same batch through one adapter (grey); running the N jobs one at a time (orange) grows linearly in N. *(b)* The ratio of the fused pass to the single-adapter batch is $1.00\times$ throughout, up to run-to-run noise of ± 0.03.

D SELECTION AND EVALUATION PROTOCOL

Validation. Every design choice was made on the Friends season-6 validation split: extraction design, context lead, tap layer, LoRA rank, learning rates, readout family, ensemble membership and combiner weights. The validation column therefore carries the selection bias of the whole search.

Table 5: **LoRA rank.** Shipped recipe otherwise unchanged (QWEN3-OMNI, linear readout, tap 32, 60 s lead, LoRA lr 6×10^{-4}, 1,429 steps), Friends s06 subject-mean after the closed-form re-fit. Rank and readout learning rate were not varied independently; the matched comparison is the two rows at 3×10^{-4}. Diverged runs are counted in parentheses.

Rank	Readout lr	Runs	s06 (mean)	s.d.
16	3×10^{-4}	4	0.3262	0.00007
32	3×10^{-4}	1	0.3268	–
32	5×10^{-4}	9	0.3257	0.00109
64	5×10^{-4}	4 (+1 diverged)	0.3237	0.00128
128	5×10^{-4}	0 (+2 diverged)	–	–

Held-out evaluation. Model selection used validation scores only. Every model was fixed on season 6 before submission, and held-out scores were only reported, never used to select, rank or tune a model. Our submissions were made to the public leaderboard in the post-challenge phase of Algonauts 2025.

Tap layer. The readout reads decoder layer 32 of 48, and the decoder is truncated there. The tap was chosen on a frozen ridge screen and checked against trained runs at the neighbouring layers (Tab. 6, App. E).

Seed variability. Each of our rows in Tab. 1 is a single run. Across seeds, LoRA + linear scores 0.32609 ± 0.00024 (15 seeds) and LoRA + cross-attn 0.33070 ± 0.00023 (3 seeds); the tabulated runs lie $+0.8$ and $+1.1$ s.d. above their arm means. Rows that differ by less than about 0.0008 are therefore not separated by the table alone, and the comparisons in §5.2 use a paired bootstrap or seed statistics.

Ridge penalties. The closed-form re-fit accumulates its moments on the training windows only. Per-parcel penalties are chosen from a fixed five-point grid on a deterministic strided 15% holdout of the training windows (128,400 fit rows and 22,800 holdout rows per subject). The chosen penalty is then re-solved on fit and holdout together. Season 6 is never used to fit or select a penalty; it is only scored, over its 250 windows.

Ensemble stacking. The per-parcel minimum-variance combiner (Perrone & Cooper, 1995) is fitted on season 6 out of fold, split by episode. Windows within an episode share stimulus and scanner state, so a window-level split would leak across folds. Rows in one half are predicted by weights fitted on the other, and one scoring pass covers every row exactly once. For an unfitted rule such as the uniform mean, out-of-fold and in-sample scores coincide by construction, which serves as a check on the procedure. The ensemble rows of Tab. 1 are the out-of-fold stack. For the 261-model pool the uniform mean scores 0.34007 and the stack 0.34834.

Subset search. The 15-model subset is chosen by greedy selection with one-swap refinement under a nested protocol: the subset and its weights are both chosen on one episode half and scored on the other, and the two quarters within each half set greedy's stopping step. The assembled out-of-fold predictions again cover every row once. The procedure scores 0.34651, and the selection bias of the reported roster is $+0.00035$.

E HYPERPARAMETER SELECTION

All values below were chosen on Friends season 6 (App. D). We report the full ladders rather than only the chosen values.

LoRA rank (Tab. 5). We hold $\alpha = 2r$ fixed, so raising the rank also raises the effective scale of the update at a fixed LoRA learning rate; rank and learning rate are therefore coupled. At the one matched setting (readout lr 3×10^{-4}), rank 32 is $+0.0006$ above rank 16, inside the 0.00077 seed spread of this configuration. The rank-32 arms also vary more across seeds (0.00109 over nine seeds,

Table 6: **Tap layer.** *Frozen* is the closed-form ridge readout on unadapted features (no adapter, no training); *trained* is the full recipe, one seed per rung, after the re-fit. The frozen screen peaks at layer 30; the trained runs at layers 31–34 lie within 0.0006 of one another, inside the 0.00077 seed spread of this configuration.

Tap layer	Frozen	Trained
16	0.2682	–
24	0.2839	–
30	0.2889	0.3252
31	0.2885	0.3260
32	0.2881	0.3263
33	0.2872	0.3258
34	0.2876	0.3264
40	0.2836	–
44	0.2784	–

Table 7: **Learning rates.** Measured in an earlier configuration without context lead or closed-form re-fit, so these are pre-re-fit validation scores about 0.013 below the final recipe, comparable to each other but not to Tab. 1. The per-modality projection uses $2\times$ the readout rate. The first row freezes the adapter (readout-only control).

LoRA lr	Readout lr	s06 (pre-re-fit)
0	2e-5	0.2818
1e-3	1e-4	0.3071
1e-3	3e-4	0.3131
6e-4	1e-3	0.3076
6e-4	3e-4	0.3134

against 0.00007 over four rank-16 runs). Rank 64 loses 0.0025 and one of its five seeds diverged; both rank-128 seeds diverged. We use rank 16, the smallest rank that no other rank beats beyond seed noise. Higher ranks were run at the learning rate tuned for rank 16 and may train better at a lower rate; on MING-FLASH-OMNI, for instance, rank 48 diverges at this rate and trains at 3×10^{-4}.

Tap layer (Tab. 6). A closed-form ridge readout on unadapted features peaks at layer 30, and the full recipe peaks at layer 34. The trained runs at layers 31–34 lie within 0.0006 of one another, inside the 0.00077 seed spread, so layer 32 is a good tap within a flat neighbourhood rather than a verified optimum. The frozen screen does locate the neighbourhood: it falls to 0.2682 at layer 16 and 0.2784 at layer 44. The neighbouring trained runs are included in the ensemble pool. On QWEN3.8-27B, where we swept the whole stack, the frozen screen ranked taps nearly opposite to trained runs, so we do not use a frozen screen alone to select taps.

Schedule length (Tab. 8). We train for exactly one epoch. A two-epoch run peaks at step 1,400, the end of the first pass over the 1,429 training windows, and stays below that peak throughout the second epoch. Two matched one- versus two-epoch pairs give -0.0147 and -0.0047. The training set already contains all of Friends seasons 1–5 and *movie10* (1,429 windows after removing windows that extend past their media or lack fMRI), so a second epoch only revisits the same windows. Alternating training and re-fitting also loses (-0.0017, both seeds), which indicates that the limit is the number of training windows. This axis was measured in the earlier configuration described below and was not repeated at the 60 s lead.

Learning rates (Tab. 7). The per-modality projection is held at twice the readout rate, so the grid is two-dimensional. The readout rate has an interior optimum at 3×10^{-4}, which beats 1×10^{-4} by 0.0060 at LoRA lr 1×10^{-3} and 1×10^{-3} by 0.0058 at LoRA lr 6×10^{-4}. The LoRA rate is flat between 6×10^{-4} and 1×10^{-3} at this readout rate (0.3134 against 0.3131, with a two-seed spread of 0.0004); we use 6×10^{-4}. Freezing the adapter and training only the readout gives 0.2818. These runs predate the context lead and the closed-form re-fit, so they are pre-re-fit scores about 0.013

Table 8: **A second epoch overfits.** Validation score through a 2858-step (two-epoch) run, same configuration as Tab. 7. The peak is at step 1400, the end of the first pass over the 1,429 training windows. Matched one- versus two-epoch pairs give -0.0147 at readout lr 3×10^{-4}, -0.0047 at readout lr 1×10^{-4}.

Step	200	400	600	800	1000	1200	1400	1600	1800	2000	2200	2400	2600	2800	2858
s06	0.2789	0.2856	0.2877	0.2921	0.2977	0.3007	**0.3052**	0.3009	0.2999	0.2996	0.3017	0.3019	0.3031	0.3030	0.3028

below the final configuration and are comparable only to one another. The rates were not re-tuned afterwards.

Interaction with the re-fit. The readout-rate effect (0.0058) is close to the value of the closed-form re-fit ($+0.0061$, §5.3). Part of what the readout rate controls is how far W drifts from the ridge optimum during training, and the re-fit removes that drift. After the re-fit, the same readout-rate comparison is $+0.00025$, and the rank-32 advantage shrinks in the same way. This is why the re-fit outweighs the training-side hyperparameters combined (§5.3).

F Is a single pass the same as re-encoding the prefix?

Single-pass extraction assumes that a state at time t depends only on the stimulus up to t. We test this directly. We encode a span once, then re-encode strict prefixes of the same span from the same start, so the shared tokens and their position ids are identical. If causal truncation held exactly, tap-layer states at the shared timesteps would agree.

As a control, the same span encoded twice agrees at cosine 1.0000 (minimum 0.99999) on all three streams, so the pipeline is deterministic. Cuts are placed on chunk boundaries, and the final chunk before each cut is excluded, because a truncated span ends with its own closing tokens where the longer span continues with content.

Table 9: Cosine between tap-layer states at shared timesteps, prefix pass against single pass, on *friends_s01e01a* with a 209 s span, tap 32.

Prefix ends at	audio	vision	text
140 s	0.9579	0.9895	0.9960
160 s	0.9569	0.9908	0.9970
180 s	0.9553	0.9901	0.9969
200 s	0.9584	0.9906	0.9973
same span twice	1.0000	1.0000	1.0000

The states differ by more than the control allows (Tab. 9). The location of the deviation shows that it is not future stimulus leaking backwards. The decoder is causally masked, the audio encoder attends within 8 s blocks, and the vision encoder attends within 1 s patches. A leak would therefore be confined to the chunk next to the cut, which is excluded. Instead, the *first* chunk deviates most: per 20 s chunk of the 140 s prefix, audio cosine is 0.919, 0.980, 0.979, 0.969, 0.958 and 0.942, from the start of the span onwards. The first chunk lies 120 s before the cut, where both passes see identical tokens. We attribute the deviation to length-dependent bf16 numerics: attention kernels tile different sequence lengths differently, and top-k expert routing can turn a rounding difference into a different expert.

A feature at t can thus depend on stimulus after t only up to the end of its 8 s audio block. The two extraction protocols differ mainly in which prefix they give the model, which is what the context-lead ladder measures. At matched media, replacing the sliding window with one whole-clip pass costs 0.0038 under the frozen MIRAGE head (Tab. 3), and the context lead recovers more than that. The ladder itself compares prefixes under one numerical regime, since train and evaluation spans have the same length throughout.

G ENSEMBLE ROSTER

The y axis of Fig. 5b is $-\log_{10}(m_\infty - r)$. Equal steps are equal fractions of the remaining gap, and $m(n) = m_\infty - An^{-b}$ is a line of slope b. A straight line therefore describes the rate of convergence, not unbounded growth. Here m_∞ is the limit of the *mean* subset, not a ceiling. Subsets scoring above it lie at $-\infty$ on this axis and are not shown; this happens from $n = 3$ on, and for 16% of subsets at $n = 20$. At $n = 20$ the 5th–95th percentile range is 0.3360–0.3409 around a mean of 0.3388.

Tab. 10 lists every model in the 261-member pool with the configuration that distinguishes it. All members share the training recipe of §4 unless a column says otherwise.

Table 10: **The 261-model pool.** Each row is a distinct configuration; members differing only in random seed are grouped, with the seed count given. Entries list only where a member departs from the shipped recipe (`Qwen3-Omni` backbone, linear readout, tap layer 32, rank 16, 60 s lead, 1,429 steps). Counts sum to 261.

Configuration	Members	Seeds
shipped recipe	34	15 seeds
rank 32, readout lr 0.0005	27	13 seeds
lead 30	16	8 seeds
backbone `Qwen3.8-27B`, chunk 2, no audio	10	6 seeds
readout cross-attn (tokens), tokens stream	8	3 seeds
bag 0.6	6	2 seeds
readout attn_pq	6	4 seeds
backbone `Gemma 3 27B`, tap 35, chunk 2, no audio	5	4 seeds
backbone `Qwen3.8-27B`, tap 36, chunk 2, no audio	5	5 seeds
rank 64, readout lr 0.0005	4	4 seeds
rank 8, lead 30	4	2 seeds
readout cross-attn (tokens), tokens raw	4	3 seeds
backbone `Ming-flash 104B`, readout cross-attn (tokens), tap 24, rank 48, LoRA lr 0.0003, scope attn_shared, tokens stream	3	3 seeds
backbone `Qwen3.8-27B`, rank 32, chunk 2, no audio	3	2 seeds
bag 0.4	3	1 seed
LoRA lr 2e-05, readout lr 1e-05, steps 715	2	2 seeds
backbone `Gemma 3 27B`, readout mlp, tap 35, chunk 2, no audio	2	2 seeds
backbone `LCO-Omni 7B`	2	1 seed
backbone `LCO-Omni 7B`, steps 1	2	1 seed
backbone `Ming-flash 104B`, tap 20, scope attn_shared	2	1 seed
backbone `Ming-flash 104B`, tap 24, rank 48, LoRA lr 0.0003, scope attn_shared	2	2 seeds
backbone `Ming-flash 104B`, tap 24, rank 48, LoRA lr 0.0003, scope attn_shared_router	2	1 seed
backbone `MiniCPM-o`	2	2 seeds
backbone `MiniCPM-o`, steps 1	2	1 seed
backbone `Nemotron Omni`	2	2 seeds
backbone `Qwen2.5-Omni 7B`	2	1 seed
backbone `Qwen2.5-Omni 7B`, steps 1	2	1 seed
backbone `Qwen3.8-27B`, no audio	2	2 seeds
backbone `Qwen3.8-27B`, tap 40, chunk 2, no audio	2	2 seeds
backbone `Qwen3.8-27B`, tap 44, chunk 2, no audio	2	2 seeds
backbone `Qwen3.8-27B`, tap 48, chunk 2, no audio	2	2 seeds
backbone `Qwen3.8-27B`, tap 52, chunk 2, no audio	2	2 seeds
backbone `Qwen3.8-27B`, tap 56, chunk 2, no audio	2	2 seeds
backbone `Qwen3.8-27B`, tap 60, chunk 2, no audio	2	2 seeds
backbone `Qwen3.8-27B`, tap 64, chunk 2, no audio	2	2 seeds
rank 32, lead 90, readout lr 0.0005	2	2 seeds
rank 32, readout lr 0.0005, chunk 2	2	2 seeds
readout cross-attn (layers)	2	1 seed
readout cross-attn (tokens), tokens stream_sp2x1	2	2 seeds
readout mlp	2	1 seed
steps 1466	2	2 seeds
LoRA lr 0, readout lr 0, steps 1	1	1 seed
LoRA lr 0.0001, readout lr 5e-05, steps 715	1	1 seed

continued on the next page

Table 10: *The 261-model pool, continued.*

Configuration	Members	Seeds
LoRA lr 0.0002, readout lr 0.0001, steps 715	1	1 seed
LoRA lr 5e-05, readout lr 2.5e-05, steps 715	1	1 seed
backbone `Gemma 3 27B`, tap 30, chunk 2, no audio	1	1 seed
backbone `Gemma 3n E4B`, readout cross-attn (tokens), tap 25, tokens stream	1	1 seed
backbone `Gemma 3n E4B`, readout cross-attn (tokens), tap 30, tokens stream	1	1 seed
backbone `Gemma 3n E4B`, tap 25	1	1 seed
backbone `Gemma 3n E4B`, tap 30	1	1 seed
backbone `LCO-Omni 7B`, readout cross-attn (tokens), tap 26, tokens stream	1	1 seed
backbone `LCO-Omni 7B`, tap 20	1	1 seed
backbone `LCO-Omni 7B`, tap 26	1	1 seed
backbone `LCO-Omni 7B`, tap 28	1	1 seed
backbone `Ming-flash 104B`	1	1 seed
backbone `Ming-flash 104B`, readout attn_branch, tap 24, rank 48, LoRA lr 0.0003, scope attn_shared	1	1 seed
backbone `Ming-flash 104B`, readout cross-attn (tokens), tap 16, tokens grid	1	1 seed
backbone `Ming-flash 104B`, readout cross-attn (tokens), tap 16, tokens raw	1	1 seed
backbone `Ming-flash 104B`, readout cross-attn (tokens), tap 16, tokens stream	1	1 seed
backbone `Ming-flash 104B`, readout cross-attn (tokens), tap 20, tokens stream	1	1 seed
backbone `Ming-flash 104B`, readout cross-attn (tokens), tap 24, rank 48, LoRA lr 0.0003, scope attn_shared, tokens grid	1	1 seed
backbone `Ming-flash 104B`, readout cross-attn (tokens), tap 24, tokens stream	1	1 seed
backbone `Ming-flash 104B`, tap 16	1	1 seed
backbone `Ming-flash 104B`, tap 16, steps 2858	1	1 seed
backbone `Ming-flash 104B`, tap 20	1	1 seed
backbone `Ming-flash 104B`, tap 20, LoRA lr 0.0002, scope attn_shared	1	1 seed
backbone `Ming-flash 104B`, tap 20, rank 48, LoRA lr 0.0002, scope attn_shared	1	1 seed
backbone `Ming-flash 104B`, tap 20, rank 48, LoRA lr 0.0003, scope attn_shared	1	1 seed
backbone `Ming-flash 104B`, tap 20, rank 48, LoRA lr 0.00045, scope attn_shared	1	1 seed
backbone `Ming-flash 104B`, tap 24	1	1 seed
backbone `Ming-flash 104B`, tap 24, scope attn_shared	1	1 seed
backbone `Ming-flash 104B`, tap 28	1	1 seed
backbone `MiniCPM-o`, readout attn_branch, tap 28	1	1 seed
backbone `MiniCPM-o`, readout cross-attn (tokens), tap 28, tokens stream	1	1 seed
backbone `MiniCPM-o`, tap 28	1	1 seed
backbone `Nemotron Omni`, readout attn_branch, tap 44	1	1 seed
backbone `Nemotron Omni`, readout cross-attn (tokens), tap 44, tokens stream	1	1 seed
backbone `Nemotron Omni`, steps 1	1	1 seed
backbone `Nemotron Omni`, tap 44	1	1 seed
backbone `Phi-4-MM`, readout attn_branch, tap 20	1	1 seed
backbone `Phi-4-MM`, readout cross-attn (tokens), tap 20, tokens stream	1	1 seed
backbone `Phi-4-MM`, tap 20	1	1 seed
backbone `Qwen2.5-Omni 3B`, readout attn_branch, tap 30	1	1 seed
backbone `Qwen2.5-Omni 3B`, readout cross-attn (tokens), tap 30, tokens stream	1	1 seed
backbone `Qwen2.5-Omni 3B`, tap 30	1	1 seed
backbone `Qwen2.5-Omni 7B`, readout attn_branch, tap 24	1	1 seed
backbone `Qwen2.5-Omni 7B`, readout cross-attn (tokens), tap 24, tokens stream	1	1 seed
backbone `Qwen2.5-Omni 7B`, tap 24	1	1 seed
backbone `Qwen3.8-27B`, LoRA lr 0, readout lr 0, chunk 2, no audio, steps 1	1	1 seed
backbone `Qwen3.8-27B`, LoRA lr 0.001, chunk 2, no audio	1	1 seed
backbone `Qwen3.8-27B`, chunk 2, no audio, scope attn	1	1 seed
backbone `Qwen3.8-27B`, chunk 2, no audio, steps 1	1	1 seed
bag 0.8	1	1 seed
chunk 10	1	1 seed
chunk 2	1	1 seed
chunk 2, no audio	1	1 seed
chunk 5	1	1 seed
lead 120	1	1 seed

continued on the next page

Table 11: **Three PEFT families at one protocol.** Friends s06 subject-mean after the closed-form re-fit, at the shipped configuration (QWEN3-OMNI, linear readout, tap 32, 60 s lead, 1,429 steps, one seed), with only the adaptation method changed. Rows are ordered by adapter budget; the scores do not follow it. *Params* counts backbone-side trainable parameters, read from each checkpoint; the 52.3M readout is common to every row and is excluded. *of LoRA* is the gain over the frozen backbone as a fraction of LoRA's, and the last column is that gain per million adapter parameters. IA^3 learns three element-wise rescaling vectors per block (keys, values, FFN intermediate) and adds no weights; a bottleneck adapter inserts $x + W_{\text{up}}\,\text{gelu}(W_{\text{down}}x)$ after the attention sub-layer and the FFN (Houlsby) or after the FFN alone (Pfeiffer). Each family runs at its own screened rate (1×10^{-2}, 3×10^{-4}, 6×10^{-4}), since a PEFT rate does not transfer between families.

Method	Params	s06	Δ frozen	of LoRA	Δ / M params
Frozen backbone (readout only)	0	0.2907	—	—	—
IA^3, attention + FFN	0.057M	0.3127	+0.0220	62%	0.3833
IA^3, + MoE experts	3.18M	0.3177	+0.0270	76%	0.0085
Bottleneck adapter, FFN only (r=32)	4.26M	0.3164	+0.0257	72%	0.0060
Bottleneck adapter, attention + FFN (r=32)	8.52M	0.3190	+0.0283	79%	0.0033
LoRA r=16 (shipped)	8.91M	0.3263	+0.0356	100%	0.0040

Table 10: *The 261-model pool, continued.*

Configuration	Members	Seeds
lead 45	1	1 seed
lead 75	1	1 seed
lead 90	1	1 seed
no audio	1	1 seed
rank 32	1	1 seed
rank 32, lead 120	1	1 seed
rank 32, lead 75, readout lr 0.0005	1	1 seed
readout cross-attn (tokens), tokens grid	1	1 seed
readout lr 0.0005	1	1 seed
tap 30	1	1 seed
tap 31	1	1 seed
tap 33	1	1 seed
tap 34	1	1 seed
tap 34, rank 32, readout lr 0.0005	1	1 seed
Total	261	

H ALTERNATIVE PEFT METHODS

All other adaptation results in this paper use LoRA. To test whether the gain depends on the method, we ran two further PEFT families under the same protocol: same backbone, tap layer, context lead, readout, readout learning rate, step budget and seed. Only the adaptation method changes, and all arms are compared with the same frozen control.

Methods. IA^3 (Liu et al., 2022) learns three element-wise rescaling vectors per decoder block, on the keys, the values and the FFN intermediate activations, and adds no weights. We run the standard attention + FFN scope and a variant that also rescales the intermediate activations of the MoE experts, which LoRA does not touch. Bottleneck adapters (Houlsby et al., 2019) insert $x \mapsto x + W_{\text{up}}\,\text{gelu}(W_{\text{down}}x)$ into the residual stream. Unlike LoRA, this update is nonlinear and cannot be merged into the frozen weights. We run the Houlsby placement (after both the attention sub-layer and the FFN) and the FFN-only placement of Pfeiffer et al. (2021), both at rank 32. At this rank the Houlsby adapter has $8{,}521{,}728$ parameters against LoRA's $8{,}912{,}896$, so the two are budget-matched.

Gain over the frozen backbone. All four non-LoRA arms beat the frozen backbone by margins tens of times the 0.00077 seed spread of this configuration. Each improves 998 of $1{,}000$ parcels

Table 12: **The nine omni-modal backbones.** Total parameters from each checkpoint's own index (QWEN3-OMNI and MING are mixture-of-experts, with 3B and 6B active per token); tap is the layer the readout reads, chosen per backbone. All arms share the recipe of §5.2: rank-16 LoRA, 60 s context lead, closed-form re-fit, Friends s06 subject-mean.

Backbone	Params	Tap	Frozen	+LoRA	Δ	+cross-attn
Phi-4-MM	4.7B	20	0.2521	0.2841	+0.0320	0.2888
Qwen2.5-Omni 3B	6.0B	30	0.2807	0.3139	+0.0331	0.3175
Gemma 3n E4B	7.8B	25	0.2631	0.3026	+0.0395	0.3049
LCO-Omni 7B	8.9B	26	0.2683	0.3199	+0.0517	0.3200
MiniCPM-o	9.4B	28	0.2594	0.3249	+0.0655	0.3265
Qwen2.5-Omni 7B	11.2B	24	0.2748	0.3178	+0.0431	0.3214
Qwen3-Omni 30B	31.7B	32	0.2907	0.3263	+0.0356	0.3310
Nemotron Omni	33.0B	44	0.2780	0.3227	+0.0447	0.3257
Ming-flash 104B	104.3B	24	0.2856	0.3227	+0.0371	0.3253

(pooled over subjects, paired block bootstrap against the frozen control) and every subject individually ($+0.021$ to $+0.031$). The benefit of adaptation is therefore not specific to LoRA, although LoRA gives the largest gain.

Budget-matched comparison. At 96% of LoRA's parameters, the Houlsby adapter recovers 79% of its gain, a deficit of 0.0073, about ten times the seed spread. Adding a nonlinearity and giving up mergeability does not help at matched budget. The FFN-only placement is within 0.0026 of the Houlsby placement at half the parameters, a difference its bootstrap does not resolve. This matches the redundancy between adapter sites reported by Pfeiffer et al. (2021).

Parameter count. Scores do not follow adapter size. The three middle arms span 3.18M to 8.52M parameters and lie within 0.0026 of one another, with overlapping bootstrap intervals, so we do not rank them. Going from the smallest non-LoRA adapter to the largest multiplies parameters by 149 and adds 0.0063, whereas LoRA exceeds the Houlsby adapter by 0.0073 at the same size. Per parameter, IA^3 is the most efficient: its $57{,}344$ coefficients (0.64% of the LoRA adapter) recover 62% of LoRA's gain, about $96\times$ the gain per parameter. This matters for the scale-out setting of §5.5, where the adapter is what is replicated per subject.

Learning rates. Each family runs at its own tuned rate: 1×10^{-2} for IA^3, 3×10^{-4} for the bottleneck adapter and 6×10^{-4} for LoRA. Both non-LoRA rates are interior optima of their grids (IA^3: six values from 3×10^{-4} to 1×10^{-1}; bottleneck: four values from 1×10^{-4} to 3×10^{-3}). With LoRA's rate, IA^3 would sit near the bottom of its curve. A plausible reason is numerical precision: a multiplicative rescaling smaller than $\sim 0.4\%$ cannot be represented in a bf16 activation, which pushes a multiplicative adapter to a high rate, while an additive update has no such floor. A PEFT comparison at one shared rate therefore partly measures the rate.

Step time. A smaller adapter does not train faster. The IA^3 arm takes 13.1 s per step and the $149\times$ larger Houlsby adapter 11.4 s, since step time is dominated by the backbone's forward and backward passes. A small adapter saves optimiser state and storage, not training time.

Limitations. Every non-LoRA arm uses a single seed. LoRA's lead over each of them is nine to eighteen times the fifteen-seed spread of the LoRA arm, but the ordering among the middle three is not established. All arms use the linear readout, and we do not claim the ordering holds under cross-attention. The non-LoRA arms and the frozen control ran in a different software environment from the LoRA arm (torch 2.8 against 2.9), with identical hyperparameters, readout initialization and re-fit. Comparisons among the non-LoRA arms and with the frozen control are within one environment; comparisons with LoRA cross environments, though the gaps are two orders of magnitude larger than the run-to-run jitter we measure within one.

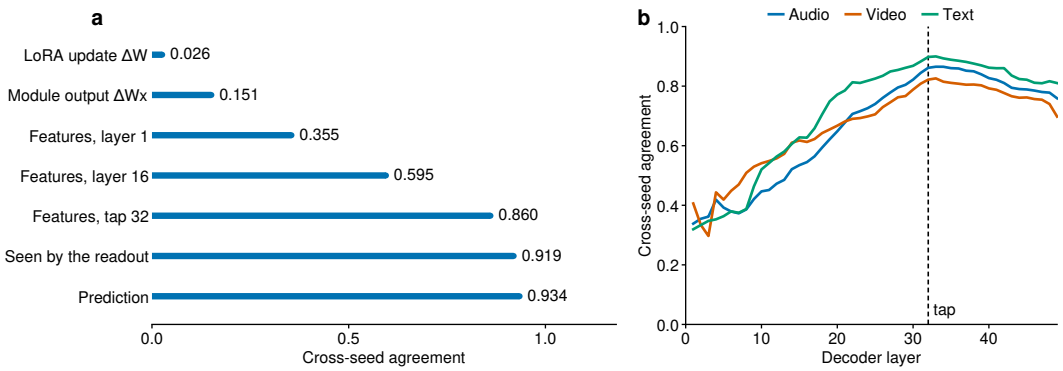

Figure 9: **Cross-seed agreement.** Mean agreement between adapters over the 15 seeds of the main configuration (105 pairs). *(a)* At successive levels of the model, from near zero for the weights to near one for the prediction. Levels below the first are cosines between the two seeds' *changes* relative to the frozen model, averaged over the audio, video and text streams. *(b)* Agreement at every decoder layer, per stream. It rises through the adapted stack, peaks at the tapped layer, and declines slowly through the 16 frozen layers above it.

I THE BACKBONE COMPARISON

Tab. 12 lists the arms behind Fig. 3. Each arm uses the recipe of §5.2 with only the backbone changed, at that backbone's own tap layer, chosen from a frozen tap screen. The frozen row is the same run with the adapter's learning rate set to zero. Its readout and per-modality projection train at the arm's own rates for the same 1,429 steps, followed by the same closed-form re-fit, so the frozen and adapted rows differ only in the adapter. Each backbone contributes one fixed run covering all four subjects, rather than its best-scoring run, because the number of runs differs widely across backbones.

QWEN3.8-27B and GEMMA-3-27B appear only in Tab. 2. They have no audio tower, so their two-modality scores are not comparable here: removing audio alone costs 0.0232 on QWEN3-OMNI (App. P).

J WHY SEEDS ARE NOT A SOURCE OF DIVERSITY

§5.5 reports that changing the seed gives almost no ensemble gain, whereas changing the readout or the backbone does. Using the 15 seeds of the main configuration, we show that adapters trained from different seeds compute nearly the same function, so their predictions have little independent content to average.

Weights. Updates from two seeds are nearly orthogonal in weight space: $\cos(\Delta W_i, \Delta W_j) = 0.026$, at a relative distance of $1.396 \approx \sqrt{2}$. This follows from the parameterization. $\Delta W = BA$ lies in the row space of A, which is drawn at random and changes little during training, so any two seeds are orthogonal regardless of what they compute. The update is not small: its largest singular value is 0.6–0.8 of the frozen matrix's. Adapters should therefore be compared on activations rather than weights.

Activations. Fig. 9 compares each seed's change relative to the frozen backbone at increasing depth. On real inputs, a single module's output change agrees at 0.151. The composed feature change agrees at 0.355 after one layer, 0.595 after sixteen, and 0.860 at the tap layer, where it is still 45% of the feature norm. Agreement falls back to 0.74–0.82 by the final layer (Fig. 9b). The rise is not explained by independent per-layer noise averaging out: fitted at layer 1, that model predicts 0.82–0.87 by layer 8, against an observed 0.43–0.54.

Readout and prediction. Pairing an adapter with another seed's jointly trained readout costs 0.0012, and with another seed's re-fit readout 0.0024, both inside the seed spread. Under one shared readout, the prediction change is 0.418 of the prediction norm and 93.4% correlated across seeds, with a shared

fraction of 0.939. Each seed's deviation from the consensus is therefore $0.418\sqrt{1-0.939} = 0.104$ of a prediction. The ensemble analysis measures this member-specific component independently, from the members' own predictions, and finds that it correlates with the fMRI target at -0.003.

The seed-specific part of a prediction is thus small and carries no signal, and it is what averaging over seeds removes. This is why seed-only pools saturate by $\approx$12 members while a member with a different readout or backbone still adds accuracy (App. G).

K Additional negative results

We summarize changes that did not improve on the recipe.

Gated-MLP readout. After the re-fit it scores $+0.0007$ over the linear readout, inside the 0.00077 seed spread.

Cross-attention over pooled features. Cross-attention over the same 894 frame-grid vectors that the linear readout uses gives $+0.00030$ [$+0.00017$, $+0.00045$], inside the seed spread. The gain of the cross-attention readout comes from the larger token set, not from attention itself.

Raw token sequence. Widening the token set from $\sim$3.2k to the full 19,099-token sequence, including every vision patch token, gives no gain ($+0.00000$, -0.00075 and $+0.00009$ over three seed-matched pairs).

Training-data subsets. Dropping whole stimuli so that two members share 44.1% of their training stimuli instead of 99.3% changes their residual correlation by only $+0.00006$. Training on random fractions of the data lowers accuracy monotonically (0.8/0.6/0.4 of the data $\rightarrow$ 0.3236/0.3161/0.3109).

Per-subject adapters. Restricting each adapter to one subject changes the score by -0.0019 [-0.0022, -0.0017] relative to the shared adapter, replicated at a second seed. A compute-matched control, with the same extra training but no subject restriction, loses by a similar amount, so the loss comes from continued training rather than from per-subject specialization.

L Is the adapter's gain stimulus-specific?

The frozen-backbone control of Tab. 2 shows that the adapter's $+0.0356$ does not come from readout training. It does not show that the gain depends on the stimulus. An adapter could in principle profit from any fMRI-shaped supervision, for example the parcels' spatial covariance, their autocorrelation or the per-run scale. We test this with a negative control: we train the same recipe on targets whose stimulus–response pairing is broken, then re-fit the readout in closed form on correctly aligned training data and score it on the usual validation windows.

Setup. For each subject, the training runs are concatenated in a seeded random order, rolled by a random offset (between 7.3 and 54.5 hours of scan time), and cut back into runs of their original lengths. Every target is still a real response of that subject. Window geometry, per-run z-scoring and within-run autocorrelation are preserved (mean lag-1 autocorrelation changes by less than 0.01), which makes this a stricter control than shuffling timepoints. Only the pairing with the stimulus is broken. Because each *movie10* film is scanned twice, we check that no shifted target comes from a stimulus span that overlaps the window the model encodes. The re-fit and all evaluations use aligned data only.

The shifted adapter arm also trains its per-modality projections on misaligned targets. Its matched control is therefore a second run at the same shift with the adapter's learning rate set to zero, so the two arms differ only in the adapter.

Results. Trained on shifted fMRI, the adapter does not just fail to help; it degrades the features. It scores 0.0733 below its matched control and loses on 996 of 1,000 parcels (Tab. 13), an effect about 95 times the seed spread of the aligned arm (0.00077). None of the $+0.0356$ survives once

Table 13: **Shifted-target negative control.** Friends s06 subject-mean after the closed-form re-fit on aligned training data, one run per arm. Differences are paired per parcel with a functional-block bootstrap interval; the last column counts the parcels on which the first arm scores higher.

Arm	Training targets	s06
LoRA + linear (ours)	aligned	0.3263
Frozen backbone	aligned	0.2907
Frozen backbone	shifted	0.2617
LoRA + linear	shifted	0.1874

Comparison	Δ [95% CI]	parcels
LoRA vs. frozen, both shifted	-0.0733 [-0.0799, -0.0675]	4 / 1,000
shifted LoRA vs. aligned frozen	-0.1032 [-0.1111, -0.0959]	1 / 1,000
shifted vs. aligned frozen	-0.0299 [-0.0319, -0.0280]	0 / 1,000
aligned vs. shifted LoRA	$+0.1389$ [$+0.1300$, $+0.1484$]	1,000 / 1,000

the stimulus–response pairing is removed, so the gain does not come from the marginal statistics of fMRI. The degradation is expected: an adapter that minimises squared error against targets unrelated to its input moves the backbone's features toward the noise, and a re-fit of the readout alone cannot undo that. Misaligned training of the projections alone has the same effect at a smaller size (-0.0299, on every parcel).

Sanity check. Before the re-fit, the shift-trained readouts score 0.0025 (adapter) and 0.029 (frozen) on aligned validation, down from 0.261 at step 0. Any stimulus-locked signal leaking into the shifted targets would have kept these scores above zero.

Scope. Because misaligned training is harmful, the shifted adapter arm does not estimate what the backbone is worth without a useful adapter; the aligned frozen row does. Each shifted arm is a single run at a single shift, which the effect size makes sufficient. The cross-attention readout was not tested.

M Does the adapter transfer to an unseen subject?

The shared adapter is trained on all four subjects, so the main results do not tell whether it learns about the stimulus or about these four people. We test this with leave-one-subject-out training. The shipped recipe is trained with the loss restricted to three subjects. The fourth subject's readout is then solved in closed form on the adapted features (the same re-fit as in §5.3), and that subject is compared against itself under the frozen backbone and under adapters that did see it. There are four arms, one per held-out subject, each with one seed and all other settings unchanged.

Table 14: **Transfer to an unseen subject.** Validation score per subject, with the readout solved in closed form, under: the frozen backbone; an adapter trained on the *other three* subjects only (bold, the transfer); the mean of the three 3-subject adapters that *did* train on the subject (in-sample); and the shared 4-subject adapter. The last three columns compare the held-out score with each of the others.

Held-out	frozen	*3-subj., held out*	3-subj., in-sample	4-subject	vs. frozen	vs. in-sample	vs. 4-subj.
sub-01	0.2875	**0.3210**	0.3215	0.3229	+0.0335	-0.0005	-0.0019
sub-02	0.2968	**0.3286**	0.3286	0.3301	+0.0317	-0.0000	-0.0015
sub-03	0.3130	**0.3485**	0.3490	0.3497	+0.0355	-0.0005	-0.0012
sub-05	0.2734	**0.3034**	0.3014	0.3025	+0.0301	+0.0021	+0.0010

Results. The adapter's gain transfers to an unseen subject (Tab. 14). A held-out subject gains $+0.0301$ to $+0.0355$ over the frozen backbone, on 995 to 998 of 1,000 parcels. The remaining deficit against the four-subject adapter (-0.0019 to $+0.0010$) reflects the amount of supervision rather than the subject being unseen. Each three-subject adapter also scores its three in-sample subjects 0.0007 to 0.0015 below the four-subject adapter (paired block bootstrap, all intervals clear of zero). Within

an arm, the held-out subject scores the same as the in-sample ones (-0.0005 to $+0.0021$; sub-05 is better when held out).

Two conclusions follow. Supervision from other people adapts the backbone as well as a subject's own data does. This is also consistent with the negative result for per-subject adapters (App. K): if three other subjects' data provide a subject's full gain, the subject's own data are not what the adapter learns from. This test does not cover transfer across scanners, protocols or datasets, which App. N addresses.

N DOES THE ADAPTER TRANSFER TO A SECOND DATASET?

Leaving a subject out (App. M) changes the person but keeps the site, scanner, preprocessing, repetition time and stimulus fixed. The Naturalistic Neuroimaging Database (NNDb; Aliko et al., 2020) changes all of them. Each participant watched one full-length feature film at a repetition time of 1 s. The data are distributed in volumetric MNI space, so they can be parcellated into the same 1,000 Schaefer parcels our readout predicts. We use the 20 participants who watched the same film (5,470 volumes each) and score the frozen and LoRA models of Tab. 1 without retraining the backbone or adapter.

Stimulus alignment. NNDb does not distribute the films, and our copy differs from the scanned one in rate and offset. The rate difference is a PAL transfer (a 23.976 fps master played at 25 fps), so local timestamps map to scanner time by the ratio $960/1001$. We estimate the offset from NNDb's word-level annotations of the film. We verify the alignment in four ways. Re-fitting the clock on the retimed stimulus gives a rate of 1.00002 and a residual offset of -0.075 s. The speech envelope of our media peaks in auditory cortex within 0.09 s of the lag at which NNDb's word annotations peak. Frame-to-frame visual change peaks in V1 and MT. Finally, a lag scan of both models' predictions against the brain peaks at zero.

Readouts. The *transferred* readout applies the Algonauts-trained weights unchanged, averaging the four subject-specific rows since NNDb participants have none. This matches the validation setting but tests two things at once: whether the features transfer and whether a readout fitted to other people does. The *re-fitted* readout separates the two. It keeps the backbone and projections from Algonauts and solves a fresh ridge readout per NNDb participant, with five-fold cross-validation over contiguous blocks of film time and the penalty chosen on a holdout inside each training fold. All NNDb scores are computed on the model's 1.49 s grid.

Noise ceiling. Correlations measured on noisier data shrink for both models by the same factor. A raw gain in r is therefore not comparable across datasets, and a relative gain can be inflated by noise. We report each model as a percentage of each dataset's own noise ceiling. The usual estimate, leave-one-out inter-subject correlation, correlates a subject with the mean of the others, and that mean is itself noisy. If a shared signal explains a fraction f of the variance, it converges to $f/\sqrt{f + (1-f)/k}$ with k other subjects, whereas an ideal stimulus model reaches $\sqrt{f}$. We invert the first expression for f per parcel and report against $\sqrt{f}$. The correction matters when k is small: against the raw estimate, the LoRA model would score 179% of the "ceiling" on Algonauts, where $k = 3$.

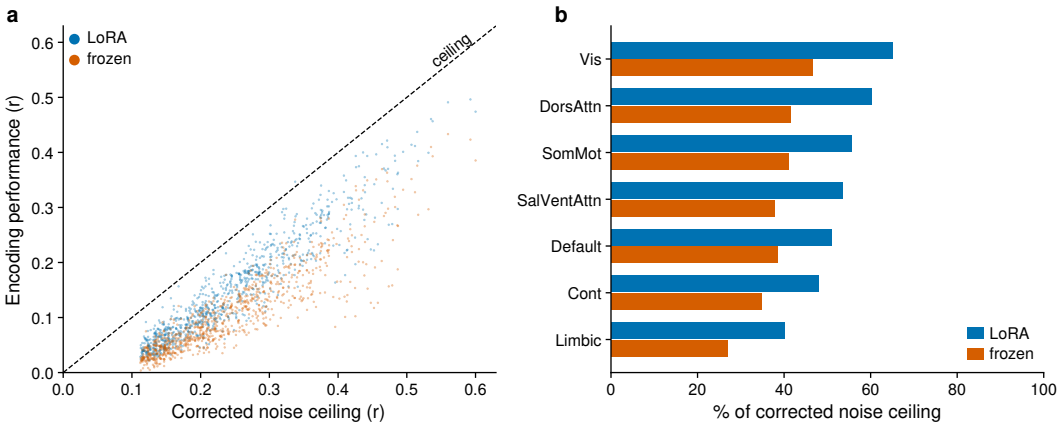

Figure 10: **Performance against the noise ceiling on NNDb.** 20 participants, readout re-fitted per participant on the fixed backbone, over the 816 parcels with an estimable ceiling. *(a)* Subject-mean r per parcel against the corrected noise ceiling (dashed line). *(b)* Mean percentage of the ceiling per Yeo network.

Table 15: **Transfer to a second dataset.** The same LoRA and frozen-backbone models on Algonauts validation and on NNDb (20 participants, one film; different site, scanner, preprocessing and TR). *Transferred* applies the Algonauts readout unchanged; *re-fitted* solves a fresh readout per NNDb participant under cross-validation over contiguous film time. r columns are subject means over all 1,000 parcels. Ceiling columns give r as a percentage of each dataset's noise ceiling, over the 922 (Algonauts) and 816 (NNDb) parcels with an estimable ceiling, with a 95% parcel-block bootstrap on the gain. Paired over the 800 parcels both datasets admit, transferred minus Algonauts is $+0.55$ $[+0.01, +1.06]$ points.

	subject-mean r			% of noise ceiling		
	frozen	LoRA	gain	frozen	LoRA	gain, points [95% CI]
Algonauts, validation	0.2907	0.3263	+0.0356	81.3	92.0	+10.7 [+10.0, +11.4]
NNDb, readout transferred	0.0991	0.1223	+0.0232	43.5	54.6	**+11.1** [+10.5, +11.6]
NNDb, readout re-fitted	0.0914	0.1252	+0.0338	40.3	55.7	+15.4 [+14.6, +16.2]

Transferred readout. The adapter's gain is $+11.1$ points of NNDb's ceiling, compared with $+10.7$ on Algonauts validation (Tab. 15). Paired over the parcels both datasets admit, the difference is $+0.55$ points $[+0.01, +1.06]$. The adapter improves 989 of 1,000 parcels and all 20 participants. In raw r the gain looks smaller ($+0.0232$ against $+0.0356$) and in relative terms larger (23% against 12%). Both differences reflect where each dataset's frozen baseline sits relative to its ceiling, not the adapter.

Re-fitted readout. The re-fitted readout shows a larger gain, $+15.4$ points of the ceiling and $+0.0338$ in r ($[+0.0293, +0.0386]$ by parcel-block bootstrap; 990 of 1,000 parcels, all 20 participants). The difference from the transferred readout lies almost entirely on the frozen side: re-fitting changes the LoRA model by $+0.0029$ and the frozen model by -0.0077. A readout solved on four fifths of one film is worse than the Algonauts readout for the frozen features but not for the adapted ones. This is not only a matter of data quantity. On a learning curve over training windows, the LoRA features with half of the data (0.105) already exceed the frozen features with all of it (0.091), and the gap widens with more data. Choosing the penalty per parcel instead of globally lowers both arms and leaves the gain at $+0.033$. We interpret the larger gain as the adapted features being easier to read out, not as a larger effect.

Spatial pattern. Across parcels, the gain map on NNDb correlates 0.620 with the gain map on Algonauts. Both maps are reliable (split-half reliability 0.891 on Algonauts and 0.940 on NNDb), giving a disattenuated correlation of 0.677. About 46% of the spatial pattern is therefore shared, and

the rest is dataset-specific structure rather than noise. At the network level (Fig. 10b), the adapter improves every network. The adapted model is closest to the ceiling in the visual and dorsal attention networks and furthest in the limbic network.

Scope. NNDb contributes one film, so effects specific to the dataset cannot be separated from effects specific to the film. Both datasets record fMRI during audiovisual narrative, so this shows transfer across datasets within that task. The extraction layer was chosen for the adapted model on Algonauts; if the frozen backbone's best layer differs, this affects the size of the gain on both datasets alike.

O WHERE IN THE BRAIN THE ADAPTER HELPS

The first two arms of Tab. 2 share a configuration and differ only in LoRA, so their per-parcel scores are paired and their difference isolates the adapter. The adapter adds $+0.0356$ on average and improves every one of the 1,000 parcels. The four subjects' gain maps agree at a mean pairwise r of 0.673, and the orderings below are unchanged when the LoRA map is averaged over 15 seeds. Tab. 16 and Fig. 11 break the scores down by the seven Yeo networks and by ten ROIs defined from parcel centroids.

Table 16: **Gain from the adapter by region.** Validation r with and without LoRA in the same configuration (the frozen arm trains the readout with the same rates and steps), over the n parcels of each region: the seven Yeo networks, then ten ROIs defined from parcel centroids. Arms are paired per parcel; gain intervals are functional-block bootstraps over parcels, and the last column is the gain relative to the frozen score. One run per arm; averaging the LoRA map over 15 seeds leaves both orderings unchanged.

region	n	frozen	LoRA	gain [95% CI]	rel. %
Yeo 7 networks					
DorsAttn	122	0.330	0.380	+0.0502 [+0.0468, +0.0533]	15
Vis	162	0.398	0.438	+0.0401 [+0.0364, +0.0440]	10
Cont	129	0.279	0.317	+0.0379 [+0.0347, +0.0412]	14
SalVentAttn	121	0.241	0.277	+0.0352 [+0.0319, +0.0397]	15
Default	212	0.318	0.351	+0.0330 [+0.0306, +0.0355]	10
SomMot	194	0.234	0.265	+0.0307 [+0.0288, +0.0336]	13
Limbic	60	0.131	0.146	+0.0147 [+0.0121, +0.0198]	11
coordinate-defined ROIs					
MT+ / lat. occipitotemporal	10	0.490	0.545	+0.0547 [+0.0468, +0.0598]	11
IPS / superior parietal	66	0.294	0.348	+0.0543 [+0.0499, +0.0587]	18
TPJ / supramarginal	46	0.321	0.370	+0.0494 [+0.0445, +0.0538]	15
early visual (V1/V2)	54	0.412	0.455	+0.0433 [+0.0401, +0.0493]	10
precuneus / PCC (DMN)	34	0.307	0.347	+0.0399 [+0.0340, +0.0436]	13
IFG / frontal operculum	59	0.260	0.295	+0.0350 [+0.0327, +0.0378]	13
somatomotor strip	61	0.171	0.202	+0.0316 [+0.0285, +0.0366]	19
early auditory (HG/PT/STG)	35	0.403	0.434	+0.0307 [+0.0275, +0.0322]	8
STS / lateral temporal	42	0.408	0.438	+0.0299 [+0.0278, +0.0326]	7
medial PFC (DMN)	39	0.194	0.218	+0.0246 [+0.0168, +0.0318]	13

Regional gains. Every region improves, with the largest gains in parietal and visual cortex. Among the networks, dorsal attention gains most ($+0.050$ on a frozen 0.330) and limbic least ($+0.015$ on 0.131). Among the ROIs, MT+ ($+0.055$), the intraparietal sulcus ($+0.054$) and the temporoparietal junction ($+0.049$) lead, and medial prefrontal cortex trails ($+0.025$). Relative to the frozen score, which varies threefold across regions, the ordering changes: the somatomotor strip and the intraparietal sulcus gain 19% and 18%, while MT+ and early visual cortex gain 11% and 10%.

Language regions. Early auditory cortex and the STS/lateral temporal cortex, where an adapted language model might be expected to help most, have the smallest relative gains: $+0.031$ (8% of a

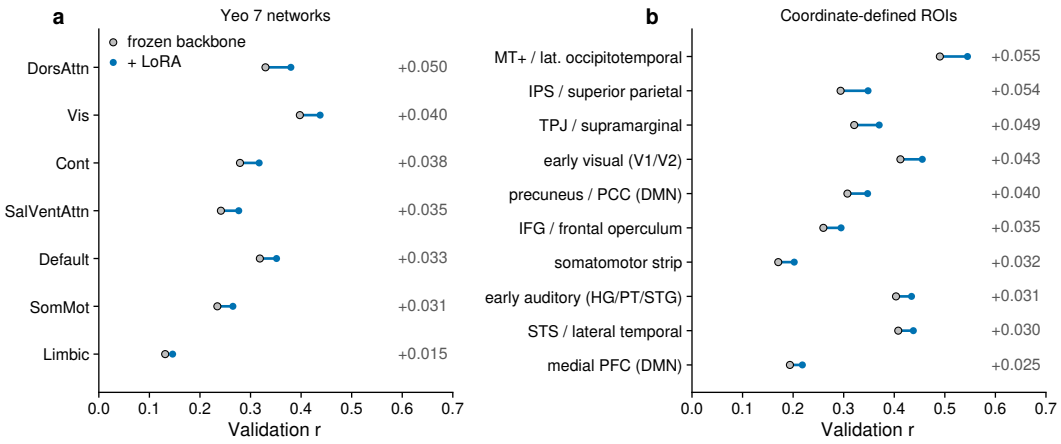

Figure 11: **Validation r by region, with and without the adapter.** Frozen backbone and the same configuration with LoRA, over the n parcels of each region (Tab. 16); bars show the gain, printed at the right. *(a)* The seven Yeo networks. *(b)* Ten coordinate-defined ROIs. One run per arm, paired per parcel; every gain's 95% functional-block bootstrap interval excludes zero.

frozen 0.403) and $+0.030$ (7% of 0.408). Both already have among the highest frozen scores. There is also no hemispheric asymmetry (LH $-$ RH -0.0010 $[-0.0030, +0.0008]$).

Reliability. Across all 1,000 parcels the gain correlates $+0.379$ with the frozen score, but only -0.016 across the 757 parcels whose *movie10* test–retest reliability exceeds 0.1. Poorly measured parcels have both low frozen scores and small gains, because measurement noise attenuates both. The gain is therefore not simply larger where the model was already good.

P Input-level modality attribution

The attribution of §5.4 is computed on the LoRA + linear model of Tab. 1 and on its frozen-backbone counterpart, over the same 250 validation windows. A modality is removed by replacing its input embeddings to the language model with their mean over training windows. For video, this covers the visual embeddings and their multi-scale copies (which would otherwise reintroduce video content in the first layers); for audio, the audio tower's output; for text, the transcript tokens, leaving template tokens untouched. Token counts, positions and timing are unchanged, so only stimulus content is removed. We evaluate all 2^3 subsets per window, which makes the three Shapley values exact. The full-subset predictions match the run's evaluation output exactly, and the Shapley values sum to the prediction (residual $< 10^{-12}$).

Modality subsets. Fig. 12 shows the subsets themselves in addition to the decomposition. The streams are largely redundant: video alone recovers 71% of the full score and audio alone 76%. Removing video or audio costs about the same (0.0675 and 0.0665), but after adaptation audio alone outscores video alone (0.2478 against 0.2303), whereas the frozen backbone ties them (0.1974 and 0.1975). Removing the transcript costs little (0.0103), although it reaches 0.0994 on its own, because the audio carries the same information.

Adaptation sharpens this. Removing the transcript costs 0.0164 with the frozen backbone and 0.0103 after LoRA. As single streams, audio gains $+0.0503$ from adaptation and video $+0.0329$, while the transcript loses 0.0065. The adapted model thus draws from the speech audio what the frozen model obtained from the transcript, matching the Shapley shares of §5.4.

These removal costs are not comparable to the *No audio* row of Tab. 2 (0.3026). That arm is trained without audio from the start, so its adapter and readout compensate for the missing stream, and it loses 0.0232 against its matched control. Removing audio from a model trained with it costs 0.0665.

Cross-stream reading. The readout takes one hidden state per stream every 0.5 s, at a video, an audio and a transcript token position. By tap layer 32 the decoder has mixed the sequence, so the content at a position need not be the content that entered there. Running the Shapley decomposition

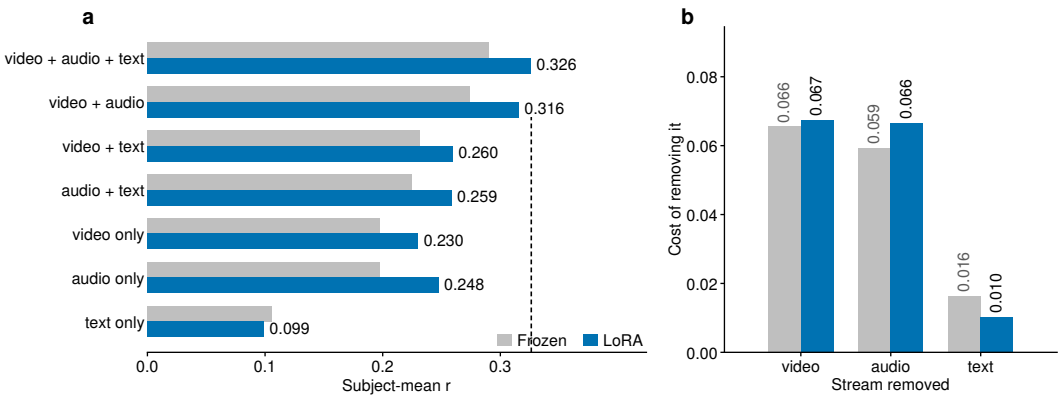

Figure 12: **All subsets of the three input streams.** Subject-mean Pearson r on the 250 validation windows used for attribution, with absent streams replaced by their training-mean embeddings. The top bar of *(a)* is the full model (dashed line). *(a)* Any single stream recovers most of the full score, and removing one costs far less than that stream achieves alone: the streams are largely redundant. *(b)* Cost of removing each stream, before and after adaptation. The video cost barely changes, the audio cost grows, and the transcript cost falls by more than a third. This removes a stream from a *trained* model and differs from the no-audio arm of Tab. 2, which is trained without audio throughout.

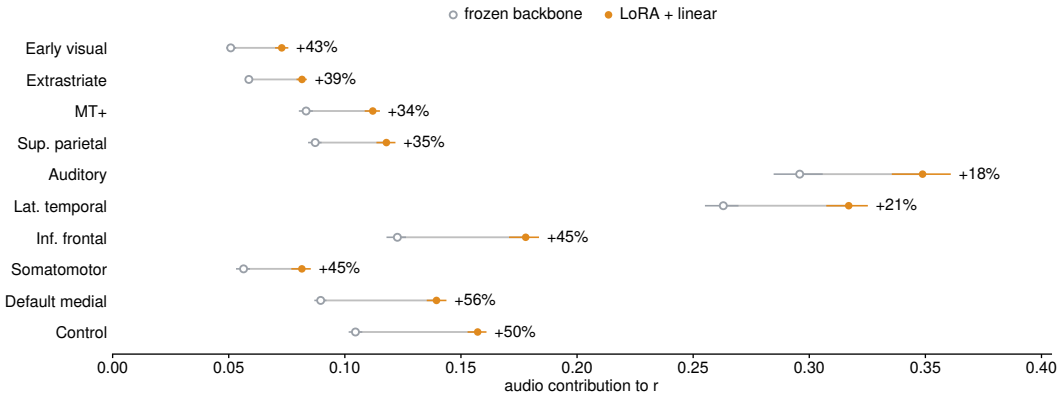

Figure 13: **Audio contribution by region, before and after adaptation.** Subject-mean contribution of the audio stream to encoding accuracy for the frozen and LoRA + linear models, with episode block-bootstrap intervals; labels give the relative change. Auditory cortex gains least; the gain is largest in regions that relied least on audio before adaptation.

within each position group gives the 3×3 flow of Fig. 4d, from stimulus modality (left) to the token type whose hidden state carries it (right). *Layout* is the contribution of the token arrangement with all content removed.

Adaptation nearly doubles cross-stream reading, from 8.9% [8.6, 9.2] of the prediction with the frozen backbone to 15.3% [14.9, 15.6] after LoRA (paired difference $+6.4$ points [$+6.1$, $+6.6$]). The flow is asymmetric: video content read at audio positions accounts for 7.5%, against 3.9% in the other direction (frozen: 5.0% and 1.1%). Audio positions carry 35.1% of the frozen prediction and 45.8% of the adapted one, while transcript positions fall from 22.5% to 14.2%. This is the same shift toward audio seen in cortex (Fig. 4c, Fig. 13), here at the token level. Text content read at audio positions rises from 0.5% to 2.0%, and audio read at text positions falls from 1.8% to 1.5%. Negative flows are clipped in the diagram; they amount to 0.2% of the frozen prediction and none of the adapted one. The percentages in the text are exact signed means.

Statistics. We report two statistics per parcel. The *share* $\mathrm{cov}(\phi_m, \hat{y})/\mathrm{var}(\hat{y})$ describes what the prediction is made of and sums to one. The *contribution to accuracy* $\mathrm{cov}(\phi_m, y)/(\sigma_{\hat{y}}\sigma_y)$ describes which part of it tracks the response and sums to r. The text quotes shares; maps meant to localise use the contribution, since a share is defined even where the model predicts poorly. Intervals are block bootstraps over validation episodes.

Robustness. As an alternative baseline, we replace an absent modality's embeddings with those of a window from a different episode, which is in distribution per token but wrong in content. On the LoRA + linear model the two baselines give share maps correlated at 0.994 and agree on the dominant modality for 98.2% of parcels.

The frozen-to-adapted comparison of §5.4 is between two models, so we compare it with the agreement between two adapted seeds. Maps are compared over the 971 parcels that both models predict at $r \geq 0.1$. Two adapted seeds agree at 0.996–1.000 on the contribution maps and on the dominant modality for 97.2% of parcels. The frozen and adapted models agree at only 0.963–0.990 and on 74.4% of parcels, outside the seed range with bootstrap intervals clear of it. With the LoRA + cross-attention model, the dominant modality changes on fewer parcels (8.0% against 25.6%) but in the same direction, and cross-stream reading rises from 11.9% [11.5, 12.2] to 15.8% [15.4, 16.3]. The reallocation is therefore a property of the adaptation, not of the readout.

Q WHAT THE POOL STILL MISSES

A validation correlation measures how good a model is, but not what it misses. A residual mixes model error, missing signal and measurement noise, and the validation season has no test-retest to separate them. Two properties of our setup allow the separation. First, the member-specific part of each member's prediction is noise (it correlates with the target at -0.003, App. J), so the *uniform* mean of the 261 members is the pool's shared function with that noise averaged out. Second, all four subjects watched the same 250 validation windows. Scanner and idiosyncratic noise cannot correlate across people, so a residual that is correlated *across subjects* is stimulus-locked signal that the pool does not produce.

For subject s and parcel p, let y be the z-scored fMRI, $\hat{y}$ the pool mean rescaled by its least-squares gain, $e = y - \hat{y}$, and t the mean of the other three subjects, over the 25,000 validation TRs. Then

$$\underbrace{\mathrm{cov}(y_s, y_t)}_{\text{ISC}} = \underbrace{\mathrm{cov}(\hat{y}_s, \hat{y}_t)}_{\text{explained}} + \underbrace{\mathrm{cov}(e_s, e_t)}_{\text{missed}} + \underbrace{\mathrm{cov}(\hat{y}_s, e_t) + \mathrm{cov}(e_s, \hat{y}_t)}_{\text{cross}}.$$

The left side is the inter-subject covariance, the stimulus-locked variance available to any stimulus-computable model. The "missed" term is the part of it the pool does not explain. The cross term is 1.3% of ISC.

Table 17: **What the pool still misses.** The cross-subject partition of the stimulus-locked (inter-subject) covariance on the validation episodes, as a percentage of it: the part shared by the predictions (explained), the part shared by the residuals (missed) and the cross terms, for three pools; the interval is a functional-block bootstrap over parcels, and the last column is the missed covariance in units of a unit-variance parcel.

pool	val. r	explained %	**missed %**	cross %	missed
shipped single model (LoRA + linear)	0.3263	60.5	36.2	3.3	0.0480
3-member mean	0.3262	60.8	36.5	2.6	0.0485
261-member uniform mean	0.3401	66.2	32.5 [30.0, 35.3]	1.3	0.0431

Results. The pool leaves 32.5% [30.0, 35.3] of the cross-subject stimulus-locked variance unexplained (Tab. 17). A single model leaves 36.2%, so averaging 261 models recovers 3.7 points. The four subjects agree to within 1.4 points. In variance units the pool misses 0.043 of a unit-variance parcel, against an ISC of 0.133.

This is a lower bound on what is missed relative to all stimulus-driven variance, because ISC does not count stimulus-driven signal that is subject-specific. Within a subject the pool explains variance

equal to 94% of ISC. Only 66 points of this are shared across subjects' predictions; the other 28 are subject-specific stimulus-driven signal. Measured against captured plus missed variance, the missed share is 26%.

The same pool reaches 97% of the ISC ceiling in correlation ($\sqrt{\mathrm{ISC}/\mathrm{var}}$), which corresponds to 66% in variance. Distance to the ceiling is therefore better stated in variance.

Characterising the residual. Nine low-level stimulus statistics (luminance, contrast, motion energy and cuts from the frames; RMS, envelope modulation and spectral centroid from the audio; words per TR and speech on/off from the transcript), lagged 0–9 s, explain the shared residual at only $r = 0.027$ under episode-level cross-validation. The same features explain the pool's prediction at 0.48, so the pool already carries this information. Lagged copies of the pool's own prediction (-4 to $+4$ TRs) explain 0.048, so the residual is not a latency error. It is also not low-dimensional: the top ten components hold 36% of its variance, against 91% for the prediction. Its spectrum follows that of the ISC (66% of the missed covariance lies below 0.05 Hz, against 70% of ISC). The residual is thus slow, distributed, high-dimensional and stimulus-locked, which fits narrative or event-level structure better than a missing low-level feature. This agrees with §5.5: even the levers that changed the pool's account of the stimulus left about a third unexplained.

R VALIDATION-TO-HELD-OUT TRANSFER

Transfer laws. We fit the relation between validation and held-out scores on twelve submissions (three single models and nine ensembles, spanning 0.061 of validation score). Validation predicts both held-out benchmarks closely, but with different functional forms. Friends S7 is proportional to validation (slope 0.985, R^2 0.999, leave-one-out residual 0.00056), so in-distribution gains transfer about one for one. The out-of-distribution films score validation *minus a constant* 0.102 (R^2 0.998, leave-one-out residual 0.00095): gains again transfer one for one, with a fixed penalty. An affine fit has the best in-sample residual but the worst leave-one-out residual on both targets.

Out-of-sample checks. Three models were submitted after the fit was fixed.

- *261-model pool* (validation 0.34834), just above the fitted range. In distribution it predicts 0.34304 against 0.34277 measured (0.5 leave-one-out residual s.d.). Out of distribution it predicts 0.24658 against 0.24827 measured (1.8 s.d.).
- *LoRA + MIRAGE head* (validation 0.32143), which pairs our adapter with the published attention readout and is the only held-out model whose readout is neither linear nor token cross-attention. Out of distribution it lands within tolerance (0.21837 measured, 0.21963 predicted, 1.3 s.d.). In distribution it scores 0.31337 against 0.31654 predicted, 5.7 s.d. low.
- *Frozen backbone with the cross-attention readout* (validation 0.31212), a configuration the fit did not contain but a readout family it did. It predicts 0.30737 in distribution against 0.30708 measured (0.5 s.d.), and 0.21036 out of distribution against 0.21159 measured (1.3 s.d.).

The laws therefore hold for new configurations within the fitted readout families. The proportional S7 law may not hold across readout families, and we do not use it that way.

Resolution floor. All three functional forms agree that a validation difference below ~ 0.001 cannot be ranked in distribution.

S REPRODUCIBILITY: THE EXACT CONFIGURATION

Hardware and precision. One NVIDIA A800-80GB per run, for training and extraction. The backbone runs in bfloat16 with FlashAttention-2 and gradient checkpointing. The vision patch embedding's 3-D convolution runs in float32, because cuDNN 9 executes it in bfloat16 roughly four orders of magnitude slower.

Table 18: **Lag blocks.** Each block is a (delay, width) pair in seconds. Block b of TR j uniformly averages the feature frames in $[c_j - (\delta_b + w_b)f, c_j - \delta_b f)$, where c_j is the TR's block centre and $f = 2$ Hz. A block that reaches before the window start is clipped, or is a zero row if it lies entirely before it. The last three columns count, over the 100 TRs of a window, the rows at full width, clipped, and zero (158 of 800 rows are zero).

Block	δ_b (s)	w_b (s)	covers (s)	full	clipped	zero
1	0.5	2.0	$[0.5, 2.5)$	98	1	1
2	2.5	2.0	$[2.5, 4.5)$	99	1	0
3	4.5	4.0	$[4.5, 8.5)$	97	2	1
4	8.5	8.0	$[8.5, 16.5)$	92	5	3
5	16.5	16.0	$[16.5, 32.5)$	81	10	9
6	32.5	32.0	$[32.5, 64.5)$	60	21	19
7	64.5	64.0	$[64.5, 128.5)$	17	42	41
8	128.5	64.0	$[128.5, 192.5)$	0	16	84

Optimisation. AdamW with default β, weight decay 0.01, gradient clipping at global norm 1.0. Batch size 1 window, no gradient accumulation; one window is 100 TRs for all four subjects, so a step carries 400 target rows. 1,429 steps (one epoch over the 1,429 training windows), 30 warm-up steps, cosine decay to zero. Learning rates: LoRA 6×10^{-4}, per-modality projection 6×10^{-4}, readout 3×10^{-4}. No run is resumed from a checkpoint.

LoRA. Rank 16, $\alpha = 32$, dropout 0, on `q_proj`, `k_proj`, `v_proj` and `o_proj` of every decoder layer (`model.layers.\d+.self_attn.(q|k|v|o)_proj`). PEFT default initialisation (A Kaiming-uniform, B zero), so training starts from the frozen model. 8.91M parameters. The decoder is truncated above the tap layer, so layers past 32 are neither run nor adapted.

Full finetuning. The full-finetuning arms of §5.2 train all 20.25B parameters below the tap with the settings above, except the backbone learning rate. We screened 3×10^{-6} to 1×10^{-4} in half-decades (constant schedule, 200 steps, 42-window selection subset). 1×10^{-5} was best (0.2753, against 0.2684 for the LoRA reference); 3×10^{-6} was still improving and the two higher rates stalled or degraded. A step takes 19.8 s against 12.0 s for LoRA. Both arms are single-seed, and the cross-attention arm reuses the linear arm's rate.

Clock and lag blocks. $\mathrm{TR} = 1.49$ s and the feature grid is 2 Hz, so 100 TRs is 149 s and 298 frames. Hemodynamic alignment uses MIRAGE's fixed shift of 4.47 s, i.e. an offset of $\mathrm{round}(4.47 \times 2) = 9$ frames: TR j's block centre is at frame $c_j = 9 + 2.98j$. Each of the 8 blocks is a (delay, width) pair (δ_b, w_b) in seconds, and block b of TR j averages frames $[\lceil c_j - (\delta_b + w_b)\, f \rceil, \lceil c_j - \delta_b f \rceil)$, clipped to the window, with a zero row where the block lies entirely before the window. The operator is one dense matrix shared by the training path and the closed-form solve.

Tab. 18 lists the eight pairs. They tile $[0.5, 192.5)$ s contiguously with widths $2, 2, 4, 8, 16, 32, 64, 64$ s. Rebuilding the operator from these pairs reproduces the 800×298 matrix in the released checkpoint exactly. Because the deepest block reaches 192.5 s back, beyond the 149 s window, 158 of the 800 design rows (19.8%) are zero and block 8 never reaches full width. Every non-zero row sums to one.

Readout. Per-frame linear projection $2048 \to 512$ per modality (shared across frames), then the lag average, giving $8 \times 3 \times 512 = 12{,}288$ features per TR, then a per-subject linear map to 1000 parcels. Projecting before lag-averaging keeps empty blocks exactly zero. 52.30M parameters. Registered buffers (lag operator and centring statistics, 0.26M) are fixed and not counted.

Cross-attention readout. Replaces the lag operator with per-TR cross-attention over the language-model token sequence: 8 learned queries per modality, key/query width 128, a causal cutoff at each TR's block centre, and a per-block additive bias (floored at -12) initialised to reproduce the lag operator's uniform weighting. It reads the `stream` token set, about 3.2k tokens per span: audio at its native 13 Hz, all text tokens, and one mean-pooled vector per second of video. It adds 0.2M parameters and uses the same schedule. A stream with no token before a TR's cutoff is masked explicitly.

Table 19: **Backbone revisions.** Hugging Face repository and commit (12 hex digits) for every backbone in Tab. 12, as loaded by our runs.

Backbone	Hugging Face repository	Revision
Phi-4-MM	`microsoft/Phi-4-multimodal-instruct`	`467a5d77dcfa`
Qwen2.5-Omni 3B	`Qwen/Qwen2.5-Omni-3B`	`f75b40e3da20`
Gemma 3n E4B	`unsloth/gemma-3n-E4B-it`	`45e9fb1dd0e3`
LCO-Omni 7B	`LCO-Embedding/LCO-Embedding-Omni-7B`	`3d38f58aae12`
MiniCPM-o	`openbmb/MiniCPM-o-4_5`	`503e754207c9`
Qwen2.5-Omni 7B	`Qwen/Qwen2.5-Omni-7B`	`ae9e1690543f`
Qwen3-Omni 30B	`Qwen/Qwen3-Omni-30B-A3B-Thinking`	`2f443cfc4c54`
Nemotron Omni	`nvidia/Nemotron-3-Nano-Omni-30B-A3B-Reasoning-BF16`	`e5e9932441de`
Ming-flash 104B	`inclusionAI/Ming-flash-omni-2.0`	`6a2e1dec0706`

Ridge penalties. $\{10^3, 10^4, 10^5, 10^6, 10^7\}$, chosen per parcel and subject on the training-split holdout of App. D. For the shipped model 3,829 of 4,000 parcel-subject pairs select 10^4, 110 select 10^5, 60 select 10^3 and one selects 10^6.

Ensemble combiner. Per-parcel minimum-variance weights (Perrone & Cooper, 1995) on the members' residual covariance S: $w = S^{-1}\mathbf{1}/(\mathbf{1}^\top S^{-1}\mathbf{1})$, the minimiser of $w^\top S w$ subject to $\mathbf{1}^\top w = 1$. The constraint is free because per-parcel Pearson is invariant to positive rescaling. Weights may be negative. S is shrunk toward its diagonal with $\lambda \in \{0.01, 0.03, 0.1, 0.3, 0.6, 0.9\}$, chosen on an inner episode split, never on the rows the weights are scored on. The outer split is in App. D.

Backbone revisions. Tab. 19 gives the Hugging Face repository and commit for every backbone in Tab. 12. Four ports pin a commit in code; for the other five we record the commit the hub resolved at run time. For PHI-4-MM and LCO-EMBEDDING-OMNI, upstream `main` has since moved past the commit we used.

What a rerun needs. The stimulus media cache at 336×224, the Algonauts 2025 fMRI, the backbone weights at the revisions in Tab. 19, and a readout initialisation: a ridge solution on frozen features at the same context lead, with the lag basis of Tab. 18. Starting a lead run from a no-lead initialisation costs about 0.004. Stage 1 is the 1,429-step run above; stage 2 is one forward-only pass that re-solves the readout in closed form.