

A Technical Appendices and Supplementary Material

A.1 Dataset Processing

Allen Human Brain Atlas. Transcriptomic data is preprocessed from the Allen Human Brain Atlas (AHBA) using the *abagen* toolbox [53], following a set of best-practice steps to standardize expression across donors and reduce noise from technical and biological sources. Unless otherwise specified, all processing steps follow established workflows [24, 3, 23].

Within-Donor Normalization. Gene expression values for all tissue samples are normalized within each donor to account for individual differences in signal scale or distribution. A scaled robust sigmoid transformation is applied to each gene expression vector x_g , defined across regions for a given donor:

$$x_{\text{norm}} = \frac{1}{1 + \exp\left(-\frac{x_g - \text{median}(x_g)}{\text{IQR}(x_g)}\right)}$$

This transformation preserves rank while being robust to outliers. A subsequent min-max rescaling step standardizes the values to the unit interval:

$$x_{\text{scaled}} = \frac{x_{\text{norm}} - \min(x_{\text{norm}})}{\max(x_{\text{norm}}) - \min(x_{\text{norm}})}$$

This two-step procedure is distribution-free and avoids assumptions of normality, unlike traditional z-scoring.

Sample-to-Region Assignment. Tissue samples are assigned to regions in a reference parcellation using their MNI coordinates, constrained by hemisphere and cortical/subcortical classification. Expression values for samples mapping to the same region are averaged, resulting in a subject-specific region-by-gene matrix of shape $r \times g$.

Gene Selection and Filtering. To reduce noise and retain biologically informative genes, we follow a multi-stage filtering process. First, genes expressed below background noise are excluded. Next, we quantify spatial reproducibility via differential stability, defined as:

$$\Delta_S(g) = \frac{1}{\binom{N}{2}} \sum_{i=1}^{N-1} \sum_{j=i+1}^N r(B_i(g), B_j(g))$$

where $B_i(g)$ is the regional expression profile of gene g for donor i , and $N = 6$ is the number of AHBA donors. Genes with high $\Delta_S(g)$ show consistent regional variation across brains. We use the strictest threshold recommended for the Schaefer-400 parcellation, retaining a final set of 7,380 genes per region.

Cross-Donor Aggregation. After normalization and filtering, gene expression values are averaged across donors to produce a population-level matrix. To address incomplete sampling across hemispheres, we employ *abagen*’s mirror-and-interpolate strategy: missing regions are imputed using their contralateral counterparts and smoothed via weighted averaging over the 10 nearest neighbors in the source hemisphere. This yields a complete $r \times g$ matrix aligned to the target parcellation, suitable for downstream modeling.

See [53] for licensing.

UK Biobank. rs-fMRI scans were acquired on a Siemens Skyra 3T scanner (TR = 735 ms, TE = 39 ms, multiband factor = 8, 2.4 mm isotropic resolution, scan duration = 6:10 min) [19]. Preprocessing followed the UKBB minimal pipeline [19] and was further refined using XCP-D [54], implemented in Nipype [55]. This included the removal of non-steady-state volumes, notch filtering of motion regressors, global signal regression, despiking via 3dDespike, and bandpass filtering (0.01–0.1 Hz) applied to both data and confounds [56, 57]. Population averaging followed the parcellation of individual rs-fMRI scans into connectivity matrices. The S456 atlas used can be found at https://github.com/PennLINC/AtlasPack/blob/main/tpl-fsLR_atlas-4S456Parcels_dseg.json.

Human-Connectome-Project Young Adult (HCP-YA). The HCP-YA dataset contains rs-fMRI data processed identically to the UKBB dataset in the S456 parcellation. Despite substantial demographic shift (n=1065; mean age = 29; age range = 22–35; 54% female) between cohorts, we

observe a stable backbone connectivity structure in the population average connectomes between UKBB and HCP (Pearson- $r = 0.90$, Figure 4). Further details for the HCP-YA cohort can be found at <https://pennlinc.github.io/AI2D/docs/datasets/HCP-YA/> [20, 58, 59].

Max Planck Institut Leipzig Mind-Brain-Body Dataset (MPI-LEMON). MPI-LEMON is a comprehensive neuroimaging dataset ($n = 136$; age range = 20-30; 72% male). The MPI dataset uses an unsupervised voxel-level clustering approach to generate parcellations at various scales including 183, 391, and 729 region resolutions from which functional connectivity can be computed. MPI-LEMON comes with paired, minimally processed AHBA gene expression data at each resolution. Processing was done with the recommended *abagen* [53] parameters. Bi-hemispheric imputation was not done for missing regions in this dataset, which differs from the paired AHBA dataset used for UKBB. See Jimenez-Marin et al. [21] for further details, licensing, and data access.

A.2 Experimental Details

Human reference genome ordering. Genes are sorted by their transcription start site (TSS) in the SMT tokenization procedure. For each protein-coding gene, the earliest TSS is selected across all transcripts, regardless of strand orientation. Genes are then ordered globally—first across chromosomes, then within each chromosome—yielding a biologically grounded input sequence. This ordering is intended to enhance interpretability and biological plausibility for SMT. The reference genome used is available at ncbi.nlm.nih.gov/datasets/genome/GCF_000001405.40. Implementation details are provided in `GeneEx2Conn/data/enigma/gene_lists/human_refgenome_README.md`.

Parcel centroids. Spatial coordinates for brain regions are used throughout our analyses. For each parcellation, we determine the centroid of each anatomical region using provided metadata files. Coordinates are defined in the Montreal Neurological Institute (MNI) standard space. For the MPI-LEMON dataset, these centroids are available via the original release [21]. For the UKBB dataset, we provide a reformatted metadata file directly at `/data/UKBB/atlas-4S456Parcels_dseg_reformatted.csv`. Centroids for this custom parcellation were computed using `nilearn.plotting.find_parcellation_cut_coords`.

Stratified metrics ranges. Brain region centroids serve two main purposes in our analyses: (1) as input features for models incorporating spatial information, such as the SMT with [CLS] token, and (2) to stratify model performance by inter-regional distance. We define short-, mid-, and long-range Pearson correlations by dividing the brain’s maximum inter-regional distance (180 mm) into three equal intervals: short-range (< 60 mm), mid-range (60–120 mm), and long-range (120–180 mm). In addition to these distance-stratified metrics, we also compute performance over connection strength categories: strong positive ($r > 0.3$), strong negative ($r < -0.3$), and weak connections ($-0.3 \leq r \leq 0.3$).

A.3 Model Optimization

Hyperparameter selection. To select the optimal hyperparameter configurations for each model in Table 1 and Table 2, we perform nested inner cross-validation using random sampling from a hyperparameter grid defined from wider grid searches and manual fine-tuning. For each model, 3–6 hyperparameter combinations are sampled, with the exact number depending on the model’s complexity and the size of the grid. Each model also contains a best default parameters configuration, which is guaranteed to be sampled in addition to the randomly sampled configurations. The combination yielding the lowest validation loss on a dedicated subset of validation nodes, is then used for full training and performance evaluation on the test set. Full hyperparameter grids for all models are available in our repo at `/GeneEx2Conn/models/configs`. All experiments are run on A100 or H100 NVIDIA GPUs on a high-performance compute cluster, taking no more than 1 hour per fold.

Optimization details. All gradient-based models, including both linear baselines and MLP architectures, are trained using the AdamW optimizer. The mean squared error (MSE) between predicted and ground-truth connectivity values are used as the training loss. We perform model selection based on validation-set performance across cross-validation folds. All model architectures and hyperparameter configurations can be found in our repo, github.com/neuroinfolab/GeneEx2Conn, under `/models/configs`.

Rules-based baselines. We adopt several simple rules-based baselines for predicting inter-regional connectivity Y_{ij} from spatiomolecular properties. The *exponential decay* model assumes decay of connectivity strength with Euclidean distance d_{ij} , given by $Y_{ij} = \text{SA}_\infty + (1 - \text{SA}_\infty) \exp(-d_{ij}/\text{SA}_\lambda)$, where SA_∞ and SA_λ control the asymptotic offset and decay rate [28].

Bilinear Connectome Model. Optimizing gene-gene interaction weights O is challenging due to its quadratic scaling with the number of genes and the large combinatorial space it spans. Kovács et al. [11] restrict their analysis to a subset of 19 innexin genes and estimate O by minimizing $\|\text{vec}(Y) - (X \otimes X)\text{vec}(O)\|_2^2 + \alpha\|\text{vec}(O)\|_2^2$. We adopt a more scalable formulation that minimizes $\min_O \|Y - XOX^\top\|_F^2 + \lambda\|O\|_F^2$, using a closed-form solution described in Section A.4, enabling efficient learning of interaction matrices on the full gene set or on dimensionality-reduced features.

Bilinear Low-rank Model. To reduce parameter count and enforce symmetry in the predicted connectome, we further simplify the bilinear model by exploiting the symmetry of Y , and optimizing a shared projection matrix $\hat{E} \in \mathbb{R}^{d \times k}$. This yields the objective $\min_{\hat{E}} \|Y - X\hat{E}\hat{E}^\top X^\top\|_F^2 + \lambda\|\hat{E}\|_F^2$, which preserves representational power while significantly lowering computational complexity.

Partial Least Squares (PLS). PLS simultaneously learns projections for inputs X and outputs Y to maximize covariance in a shared latent space, providing a powerful framework for decomposing high-dimensional, collinear data into low-dimensional latent components [60]. The general form of the PLS model can be written as $X = TP^\top + F$ and $Y = UQ^\top + E$, where $P \in \mathbb{R}^{g \times k}$ and $Q \in \mathbb{R}^{r \times k}$ are projection matrices, $T, U \in \mathbb{R}^{r \times k}$ are the corresponding score matrices, and E, F are error terms. Each latent component k of P and Q is learned by maximizing $\max_{p,q} (Xp)^\top (Yq)$, followed by solving the regression $T\beta = U$, yielding the overall model $Y = XP^\top\beta Q^\top$.

PLS-based encoder-decoder adaptation. To adapt PLS for our edge-wise prediction task, we reformulate it into an encoder-decoder framework. We first use its learned projections to create region-level embeddings $t_i = XP^\top$ from P learned on a training set $(X_{\text{train}}, Y_{\text{train}})$. Connection strength is then predicted via a bilinear decoder by minimizing the mean squared error between predicted and observed connectivities, $\min_O \frac{1}{r^2} \sum_{i,j=1}^r (Y_{i,j} - t_i^\top O t_j)^2$. The latent dimensionality k and regularization parameter λ are selected via cross-validation.

Multilayer Perceptron. The input to the fully-connected MLP is of shape 2×7380 with 2-4 hidden layers. Each hidden layer includes batch normalization, dropout, and ReLU activations. The model is trained using the AdamW optimizer with weight decay for regularization. All hyperparameters, including network depth, hidden layer size, dropout rate, and learning rate, are selected via cross-validation to mitigate overfitting given the high-dimensional input and large parameter space.

A.4 Spatiomolecular Null Shuffle Details

Implementation. Null brain maps were developed to address inflated false-positive rates that arise when comparing spatially structured brain data. Instead of naive permutations, spatially informed spin methods generate surrogate maps that preserve spatial autocorrelation while disrupting correspondence with true data, enabling statistical null distributions that isolate the contribution of spatial structure.

We adopt the spin-based procedure of Váša et al. [43], which projects cortical parcel centroids onto a sphere, applies a random rotation, and reassigns each parcel’s data to the region it lands on. This method preserves inter-hemispheric contiguity and has been shown to be robust across parcellation resolutions [42]. For subcortical structures, we instead apply the spin directly in coordinate space for the subcortex and cerebellum, ensuring bilateral symmetry, to overcome the non-spherical topology of the subcortex.

Even with modifications for subcortical structures, a naive spatial spin, though preserving distance-based spatial autocorrelation, may disrupt the genetic autocorrelation structure of the transcriptome. Genetic autocorrelation reflects the principle that regions with similar gene expression profiles tend to exhibit stronger functional connectivity. This pattern partially overlaps with spatial proximity, but may include more complex distance-based molecular relationships.

To quantify transcriptomic autocorrelation, we examine how correlated gene expression (CGE) decays with distance. We bin all region pairs into 5mm intervals based on Euclidean distances and compute the mean CGE within each bin, yielding a CGE–distance decay curve. The canonical curve observed in human data [61] and across species [6] typically follows an exponential decline. Notably, we observe a secondary rise in CGE around 120mm, likely driven by long-range associations between cytoarchitectonically similar regions. Negative CGE values at larger distances often correspond to cortico-subcortical interactions [23].

Algorithm 1 CGE-Matched Spatial Null Brain Map Generation

- 1: **Input:** True transcriptome $X^{\text{true}} \in \mathbb{R}^{r \times g}$, region coordinates $C \in \mathbb{R}^{r \times 3}$, number of spins N , number of top spins to return K
- 2: **Define:** Exponential CGE curve $f(d) = \text{SA}_{\infty} + (1 - \text{SA}_{\infty})e^{-d/\text{SA}_{\lambda}}$
- 3: **Define:** 3rd-order polynomial CGE curve $g(d) = a_1 d^3 + a_2 d^2 + a_3 d + a_4$
- 4: Compute true CGE parameters $(\text{SA}_{\lambda}^{\text{true}}, \text{SA}_{\infty}^{\text{true}}, \{a_1^{\text{true}}, \dots, a_4^{\text{true}}\})$ from X^{true}
- 5: **for** $i = 1$ to N **do**
- 6: **(a) Cortical Spin:** Project cortical coordinates C_{ctx} to the unit sphere
- 7: Generate rotation matrix $R_i \in \text{SO}(3)$ and compute $C_i^{\text{rot}} = C_{\text{ctx}} R_i$
- 8: Construct bijective mapping $\pi_i^{\text{ctx}} : \{1, \dots, r_{\text{ctx}}\} \rightarrow \{1, \dots, r_{\text{ctx}}\}$ based on Váša et al. [43]:
 - Initialize unassigned set $\mathcal{U} = \{1, \dots, r_{\text{ctx}}\}$
 - While $\mathcal{U} \neq \emptyset$:
 1. Select the most distant pair $(p, q) \in \mathcal{U}$
 2. Assign: $\pi_i^{\text{ctx}}(p) = \arg \min_j \|C_i^{\text{rot}}[p] - C_{\text{ctx}}[j]\|_2$, likewise for q
 3. Remove p, q from \mathcal{U}
- 9: **(b) Subcortical Spin:** Define π_i^{sub} by applying a symmetric spin on subcortical coordinates
- 10: Construct null matrix: $X_i^{\text{null}} = \text{concat}(X_{\text{ctx}}[\pi_i^{\text{ctx}}], X_{\text{sub}}[\pi_i^{\text{sub}}])$
- 11: Compute CGE vs. distance from X_i^{null} and C
- 12: Fit $f_i(d)$ to extract $(\text{SA}_{\lambda}^i, \text{SA}_{\infty}^i)$
- 13: Fit $g_i(d)$ to obtain $(a_1^i, a_2^i, a_3^i, a_4^i)$
- 14: Compute total reassignment cost $c_i = \sum_{j=1}^{r_{\text{ctx}}} \|C_{\text{ctx}}[j] - C_{\text{ctx}}[\pi_i^{\text{ctx}}(j)]\|_2$
- 15: **end for**
- 16: Standardize each CGE parameter and cost across null spins using z-scores
- 17: Compute total standardized error for each spin i :

$$e_i = |z(\text{SA}_{\lambda}^i) - z(\text{SA}_{\lambda}^{\text{true}})| + |z(\text{SA}_{\infty}^i) - z(\text{SA}_{\infty}^{\text{true}})| + \sum_{j=1}^4 |z(a_j^i) - z(a_j^{\text{true}})| + |z(c_i) - z(c^{\text{true}})|$$

- 18: Rank spins by ascending e_i

- 19: **Return:** Top K lowest-error null transcriptomes $\{X_{i(1)}^{\text{null}}, \dots, X_{i(K)}^{\text{null}}\}$
-

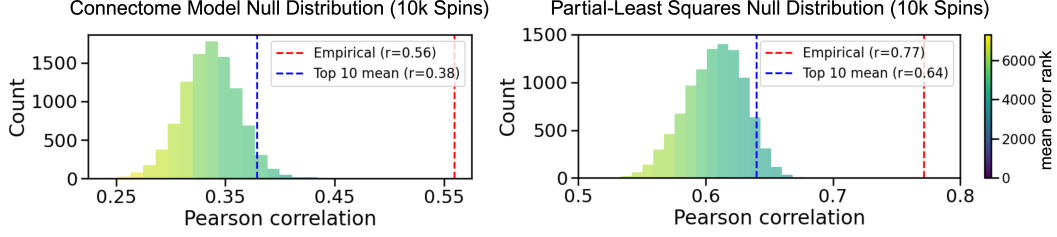


Figure 6: **Null distributions of Connectome Model and Partial Least Squares.** Each histogram shows the distribution of Pearson correlation between predicted and observed connectomes across 10,000 spatially spun/shuffled gene expression matrices. Left: Connectome Model (CM). Right: Partial Least Squares (PLS). Bars are colored by the average mean error rank across 25 bins (darker = better rank). Red dashed lines mark empirical performance on the true gene expression matrix; blue dashed lines indicate the mean performance using the 10 lowest error nulls.

Figure 3 illustrates that spin-based nulls generated by Algorithm 1 produce highly variable CGE-distance curves, ranging from biologically realistic exponential decays to distorted, spatially disordered profiles. To address this, we implement a rejection sampling procedure that selects null transcriptomes best preserving both spatial and transcriptomic autocorrelation, enabling stringent null testing without requiring full null distributions. The left panel of Figure 3 shows the empirical CGE decay from the true brain, while representative outcomes of our procedure are labeled as “*Low-error null spin*” and “*High-error null spin*”.

In our experiments, we set $N = 10,000$ spatial spins and select the top $K = 10$ based on Algorithm 1. The full implementation is provided in our repository at `/sim/null.py` within the `generate_null_spins()` function. The procedure for fitting the exponential decay and polynomial parameters are also outlined in this Python file.

Effect of mean error rank. Despite the mismatch between features and targets introduced by spatial spinning, we hypothesize that gene expression inputs preserving realistic spatial and genetic structure will yield more realistic connectome predictions. If autocorrelation is indeed predictive of connectivity, null brain transcriptome datasets with more realistic CGE profiles should result in predictions closer to the true value.

We test this by fitting two linear models—the Connectome Model (CM) and Partial Least Squares (PLS)—using 10,000 spatiomolecular null spun/shuffled brains. For each spin, we fit the models using the null gene expression matrix X' and evaluate the Pearson correlation between predicted \hat{Y} and observed Y . For PLS, we use 10 latent components (selected via grid search over 0–25 dimensions selecting the elbow point based on Pearson r); for CM, we use PCA-reduced gene expression with 27 components (capturing 95% of the variance), enabling efficient training via the closed-form solution outlined below. We train and test on the full dataset as is standard for generating null distributions under null spin tests [42].

Figure 6 indeed confirms that model performance across spins is sensitive to the quality of the CGE profile. Histograms of prediction accuracy (Pearson- r) reveal that the top 10 most CGE-consistent nulls produce better fits than the bulk of nulls. This effect is consistent across both models, and supports the idea that realistic gene-distance relationships are a critical determinant of null predictive accuracy.

Closed form solution for the Connectome Model. We solve the ridge-regularized bilinear regression problem:

$$\min_O \|Y - XOX^\top\|_F^2 + \lambda \|O\|_F^2,$$

where $X \in \mathbb{R}^{n \times d}$, $Y \in \mathbb{R}^{n \times n}$, $O \in \mathbb{R}^{d \times d}$, and $\lambda > 0$. λ is set following Kovács et al. [11]. Using $\|A\|_F^2 = \text{Tr}(A^\top A)$, we expand the objective:

$$\mathcal{L}(O) = \text{Tr}(Y^\top Y) - 2 \text{Tr}(OX^\top YX) + \text{Tr}(O^\top X^\top XOX^\top X) + \lambda \text{Tr}(O^\top O).$$

Letting $A = X^\top X$ and $M = X^\top Y X$, we simplify:

$$\mathcal{L}(O) = \text{const} - 2\text{Tr}(OM) + \text{Tr}(O^\top AOA) + \lambda \text{Tr}(O^\top O).$$

Taking the gradient with respect to O gives:

$$\nabla_O \mathcal{L} = -2M + 2AOA + 2\lambda O.$$

Setting the gradient to zero and rearranging:

$$AOA + \lambda O = M.$$

Assuming A is symmetric and positive definite (e.g., after PCA), we pre- and post-multiply by $(A + \lambda I)^{-1}$ to obtain:

$$O^* = (A + \lambda I)^{-1} M (A + \lambda I)^{-1},$$

or equivalently,

$$O^* = (X^\top X + \lambda I)^{-1} X^\top Y X (X^\top X + \lambda I)^{-1}.$$

This formulation enables vast speedups over [11]’s original Kronecker based formulation, enabling fitting of the Connectome Model for many iterations (e.g. 10,000 spins) and for larger dimension, d .

Train-test split with spatiomolecular null comparison. As described in 4, each model in Table 1 is tested across 10 random or spatial four-fold splits. For each split, a corresponding unique spatiomolecular null gene expression matrix (from the top 10 out of 10,000) is used to refit the model and predict held out test set. This results in 40 test performance metrics for the true gene expression and 40 test performance metrics for the null gene expression. Aggregate metrics are computed to compare the true vs null model performance. This procedure circumvents the need to refit the model thousands of times using many, potentially suboptimal, null spins.

MLP/SMT performance with coordinates & null gene expression. In Table 1 we observe a substantial true vs. null gap in Pearson- r , however, this gap is markedly smaller for the SMT w/ [CLS] under the random split. We posit that this trend is due to the fact that the MLP w/ coords and the SMT w/ [CLS] have access to the true coordinates in both the true and null case. Thus, Euclidean coordinates alone might be predictive of connectivity even without molecular information. To deduce the effect of spatial position alone, we fit several variations of null models in Table 3.

Table 3: Null model performance comparison with varying feature access on UKBB dataset (mean and standard deviation over 10 random 4-fold splits)

Model	Features	Pearson-r	Short r	Mid r	Long r	R ²	MSE	Geodesic
MLP	coords	.72 ± .06	.76 ± .06	.67 ± .07	.63 ± .10	.52 ± .09	.016 ± .003	13.12 ± 1.01
MLP	permuted + coords	.29 ± .07	.28 ± .08	.20 ± .06	.16 ± .09	-.05 ± .11	.035 ± .004	16.45 ± .80
MLP	SM + coords	.47 ± .08	.48 ± .08	.39 ± .08	.36 ± .11	.11 ± .13	.030 ± .005	14.31 ± 1.2
SMT w/ CLS	permuted + coords	.68 ± .06	.70 ± .07	.64 ± .06	.61 ± .10	.41 ± .11	.020 ± .004	11.79 ± .92
SMT w/ CLS	SM + coords	.71 ± .05	.73 ± .05	.67 ± .05	.65 ± .08	.47 ± .09	.018 ± .003	11.52 ± .84

SM indicates gene expression has been permuted under the spatiomolecular null procedure. Permuted indicates the gene expression is a purely random permutation reassignment.

The MLP trained with coordinates alone demonstrates that spatial position is a strong predictor of functional connectivity, achieving Pearson- $r > 0.7$. This likely accounts for the similarly high performance of the SMT w/ [CLS] model under the random split spatiomolecular null. Aside from architectural differences, the only distinction between these models is that the SMT receives additional (but spatially spun) gene expression input. Still, the SMT w/ [CLS] matches the performance of MLP w/ coords across all metrics. The difference observed between the permuted gene expression and spun gene expression highlights the stringency of the spatiomolecular null as argued in A.4. Overall, an MLP trained just on coordinates is effective at connectome reconstruction.

These results reinforce that any model exceeding the SMT with [CLS] null reflects learning of true transcriptomic patterns, rather than spatially autocorrelated structure. Notably, the SMT w/ [CLS] architecture appears to leverage spatial information more explicitly than the MLP, which has reduced ability to isolate positional cues from the high-dimensional input. Attention heads in Figure 13 support strong emphasis of the [CLS] token during learning. In the UKBB dataset, all learned models surpass this stringent null baseline, underscoring their capacity to extract meaningful molecular information beyond what is encoded by Euclidean distance.

A.5 Additional Experiments

Hyperparameter experiments. To isolate the effect of key design choices in the SMT architecture, we conduct a series of hyperparameter experiments using 10 fixed inner cross-validation splits with the UKBB dataset under our random train-test split. These experiments apply each setting independently to the default SMT configuration. Figure 7 summarizes the effects of four components: token encoder dimension, token encoder output dimension, ALiBi slopes, and target augmentation probability. Table 4 displays the hyperparameter grid for the Spatiomolecular Transformer with default parameters bolded.

Inner-CV Experiments

Token encoder dimension performs best at 60, grouping larger gene chunks into fewer tokens. This improves speed (reducing quadratic attention cost) with minimal performance trade-off. Advances in efficient attention may eventually allow single-gene tokenization at scale.

Fixing the token encoder dimension to 60 dimension gene chunk bins and varying the output dimensionality after the transformer shows relative stability up to an output dimensionality of 3 per token. A token encoder output dimensionality of 10 is optimal on the validation set.

ALiBi slopes show a simple and clear trend improving both Pearson- r and MSE when incorporated into the MHSA mechanism, pointing towards the utility of multi-head representations of the input sequence. In the next section we explore the effect of alternate sorting strategies under the random and spatial split. Modest amounts of target augmentation probability 0.1-0.3 are optimal over more aggressive strategies. This is tested for the standard linear decay setting. We explore target-side augmentation strategies further in the subsequent sections.

In general it is worth noting that differences between hyperparameters are nominal indicating nuanced effects of each on overall learning, as well as potential combinatorial effects.

Tokenization strategy

To evaluate the effect of our TSS based tokenization strategy we conduct a post-hoc experiment, where all hyperparameters are fixed to default. We then systematically vary the gene bin ordering and the size of the bins. For each gene bin size, k , genes are binned together randomly, by TSS, or by mean global expression patterns. The mean expression based setup may facilitate the co-embedding of genes based on coexpression and is commonly used in single-cell transformer based methods [31]. Figure 8 shows changes in performance when varying the token size and tokenization strategy. The first notable effect is that the SMT performs best on both splits with gene groups of 60. There is no clear added benefit to smaller gene bins, which aligns with the inner-CV experiments in 7. Interestingly, even when genes are sorted and binned completely randomly, with no biological prior information, SMT performance does not drop. Performance only drops when ALiBi slopes is removed from the model. This points towards a robustness of the *value projection* transformer style model and that there are likely alternate architectural modifications that would introduce biological priors more strongly into the SMT. ALiBi

Table 4: Hyperparameter search space for the Spatiomolecular Transformer. Default values are in **bold**.

Hyperparameter	Values
token_encoder_dim	[20, 60 , 180]
d_model	[64, 128]
encoder_output_dim	[5, 10]
use_alibi	[True , False]
nhead	[2, 4]
num_layers	[2, 4]
deep_hidden_dims	[[256, 128], [512 , 256 , 128]]
transformer_dropout	[0.1, 0.2]
dropout_rate	[0.1, 0.2 , 0.3]
learning_rate	[0.00009, 0.0001]
weight_decay	[0.0001 , 0.001]
batch_size	[512 , 1024]
aug_prob	[0, 0.1, 0.3]
aug_style	[linear_decay, linear_peak, curriculum_swap_constant, curriculum_swap_linear_decay]
epochs	[90, 110]
num_workers	[2]
prefetch_factor	[4]

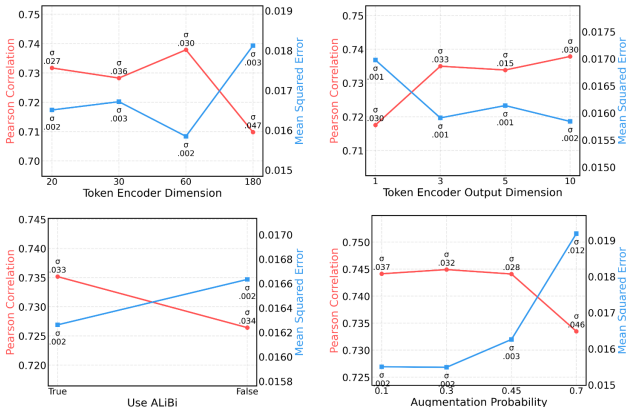


Figure 7: **Validation set hyperparameter experiments for SMT.** Performance varies with token dimensionalities, ALiBi slopes, and augmentation probabilities.

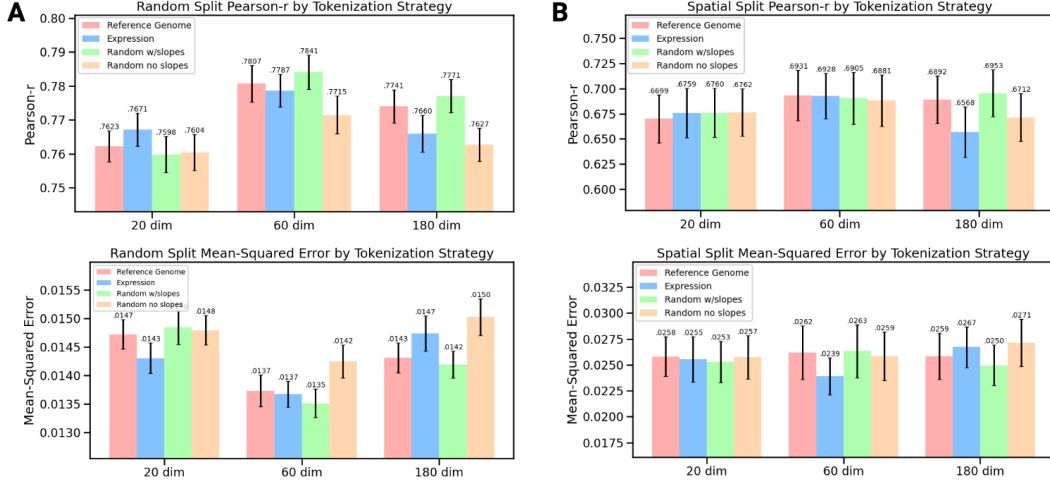


Figure 8: **Test set tokenization strategy analysis.** TSS-based, mean global expression-based, and randomly sorted tokenization with and without ALiBi slopes are evaluated on the test set.

slopes, however, seems to have a positive effect on model performance likely due to its enforced hierarchical representation of the gene expression sequence agnostic of tokenization sorting strategy.

Target-side augmentation strategy

Given the data-scarce regime of population-level transcriptome-to-connectome prediction, we introduce a distributional target-side augmentation strategy to improve generalization. Our approach combines ideas from curriculum learning [40, 22] and synthetic minority oversampling techniques [41], modifying training batches throughout the learning process while injecting signal from the full population distribution.

Algorithm 2 outlines this procedure, which defines three possible augmentation paths: (i) using targets from the population-average connectome Y (i.e., no augmentation), (ii) randomly replacing targets in the current batch with individual-level values from the population, and (iii) replacing targets only if their original population value satisfies $|y| > \theta$, where we fix $\theta = 0.3$. This threshold ensures that only strong positive or negative connections are overrepresented during training—corresponding to a minority class in typical connectome distributions (see the density function in the bottom right of Fig. 1).

Algorithm 2 Distributional Target-Side Augmentation Protocol

Require: Current mini-batch $\mathcal{B} = \{(X_i, y_i)\}_{i=1}^B$, epoch e , total epochs E , augmentation style `aug_style`, probability schedule $p(e)$, population matrix \mathcal{Y}_{pop}

- 1: Draw $r \sim \mathcal{U}(0, 1)$
 - 2: **if** $r < p(e)$ **then**
 - 3: **if** `aug_style` = `curriculum_swap` **then**
 - 4: Identify indices in $\mathcal{D}_{\text{train}}$ where $|y| > \theta$
 - 5: Sample new batch \mathcal{B} from strong edges
 - 6: Replace all y_i in \mathcal{B} with subject-level values from \mathcal{Y}_{pop}
 - 7: **else**
 - 8: **for** each target y_i in \mathcal{B} **do**
 - 9: Replace y_i with a subject-level value from \mathcal{Y}_{pop}
 - 10: **end for**
 - 11: **end if**
 - 12: **else**
 - 13: No augmentation is applied; original batch is used
 - 14: **end if**
 - 15: Compute loss: $\mathcal{L}_{\text{MSE}} = \frac{1}{B} \sum_{i=1}^B \|f_{\theta}(X_i) - y_i\|^2$
-

The augmentation schedule $p(e)$ determines the probability of performing target-side augmentation at epoch e . Various scheduling strategies—such as linear increase, decrease, constant, and peak—are visualized in Fig. 9. This mechanism allows us to inject task-relevant population signal at different stages of training. In our main experiments, we select the augmentation strategy using cross-validation, and Fig. 10 analyzes performance differences across augmentation protocols. For all styles, the maximum augmentation probability is fixed at $p = 0.3$, based on inner cross-validation results using linear decay.

Nearly all augmentation strategies improve model performance with respect to Pearson correlation and mean squared error (MSE). Gains are especially pronounced when evaluating Pearson- r restricted to strong positive edges ($y > 0.3$), where both constant and linearly decaying augmentation schedules yield an improvement of ~ 0.05 . This effect is intuitive, as curriculum-based sampling emphasizes edges with strong signal, which are otherwise underrepresented. Overall, our findings suggest that curriculum-guided target-side augmentation improves model robustness across both random and spatial data splits, and should be considered as a general-purpose regularization strategy in transcriptome-connectome modeling tasks in future work.

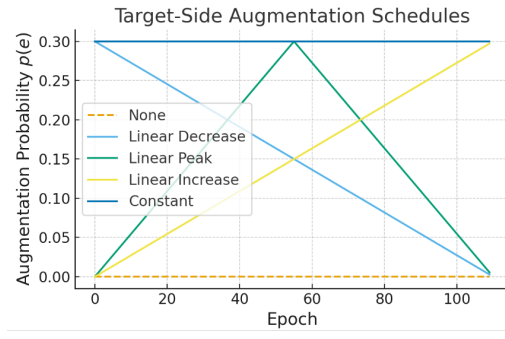


Figure 9: **Augmentation schedules defining $p(e)$ over training epochs.**

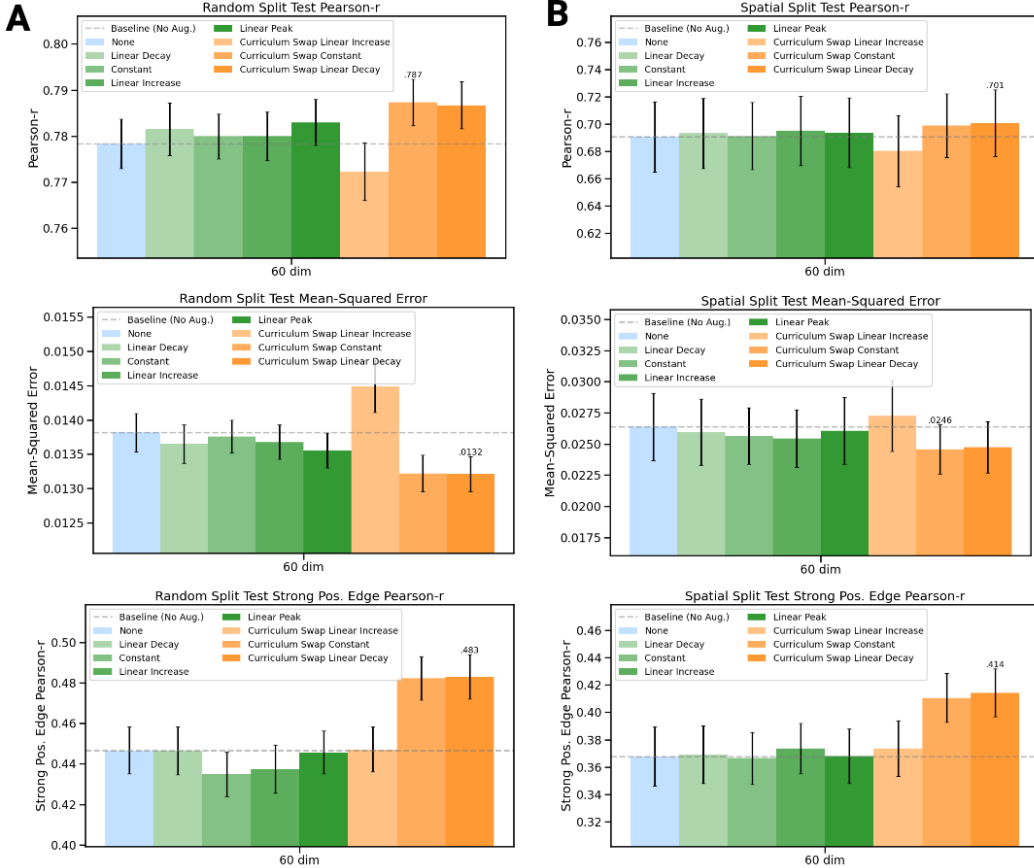


Figure 10: **Distributional target-side augmentation analysis.** Augmentation style test-set performance for Pearson- r , MSE, and strong-positive Pearson- r (>0.3) for [A] random split and [B] spatial split.

A.6 Replication Analysis

Intra-resolution replication analysis. For all resolutions of the MPI-LEMON and HCP dataset, the SMT outperforms its corresponding null (Table 5 and 7) under both the random and spatial split setting. The raw performance values are lower than those reported for UKBB, which could be attributed to a variety of fundamental differences between the datasets such as data quality, processing steps, parcellation style, or weaker signal between the AHBA population and the HCP and MPI-LEMON populations. Alternatively, despite using cross-validated hyperparameter searching strategy, a more optimal search space may exist for the MPI-LEMON dataset as compared to UKBB. Another primary difference between the datasets is that the MPI-LEMON dataset contains more positively skewed values whereas UKBB contains strong signal in both the positive and negative direction.

Tables 6 and Table 8 similarly illustrates performance above the spatiomolecular null, but at a smaller margin for MPI-LEMON. Strong spatial effects are present at all resolutions. This gap is in line with our SMT w/ [CLS] results from 1 which shows a much smaller gap when true coordinates are introduced to the null model. In sum, Tables 5 through 8 confirm a robust relationship between gene expression and functional connectivity in the HCP-YA and MPI-LEMON dataset, but there may be more weakly aligned genetic signal in the MPI-LEMON dataset.

Cross-dataset and cross-resolution analysis. We use a full dataset framework to test cross-dataset and cross-resolution generalization simultaneously. Here, we train an SMT w/ [CLS] model on a source dataset-resolution pair (e.g., UKBB S456) and reconstruct the full connectome of a different dataset-resolution pair (e.g., MPI-LEMON 729). To evaluate whether performance exceeds chance, we benchmark against an SMT model trained on one of the top 10 spatiomolecular spins for the source dataset, keeping spatial coordinates fixed. This model represents the baseline explained by spatial proximity and transcriptomic autocorrelation.

For example, we first train the SMT w/ [CLS] model on the true data and a top K null spin of the UKBB S456. If a model trained on MPI-LEMON 729 reconstructs the UKBB S456 connectome more accurately than the UKBB-based null model, it implies generalization beyond spatial and genetic autocorrelation. Default model hyperparameters are selected prior to application on the target dataset.

Table 5: Multi-resolution test set performance for SMT with random CV on MPI-LEMON and HCP-YA dataset (mean and standard deviation over 10 4-fold splits). (*null*) indicates model trained on 10 lowest-error spatiomolecular nulls at each resolution.

Model	Resolution	Pearson- r	Short r	Mid r	Long r	R^2	MSE	Geodesic
SMT (<i>null</i>)	183	0.26 ± 0.10	0.16 ± 0.12	0.17 ± 0.10	0.24 ± 0.13	-0.11 ± 0.23	0.030 ± 0.005	6.62 ± 0.87
SMT	183	0.50 ± 0.09	0.43 ± 0.11	0.41 ± 0.11	0.45 ± 0.09	0.17 ± 0.16	0.022 ± 0.005	5.62 ± 0.98
SMT (<i>null</i>)	391	0.28 ± 0.06	0.22 ± 0.06	0.20 ± 0.05	0.20 ± 0.10	-0.06 ± 0.07	0.025 ± 0.002	9.19 ± 0.63
SMT	391	0.57 ± 0.05	0.56 ± 0.06	0.49 ± 0.06	0.48 ± 0.08	0.28 ± 0.08	0.017 ± 0.002	8.17 ± 0.92
SMT (<i>null</i>)	729	0.30 ± 0.05	0.24 ± 0.05	0.21 ± 0.05	0.25 ± 0.07	-0.08 ± 0.08	0.022 ± 0.002	11.90 ± 0.77
SMT	729	0.54 ± 0.04	0.53 ± 0.04	0.45 ± 0.04	0.45 ± 0.06	0.23 ± 0.06	0.016 ± 0.001	10.67 ± 0.77
SMT (<i>null</i>)	HCP	0.27 ± 0.07	0.28 ± 0.08	0.21 ± 0.07	0.20 ± 0.10	-0.22 ± 0.11	0.030 ± 0.004	14.41 ± 0.87
SMT	HCP	0.71 ± 0.04	0.75 ± 0.04	0.68 ± 0.04	0.58 ± 0.08	0.47 ± 0.08	0.013 ± 0.002	10.89 ± 0.77

Table 6: Multi-resolution test set performance for SMT w/ [CLS] on random split (mean and standard deviation over 10 4-fold splits).

Model	Resolution	Pearson- r	Short r	Mid r	Long r	R^2	MSE	Geodesic
SMT w/ CLS (<i>null</i>)	183	0.50 ± 0.09	0.44 ± 0.12	0.39 ± 0.11	0.45 ± 0.12	0.04 ± 0.18	0.026 ± 0.004	6.18 ± 0.93
SMT w/ CLS	183	0.57 ± 0.09	0.50 ± 0.14	0.48 ± 0.12	0.50 ± 0.12	0.14 ± 0.24	0.023 ± 0.007	5.80 ± 1.11
SMT w/ CLS (<i>null</i>)	391	0.61 ± 0.06	0.60 ± 0.06	0.54 ± 0.06	0.54 ± 0.09	0.30 ± 0.10	0.016 ± 0.002	7.76 ± 0.80
SMT w/ CLS	391	0.68 ± 0.04	0.68 ± 0.04	0.61 ± 0.05	0.61 ± 0.06	0.42 ± 0.07	0.013 ± 0.001	7.61 ± 0.81
SMT w/ CLS (<i>null</i>)	729	0.66 ± 0.04	0.65 ± 0.04	0.59 ± 0.05	0.59 ± 0.06	0.38 ± 0.07	0.013 ± 0.002	9.63 ± 0.71
SMT w/ CLS	729	0.70 ± 0.04	0.70 ± 0.03	0.63 ± 0.03	0.64 ± 0.06	0.45 ± 0.07	0.011 ± 0.001	9.67 ± 1.03
SMT w/ CLS (<i>null</i>)	HCP	0.59 ± 0.05	0.63 ± 0.05	0.53 ± 0.06	0.53 ± 0.10	0.26 ± 0.09	0.018 ± 0.003	12.59 ± 0.72
SMT w/ CLS	HCP	0.74 ± 0.05	0.77 ± 0.04	0.70 ± 0.05	0.66 ± 0.10	0.52 ± 0.08	0.012 ± 0.002	11.27 ± 0.81

Table 7: Multi-resolution test set performance for SMT on spatial split (mean and standard deviation over 10 4-fold splits).

Model	Resolution	Pearson- r	Short r	Mid r	Long r	R^2	MSE	Geodesic
SMT (<i>null</i>)	183	0.13 ± 0.10	0.09 ± 0.08	0.08 ± 0.10	0.11 ± 0.14	-0.25 ± 0.41	0.041 ± 0.013	7.02 ± 0.84
SMT	183	0.38 ± 0.15	0.35 ± 0.16	0.31 ± 0.12	0.27 ± 0.18	0.02 ± 0.28	0.032 ± 0.008	6.27 ± 0.81
SMT (<i>null</i>)	391	0.19 ± 0.08	0.15 ± 0.07	0.11 ± 0.09	0.08 ± 0.16	-0.15 ± 0.11	0.034 ± 0.006	9.34 ± 0.94
SMT	391	0.49 ± 0.10	0.47 ± 0.11	0.36 ± 0.09	0.28 ± 0.21	0.18 ± 0.17	0.024 ± 0.006	8.09 ± 0.84
SMT (<i>null</i>)	729	0.22 ± 0.09	0.18 ± 0.07	0.09 ± 0.07	0.10 ± 0.09	-0.16 ± 0.12	0.031 ± 0.005	11.98 ± 0.78
SMT	729	0.48 ± 0.09	0.45 ± 0.06	0.34 ± 0.09	0.28 ± 0.14	0.17 ± 0.11	0.022 ± 0.004	10.73 ± 0.83
SMT (<i>null</i>)	HCP	0.25 ± 0.09	0.25 ± 0.10	0.15 ± 0.08	0.09 ± 0.10	-0.20 ± 0.18	0.038 ± 0.011	13.91 ± 1.18
SMT	HCP	0.63 ± 0.15	0.63 ± 0.17	0.55 ± 0.14	0.44 ± 0.22	0.20 ± 0.81	0.024 ± 0.014	11.13 ± 1.80

Table 8: Multi-resolution test set performance for SMT w/ [CLS] on spatial split (mean and standard deviation over 10 4-fold splits).

Model	Resolution	Pearson- r	Short r	Mid r	Long r	R^2	MSE	Geodesic
SMT w/ CLS (<i>null</i>)	183	0.29 ± 0.13	0.24 ± 0.13	0.14 ± 0.14	0.22 ± 0.13	-0.35 ± 0.59	0.044 ± 0.019	7.28 ± 2.47
SMT w/ CLS	183	0.32 ± 0.12	0.27 ± 0.14	0.19 ± 0.14	0.34 ± 0.16	-0.28 ± 0.39	0.042 ± 0.015	7.01 ± 1.43
SMT w/ CLS (<i>null</i>)	391	0.43 ± 0.10	0.40 ± 0.10	0.28 ± 0.13	0.35 ± 0.15	0.00 ± 0.19	0.030 ± 0.006	8.02 ± 0.81
SMT w/ CLS	391	0.50 ± 0.09	0.48 ± 0.10	0.34 ± 0.10	0.35 ± 0.17	0.15 ± 0.15	0.024 ± 0.006	7.95 ± 0.83
SMT w/ CLS (<i>null</i>)	729	0.44 ± 0.11	0.41 ± 0.10	0.25 ± 0.12	0.23 ± 0.16	-0.08 ± 0.27	0.028 ± 0.007	10.19 ± 0.72
SMT w/ CLS	729	0.52 ± 0.09	0.49 ± 0.07	0.36 ± 0.12	0.31 ± 0.22	0.08 ± 0.21	0.025 ± 0.007	9.87 ± 0.88
SMT w/ CLS (<i>null</i>)	HCP	0.37 ± 0.11	0.37 ± 0.12	0.21 ± 0.12	0.18 ± 0.13	-0.27 ± 0.58	0.040 ± 0.017	13.14 ± 1.56
SMT w/ CLS	HCP	0.57 ± 0.13	0.57 ± 0.14	0.45 ± 0.14	0.47 ± 0.23	0.17 ± 0.30	0.026 ± 0.012	12.05 ± 2.21

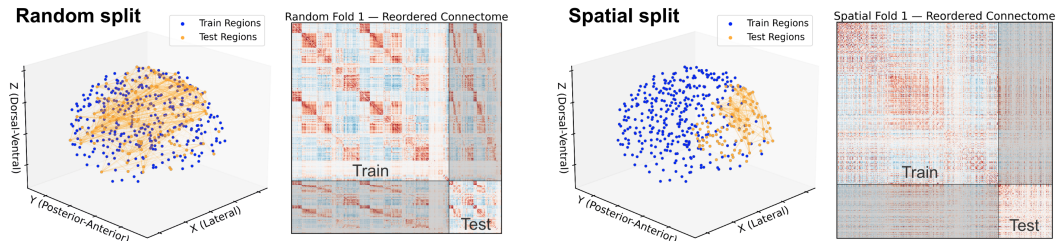


Figure 11: Example connectome random and spatial splits.

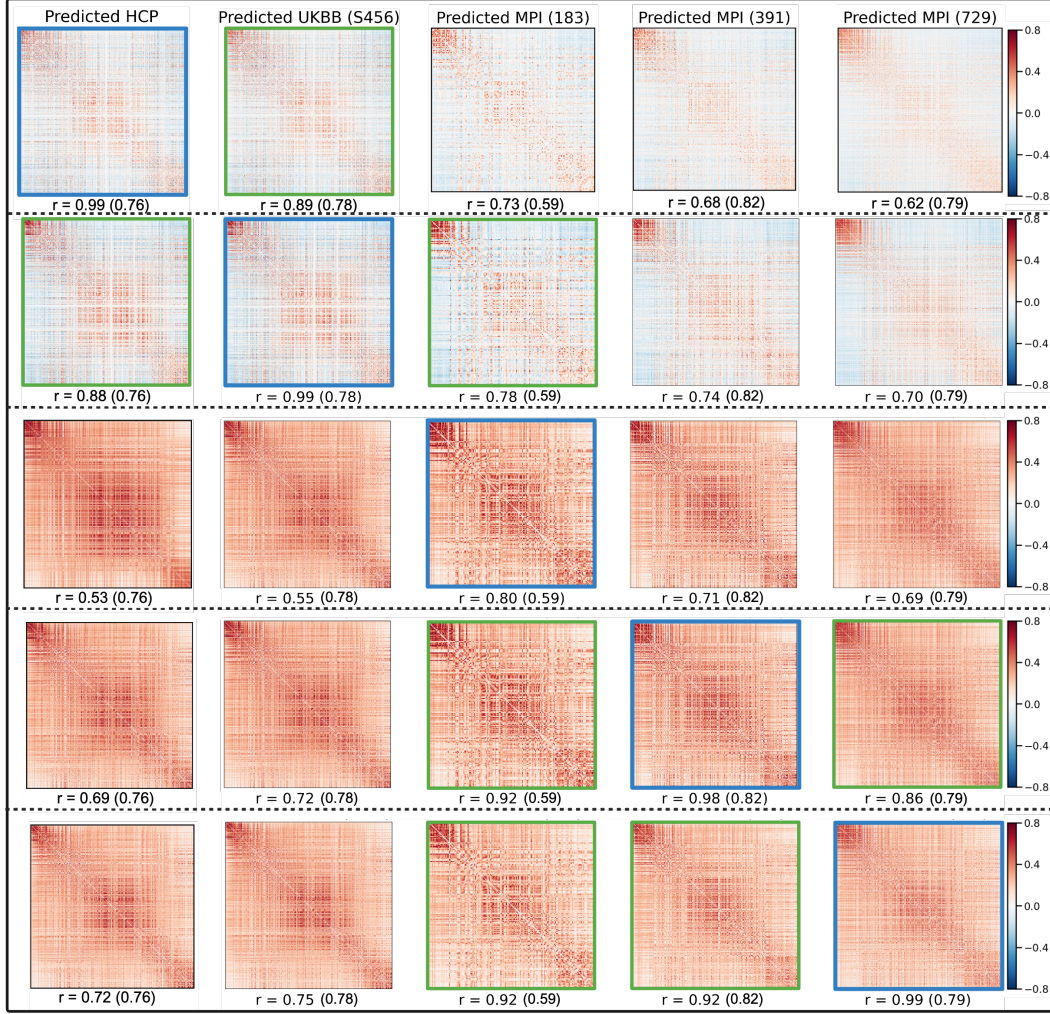


Figure 12: **Cross-dataset generalization of SMT with CLS.** The diagonal shows population-average functional connectomes from UK Biobank (S456), Human Connectome Project (S456), and MPI-LEMON (183, 391, 729 resolutions). Connectomes are sorted by the anterior-posterior axis across datasets and resolutions. Each row shows reconstructions from SMT w/ [CLS] models trained on a source dataset (highlighted with a blue border) and tested on external datasets or resolutions. Pearson correlation r between predicted and true connectomes is shown below each matrix, with the source dataset spatiomolecular null performance in parentheses. Null performance is based on SMT w/ [CLS] models fit on a low-error spatiomolecular null gene expression matrix with true coordinates. Connectomes highlighted with a green border indicate generalization above native SMT [CLS] null chance. (Non-parenthesized Pearson correlations along the diagonal correspond to training fit and thus do not indicate significance over the null. See intra-resolution effects in Tables 5–8).

As shown in Figure 12, across all experiments we observe consistently high reconstruction performance and a backbone connectome structure at all resolutions. Meaningful generalization above the null is observed in a few cases. Both the UKBB and HCP trained models are able to predict MPI-LEMON 183 beyond the spatiomolecular null baseline, suggesting a possible conserved molecular–connectomic relationship across adult populations. Notably, UKBB participants are older on average than those in MPI-LEMON (see Section A.1), supporting the hypothesis that shared transcriptomic gradients contribute to functional brain organization across the adult lifespan. This is further supported by the fact that the HCP trained model significantly predicts MPI-LEMON 183, at a value slightly worse than UKBB. Differences in acquisition and population demographics may further influence our generalization results. HCP to UKBB generalization exceeds null chance, which is expected given the targets of UKBB are highly similar.

We also find that while UKBB and HCP connectomes contain both positive and negative values, MPI-LEMON matrices are strictly positive. This discrepancy may limit the model’s ability to capture absolute connectivity values across datasets (as reflected in MSE), though relative connection strengths remain well preserved (as reflected in Pearson- r). Parcellation differences likely also play a role: the UKBB parcellation is functionally defined based on resting-state networks, whereas the MPI-LEMON atlas is derived from unsupervised voxel-level clustering.

Within the MPI-LEMON dataset, we find strong generalization across parcellation resolutions. Five cross-resolution reconstructions exceed the null baseline, including one case where a model trained at lower resolution (391) generalizes successfully to a higher resolution (729). These results suggest that while spatial position explains much of the variance, gene expression captures complementary structure. As resolution increases and more training data becomes available, models are better able to learn complex molecular–connectomic relationships.

In sum, our cross-dataset generalization analysis demonstrates that our predictive accuracy on UKBB is not an artifact of the particular dataset properties. There are common properties preserved across datasets, demographics, and parcellation resolutions that the SMT captures.

A.7 Attention Weights Analysis

To interpret how the SMT attends to gene tokens, we extract and average attention weights from the final layer of the trained transformer encoder. For a given region, the input is a sequence of tokenized gene expression values $X \in \mathbb{R}^{\ell \times d}$, where ℓ is the number of tokens and d is the embedding dimension. In each layer, X is linearly projected to queries, keys, and values: $Q = XW_Q$, $K = XW_K$, $V = XW_V$, where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d_h}$, and $d_h = d/h$ is the head dimension for h heads. Each head computes scaled dot-product attention with ALiBi biases B :

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^\top + B}{\sqrt{d_h}} \right) V,$$

where $\text{softmax} \left(\frac{QK^\top + B}{\sqrt{d_h}} \right) \in \mathbb{R}^{\ell \times \ell}$ defines the attention weights.

After model training (on the full dataset using the default hyperparameters identified via cross-validation), we pass in the full training data to compute these attention weights. For each sample, we extract the self-attention matrix from the final transformer layer for each head. We then average these matrices across all samples and all heads to obtain a global attention score matrix:

$$\bar{A} = \frac{1}{hN} \sum_{n=1}^N \sum_{j=1}^h A_n^{(j)},$$

where $A_n^{(j)} \in \mathbb{R}^{\ell \times \ell}$ is the attention matrix for the j -th head on the n -th region, and N is the total number of regions.

For subnetwork-level analysis, we instead compute attention weights restricted to samples belonging to canonical functional subnetworks defined by our UK Biobank 456 region, 9 network parcellation. We compute the attention weights for each network $k \in \{1, \dots, 9\}$ by passing in its regions to the encoder individually. These attention matrices are aggregated in the same way, producing a network-specific average attention map \bar{A}_k . This procedure allows us to characterize how attention is distributed over gene tokens in general, and how different brain networks emphasize distinct transcriptomic features during embedding.

Figure 13A shows that, in the absence of the [CLS] token, self-attention patterns differ across heads, with specific gene tokens receiving higher attention in some heads more than others. These variations also exhibit different spatial frequencies, possibly reflecting local versus global transcriptomic interactions. Such diversity may be encouraged by the ALiBi slope mechanism.

Figure 13B presents the mean attention scores for the SMT w/ [CLS] model. The first token (expanded for visibility) is the [CLS] token, which consistently receives the highest attention across all heads, indicating strong influence from spatial coordinates. Interestingly, the 54th gene token also receives high attention in all heads, suggesting despite the strong emphasis on spatial position, some gene modules such as those related to white matter microstructure, neurodevelopment, or metabolic activity, may selectively co-embed with spatial information to improve prediction. The stability of

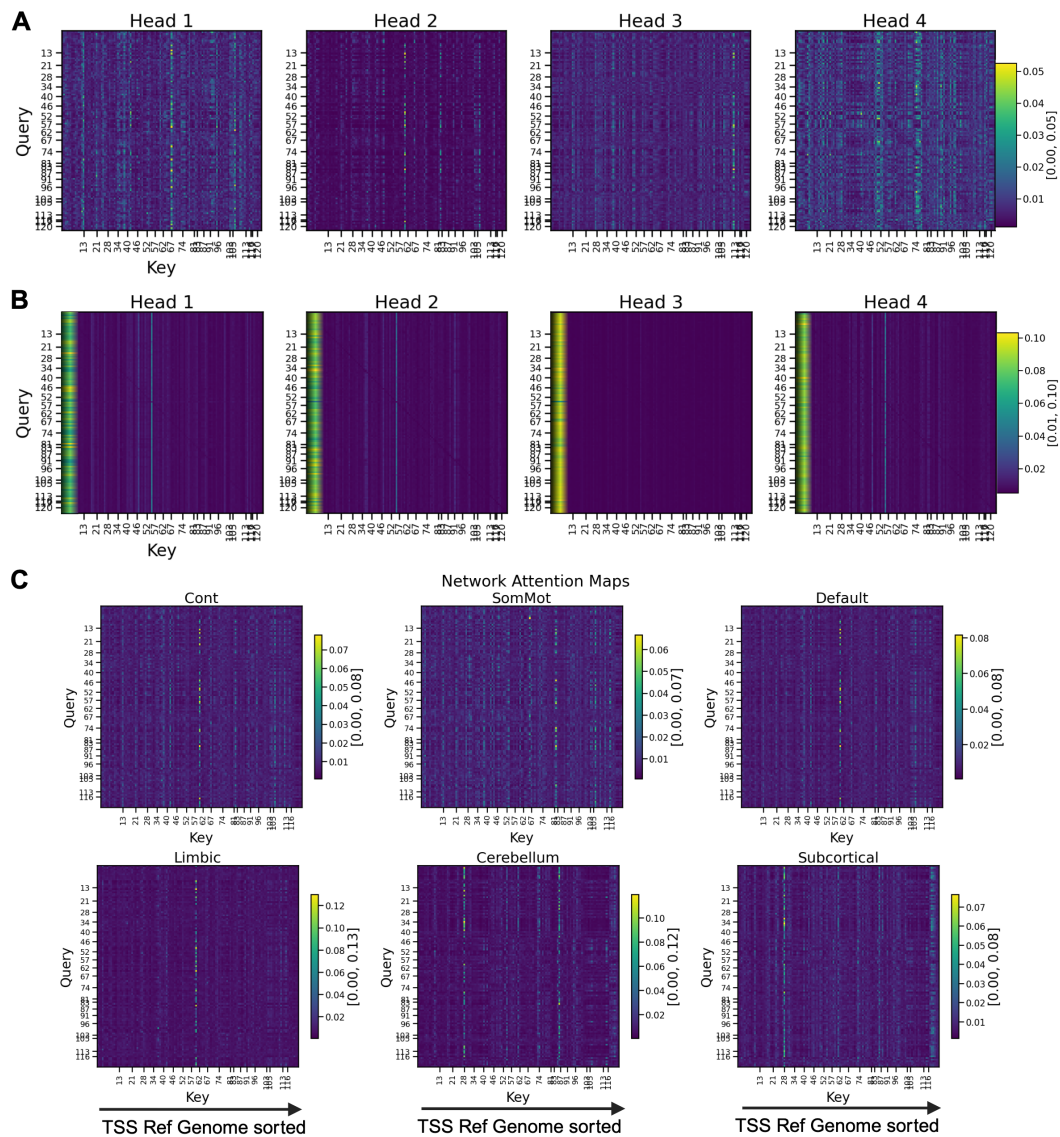


Figure 13: **SMT multi-head self-attention weights.** [A] Attention scores for the Spatiomolecular Transformer (SMT). Tick marks on both Query and Key axes denote token positions at chromosome boundaries. [B] Attention scores for SMT with [CLS] token. The CLS token key column is artificially expanded for visualization. [C] Subnetwork-specific attention weights for SMT, averaged over all heads for a subset of six functional subnetworks.

these attention heads must be further validated to link the gene sets highlighted by the SMT with biologically relevant processes.

Figure 13C shows average attention distributions when regions from specific functional subnetworks are input into the SMT model. Across a subset of six functional networks, we observe the subcortex and cerebellum show similar profiles as compared to cortical networks. Certain gene tokens consistently receive elevated attention across all subnetworks. Further evaluation of global and subnetwork-specific gene tokens may inform if distinct transcriptomic modules underlie functional connectivity across the brain.

Table 9: Author contributions.

	AR	SG	JW	CD	EV
Study concept or design					
Data acquisition or processing					
Data analysis or interpretation					
Drafting/revision of manuscript for content					
Funding acquisition					

Note: Bolded initials indicate the primary contributor. Black cells indicate a documented contribution to the corresponding activity.

NeurIPS Paper Checklist

1. Claims

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

Answer: [\[Yes\]](#)

Justification: The abstract and introduction lay out the context of the problem and list the contributions made to solve the problem in the results section.

2. Limitations

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

Answer: [\[Yes\]](#)

Justification: The assumptions and limitations of the described method are described in the discussion section.

3. Theory assumptions and proofs

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

Answer: [\[NA\]](#)

Justification: The work is mostly empirical, not theoretical, building on pre-existing methods.

4. Experimental result reproducibility

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

Answer: [\[Yes\]](#)

Justification: The paper provides details on the datasets, preprocessing pipelines, train/test split methodology, model architecture, and evaluation metrics. These are sufficient to reproduce the claims even without code provided, however our code will be made publicly available.

5. Open access to data and code

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

Answer: [\[Yes\]](#)

Justification: The AHBA and MPI-LEMON datasets are publicly available as described in the datasets section. The HCP-YA and UK Biobank dataset are available through application. We will provide anonymized code zip file in supplementary materials and publicly release code.

6. Experimental setting/details

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

Answer: [\[Yes\]](#)

Justification: All details of the experimental setup (data split, hyperparameters, etc.) are specified in the main body or appendix of the paper.

7. Experiment statistical significance

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

Answer: [\[Yes\]](#)

Justification: The paper covers statistical significance of the experiments in the sections describing the null model method, covered in the results and technical appendix.

8. Experiments compute resources

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

Answer: [Yes]

Justification: These details are covered in the supplementary material.

9. Code of ethics

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

Answer: [Yes]

Justification: The authors adhere to all standards described in the NeurIPS Code of Ethics.

10. Broader impacts

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

Answer: [NA]

Justification: This is not applicable as it is foundational basic research.

11. Safeguards

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

Answer: [NA]

Justification: The models described in this paper pose no such risks.

12. Licenses for existing assets

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

Answer: [Yes]

Justification: All uses of data or code are appropriately cited in the references section. The licenses are detailed in the supplementary materials.

13. New assets

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

Answer: [Yes]

Justification: New assets (namely trained model checkpoints) will be included in the code repository provided in the supplementary material.

14. Crowdsourcing and research with human subjects

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

Answer: [NA]

Justification: This paper relies on previously collected, public neuroimaging and transcriptomic datasets. No new data gathering was conducted.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

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

Answer: [NA]

Justification: This paper relies on previously collected, public neuroimaging and transcriptomic datasets. No new data gathering was conducted. Approval licenses can be made available upon publication.

16. Declaration of LLM usage

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

Answer: [NA]

Justification: The method described in this paper does not involve LLMs in any capacity.