REPRESENTATION OVER CAPACITY: SPARSE TEMPORAL ACCESS FOR FUNCTIONAL CONNECTIVITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Resting-state fMRI is inherently temporal, yet most brain-network models operate on a static summary of the signal. Standard functional connectivity (FC) compresses an entire BOLD scan into a single matrix, averaging away how interactions between brain regions evolve over time. We argue that this creates a representational bottleneck: once temporal variation is discarded, increasing downstream model capacity cannot recover it. We introduce **TempLasso**, a deliberately low-capacity temporal FC model that selects a small set of discriminative connections, tracks them across local FC windows, and models their trajectories with a lightweight BiGRU. Across three rs-fMRI cohorts and five static FC backbones, TempLasso outperforms 13 of 15 substantially more complex static models. When used as a plug-and-play temporal branch, **X+TempLasso** improves all 15 backbone–dataset combinations, with an average gain of $+4.63$ AUC points. In contrast, directly applying the full static backbone to every temporal window is markedly less reliable, showing that greater per-window model capacity does not necessarily translate into better temporal modeling. A complementary theoretical analysis formalizes static FC as a many-to-one temporal compression: Bayes-optimal prediction with temporal access cannot be worse than prediction restricted to static FC, and can be strictly better when discarded temporal structure is task-relevant. Together, these results suggest that progress in brain-network learning may depend less on increasingly elaborate models of a static connectome and more on preserving and exploiting the temporal structure already present in the signal.

1 INTRODUCTION

Resting-state fMRI is a temporal measurement, yet most brain-network learning begins only after time has been largely removed. A standard pipeline compresses an entire BOLD scan into a single functional-connectivity (FC) matrix, and increasingly sophisticated models are then built on top of this static representation. This paradigm has produced connectome-specific CNNs (Kawahara et al., 2017), graph neural networks (Parisot et al., 2017; Li et al., 2021), graph Transformers (Kan et al., 2022; Yu et al., 2024; Peng et al., 2025), and quadratic networks designed specifically for dense FC matrices (Yang et al., 2025). The architectures have changed dramatically; the representation has not. In most cases, one subject is still reduced to one connectome before learning begins.

We revisit this design choice. Functional coupling between brain regions varies over the course of a scan (Hutchison et al., 2013; Preti et al., 2017; Saggar et al., 2018). Two subjects can therefore have similar full-scan FC matrices while exhibiting substantially different connectivity trajectories. Once this temporal variation is collapsed into a single full-scan FC matrix, no increase in downstream model capacity can reconstruct distinctions that have already been discarded. This suggests a different route to progress: rather than asking only how to build a better model for static FC, we should ask whether we are giving the model the right representation in the first place.

Dynamic functional connectivity provides a natural way to preserve this missing information by replacing the full-scan matrix with a sequence of local connectivity states, $\mathbf{C}_1, \ldots, \mathbf{C}_L$ (Allen et al., 2014a). Existing approaches such as STAGIN (Kim et al., 2021) and T3former (Uddin et al., 2026) show that temporal brain-network structure can be useful, but typically address it with dedicated spatiotemporal architectures. We ask a deliberately simpler question:

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.

How much of the apparent need for complex FC models comes from modeling capacity, and how much comes from information already lost by static aggregation?

To answer this, we introduce **TempLasso**, a deliberately low-capacity temporal FC model built around a strong information bottleneck. Using training data only, TempLasso identifies a small set of discriminative functional connections, follows those same connections through a sequence of local FC matrices, and models their trajectories with a compact BiGRU. There is no sophisticated spatial encoder, graph propagation, or connectome-specific attention mechanism. The model simply determines *which connections to track* and then learns *how they evolve*.

This simplicity is intentional. It provides a controlled test of whether preserving temporal information can matter more than increasing model complexity. Despite its restricted capacity, TempLasso is highly competitive with, and often outperforms, substantially more complex static FC models. The result points to a representational bottleneck: a simple temporal model can exploit predictive structure obscured by static aggregation.

Static and temporal FC are also complementary rather than competing representations. The full-scan matrix provides a stable view of persistent connectivity, while TempLasso captures within-scan variation. We therefore use TempLasso as a plug-and-play temporal branch for any static FC backbone **X**, yielding **X+TempLasso** hrough lightweight static–temporal fusion. The static model is left unchanged, and the temporal branch is lightweight: it operates only on a small set of selected connection trajectories rather than repeatedly evaluating the full backbone on every temporal window. Across all 15 evaluated backbone–dataset combinations, this simple augmentation improves the corresponding static model.

A natural alternative is to temporalize the full static model itself by applying its encoder to every local FC matrix. We denote this higher-capacity alternative by **TempX**. Surprisingly, TempX is substantially less reliable: standalone TempX often underperforms the corresponding static model, while $\mathbf{X}+\mathrm{TempX}$ produces architecture- and dataset-dependent gains. Thus, temporal access alone is not enough; how temporal information is represented matters. Sparse, task-guided temporal tracking provides a more reliable complement to the full-scan connectome than simply increasing per-window model capacity.

This observation leads to the broader message of the paper: *for functional connectivity, preserving the right information may be more important than making the model more complex.* We formalize this intuition by showing that static FC is a many-to-one compression of the temporal signal: Bayes-optimal prediction with temporal access cannot be worse than prediction restricted to static FC and can be strictly better when discarded temporal structure changes the optimal decision. Our controlled experiments then test the practical consequences of this gap through windowing sensitivity, order-free aggregation, shuffled-sequence controls, and direct full-backbone temporalization.

We evaluate TempLasso on three public rs-fMRI cohorts spanning autism spectrum disorder, Alzheimer's disease, and Parkinson's disease, and compare against convolutional, Transformer, graph-based, and quadratic FC models. TempLasso matches or exceeds 13 of the 15 static models evaluated despite its restricted capacity. More strikingly, **X+TempLasso** improves all 15 backbone–dataset combinations over their corresponding static models, with a mean gain of $+4.63$ AUC points (sign test, $p < 0.001$).

Contributions.

- **TempLasso.** A sparse, low-capacity temporal FC model that tracks selected connections through time with a lightweight BiGRU, matching or outperforming 13 of 15 substantially more complex static models.
- **X+TempLasso.** A plug-and-play temporal branch that improves all 15 evaluated static backbone–dataset combinations through lightweight static–temporal fusion at low additional cost.
- **Complexity is not enough.** Directly temporalizing the full backbone (**TempX**) is markedly less reliable, showing that richer per-window encoding need not make better use of temporal information.
- **Theory and controls.** We formalize static FC as a many-to-one temporal compression, showing that Bayes-optimal risk with temporal access cannot exceed that of static FC and can be strictly lower. Controlled experiments isolate the value of temporal structure.

2 Related Work

Learning from Static Functional Connectivity. Resting-state fMRI is commonly summarized by a functional-connectivity matrix $\mathbf{C} \in \mathbb{R}^{N \times N}$, whose entries are correlations between ROI BOLD signals. Existing models process this dense connectome in several ways: as an image with specialized convolutions (Kawahara et al., 2017), as a graph with message passing (Parisot et al., 2017; Li et al., 2021), or as a set of ROI tokens with Transformer-style architectures (Kan et al., 2022; Yu et al., 2024; Peng et al., 2025). Most recently, Brain Quadratic Networks (BQN) revisit this design space and show that a Hadamard-product quadratic model outperforms message passing on the dense FC matrix altogether, suggesting that the field's growing architectural complexity is not obviously necessary (Yang et al., 2025).

Despite their different inductive biases, these methods share the same basic interface: a model $\mathbf{X}$ maps a single static connectivity matrix to a representation $\phi_X(\mathbf{C})$. Our framework treats this mapping as a fixed, reusable component rather than a target for redesign, and asks instead what can be added to it.

Deep Learning for Dynamic Functional Connectivity. Static FC summarizes connectivity over the full scan and therefore removes explicit information about how functional coupling changes over time (Hutchison et al., 2013; Preti et al., 2017; Saggar et al., 2018). Sliding-window dynamic FC addresses this by estimating a sequence of local connectivity matrices $\mathbf{C}_1, \ldots, \mathbf{C}_L$ rather than a single full-scan summary (Allen et al., 2014a).

Methods that learn directly from this sequence largely follow one recipe: pair a spatial encoder, applied per window, with a temporal aggregator. STAGIN applies a graph encoder to each window and aggregates the resulting representations with spatio-temporal attention (Kim et al., 2021); T3former extracts topological and spectral descriptors per window and fuses them via a Descriptor-Attention transformer, evaluated on brain, social, and traffic temporal graphs alike (Uddin et al., 2026). State-space architectures have recently been adopted for the same recipe: FST-Mamba applies hierarchical Mamba encoders separately along the spatial and temporal axes of the dynamic FNC matrix (Wei et al., 2025), CP-SSM combines a Mamba block with a core-periphery-guided mixture-of-experts for connectome classification (Chen et al., 2025a), and dFCExpert learns dFC representations with modularity and state experts built on GNNs and transformers/RNNs (Chen et al., 2025b). Closest in spirit to our own model, a convolutional bidirectional GRU has been applied directly to windowed dFC matrices for disease classification (Ji et al., 2024). These methods generally increase modeling capacity along the spatial axis, the temporal axis, or both; they do not directly isolate whether a constrained, low-capacity temporal representation can be sufficient on its own.

Self-Supervised Foundation Models for BOLD Dynamics. A separate line of work scales capacity along a different axis entirely: rather than hand-designing a spatial or temporal encoder, these methods pretrain large backbones directly on raw BOLD time series across tens of thousands of subjects and fine-tune for downstream prediction. BrainLM pretrains a Transformer via spatiotemporal masked reconstruction on fMRI recordings (Ortega Caro et al., 2024); SwiFT applies 4D windowed self-attention directly to raw fMRI volumes (Kim et al., 2023); Brain-JEPA combines gradient-based positional encoding with spatiotemporal masking in a joint-embedding predictive architecture (Dong et al., 2024). These models show that scale and pretraining can exploit temporal structure directly from BOLD signals, at the cost of large-scale pretraining data and compute. Our setting is complementary: no pretraining, ordinary per-cohort training-set sizes rather than tens of thousands of subjects, and the question of how much temporal benefit is recoverable with a small, targeted, task-specific model rather than a larger one.

Relation to Prior Work. Our work differs from the architectures above in treating temporal modeling as a controlled test of representation versus capacity, rather than as an encoder-design problem. Given an existing static backbone $\mathbf{X}$, the most direct way to lift $\mathbf{X}$ into the temporal domain is to apply its representation ϕ_X with shared weights to every window and aggregate the resulting sequence. We call this construction **TempX**; it follows the same broad spatial-encoder–plus–temporal-aggregator pattern used by STAGIN, T3former, FST-Mamba, CP-SSM, and dFCExpert, while retaining the original static backbone as the per-window encoder. We use TempX in our experiments as a strong, natural baseline rather than as our proposed method.

TempLasso, in contrast, deliberately withholds encoder capacity: it fixes a sparse, training-selected set of discriminative connections and models only their trajectories with a lightweight BiGRU, independent of any static backbone's complexity. This isolates whether preserving temporal *access* — rather than increasing model capacity along the spatial or temporal axis — is what drives predictive gains. **TempLasso** is closest to the convolutional BiGRU of (Ji et al., 2024); unlike that model, it operates on a sparse, supervised edge selection rather than the full connectivity matrix at each window, and it is designed from the outset as a plug-and-play branch: **X+TempLasso** attaches to any static backbone through lightweight static–temporal fusion without repeatedly evaluating **X** at every window, whereas **X+TempX** requires a full backbone evaluation for each temporal window. Our experiments directly contrast these two ways of adding temporal information to a static model, and we support the comparison with a formal result (Theorem 3.1) showing that Bayes-optimal risk with temporal access cannot exceed that of static FC and can be strictly lower when static aggregation discards task-relevant temporal distinctions.

3 METHODOLOGY

Modern FC models devote substantial modeling capacity to a representation that has already compressed an entire BOLD scan into a single static connectome. Our approach asks whether this tradeoff is necessary. Rather than designing a more expressive static architecture, we preserve within-scan connectivity variation and model it with a deliberately simple temporal model.

We introduce **TempLasso**, which follows a small set of discriminative functional connections through time and models their trajectories with a BiGRU. **TempLasso** can be used on its own, allowing us to directly compare a lightweight temporal model with substantially more complex static FC architectures. It can also be attached to any static backbone **X** as a lightweight auxiliary branch, yielding **X+TempLasso** without modifying the backbone itself.

3.1 FROM ONE CONNECTOME TO A CONNECTIVITY TRAJECTORY

Let $\mathbf{S}^{(s)} = [\mathbf{s}_1^{(s)}, \ldots, \mathbf{s}_N^{(s)}]^\top \in \mathbb{R}^{N \times T_s}$ denote the BOLD signal of subject s, where N is the number of ROIs and T_s is the scan length. Conventional functional connectivity compresses the entire recording into a single matrix $\mathbf{C}_{\text{stat}}^{(s)} \in \mathbb{R}^{N \times N}$. This representation captures persistent pairwise association, but removes when and how connectivity changes within the scan.

We instead divide the signal into overlapping temporal windows and estimate one local FC matrix in each window. Because these estimates contain substantially fewer observations than the full-scan matrix, we use Ledoit–Wolf shrinkage covariance, convert it to correlation, and apply the Fisher-z transform. This yields

$$\mathcal{C}^{(s)} = \left(\mathbf{C}_1^{(s)}, \ldots, \mathbf{C}_{L_s}^{(s)}\right), \qquad \mathbf{C}_t^{(s)} \in \mathbb{R}^{N \times N}.$$

The ROI set remains fixed while the connectivity pattern evolves through time. Subjects retain all valid windows, so the sequence length L_s may vary across scans. All data-dependent transformations are estimated using training subjects only; windowing and preprocessing details are given in Section 4 and the Appendix.

3.2 TEMPLASSO: A SIMPLE TEMPORAL FC MODEL

TempLasso is intentionally low-capacity. Its purpose is not to build a more sophisticated spatial encoder for every local connectome, but to test whether preserving a small number of informative connectivity trajectories is sufficient to recover predictive information discarded by static FC.

Using training subjects only, we fit an ℓ_1-regularized linear model and identify a fixed set of discriminative connections

$$\mathcal{E}_{\text{sel}} = \{(i_1, j_1), \ldots, (i_K, j_K)\}.$$

The same connections are then extracted from every local FC matrix:

$$\mathbf{v}_t^{(s)} = \left[C_{t,i_1 j_1}^{(s)}, \ldots, C_{t,i_K j_K}^{(s)}\right] \in \mathbb{R}^K.$$

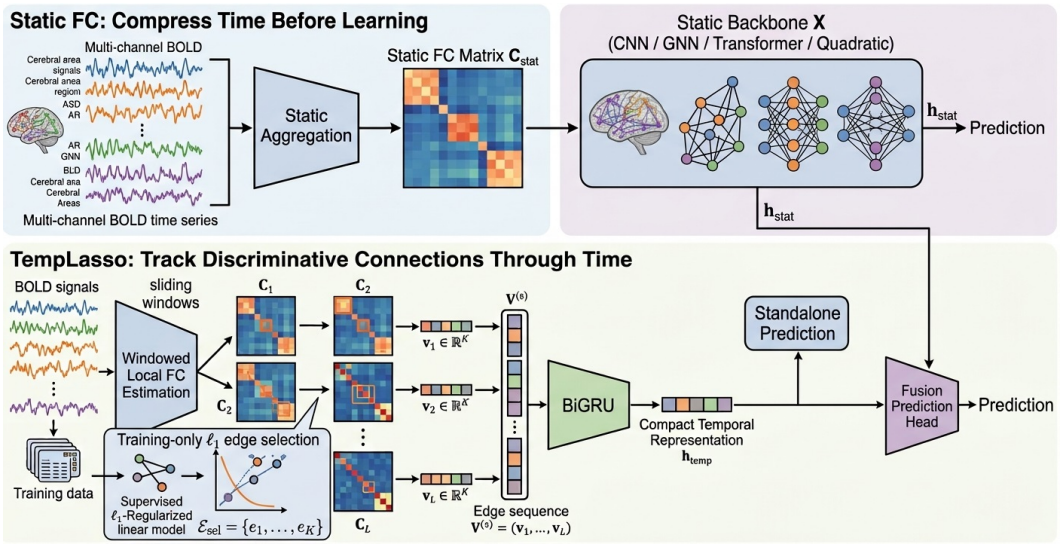

Figure 1: **Overview of TempLasso.** Static FC compresses the full BOLD scan into a single connectome $\mathbf{C}_{\mathrm{stat}}$ before learning. **TempLasso** instead constructs local FC matrices $\mathbf{C}_1, \ldots, \mathbf{C}_L$, tracks the same ℓ_1-selected connections across windows, and summarizes their trajectories with a lightweight BiGRU. The resulting temporal representation can be used alone or fused with the unchanged static backbone representation $\mathbf{h}_{\mathrm{stat}}$ to form **X+TempLasso**.

Thus, coordinate k always represents the same functional connection, and the sequence

$$\mathbf{V}^{(s)} = \left(\mathbf{v}_1^{(s)}, \ldots, \mathbf{v}_{L_s}^{(s)}\right)$$

describes how the selected connectivity pattern evolves throughout the scan.

We model this trajectory with a bidirectional GRU, $\mathbf{h}_{\mathrm{temp}}^{(s)} = \mathrm{BiGRU}\left(\mathbf{v}_1^{(s)}, \ldots, \mathbf{v}_{L_s}^{(s)}\right)$,

and obtain the standalone prediction $\hat{\mathbf{y}}^{(s)} = g_{\mathrm{temp}}\left(\mathbf{h}_{\mathrm{temp}}^{(s)}\right)$.

This construction is deliberately simple: sparse supervised edge selection determines *what* to track, while the BiGRU determines *how those selected connections evolve*. Comparing **TempLasso** with complex static FC architectures therefore directly tests whether preserving temporal information can matter more than increasing downstream model complexity.

3.3 TempLasso as a Plug-and-Play Temporal Branch

Temporal and static FC describe complementary aspects of the same scan. The full-scan matrix provides a stable summary of persistent connectivity, while **TempLasso** captures temporal variation in a small set of discriminative connections. We therefore augment existing static models rather than replace them.

Let $\mathbf{X}$ be any static FC backbone and let $\phi_X(\mathbf{C})$ denote its representation immediately before the prediction head. Its conventional static representation is $\mathbf{h}_{\mathrm{stat},X}^{(s)} = \phi_X\left(\mathbf{C}_{\mathrm{stat}}^{(s)}\right)$.

We combine the static and temporal branches using lightweight static–temporal fusion, with the fusion strategy selected using validation data only. We denote the resulting temporally augmented model by **X+TempLasso**; full fusion details are provided in Appendix B.4.

The construction is plug-and-play: the architecture and computation of $\mathbf{X}$ are unchanged, and the same **TempLasso** branch can be attached to convolutional, graph-based, transformer, or quadratic FC models. Importantly, the temporal branch does not require evaluating $\mathbf{X}$ on every local FC matrix. Its additional cost consists only of extracting K selected edges per window and processing the resulting low-dimensional sequence.

This asymmetric design is intentional. The static backbone is free to use substantial capacity to model the relatively stable full-scan connectome, while the temporal branch uses a strong information bottleneck to model noisier short-window connectivity estimates. In Section 4, we contrast this design with the natural alternative of applying the full backbone to every temporal window.

3.4 Why Temporal Information Can Matter More Than Model Capacity

The motivation for **TempLasso** is representational rather than architectural. Static FC is a many-to-one compression of the BOLD signal and is invariant to a common reordering of the time index (Appendix D.1). Therefore, distinct temporal trajectories can share the same static connectome, and no increase in the capacity of a downstream static model can recover distinctions that have already been removed.

Theorem 3.1 (Information advantage of temporal access)**.** *Let* $\mathbf{S}$ *denote the BOLD signal,* $\mathbf{C}_{\mathrm{stat}}(\mathbf{S})$ *its static FC matrix, and* Y *the prediction target. For any loss function* ℓ*, the Bayes-optimal risks satisfy* $R^*_{\mathrm{temp}} \le R^*_{\mathrm{stat}}$*, where* R^*_{stat} *restricts prediction to functions of* $\mathbf{C}_{\mathrm{stat}}(\mathbf{S})$*. The inequality is strict whenever the Bayes-optimal decision cannot be expressed as a function of* $\mathbf{C}_{\mathrm{stat}}(\mathbf{S})$ *alone on a set of positive probability.*

Proof and a strictness characterization are given in Appendices D.2–D.3. A related statement applies directly to static–temporal fusion. Let $\mathbf{Z}_{\mathrm{temp}}$ denote any temporal representation of the scan and define $\mathbf{Z}_{\mathrm{fuse}} = (\mathbf{C}_{\mathrm{stat}}, \mathbf{Z}_{\mathrm{temp}})$. Because $\mathbf{Z}_{\mathrm{fuse}}$ explicitly contains the static connectome, $R^*_{\mathrm{fuse}} \le R^*_{\mathrm{stat}}$, with strict inequality whenever the temporal representation resolves task-relevant distinctions that static FC collapses.

Two scope caveats are important. First, Theorem 3.1 ranges over all measurable functions of the raw signal $\mathbf{S}$, whereas **TempLasso** operates on a windowed, edge-selected representation. It therefore establishes an information limit, not a performance guarantee for a particular architecture (Appendix D.4). Second, the theorem is a population statement and does not account for finite-sample estimation. **TempX** uses a substantially higher-capacity representation of each temporal window than **TempLasso** yet is empirically less reliable. This behavior is consistent with a bias–variance tradeoff: expressive encoders must operate on noisy short-window FC estimates in a limited-data regime (Appendix D.5).

4 Experiments

4.1 Setup

Datasets. We evaluate on three public resting-state fMRI cohorts (Table 1): **ABIDE** for autism spectrum disorder classification (Martino et al., 2014), **ADNI** for Alzheimer's disease classification (Weiner et al., 2015), and **PPMI** for four-class Parkinson's disease classification (Marek et al., 2011). All experiments use the **Schaefer-100** atlas (Schaefer et al., 2018), giving $N = 100$ ROIs for every dataset. The same ROI time series are used to construct both the full-scan static FC matrix and the temporal FC sequence. Dataset composition and additional cohort details are provided in Appendix B.1; preprocessing and FC construction are described in Appendix B.

Table 1: Cohort summary.

Dataset	Condition	#Subjects	Class Counts	Task
ABIDE	Autism	1025	537/488	Binary
ADNI	Alzheimer's	138	80/58	Binary
PPMI	Parkinson's	195	15/113/14/53	4-class

Models and comparisons. We compare **TempLasso** against five static FC backbones spanning diverse architectural families: two established baselines, GCN (Kipf & Welling, 2017) and BrainNetCNN (Kawahara et al., 2017), and three recent models, BNT (Kan et al., 2022), ALTER (Yu et al., 2024), and BQN (Yang et al., 2025). Our evaluation focuses on two complementary questions.

(i) TempLasso vs. static FC models. We compare standalone **TempLasso** with these substantially more complex static architectures to test whether preserving temporal connectivity can compensate for lower model capacity.

Table 2: **Standalone TempLasso and plug-and-play temporal augmentation.** Mean±std AUC over five matched splits for ABIDE and ADNI, and macro-AUC for PPMI. **TempLasso** is evaluated standalone; for each static backbone **X**, we compare **X** with **X+TempLasso** (**X+TL**). Δ reports the mean AUC gain over **X**. Within each dataset block, **bold** marks the best standalone/backbone performance and underline marks the second best; gray marks the best overall performance (among X and X+TL) and the largest Δ. Statistical significance tests are reported in Appendix C.1.

	ABIDE			ADNI			PPMI		
Model	**X**	**X+TL**	Δ	**X**	**X+TL**	Δ	**X**	**X+TL**	Δ
TempLasso	71.91±1.81			80.52±9.70			**62.12±8.30**		
BQN	70.38±1.43	73.44±2.26	+3.06*	79.80±4.37	82.82±6.89	+3.02	55.40±4.96	62.48±7.68	+7.08
ALTER	**72.04±4.25**	**73.50±2.51**	+1.46	79.90±5.68	81.98±9.67	+2.08	52.51±4.27	60.00±4.77	+7.49
BrainNetCNN	71.76±2.18	73.42±1.90	+1.66*	**83.64±3.48**	**84.60±4.51**	+0.96	56.48±4.33	61.94±7.81	+5.46
BNT	67.89±1.71	70.55±2.42	+2.66	75.42±6.11	83.65±6.84	**+8.23**	51.50±8.60	61.87±8.03	**+10.38****
GCN	66.94±2.62	71.88±2.25	**+4.94***	78.24±8.05	82.00±7.43	+3.76	54.12±7.33	61.34±6.86	+7.22

(ii) Plug-and-play temporal augmentation. For each backbone **X**, we compare **X** with **X+TempLasso** to test whether the same lightweight temporal branch provides complementary predictive information beyond the static connectome.

We additionally evaluate direct full-backbone temporalization **TempX** and **X+TempX** as an ablation, asking whether repeatedly applying a high-capacity static encoder to local FC windows is preferable to the sparse temporal representation of **TempLasso**. All models are trained and evaluated from scratch under the same subject splits; no results are copied from prior work.

Training protocol. We use stratified 70/10/20 subject-level train/validation/test splits and repeat every experiment over five random seeds. The same splits are shared across all methods, enabling paired comparison between **X** and **X+TempLasso**. All data-dependent operations, including normalization and Lasso edge selection, are estimated using training subjects only. Hyperparameters and checkpoints are selected using the validation split; the test split is never used for model selection. Models are trained for up to 200 epochs with Adam / AdamW, batch size 16, and initial learning rate 10^{-4}. Complete optimization settings, **TempLasso** hyperparameters, and backbone configurations are given in Appendix B.4.

Evaluation. AUC is the primary metric for ABIDE and ADNI, and macro-AUC is used for the four-class PPMI task. We report mean±standard deviation across five runs. Statistical significance results are reported in Appendix C.1, and computational-cost and runtime comparisons are given in App. B.6. Our code is available at `https://anonymous.4open.science/r/TempLasso-EA65`

4.2 Main Results

Table 2 summarizes our two primary comparisons: standalone **TempLasso** against substantially more complex static FC models, and each backbone **X** against its temporally augmented counterpart **X+TempLasso** (**X+TL**).

Representation can rival model capacity. The first result is striking: despite its deliberately restricted architecture, standalone **TempLasso** is competitive with the strongest static FC models across all three cohorts. On ABIDE, it outperforms four of five static backbones and is within 0.13 AUC points of the best one. On ADNI, it again exceeds four of five static models. On PPMI, **TempLasso** is the strongest standalone model, reaching 62.98 macro-AUC compared with 56.48 for the best static backbone (BrainNetCNN). This comparison is intentionally asymmetric. **TempLasso** uses no graph propagation, connectome-specific convolution, Transformer, or quadratic spatial encoder; it follows a sparse set of selected connections through time with a lightweight BiGRU. Its competitiveness therefore points to a different source of predictive power: preserving temporal structure that sophisticated static models never observe. In this setting, access to a richer representation can compensate for substantially less model capacity.

Temporal information consistently complements static FC. The stronger result emerges when **TempLasso** is used as a plug-and-play temporal branch. Adding the same lightweight module improves every evaluated static backbone on every dataset: all 15 of 15 backbone–dataset combinations show a positive mean gain. The average improvement is +2.76 AUC points on ABIDE, +3.61 on

Table 3: **Temporalization strategy.** Mean AUC over five runs for ABIDE and ADNI, and macro-AUC for PPMI TempX (**TX**) applies the full backbone **X** to every temporal window, whereas **TempLasso** (**TL**) tracks a sparse set of selected connections. Best result within each backbone–dataset block is shown in bold.

	ABIDE				ADNI				PPMI			
Backbone	**X**	**TX**	**X+TX**	**X+TL**	**X**	**TX**	**X+TX**	**X+TL**	**X**	**TX**	**X+TX**	**X+TL**
BQN	70.38	65.60	69.22	**73.44**	79.80	76.68	80.82	**82.82**	55.40	56.00	57.80	**62.48**
ALTER	72.04	67.78	71.64	**73.50**	79.90	81.24	**83.34**	81.98	52.51	51.57	53.09	**60.00**
BrainNetCNN	71.76	69.72	72.56	**73.42**	83.64	82.28	**85.08**	84.60	56.48	52.75	56.53	**61.94**
BNT	67.89	64.75	67.36	**70.55**	75.42	70.83	76.77	**83.65**	51.50	59.08	55.14	**61.87**
GCN	66.94	67.58	68.36	**71.88**	78.24	75.52	77.18	**82.00**	54.12	53.00	55.27	**61.34**

ADNI, and $+7.53$ macro-AUC points on PPMI, for an overall mean gain of $+4.63$ points. The single largest gain is $+10.38$ for BNT on PPMI ($p = 0.004$), the strongest effect size observed across all backbone–dataset combinations.

The improvements are not confined to weak baselines. **TempLasso** also raises the performance of the strongest static architecture in each cohort, with gains of $+1.46$ for ALTER on ABIDE, $+0.96$ for BrainNetCNN on ADNI, and $+5.46$ for BrainNetCNN on PPMI. This consistency supports the view that static and temporal FC capture complementary information: the full-scan connectome summarizes persistent connectivity, while **TempLasso** restores within-scan variation that static aggregation suppresses.

Because each comparison is based on only five matched splits, individual tests have limited power. Nevertheless, several backbone–dataset improvements are significant under two-sided paired t-tests, while the direction of the mean effect is positive in every case. Full seed-level results and significance tests are reported in Appendix C.1.

More temporal capacity is not necessarily better. A natural alternative is to apply the full static backbone **X** to every local FC window. We denote the resulting standalone model by **TempX**, and its fusion with static FC by $X + \mathrm{TempX}$. Table 3 shows that this substantially richer temporalization strategy is much less reliable.

Standalone TempX frequently underperforms the corresponding static model. Fusing TempX with static FC helps in some cases, especially on ADNI, but the gains remain architecture- and dataset-dependent. In contrast, **X+TempLasso** improves every evaluated static model and is the strongest temporalized variant for all available backbones on ABIDE and PPMI and for three of five on ADNI. Full mean$\pm$std results for these comparisons are provided in Appendix C.

This result sharpens the central message of the paper. Temporal access is valuable, but simply increasing the capacity of the model applied to each short-window connectome is not sufficient. The more reliable strategy is to preserve a small amount of task-relevant temporal variation and model it with an appropriately constrained temporal representation. Together, the standalone, fusion, and TempX results suggest that the main bottleneck in FC learning is not merely how powerfully a connectome is modeled, but how much predictive structure is discarded before modeling begins.

4.3 Ablation Studies

The ablations in this section isolate two design choices internal to **TempLasso** itself — the windowing scheme used to build the temporal sequence, and the aggregator used to model it — independent of which static backbone it is later fused with.

To keep the significance-testing burden tractable, we run both ablations on **TempLasso** standalone, evaluated on ABIDE and PPMI rather than the full three-cohort grid used in Section 4.2; full sweeps, exact deltas, and a discussion of what does and does not survive seed noise at this sample size are provided in Appendix C.3.

Table 4: **Window/stride sensitivity for TempLasso.** The main configuration (44s/22s) is used throughout the paper.

Win./St.	**Overlap**	**ABIDE**	**PPMI**
32s / 32s	No	70.55	59.14
32s / 16s	Yes	70.57	58.83
44s / 22s	**Yes**	**71.91**	**62.98**
60s / 30s	Yes	69.22	55.59

Ablation 1: Sensitivity to window length and stride. Section 3.1 fixes the window/stride at 44s/22s following Allen et al. (2014b), a choice taken from prior dynamic-FC literature rather than tuned on either evaluation cohort. We test four alternatives spanning shorter,

non-overlapping, and longer windows. **TempLasso** is stable across the shorter and overlapping variants — window length appears to matter more than overlap at this scale — and degrades only at the longest window tested (60s/30s), where averaging over more of the signal erodes the short-timescale connectivity variation the model is designed to track. That the literature-standard 44s/22s setting also performs best on both cohorts is a favorable outcome of this sweep rather than something selected for; full numbers and a discussion of what is and is not distinguishable from seed noise are given in Appendix C.3.

Ablation 2: How important is sequential modeling? **TempLasso** uses a BiGRU to model the ordered trajectory of selected functional connections. We compare four temporal aggregators while keeping the selected edges and local FC windows fixed: (i) a shared MLP followed by mean pooling, which is permutation invariant; (ii) a unidirectional GRU; (iii) a BiGRU applied to randomly shuffled windows, which preserves model capacity but destroys the true temporal order; and (iv) the ordered BiGRU used by **TempLasso**.

Table 5: **Temporal aggregator ablation.**

Aggregator	ABIDE	PPMI
MLP	70.18	58.56
GRU	69.24	60.80
BiGRU-Shuffled	68.16	58.80
BiGRU	**71.91**	**62.98**

The ordered BiGRU wins on both cohorts.It beats the Shuffled BiGRU by 3.75 AUC points on ABIDE and 4.18 macro-AUC points on PPMI despite a matched architecture and parameter count, highlighting the value of true temporal ordering. Relative to the unidirectional GRU, the corresponding gains are 2.67 and 2.18 points. The MLP baseline is less consistent — stronger than GRU and Shuffled BiGRU on ABIDE, weaker on PPMI — suggesting capacity without correct ordering is not reliably beneficial on its own. Recovering temporal structure therefore requires both the right trajectory and a model equipped to use it; neither ingredient alone recovers what the two together provide. Full results are reported in Appendix C.3.

Together, these two ablations reinforce the paper's central claim at a smaller scale: within **TempLasso** itself, the gains come from preserving the right temporal information and modeling it with a correctly-ordered, appropriately-sized aggregator — not from maximizing capacity along either the windowing or the sequence-model axis.

4.4 Discussion and Limitations

Our results support temporal access as a useful complement to static FC across several modeling families and clinical cohorts. While we use a fixed sliding-window construction and a sparse set of task-relevant connections, the observed gains are consistent across architectures and remain stable under the windowing variations considered in our ablations. This suggests that the benefit is not tied to a single temporal configuration. At the same time, TempLasso is intentionally simple. Its sparse representation provides a strong inductive bias for the finite-sample regime studied here, while leaving open the possibility that adaptive, multiscale, or more structured temporal representations could capture additional information. Evaluating such extensions across broader parcellations and acquisition settings is a natural direction for future work.

5 Conclusion

Resting-state fMRI is temporal, but much of brain-network learning begins only after that temporal structure has been collapsed into a single static connectome. Our results suggest that this compression can be a more fundamental bottleneck than model capacity itself. We introduced **TempLasso**, a deliberately simple temporal model that follows a sparse set of discriminative functional connections through time, and showed that preserving these trajectories can match or outperform substantially more complex static FC architectures. The same lightweight temporal branch consistently improves strong static backbones through simple late fusion, while directly temporalizing the full backbone is less reliable. Together with our theoretical analysis, these findings support a simple message: information discarded before learning cannot be recovered by making the downstream model more expressive. For functional connectivity, progress may therefore depend less on building increasingly elaborate models of one static connectome and more on preserving and exploiting the temporal structure already present in the signal.

AI USE STATEMENT

Generative AI tools were used to assist with drafting and revising portions of the manuscript, primarily to improve English, clarity, organization, and readability; to assist with research-code development and debugging; and to help create and refine schematic figures. All methodological choices, model designs, preprocessing decisions, experimental protocols, statistical analyses, and scientific conclusions were made by the authors. AI-assisted code was reviewed, tested, and verified before use, and all AI-assisted text and visual material was checked and revised by the authors. Generative AI was not used to independently establish the scientific claims or validate the reported results. The authors take full responsibility for the final manuscript, code, experiments, and conclusions.

ETHICS STATEMENT

This work studies methodological questions in resting-state fMRI and functional connectivity using three existing neuroimaging resources: ABIDE, ADNI, and PPMI. We collect no new human-subject data and use only data made available through the corresponding research repositories and their established access and data-use procedures. Our experiments operate on preprocessed, region-level fMRI representations and do not attempt to identify individual participants.

The proposed methods are intended for research on brain-network representation and prediction, not for clinical diagnosis or decision making. Although the evaluated tasks involve neurological and neuropsychiatric conditions, model outputs should not be interpreted as clinical evidence without independent validation on appropriate populations and under suitable clinical protocols. As with other neuroimaging prediction systems, performance may also depend on cohort composition, acquisition site, preprocessing, and other sources of dataset shift. We therefore view the reported results as evidence about functional-connectivity representations and modeling rather than as claims of clinical utility.

REPRODUCIBILITY STATEMENT

We provide implementation code and experiment scripts in an anonymous repository, including static and temporal FC construction, sliding-window processing, Ledoit–Wolf covariance estimation, **TempLasso** edge selection, BiGRU temporal modeling, static–temporal fusion, **TempX** controls, and the reported ablations. The repository also includes training, evaluation, statistical-testing, and table-generation scripts, together with the subject splits and random seeds used in the experiments.

The main paper describes the TempLasso methodology and evaluation protocol, while Appendix B provides the experimental details. Dataset composition is reported in Appendix B.1; FC construction and temporal windowing in Appendix B.2; TempLasso construction in Appendix B.3; and training and hyperparameter selection in Appendix B.4. Evaluation details are given in Appendix B.5, and the complete paired significance analysis for **X+TempLasso** is reported in Appendix C.1. Theoretical statements and proofs are provided in Appendix D.

All data-dependent operations, including normalization and sparse edge selection, are fit using training subjects only. Competing methods use matched subject splits and evaluation protocols, and results are reported over the same five random seeds. Together, the released code, fixed splits, and appendix details are intended to reproduce the main experiments, temporal controls, ablations, and statistical comparisons reported in the paper. Our code is available at
https://anonymous.4open.science/r/TempLasso-EA65

REFERENCES

Elena A. Allen, Eswar Damaraju, Sergey M. Plis, Erik B. Erhardt, Tom Eichele, and Vince D. Calhoun. Tracking whole-brain connectivity dynamics in the resting state. *Cerebral Cortex*, 24(3): 663–676, 2014a.

Elena A. Allen, Eswar Damaraju, Sergey M. Plis, and et al. Tracking whole-brain connectivity dynamics in the resting state. *Cerebral Cortex*, 24(3):663–676, 2014b. doi: 10.1093/cercor/bhs352.

Minheng Chen, Xiaowei Yu, Jing Zhang, Tong Chen, Chao Cao, Yan Zhuang, Yanjun Lyu, Lu Zhang, Tianming Liu, and Dajiang Zhu. Core-periphery principle guided state space model for func-

tional connectome classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 236–246. Springer, 2025a.

Tingting Chen, Hongming Li, Hao Zheng, and Yong Fan. dfcexpert: Learning dynamic functional connectivity patterns with modularity and state experts. *IEEE Transactions on Medical Imaging*, 2025b.

Zijian Dong, Ruilin Li, Yilei Wu, Thuan T Nguyen, Joanna S Chong, Fang Ji, Nathanael R Tong, Christopher L Chen, and Juan H Zhou. Brain-jepa: Brain dynamics foundation model with gradient positioning and spatiotemporal masking. *Advances in Neural Information Processing Systems*, 37: 86048–86073, 2024.

R Matthew Hutchison, Thilo Womelsdorf, Joseph S Gati, Stefan Everling, and Ravi S Menon. Resting-state networks show dynamic functional connectivity in awake humans and anesthetized macaques. *Human brain mapping*, 34(9):2154–2177, 2013.

Junzhong Ji, Chuantai Ye, and Cuicui Yang. Convolutional bidirectional gru for dynamic functional connectivity classification in brain diseases diagnosis. *Knowledge-Based Systems*, 287:111450, 2024.

Xuan Kan, Wei Dai, Hejie Cui, Zilong Zhang, Ying Guo, and Carl Yang. Brain network transformer. In *NeurIPS*, 2022.

Jeremy Kawahara et al. Brainnetcnn: Convolutional neural networks for brain networks; towards predicting neurodevelopment. *NeuroImage*, 146:1038–1049, 2017.

Byung-Hoon Kim, Jong Chul Ye, and Jae-Jin Kim. Learning dynamic graph representation of brain connectome with spatio-temporal attention. In *NeurIPS*, volume 34, pp. 4314–4327, 2021.

Peter Kim, Junbeom Kwon, Sunghwan Joo, Sangyoon Bae, Donggyu Lee, Yoonho Jung, Shinjae Yoo, Jiook Cha, and Taesup Moon. Swift: Swin 4d fmri transformer. *Advances in Neural Information Processing Systems*, 36:42015–42037, 2023.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2017.

Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411, 2004.

Xiang Li, Yong Fan, and et al. Interpretable brain graph neural network for fmri analysis. *Medical Image Analysis*, 74:102233, 2021. doi: 10.1016/j.media.2021.102233.

Kenneth Marek, Danna Jennings, Shirley Lasch, Andrew Siderowf, Caroline Tanner, et al. The Parkinson's Progression Markers Initiative (ppmi). *Progress in Neurobiology*, 95(4):629–635, 2011. doi: 10.1016/j.pneurobio.2011.09.005.

Adriana Di Martino, Chaogan G. Yan, Qingyang Li, and et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular Psychiatry*, 19(6):659–667, 2014. doi: 10.1038/mp.2013.78.

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.

Sarah Parisot, Sofia Ktena, Enzo Ferrante, and et al. Disease prediction using graph convolutional networks: Application to autism spectrum disorder and alzheimer's disease. In *MICCAI—Workshop on Graphs in Biomedical Image Analysis (GRAIL)*, 2017. Extended version: arXiv:1806.01735.

Ciyuan Peng, Yuelong Huang, Qichao Dong, Shuo Yu, Feng Xia, Chengqi Zhang, and Yaochu Jin. Biologically plausible brain graph transformer. In *ICLR*, 2025.

M. A. Preti, T. A. Bolton, and D. Van De Ville. The dynamic functional connectome: State-of-the-art and perspectives. *NeuroImage*, 160:41–54, 2017. doi: 10.1016/j.neuroimage.2016.12.061.

Manish Saggar, Olaf Sporns, Javier Gonzalez-Castillo, Peter A Bandettini, Gunnar Carlsson, Gary Glover, and Allan L Reiss. Towards a new approach to reveal dynamical organization of the brain using topological data analysis. *Nature Communications*, 9(1):1399, 2018. doi: 10.1038/s41467-018-03664-4.

Alexander Schaefer, Ru Kong, Evan M Gordon, Timothy O Laumann, Xi-Nian Zuo, Avram J Holmes, Simon B Eickhoff, and BT Thomas Yeo. Local-global parcellation of the human cerebral cortex from intrinsic functional connectivity mri. *Cerebral cortex*, 28(9):3095–3114, 2018.

Md. Joshem Uddin, Soham Changani, and Baris Coskunuzer. T3former: Temporal graph classification with topological machine learning. In *AAAI*, 2026.

Yuxiang Wei, Anees Abrol, and Vince D Calhoun. Hierarchical spatio-temporal state-space modeling for fmri analysis. In *International Conference on Research in Computational Molecular Biology*, pp. 86–98. Springer, 2025.

Michael W Weiner, Dallas P Veitch, Paul S Aisen, Laurel A Beckett, Nigel J Cairns, et al. 2014 update of the Alzheimer's Disease Neuroimaging Initiative: A review of papers published since its inception. *Alzheimer's & Dementia*, 11(6):e1–e120, 2015. doi: 10.1016/j.jalz.2014.11.001.

Jiaxing Xu, Yunhan Yang, David Huang, Sophi Shilpa Gururajapathy, Yiping Ke, Miao Qiao, Alan Wang, Haribalan Kumar, Josh McGeown, and Eryn Kwon. Data-driven network neuroscience: On data collection and benchmark. *NeurIPS*, 36:21841–21856, 2023.

Liang Yang, Yuwei Liu, Jiaming Zhuo, Di Jin, Chuan Wang, Zhen Wang, and Xiaochun Cao. Do we really need message passing in brain network modeling? In *ICML*, 2025.

Shuo Yu, Shan Jin, Ming Li, Tabinda Sarwar, and Feng Xia. Long-range brain graph transformer. In *NeurIPS*, 2024.

APPENDIX

A A Brief Primer on BOLD Signals and Functional Connectivity **13**
A.1 From fMRI to Regional BOLD Signals . . . 13
A.2 Static Functional Connectivity . . . 14
A.3 Temporal Functional Connectivity . . . 14
A.4 Choosing Temporal Windows . . . 14

B Experimental Details **15**
B.1 Dataset Details . . . 15
B.2 BOLD Preprocessing and FC Construction . . . 15
B.3 **TempLasso** Construction . . . 16
B.4 Training and Hyperparameter Selection . . . 17
B.5 Evaluation and Reproducibility . . . 18
B.6 Computational Cost and Runtimes . . . 19

C Additional Results **20**
C.1 Statistical Significance Results . . . 20
C.2 Results on Temporalizing Strategy . . . 21
C.3 Ablations . . . 22

D Temporal Information and the Limits of Static FC **23**
D.1 Static FC is a Many-to-One Temporal Compression . . . 24
D.2 Proof of Theorem 3.1 . . . 24
D.3 When Is the Inequality Strict? . . . 25
D.4 Representation Versus Model Capacity . . . 25
D.5 Why More Expressive Temporal Models Need Not Be Better . . . 26
D.6 What the Result Does Not Claim . . . 26

A A BRIEF PRIMER ON BOLD SIGNALS AND FUNCTIONAL CONNECTIVITY

This section provides a short, non-technical introduction to the neuroimaging setting used in this paper. It is intended for readers familiar with machine learning but not necessarily with resting-state fMRI or functional connectivity. Dataset-specific preprocessing, parcellation, and temporal-window choices are described separately in Appendix B.

A.1 FROM FMRI TO REGIONAL BOLD SIGNALS

Functional magnetic resonance imaging (fMRI) does not directly measure neural electrical activity. Instead, it measures changes in blood oxygenation that are associated with local neural activity. The resulting measurements are known as *blood-oxygen-level-dependent* (BOLD) signals.

During a resting-state fMRI scan, the brain is repeatedly imaged over time while the subject is not engaged in an explicit experimental task. The brain is typically divided into a collection of regions of interest (ROIs) using a predefined atlas. The voxel-level measurements within each ROI are combined to produce one representative BOLD time series.

After this step, a subject can be viewed simply as N regional signals observed over T time points,

$$\mathbf{S} \in \mathbb{R}^{N \times T}.$$

Thus, the data contain both spatial structure, through the different brain regions, and temporal structure, through the evolution of their signals over the scan.

A.2 Static Functional Connectivity

Functional connectivity (FC) describes statistical dependence between the BOLD signals of different brain regions. A standard choice is Pearson correlation. For ROIs i and j,

$$C_{ij} = \mathrm{corr}(\mathbf{s}_i, \mathbf{s}_j).$$

Computing this quantity for every pair of ROIs produces a symmetric matrix

$$\mathbf{C} \in \mathbb{R}^{N \times N},$$

often called a *functional connectome*. The ROIs can be interpreted as graph nodes and the FC values as weighted edges.

In the conventional static setting, each correlation is computed using the entire scan. An otherwise high-dimensional temporal recording is therefore reduced to a single matrix $\mathbf{C}_{\mathrm{stat}}$.

This representation is compact and convenient for machine learning, but it summarizes each connection by one value over the full recording. As a result, information about how that connection changed during the scan is no longer explicitly available.

A.3 Temporal Functional Connectivity

Brain connectivity need not remain constant throughout a resting-state scan. Two regions may be strongly coupled during one portion of the recording and only weakly coupled during another. A single full-scan correlation averages these different behaviors together.

A common way to preserve within-scan variation is *sliding-window functional connectivity* (Allen et al., 2014a; Hutchison et al., 2013; Preti et al., 2017). Instead of computing one FC matrix from the complete scan, we compute FC separately over a sequence of shorter, possibly overlapping intervals:

$$\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_L.$$

The ROIs remain the same in every window; only the estimated connectivity values change. The result is therefore a trajectory of connectivity states rather than one static connectome.

A useful analogy is the difference between a photograph and a video. Static FC gives one summary of the recording, whereas temporal FC retains a sequence describing how connectivity evolves.

For example, two subjects might exhibit

$$A \to B \qquad \text{and} \qquad B \to A,$$

where A and B denote two connectivity states. The two subjects may have similar full-scan FC summaries even though their temporal evolution is different.

A.4 Choosing Temporal Windows

Window length controls a basic tradeoff. Short windows provide finer temporal resolution but estimate correlations from fewer observations, making local FC matrices noisier. Longer windows provide more stable estimates but average over more of the scan and therefore lose temporal resolution.

Conceptually,

$$\begin{aligned} \text{short windows} &\implies \text{more temporal detail, less stable FC,} \\ \text{long windows} &\implies \text{more stable FC, less temporal detail.} \end{aligned}$$

Overlap between consecutive windows creates a smoother and more densely sampled connectivity trajectory, at the cost of increased redundancy.

There is therefore no universally optimal window length or overlap. The appropriate choice depends on the temporal scale of interest and the reliability of the resulting FC estimates. In our experiments, we use a fixed default windowing scheme and separately evaluate sensitivity to alternative window lengths and overlaps. Exact choices and all preprocessing details are provided in Appendix B.

B Experimental Details

This section summarizes the preprocessing and implementation choices needed to reproduce our experiments. Additional code-level details, configuration files, and exact backbone settings will be released with the implementation. All data-dependent operations are fit using training subjects only.

B.1 Dataset Details

We use three public resting-state fMRI cohorts: ABIDE, ADNI, and PPMI. A compact summary of the prediction tasks and cohort sizes is given in Table 1; additional details are provided below.

ABIDE. The Autism Brain Imaging Data Exchange (ABIDE) (Martino et al., 2014) is a multi-site neuroimaging collection designed to support the study of autism spectrum disorder (ASD). We use the resting-state fMRI cohort for binary classification between subjects with ASD and typically developing controls. After applying our inclusion and quality-control criteria, the final cohort contains $1{,}025$ subjects, with 488 ASD subjects and 537 typically developing controls. ABIDE is the largest cohort in our evaluation and provides a heterogeneous multi-site setting in which acquisition conditions and subject characteristics vary substantially across sites.

ADNI. The Alzheimer's Disease Neuroimaging Initiative (ADNI) (Weiner et al., 2015) is a longitudinal multi-center study of Alzheimer's disease and related neurodegenerative conditions. We use resting-state fMRI for binary classification between subjects with Alzheimer's disease and cognitively normal controls. The resulting cohort contains 138 subjects, with 80 cognitively normal controls and 58 subjects with Alzheimer's disease. Compared with ABIDE, ADNI is a substantially smaller-sample setting and therefore provides a useful test of whether temporal FC representations remain effective when training data are limited.

PPMI. The Parkinson's Progression Markers Initiative (PPMI) (Marek et al., 2011) is a multi-center study focused on Parkinson's disease and related clinical phenotypes. We use the resting-state fMRI cohort for four-class classification among controls, Parkinson's disease (PD), scans without evidence of dopaminergic deficit (SWEDD), and prodromal subjects. After quality control, the dataset contains 195 subjects, with class counts of $15/113/14/53$ for Control/PD/SWEDD/Prodromal, respectively. PPMI complements the binary ABIDE and ADNI tasks by providing a multiclass and strongly imbalanced prediction problem, allowing us to evaluate temporal FC modeling beyond binary disease–control classification.

B.2 BOLD Preprocessing and FC Construction

ROI time series. We use preprocessed, parcellated Schaefer-100 ROI-level BOLD time series from the brain-network benchmark of Xu et al. (2023), with the Schaefer parcellation (Schaefer et al., 2018) and $N = 100$ cortical regions. For each subject, the ROI time series is represented as

$$\mathbf{S}^{(s)} \in \mathbb{R}^{N \times T_s}.$$

For temporal FC construction, we resample each subject's ROI time series from its native repetition time (TR) to a common temporal grid with $\mathrm{TR}_{\mathrm{common}} = 2\,\mathrm{s}$.

Static FC. For the static branch, we use the whole-scan functional-connectivity matrix provided by the source benchmark (Xu et al., 2023),

$$\mathbf{C}_{\text{stat}}^{(s)} \in \mathbb{R}^{N\times N}.$$

These matrices are used as the common static FC input across the evaluated backbones.

Temporal FC. We construct local FC matrices using sliding-window functional connectivity, following the standard dynamic-FC paradigm (Allen et al., 2014a; Hutchison et al., 2013; Preti et al., 2017). Our default window length is $W = 44$ s with a stride of 22 s, corresponding to 50% overlap between consecutive windows.

Because short-window covariance estimates can be unstable, each local FC matrix is estimated using Ledoit–Wolf shrinkage covariance (Ledoit & Wolf, 2004). The resulting covariance matrix is converted to a correlation matrix, clipped numerically to lie strictly within $(-1, 1)$, and Fisher-z transformed. This produces

$$\mathcal{C}^{(s)} = \left(\mathbf{C}_1^{(s)}, \ldots, \mathbf{C}_{L_s}^{(s)}\right), \qquad \mathbf{C}_t^{(s)} \in \mathbb{R}^{N\times N}.$$

We set the diagonal to zero and retain the $N(N-1)/2 = 4{,}950$ upper-triangular connections from each window for storage and downstream temporal modeling. Subjects retain all valid windows, so L_s may vary across scans; no temporal windows are truncated. During minibatch construction, sequences are zero-padded to the longest sequence in the batch and their true lengths are retained. The BiGRU processes the resulting variable-length sequences using packed sequences, ensuring that padded windows do not contribute to the recurrent computation.

B.3 TEMPLASSO CONSTRUCTION

Edge selection. For each training split, we pool the upper-triangular temporal-FC edge vectors from all windows of the training subjects. Each window inherits the class label of its corresponding subject, yielding a training matrix with one row per temporal window and $N(N-1)/2 = 4{,}950$ candidate connections for $N = 100$ ROIs.

Before fitting the selector, each connection is standardized using the mean and standard deviation computed over the pooled training windows only. We then fit an ℓ_1-regularized logistic regression model in the Fisher-z input space. We use the `liblinear` solver with `max_iter`=3000 and a fixed random state of 0. The inverse regularization strength C is included in the validation search over **TempLasso** candidate configurations. We search

$$C \in \{0.005, 0.01, 0.03\}$$

for ABIDE and

$$C \in \{0.01, 0.03, 0.1\}$$

for ADNI and PPMI. Candidate configurations are selected using the validation set only: by minimum validation loss for the binary ABIDE and ADNI tasks, and by maximum validation macro-AUC for the four-class PPMI task.

For the binary ABIDE and ADNI tasks, the selected edge set consists of all connections whose fitted logistic-regression coefficient is nonzero. For the four-class PPMI task, we fit one-vs-rest ℓ_1-regularized logistic regressions and take the union of the connections with nonzero coefficients across the four classifiers. Candidates retaining fewer than five connections are treated as degenerate and discarded. Thus, the number of selected connections, denoted by K, is determined by the training data and regularization strength rather than fixed in advance.

After model selection, the resulting edge set and its training-set standardization statistics are frozen and applied unchanged to the training, validation, and test sequences. For a selected connection (i_k, j_k), let μ_k and σ_k denote its mean and standard deviation estimated over the pooled training windows. For temporal window t, the input to the temporal model is

$$\mathbf{v}_t^{(s)} = \left[\frac{C_{t,i_1j_1}^{(s)} - \mu_1}{\sigma_1}, \ldots, \frac{C_{t,i_Kj_K}^{(s)} - \mu_K}{\sigma_K}\right] \in \mathbb{R}^K.$$

The same ordered set of connections and the same training-set standardization statistics are used at every window and for every data split. Thus, each coordinate of the sequence represents the same functional connection throughout the scan.

Temporal model. The sequence of selected connection vectors is processed by a single-layer bidirectional GRU. We tune the hidden dimension per direction over

$$h \in \{32, 64, 128\}$$

using the validation set. The recurrent layer uses no inter-layer dropout, and we use final-state pooling. Specifically, the final hidden states from the forward and backward directions are concatenated to form the subject-level temporal representation

$$\mathbf{h}_{\text{temp}}^{(s)} = \left[\mathbf{h}_{\text{fwd}}^{(s)} \| \mathbf{h}_{\text{bwd}}^{(s)}\right] \in \mathbb{R}^{2h}.$$

Variable-length sequences are processed using packed sequences, so padded windows do not affect the recurrent states. No additional projection or layer normalization is applied to the resulting temporal embedding.

Standalone **TempLasso** maps $\mathbf{h}_{\text{temp}}^{(s)}$ directly to the class logits using a linear prediction head.

For a static backbone **X**, the **X+TempLasso** model combines the representation produced by the static backbone with the TempLasso temporal representation using parameter-free concatenation,

$$\mathbf{h}_{\text{fuse}}^{(s)} = \left[\mathbf{h}_{\text{stat},X}^{(s)} \| \mathbf{h}_{\text{temp}}^{(s)}\right].$$

The concatenated representation is passed directly to a linear classifier. The static backbone itself is otherwise unchanged.

B.4 Training and Hyperparameter Selection

We use stratified $70/10/20$ subject-level train/validation/test splits over five random seeds,

$$\{42, 1042, 2042, 3042, 4042\}.$$

The same frozen splits are used for every method within a dataset. All preprocessing and data-dependent operations, including standardization and TempLasso edge selection, are fit using training subjects only. The validation split is used for hyperparameter selection, checkpoint selection, early stopping, and fusion selection. The test split is not accessed until the complete configuration has been selected.

Neural models are trained with Adam using weight decay 10^{-4} and batch size 16, for at most 200 epochs. We use a cosine learning-rate schedule with minimum learning rate 10^{-5} and early stopping based on validation performance. Model selection is performed independently for each seed. For the binary ABIDE and ADNI tasks, checkpoints and candidate configurations are selected by minimum validation loss. For the four-class PPMI task, selection is based on maximum validation macro one-vs-rest AUC. Once the validation-selected configuration is frozen, the corresponding held-out test split is evaluated once.

Static backbones. The static models are tuned using compact architecture-specific search spaces. For BQN, we search

$$L \in \{1, 2, 3, 4, 5\}, \qquad a \in \{\text{GELU}, \text{LeakyReLU}, \text{ELU}\}, \qquad p \in \{0, 0.1, 0.2, 0.3\},$$

with learning rate 10^{-4}. For ALTER, we use the fixed single-head configuration with correlation threshold 0.3 and no Transformer bias, and search

$$L \in \{1, 2\}, \qquad p \in \{0.1, 0.3\},$$

with learning rate 10^{-4}. For BrainNetCNN, we search

$$p \in \{0.1, 0.3, 0.5\}, \qquad \eta \in \{10^{-4}, 3 \times 10^{-4}, 10^{-3}\}.$$

For BNT, we search

$$L \in \{1, 2\}, \qquad p \in \{0.1, 0.3\}, \qquad \eta \in \{10^{-4}, 3 \times 10^{-4}\}.$$

For GCN, we use two graph-convolution layers and search

$$h_{\text{GCN}} \in \{32, 64\}, \qquad p \in \{0.1, 0.3\}, \qquad \eta \in \{10^{-4}, 10^{-3}\}.$$

TempLasso. For standalone **TempLasso**, the validation search jointly varies the ℓ_1 regularization strength described in Appendix B.3, the BiGRU hidden dimension

$$h \in \{32, 64, 128\},$$

and learning rate

$$\eta \in \{10^{-4}, 3 \times 10^{-4}\}.$$

The BiGRU contains one recurrent layer and uses final-state pooling. The number of selected connections K is not independently tuned: it is induced by the training data and the selected regularization strength C.

Full-backbone temporalization. For **TempX**, the static backbone is applied independently to each temporal FC window and its sequence of embeddings is aggregated by a single-layer BiGRU. We search

$$h \in \{32, 64, 128\}$$

for the BiGRU hidden dimension. For BQN, the temporal search uses $L \in \{1, 3\}$, dropout $p \in \{0.1, 0.3\}$, and ELU activation. For ALTER, we use $L \in \{1, 2\}$ and $p \in \{0.1, 0.3\}$. For BrainNetCNN, we use $p \in \{0.3, 0.5\}$. BNT uses $L \in \{1, 2\}$ and $p \in \{0.1, 0.3\}$, while GCN uses two layers, hidden dimension in $\{32, 64\}$, and dropout in $\{0.1, 0.3\}$. The learning-rate search is

$$\eta \in \{10^{-4}, 3 \times 10^{-4}\}$$

for BQN, ALTER, BrainNetCNN, and BNT, and

$$\eta \in \{10^{-4}, 10^{-3}\}$$

for GCN.

Fusion selection. We treat the fusion mechanism as an additional validation-selected hyperparameter. We consider two complementary fusion strategies. In representation-level fusion, the static and temporal embeddings are concatenated,

$$\mathbf{h}_{\text{fuse}}^{(s)} = \left[\mathbf{h}_{\text{stat},X}^{(s)} \| \mathbf{h}_{\text{temp}}^{(s)}\right],$$

and the concatenated representation is passed to a linear prediction head. In prediction-level fusion, the static and temporal branches are selected independently on the validation split and their predicted class probabilities are averaged,

$$\mathbf{p}_{\text{avg}}^{(s)} = \frac{1}{2}\left(\mathbf{p}_{X}^{(s)} + \mathbf{p}_{\text{temp}}^{(s)}\right).$$

For each seed, the fusion strategy is chosen using the same dataset-specific validation criterion used for the other hyperparameters. No test-set information is used to choose between fusion strategies. After the fusion rule and all component hyperparameters have been selected, the configuration is frozen and evaluated once on the held-out test subjects.

All competing methods therefore use identical subject splits and the same train-only preprocessing and validation-only model-selection protocol, while retaining architecture-specific search spaces appropriate to each backbone.

B.5 Evaluation and Reproducibility

For the binary ABIDE and ADNI tasks, area under the ROC curve (AUC) is the primary evaluation metric; we additionally report accuracy, sensitivity, and specificity. For the four-class PPMI task, we use macro one-vs-rest AUC as the primary metric. Results are reported as mean$\pm$standard deviation across the five matched random seeds

$$\{42, 1042, 2042, 3042, 4042\}.$$

Because each static backbone $\mathbf{X}$ and its temporally augmented counterpart are evaluated on identical subject splits, comparisons are performed in a paired manner. For each backbone–dataset combination, we apply a two-sided paired t-test to the five seed-level performance differences. We additionally

assess the consistency of the improvement across backbone–dataset combinations using a sign test and a combination-level one-sample t-test. Complete statistical results are reported in Appendix C.1.

Our ablation studies examine two components of **TempLasso**: (i) sensitivity to temporal window length and stride, and (ii) the temporal aggregation mechanism, including order-invariant, unidirectional, shuffled-order, and ordered bidirectional sequence models. We additionally compare against direct full-backbone temporalization (**TempX**) and its fusion with the corresponding static backbone as higher-capacity temporal controls.

All models and training pipelines are implemented in PyTorch. The anonymous implementation repository includes the static and temporal FC pipelines, frozen subject splits, model and hyperparameter configurations, training and evaluation scripts, statistical-testing code, and scripts for reproducing the reported tables and ablations. All data-dependent transformations are estimated using training data only, hyperparameter and checkpoint selection use only the validation split, and the held-out test split is evaluated only after model selection is complete. Code: [1]

B.6 Computational Cost and Runtimes

Model-level complexity. After temporal FC construction and sparse edge selection, **TempLasso** operates on a sequence of L windows with K selected connections per window. For a bidirectional GRU with hidden dimension H per direction, the recurrent computation scales as

$$O\big(L(KH + H^2)\big) = O(L(K + H)H)\,.$$

Thus, TempLasso avoids applying a high-dimensional spatial encoder to each $N \times N$ temporal connectome and instead models only the selected connection trajectories.

For a static backbone **X**, let $\mathcal{C}_X(N)$ denote the cost of one forward pass on an N-ROI connectivity matrix. The static model evaluates this backbone once per subject, with cost

$$O(\mathcal{C}_X(N))\,.$$

In contrast, **TempX** applies the same backbone independently to each of the L temporal windows before sequence aggregation, producing a backbone-computation cost of approximately

$$O(L\,\mathcal{C}_X(N))\,.$$

The exact form of $\mathcal{C}_X(N)$ depends on the backbone architecture, but every evaluated static model processes the dense $N \times N$ connectivity representation.

The same distinction carries to fusion. **X+TempLasso** evaluates the static backbone once and adds the sparse temporal branch, giving

$$O\big(\mathcal{C}_X(N) + L(KH + H^2)\big)$$

model-level computation. By contrast, **X+TempX** retains the L repeated evaluations of the static backbone and therefore incurs approximately

$$O(L\,\mathcal{C}_X(N))$$

backbone computation in addition to temporal aggregation.

These expressions describe learned-model computation after FC construction. TempLasso edge selection itself begins from the $N(N-1)/2$ candidate connections, and temporal FC construction also depends on the number of ROIs. We therefore do not claim that the complete TempLasso pipeline is independent of N; rather, its advantage is that the learned temporal model operates on a sparse K-dimensional trajectory instead of repeatedly processing the full $N \times N$ connectome.

Runtimes. Table 6 reports the average training time per epoch on ABIDE. Reporting per-epoch time avoids conflating computational cost with differences in early-stopping epoch across runs. All measurements were obtained on an NVIDIA H100 NVL GPU.

The measured runtimes provide a practical complement to the complexity analysis above. In particular, they quantify the cost of the sparse TempLasso temporal branch relative to models that operate on the full connectivity representation.

[1] https://anonymous.4open.science/r/TempLasso-EA65/

Table 6: **Training time per epoch on ABIDE** measured on an NVIDIA H100 NVL GPU.

Model	Training time / epoch (s)
BQN	0.34
ALTER	2.20
BrainNetCNN	0.87
BNT	1.79
GCN	0.73
TempLasso	1.01

C Additional Results

C.1 Statistical Significance Results

This section describes the significance testing underlying Table 7 and the summary statistic reported in Section 4. All tests are two-sided and computed over 5 seeds (42, 1042, 2042, 3042, 4042), with the same seed value corresponding to the same train/test split for every backbone within a given dataset. Thus, **X** and **X+TempLasso** are always compared on matched subject splits.

Individual backbone–dataset tests. For each of the 15 evaluated backbone–dataset combinations (5 backbones on each of ABIDE, ADNI, and PPMI), we treat the 5 matched seed-level primary-metric pairs as a paired sample and apply a two-sided paired t-test with 4 degrees of freedom:

$$t = \frac{\bar{\Delta}}{s_\Delta/\sqrt{5}}, \qquad \Delta_i = M_i(\text{X+TempLasso}) - M_i(X),$$

where M_i denotes AUC for ABIDE and ADNI and macro-AUC for PPMI, and $\bar{\Delta}$ and s_Δ are the mean and sample standard deviation of the five paired differences. Results are reported in Table 7.

At this sample size, the individual paired tests have limited power, so a non-significant result should be interpreted as inconclusive rather than as evidence of no effect. We report unadjusted per-combination p-values because each backbone–dataset comparison is also of individual interest; however, these values should be interpreted alongside the aggregate analysis below rather than as multiplicity-controlled discoveries.

Overall combined test. To assess whether **X+TempLasso** improves over **X** systematically across backbones and datasets, we summarize evidence at the level of the 15 backbone–dataset combinations rather than pooling all 75 seed-level pairs. Pooling the seed-level observations would be inappropriate because the same five subject splits are reused across backbones within each dataset, so observations from different backbones are not independent.

We therefore use each backbone–dataset combination's mean improvement $\bar{\Delta}$ as one combination-level summary and apply two complementary aggregate tests.

First, a sign test considers only the direction of the mean improvement. All 15 of 15 backbone–dataset combinations show a positive mean difference in favor of **X+TempLasso**. Under the null hypothesis that positive and negative differences are equally likely, the two-sided sign-test probability of observing this result is

$$p = \frac{2}{2^{15}} \approx 6.1 \times 10^{-5}.$$

Second, a one-sample t-test on the 15 combination-level mean differences evaluates whether their average differs from zero while incorporating effect magnitude. Across the 15 combinations, the mean improvement is 4.63 AUC points, with

$$t(14) = 6.22, \qquad p = 2.2 \times 10^{-5},$$

and standardized combination-level effect size

$$d = \frac{\bar{\Delta}}{s_\Delta} = 1.61.$$

Table 7: Paired comparison of **X** vs. **X+TempLasso** across ABIDE, ADNI, and PPMI, over 5 seeds. p-values from a two-sided paired t-test (df$= 4$). p-values with < 0.05 are bolded.

ABIDE

Model	**X**	**X+TempLasso**	Δ	p
BQN	70.38 ± 1.43	73.44 ± 2.26	+3.06	**0.046**
ALTER	72.04 ± 4.25	73.50 ± 2.51	+1.46	0.179
BrainNetCNN	71.76 ± 2.18	73.42 ± 1.90	+1.66	**0.005**
BNT	67.89 ± 1.71	70.55 ± 2.42	+2.66	0.059
GCN	66.94 ± 2.62	71.88 ± 2.25	+4.94	**0.003**

ADNI

Model	**X**	**X+TempLasso**	Δ	p
BQN	79.80 ± 4.37	82.82 ± 6.89	+3.02	0.371
ALTER	79.90 ± 5.68	81.98 ± 9.67	+2.08	0.410
BrainNetCNN	83.64 ± 3.48	84.60 ± 4.51	+0.96	0.646
BNT	75.42 ± 6.11	83.65 ± 6.84	+8.23	0.071
GCN	78.24 ± 8.05	82.00 ± 7.43	+3.76	0.393

PPMI

Model	**X**	**X+TempLasso**	Δ	p
BQN	55.40 ± 4.96	62.48 ± 7.68	+7.08	0.169
ALTER	52.51 ± 4.27	60.00 ± 4.77	+7.49	**0.012**
BrainNetCNN	56.48 ± 4.33	61.94 ± 7.81	+5.46	0.070
BNT	51.50 ± 8.60	61.87 ± 8.03	+10.38	**0.004**
GCN	54.12 ± 7.33	61.34 ± 6.86	+7.22	**0.046**

Both aggregate tests provide evidence of a systematic positive effect ($p < 0.001$). The sign test depends only on the direction of the combination-level effects, whereas the one-sample t-test additionally uses their magnitudes. Their agreement is consistent with the observed pattern that temporal augmentation produces a positive mean improvement for every evaluated backbone–dataset combination. Because backbones within a dataset share the same subject splits, these aggregate tests are intended as complementary summaries of the cross-model pattern rather than as a substitute for the matched per-combination analyses.

C.2 Results on Temporalizing Strategy

Table 8 reports the complete mean±std results for the comparison between **X**, **TempX**, **X+TempX**, and **X+TempLasso**. The higher-capacity **TempX** variants are not uniformly worse: in several cases, particularly on ADNI, **X+TempX** is competitive with or slightly better than **X+TempLasso**. The broader pattern, however, is substantially less stable. Standalone **TempX** often falls below the corresponding static backbone, and fusion with **TempX** can either help or hurt depending on the architecture and dataset. By contrast, **X+TempLasso** improves the corresponding static model in every evaluated backbone–dataset combination.

These results suggest that short-window FC should not be treated simply as a smaller version of the full-scan connectome. Local FC estimates are formed from far fewer observations and are consequently noisier, so an encoder designed for the more stable full-scan matrix may not be the best inductive bias when applied independently to every window. **TempLasso** instead imposes a strong bottleneck by following only a sparse set of task-relevant connections through time. The experiments do not imply that low-capacity temporal models are universally preferable, but they show that increased per-window expressiveness alone does not guarantee better use of temporal information. This distinction helps explain why temporal access can be valuable while direct full-backbone temporalization remains less reliable in finite-sample settings.

Table 8: Full results comparing static backbones (**X**), full-backbone temporalization (TempX), temporal fusion (**X+TempX**), and **X+TempLasso**, across ABIDE, ADNI, and PPMI. Standalone **TempLasso** is reported once per dataset. Values are mean ± std (ddof=1) over 5 seeds; – denotes runs not yet completed.

ABIDE

Model	X	TempX	X+TempX	X+TempLasso
TempLasso	71.91±1.81	–	–	–
BQN	70.38±1.43	65.60±8.59	69.22±2.59	73.44±2.26
ALTER	72.04±4.25	67.78±4.36	71.64±3.45	73.50±2.51
BrainNetCNN	71.76±2.18	69.72±2.73	72.56±2.26	73.42±1.90
BNT	67.89±1.71	64.75±2.96	67.36±1.28	70.55±2.42
GCN	66.94±2.62	67.58±3.18	68.36±2.78	71.88±2.25

ADNI

Model	X	TempX	X+TempX	X+TempLasso
TempLasso	80.52±9.70	–	–	–
BQN	79.80±4.37	76.68±4.89	80.82±6.93	82.82±6.89
ALTER	79.90±5.68	81.24±3.13	83.34±3.54	81.98±9.67
BrainNetCNN	83.64±3.48	82.28±11.45	85.08±8.09	84.60±4.51
BNT	75.42±6.11	70.83±16.90	76.77±9.51	83.65±6.84
GCN	78.24±8.05	75.52±4.50	77.18±5.62	82.00±7.43

PPMI

Model	X	TempX	X+TempX	X+TempLasso
TempLasso	62.98±7.42	–	–	–
BQN	55.40±4.96	56.00±10.44	57.80±10.43	62.48±7.68
ALTER	52.51±4.27	51.57±9.20	53.09±5.55	60.00±4.77
BrainNetCNN	56.48±4.33	52.75±2.47	56.53±5.57	61.94±7.81
BNT	51.50±8.60	59.08±4.58	55.14±7.29	61.87±8.03
GCN	54.12±7.33	53.00±5.56	55.27±5.40	61.34±6.86

C.3 Ablations

Temporal Aggregator Ablation. Table 9 isolates the contribution of the BiGRU itself, holding everything else in **TempLasso** fixed. All four aggregators operate on the same lasso-selected edges and the same windowed FC sequence; only the model applied to that sequence changes. The MLP baseline applies a shared MLP to each window's edge vector independently and mean-pools across windows, making it invariant to window order. The unidirectional GRU processes the sequence in its true order but without backward context. The Shuffled BiGRU uses the same architecture and parameter count as **TempLasso**'s BiGRU, but each seed's window sequence is randomly permuted before being passed to the model, so it retains full model capacity while losing access to the true

Table 9: **Temporal aggregator ablation.** Mean AUC (%) ± std over 5 seeds. Comparing sequence models applied to the same selected edge trajectories, holding edge selection and windowing fixed. "Uses chronological order?" indicates whether the model is sensitive to the true temporal ordering of the windows (Shuffled BiGRU retains the BiGRU's capacity but destroys ordering by permuting windows). Best result per column in bold.

Temporal Aggregator	Uses chronological order?	ABIDE	PPMI
MLP + Mean Pooling	No	70.18±1.92	58.56±6.97
GRU	Yes	69.24±2.18	60.80±8.08
Shuffled BiGRU	No	68.16±1.87	58.80±8.36
BiGRU (TempLasso)	**Yes**	**71.91±1.81**	**62.98±7.42**

Table 10: **Window/stride sensitivity for TempLasso.** Mean ± std AUC over 5 seeds. The main configuration (44s/22s) is used throughout the paper.

Window / Stride	Overlap	ABIDE AUC (%)	PPMI AUC (%)
32s / 32s	0%	70.55 ± 3.21	59.14 ± 6.92
32s / 16s	50%	70.57 ± 2.88	58.83 ± 8.88
44s / 22s (main)	50%	**71.91 ± 1.81**	**62.98 ± 7.42**
60s / 30s	50%	69.22 ± 3.53	55.59 ± 8.45

temporal ordering. This isolates ordering from capacity: any gap between the BiGRU and the Shuffled BiGRU reflects the value of correct ordering alone, not a larger or more expressive model.

The ordered BiGRU outperforms all three alternatives on both cohorts. Relative to the Shuffled BiGRU, which matches it in capacity exactly, the gap is 3.75 AUC points on ABIDE and 4.18 on PPMI, indicating that the gain comes from the true trajectory rather than from model size. Relative to the unidirectional GRU, the gap is 2.68 and 1.49 points, attributable to bidirectional context. The order-invariant MLP baseline is the least consistent of the three alternatives: it exceeds both the GRU and the Shuffled BiGRU on ABIDE, but falls behind both on PPMI. This suggests that a sequence-capable model given the wrong order is not uniformly better than a model that makes no use of order at all — capacity is not a substitute for the correct representation, echoing the paper's broader argument.

Window and Stride Sensitivity. Section 3.1 fixes the sliding-window length and stride at 44s/22s (50% overlap), following the convention of Allen et al. (2014b); this choice was made *a priori* from prior dynamic-FC literature and was not tuned on ABIDE or PPMI. Here we test how sensitive **TempLasso** is to this choice by sweeping window length and overlap while holding every other component of the pipeline — edge selection, the BiGRU, and all training hyperparameters — fixed. We compare the main configuration against a shorter, non-overlapping window (32s/32s, 0% overlap), the same 32s window at 50% overlap, and a longer window (60s/30s, 50% overlap), on ABIDE and PPMI.

The main 44s/22s configuration performs best on both cohorts. Relative to this configuration, both 32s settings show lower performance: 32s/32s decreases AUC by 1.36 points on ABIDE and 3.84 points on PPMI, while 32s/16s decreases AUC by 1.34 and 4.15 points, respectively. The near-identical performance of the two 32s settings despite their different overlap (0.02 AUC points on ABIDE and 0.31 on PPMI) suggests that, at this window length, overlap has relatively little effect. Performance drops most noticeably at 60s/30s, by 2.69 AUC points on ABIDE and 7.39 points on PPMI relative to the main configuration, consistent with longer windows averaging over more of the signal and reducing the short-timescale connectivity variation that **TempLasso** is designed to track. The main configuration was selected from prior literature rather than tuned on these datasets, so we interpret its strong performance as evidence that a reasonable literature-standard window choice is effective, rather than as a claim that 44s/22s is uniquely optimal.

As with the other ablations in this appendix, this comparison is run at $n = 5$ seeds per configuration without paired significance testing, and the gap between the two 32s settings in particular (0.02 AUC points on ABIDE and 0.31 on PPMI) is well within seed noise and should not be read as evidence for or against window overlap specifically. The degradation at 60s/30s is the most pronounced and consistent pattern across both cohorts, although we do not assign statistical significance to this ablation at $n = 5$.

D Temporal Information and the Limits of Static FC

This section formalizes the representational argument behind Theorem 3.1. The key point is independent of any particular temporal architecture: static functional connectivity is a deterministic compression of the original BOLD signal. Consequently, a predictor that receives the temporal signal has access to every decision rule available to a static-FC predictor, together with additional decision rules that distinguish signals collapsed to the same static connectome.

D.1 Static FC is a Many-to-One Temporal Compression

Let $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_N]^\top \in \mathbb{R}^{N \times T}$ denote the ROI-level BOLD signal, and let $A(\mathbf{S}) = \mathbf{C}_{\text{stat}}(\mathbf{S})$ denote the static FC map. For concreteness, assume that $\mathbf{C}_{\text{stat}}$ is obtained from pairwise Pearson correlations over the complete recording.

Static FC depends on the collection of samples observed over the scan, but not on their common temporal ordering. This immediately implies that the map A is non-injective.

Proposition D.1 (Temporal permutation invariance)**.** *Let π be any permutation of the T time indices, applied identically to every ROI. Let $\mathbf{S}_\pi$ denote the resulting BOLD signal. Then*

$$\mathbf{C}_{\text{stat}}(\mathbf{S}_\pi) = \mathbf{C}_{\text{stat}}(\mathbf{S}).$$

Therefore, distinct temporal signals can produce exactly the same static FC matrix.

Proof. For ROIs i and j, the Pearson correlation is

$$C_{ij} = \frac{\sum_{t=1}^{T}(s_{i,t} - \bar{s}_i)(s_{j,t} - \bar{s}_j)}{\sqrt{\sum_{t=1}^{T}(s_{i,t} - \bar{s}_i)^2}\sqrt{\sum_{t=1}^{T}(s_{j,t} - \bar{s}_j)^2}}.$$

Applying the same permutation π to the time index only reorders the terms appearing in each sum. The ROI means, covariance term, and marginal variance terms are therefore unchanged. Hence every pairwise correlation is unchanged, and so $\mathbf{C}_{\text{stat}}(\mathbf{S}_\pi) = \mathbf{C}_{\text{stat}}(\mathbf{S})$.

Whenever $\mathbf{S}_\pi \neq \mathbf{S}$, this gives two distinct temporal signals with identical static FC. □

The proposition is intentionally simple, but it exposes the essential limitation. A static-FC model cannot distinguish two recordings that differ only in temporal organization if their full-scan pairwise statistics are identical. Increasing the capacity of the downstream model does not change this fact: every such model receives exactly the same input for both recordings.

Sliding-window FC does not preserve the complete BOLD signal, but it can break some of these equivalences. For example, the two trajectories $\mathbf{C}_A \to \mathbf{C}_B$ and $\mathbf{C}_B \to \mathbf{C}_A$ may have similar or identical full-scan summaries while remaining distinguishable as ordered temporal sequences.

D.2 Proof of Theorem 3.1

We now express the same idea at the level of optimal prediction risk.

Let $\mathcal{F}_{\text{temp}}$ denote the class of all measurable predictors that may use the temporal signal $\mathbf{S}$, and let $\mathcal{F}_{\text{stat}}$ denote predictors restricted to the static FC matrix. Define

$$R^*_{\text{temp}} = \inf_{f \in \mathcal{F}_{\text{temp}}} \mathbb{E}\left[\ell(Y, f(\mathbf{S}))\right], \quad \text{and} \quad R^*_{\text{stat}} = \inf_{g \in \mathcal{F}_{\text{stat}}} \mathbb{E}\left[\ell\left(Y, g(\mathbf{C}_{\text{stat}}(\mathbf{S}))\right)\right].$$

Theorem 3.1. *Let $\mathbf{S}$ denote the BOLD signal, $\mathbf{C}_{\text{stat}}(\mathbf{S})$ its static FC matrix, and Y the prediction target. For any loss function ℓ, the Bayes-optimal risks satisfy $R^*_{\text{temp}} \leq R^*_{\text{stat}}$, where R^*_{stat} restricts prediction to functions of $\mathbf{C}_{\text{stat}}(\mathbf{S})$. The inequality is strict whenever the Bayes-optimal decision cannot be expressed as a function of $\mathbf{C}_{\text{stat}}(\mathbf{S})$ alone on a set of positive probability.*

Proof of Theorem 3.1. Because $\mathbf{C}_{\text{stat}}(\mathbf{S})$ is a deterministic function of $\mathbf{S}$, every predictor based on static FC can also be implemented by a predictor with access to the temporal signal.

Indeed, for any static predictor g, define $f_g(\mathbf{S}) = g\left(\mathbf{C}_{\text{stat}}(\mathbf{S})\right)$.

Then $f_g \in \mathcal{F}_{\text{temp}}$ and $\mathbb{E}\left[\ell(Y, f_g(\mathbf{S}))\right] = \mathbb{E}\left[\ell\left(Y, g(\mathbf{C}_{\text{stat}}(\mathbf{S}))\right)\right]$.

Thus, the set of risks attainable by static-FC predictors is contained within the set attainable by predictors with temporal access. Taking the infimum over the two classes gives

$$R^*_{\text{temp}} \leq R^*_{\text{stat}}.$$

The inequality is strict whenever the Bayes-optimal decision cannot be expressed as a function of $\mathbf{C}_{\mathrm{stat}}(\mathbf{S})$ alone on a set of positive probability. Equivalently, strictness occurs when there exist temporal signals that are identified by static FC but require different optimal decisions under the loss ℓ. In that case, every static predictor must assign the same decision to those signals, whereas a predictor with temporal access can distinguish them, yielding strictly lower expected risk. □

The weak inequality requires no assumption about the architecture used for temporal prediction. It follows only from the fact that static FC is computed from the temporal signal.

D.3 When Is the Inequality Strict?

The strict case is easiest to understand through the fibers of the static-FC map. For a given connectome $\mathbf{C}$, define $A^{-1}(\mathbf{C}) = \{\mathbf{S} : \mathbf{C}_{\mathrm{stat}}(\mathbf{S}) = \mathbf{C}\}$.

Every signal in this set is indistinguishable to a static model, regardless of how different their temporal organization may be.

If all signals within each such set have the same Bayes-optimal prediction, then static FC is sufficient for the task and equality may hold: $R^*_{\mathrm{temp}} = R^*_{\mathrm{stat}}$.

Temporal information becomes useful when this is not true. In particular, suppose that two groups of recordings can share the same static FC representation but have different conditional target distributions because of differences in their temporal evolution. A static model must collapse these cases, whereas a temporal model may separate them. This is precisely the situation in which

$$R^*_{\mathrm{temp}} < R^*_{\mathrm{stat}}.$$

For binary classification under 0–1 loss, one concrete example is a static-FC fiber containing temporal signals $\mathbf{S}_a$ and $\mathbf{S}_b$ such that $\mathbf{C}_{\mathrm{stat}}(\mathbf{S}_a) = \mathbf{C}_{\mathrm{stat}}(\mathbf{S}_b)$, but

$$P(Y = 1 \mid \mathbf{S}_a) > \frac{1}{2}, \qquad P(Y = 1 \mid \mathbf{S}_b) < \frac{1}{2}.$$

The Bayes-optimal temporal classifier assigns different labels to the two signals, while every static classifier must assign the same label because its inputs are identical.

D.4 Representation Versus Model Capacity

The theorem separates two distinct sources of predictive power: *information available in the representation* and *capacity of the model operating on that representation*.

A highly expressive static model computes a potentially complicated function $g(\mathbf{C}_{\mathrm{stat}})$, but the function still depends only on $\mathbf{C}_{\mathrm{stat}}$. If two BOLD recordings are mapped to the same static connectome, then every deterministic static model, regardless of depth or architectural complexity, must produce the same representation and prediction for those recordings.

In contrast, a temporal model receives a representation that may distinguish these recordings before prediction begins. It may therefore achieve better performance even with substantially lower architectural capacity.

This distinction motivates the central empirical comparison in our paper:

high-capacity model + static FC	versus	simple model + temporal FC.

Our theoretical result does not predict which finite-sample model will win this comparison. It establishes only that additional capacity on top of static FC cannot recreate temporal distinctions that static aggregation has removed.

D.5 Why More Expressive Temporal Models Need Not Be Better

The information inequality also should not be interpreted as saying that every model using temporal FC must outperform every static model. The Bayes risk assumes access to the best possible decision rule, whereas practical models are trained from finite, noisy datasets.

Windowed FC introduces its own estimation difficulty. Each local connectivity matrix is computed from substantially fewer observations than the full-scan matrix and is consequently noisier. A high-capacity encoder applied independently to every local matrix may therefore have to learn from a sequence of high-dimensional, noisy estimates.

This provides a possible explanation for the **TempX**-versus-**TempLasso** comparison in Section 4: **TempX** applies a high-capacity encoder to every local FC matrix, providing access to the full windowed connectome rather than only the sparse set of connections retained by TempLasso, yet this additional per-window modeling capacity does not translate into consistently better finite-sample performance.

A compact temporal representation can act as an information bottleneck, retaining task-relevant temporal variation while avoiding the burden of modeling every local connectome in full.

Accordingly, the theoretical result supports the motivation for temporal access, but the appropriate inductive bias for exploiting that access remains an empirical question.

D.6 What the Result Does Not Claim

For clarity, Theorem 3.1 makes no claim that

- every temporal representation contains useful information for every prediction task;
- every temporal architecture will outperform a static architecture;
- sliding-window FC preserves all information in the original BOLD signal; or
- static FC is an inadequate representation in every setting.

Equality is entirely possible when the prediction target depends only on information retained by static FC. The theorem instead establishes a one-sided limitation: static aggregation cannot increase the information available for prediction, and any temporally informative distinctions that it removes cannot subsequently be recovered merely by increasing the complexity of the static model.

Our experiments test whether this theoretical possibility has practical consequences in resting-state fMRI classification.