PANOMONTAGE: LEARNING MONTAGE FIELDS FOR ARBITRARY-LOCATION EEG SUPER-RESOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

High-density electroencephalography (EEG) captures fine-grained spatial patterns of brain activity, supporting the study of neural dynamics and the localization of abnormal activity, but its hardware and preparation requirements limit widespread use. EEG spatial super-resolution offers a way to recover dense scalp activity from sparse recordings, but most existing methods synthesize signals only at predefined electrode locations, preventing flexible reconstruction at arbitrary scalp locations. We introduce PanoMontage, which formulates EEG super-resolution as continuous montage field learning. Given sparse EEG observations, PanoMontage constructs a personalized montage field that combines physics-guided waveform estimates with learned spatial context, and queries this field to reconstruct EEG signals at arbitrary scalp locations. To enable montage field learning across heterogeneous electrode configurations, we develop a dynamic-montage self-supervised pretraining scheme that jointly optimizes field construction and query reconstruction on large-scale unlabeled EEG recordings. After fine-tuning on four downstream datasets, PanoMontage achieves the lowest NMSE in 12 of 14 dataset-density settings, with reductions of up to 36.5% relative to the strongest competing baseline in each setting. Controlled comparisons further show that the advantage of montage field learning increases as observations become sparser, highlighting its importance for accurate arbitrary-location EEG super-resolution.

1 INTRODUCTION

Electroencephalography (EEG) provides millisecond-scale measurements of brain activity through non-invasive scalp recordings. Increasing electrode density enables finer characterization of the spatial organization of scalp potentials, supporting the study of neural dynamics and the localization of abnormal activity (Michel & Murray, 2012; Srinivasan et al., 1996). However, high-density EEG requires more acquisition hardware, longer preparation, and greater recording effort, making it considerably more difficult to collect at scale. In contrast, sparse EEG is easier and less costly to acquire. EEG spatial super-resolution (EEGSR) therefore seeks to recover spatially dense EEG signals from sparse observations, providing a potential route to richer spatial analysis without equally dense acquisition.

A central limitation of most existing EEGSR methods is that reconstruction is tied to predefined electrode locations. Classical interpolation estimates signals according to fixed spatial relationships between electrodes, while many learning-based approaches learn mappings between prescribed source and target montages (Perrin et al., 1989; Tang et al., 2023; Li et al., 2025; Wang et al., 2025; Liu et al., 2026b; Kang et al., 2026). Such formulations are effective when the target montage is known in advance, but restrict flexible reconstruction at arbitrary scalp locations and complicate reuse across heterogeneous electrode configurations. Recent coordinate-conditioned approaches have begun to relax this constraint by enabling reconstruction at requested scalp positions (Liu et al., 2026a; Warner et al., 2026; Lazarovich et al., 2026), yet maintaining high reconstruction fidelity becomes increasingly challenging as observations become sparse. This raises a broader question: *rather than treating EEG super-resolution primarily as missing-channel prediction, can we learn a spatial representation of scalp activity that supports accurate reconstruction wherever it is queried?*

We address this question by introducing **PanoMontage**, which formulates EEG super-resolution as continuous montage field learning. Our formulation is motivated by the physical structure of

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.

EEG: scalp electrodes provide sparse samples of a spatial potential field shaped by volume conduction (Srinivasan et al., 1996). Instead of mapping observed electrodes directly to each requested output, PanoMontage first constructs a personalized montage field from the sparse observations. This field combines physics-guided waveform estimates with learned contextual features that capture signal-dependent spatial relationships. A query decoder then uses the learned field to reconstruct EEG waveforms at arbitrary scalp coordinates. The output montage is therefore determined by the query locations rather than by a predefined target electrode configuration.

To learn such montage fields across heterogeneous EEG recordings, we develop a dynamic-montage self-supervised pretraining strategy that varies the observed and query electrode sets during pretraining. This enables a single model to learn the underlying spatial relationships among any electrode locations, supporting both field construction and arbitrary-location reconstruction across different electrode configurations without requiring ground-truth signals at intermediate field locations. We pretrain PanoMontage on large-scale unlabeled TUEG recordings to establish a generalizable montage field representation. To evaluate the learned representation, we fine-tune the pretrained model on four downstream datasets under subject-held-out protocols. Across 14 dataset–density settings, PanoMontage achieves the lowest NMSE in 12 settings, with reductions of up to 36.5% relative to the strongest competing baseline in each setting. Further analyses show that the advantage of montage field learning becomes increasingly pronounced as observations become sparser, demonstrating its particular value under severe spatial undersampling. Our contributions can be summarized into threefold:

- We formulate EEG spatial super-resolution as **continuous montage field learning** and introduce PanoMontage, which learns a personalized scalp field from sparse observations and enables reconstruction at arbitrary scalp locations beyond predefined electrode montages.
- We develop a dynamic-montage self-supervised pretraining framework that learns transferable spatial relationships from heterogeneous EEG recordings by varying observed and query electrode configurations, allowing field construction and arbitrary-location reconstruction without supervision at intermediate field locations.
- We evaluate our approach on four datasets and across 14 dataset–density settings, where PanoMontage achieves the lowest NMSE in 12 settings with reductions of up to 36.5% over the strongest competing baseline in each setting. Controlled analyses further show that its advantage increases as observations become sparser, providing direct evidence that the learned montage field is particularly beneficial under severe spatial undersampling.

2 Related Work

EEG Spatial Super-Resolution. EEGSR estimates signals at unobserved electrode locations from sparse recordings. Classical methods use spherical splines and radial basis functions to interpolate scalp potentials (Perrin et al., 1989; Jäger et al., 2016). Learning-based approaches model spatial and temporal relationships through graph networks, Transformers, and diffusion models (Tang et al., 2023; Li et al., 2025; Wang et al., 2025; Liu et al., 2026b; Kang et al., 2026). Many of these methods assume prescribed source–target montages, limiting their reuse across electrode configurations. Recent work supports more flexible spatial synthesis. ScalpINR predicts signals at queried coordinates through a conditional neural field (Liu et al., 2026a), while EMAG renders scalp signals from a Gaussian representation of electrical sources (Lazarovich et al., 2026). PanoMontage explores progressive spatial inference by using waveforms estimated on an intermediate montage field to guide subsequent synthesis.

Self-Supervised EEG Learning. Self-supervised learning enables EEG models to learn from unlabeled recordings. BENDR, LaBraM, and EEGPT use contrastive or masked-prediction objectives to learn representations for downstream EEG tasks (Kostas et al., 2021; Jiang et al., 2024; Wang et al., 2024). For waveform generation, ZUNA combines position-aware modeling with channel dropout during pretraining on heterogeneous recordings, supporting synthesis across electrode configurations (Warner et al., 2026). Our dynamic-montage scheme uses held-out channels to jointly supervise field construction and query synthesis across varying electrode configurations.

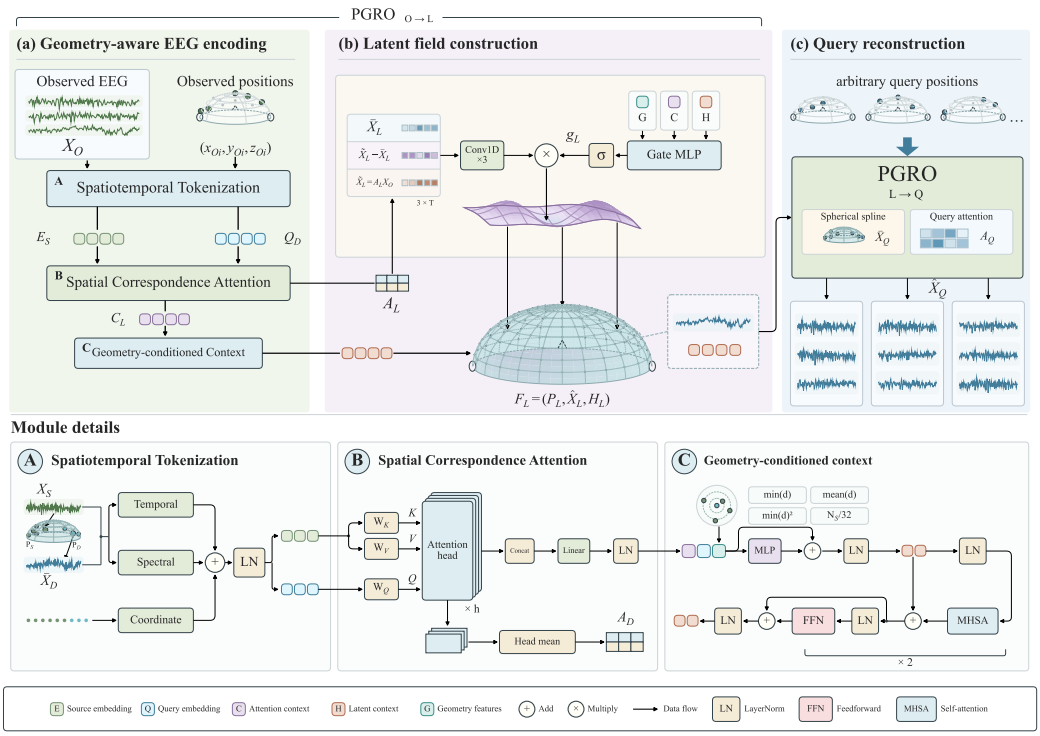

Figure 1: PanoMontage framework. (a) Observed EEG waveforms and electrode positions are encoded into tokens and spatial context. (b) The constructor adds a gated learned correction Δ_L to the spherical-spline estimate $\overline{X}_L = \mathcal{S}_{L \leftarrow O} X_O$, forming the intermediate field $\mathcal{F}_L = (P_L, \widehat{X}_L, H_L)$ with $\widehat{X}_L = \overline{X}_L + \Delta_L$. (c) PGRO queries this field at arbitrary scalp positions to reconstruct the requested EEG waveforms.

3 METHOD

Figure 1 presents an overview of PanoMontage. Given sparse EEG observations, the model first constructs an intermediate montage field and then queries this field at requested scalp locations. At both stages, a Physics-Guided Reconstruction Operator (PGRO) refines spherical-spline estimates with signal-dependent corrections. Dynamic-montage self-supervised learning jointly optimizes field construction and query reconstruction using held-out channels from varying electrode configurations.

3.1 PROBLEM FORMULATION

Our goal is to reconstruct EEG waveforms at unobserved scalp locations from sparse recordings, allowing the observed and query electrode sets to vary across examples. For each input window, $X_O \in \mathbb{R}^{N_O \times T}$ contains recordings from N_O electrodes over T time samples, with coordinates $P_O \in \mathbb{R}^{N_O \times 3}$. Given query coordinates $P_Q \in \mathbb{R}^{N_Q \times 3}$, the model predicts the corresponding waveforms $\widehat{X}_Q \in \mathbb{R}^{N_Q \times T}$. All coordinates are normalized to the unit sphere.

3.2 PROGRESSIVE MONTAGE FIELD LEARNING

PanoMontage uses estimated scalp activity on an intermediate spatial support to guide subsequent waveform synthesis. We represent this intermediate montage field as $\mathcal{F}_L = (P_L, \widehat{X}_L, H_L)$, where $P_L \in \mathbb{R}^{N_L \times 3}$ specifies its locations, $\widehat{X}_L \in \mathbb{R}^{N_L \times T}$ contains waveform estimates, and $H_L \in \mathbb{R}^{N_L \times d}$ contains contextual features, with feature dimension d.

The two stages compute

$$(\widehat{X}_L, H_L) = \mathcal{C}_{\theta_c}(X_O, P_O; P_L), \qquad \widehat{X}_Q = \mathcal{Q}_{\theta_q}(\mathcal{F}_L, P_Q), \tag{1}$$

with separate parameters θ_c and θ_q. The intermediate support varies during pretraining and is fixed within each fine-tuned target-domain model; the field values are inferred from each input window.

The constructor estimates waveforms at the intermediate locations and builds their contextual features from the observations. Spatial self-attention exchanges information across these locations, refining the features that guide waveform correction. The resulting field supports query synthesis through both its waveforms and context: $\widehat{X}_L$ supplies the source signals for spline estimation and attention-based waveform mixing, while H_L augments the source tokens with information derived from the original observations.

Each stage combines a spline estimate with a learned correction. Let $S_{D\leftarrow S}$ denote spherical-spline interpolation from source support S to destination support D, and let Δ_L and Δ_Q denote the intermediate and query corrections. Substituting the first-stage estimate into the query stage gives

$$\widehat{X}_Q = S_{Q\leftarrow L} S_{L\leftarrow O} X_O + S_{Q\leftarrow L}\Delta_L + \Delta_Q. \tag{2}$$

Because Δ_L is propagated through $S_{Q\leftarrow L}$, the intermediate correction affects the query estimate before the query-stage correction Δ_Q is applied. This correction is conditioned on both $\widehat{X}_L$ and H_L. For a given input window, the same intermediate field can be queried at different sets of scalp coordinates. Query positions and their number N_Q determine the output montage independently of the intermediate support P_L, allowing the trained model to synthesize waveforms at requested locations and sampling densities.

3.3 Physics-Guided Reconstruction Operator

PGRO implements each stage by refining a spherical-spline estimate with signal context and electrode geometry. We describe a source support S and destination support D: construction uses $(S, D) = (O, L)$ with $X_S = X_O$, while querying uses $(S, D) = (L, Q)$ with $X_S = \widehat{X}_L$. Stage-specific parameter indices are omitted below.

The regularized interpolation matrix $S_{D\leftarrow S} = S_\alpha(P_S, P_D) \in \mathbb{R}^{N_D\times N_S}$ provides an initial waveform estimate:

$$\overline{X}_D = S_{D\leftarrow S} X_S. \tag{3}$$

We use the Perrin formulation (Perrin et al., 1989; Gramfort et al., 2013). This estimate supplies the destination waveforms for tokenization and the starting point for residual refinement.

We encode source and destination waveforms with their coordinates as $T_S = \mathcal{T}_S(X_S, P_S)$ and $T_D = \mathcal{T}_D(\overline{X}_D, P_D)$. For $r \in \{S, D\}$, each encoder combines temporal, spectral, and coordinate representations:

$$\mathcal{T}_r(X, P) = \mathrm{LN}\Big(\mathcal{E}^{(r)}_{\mathrm{time}}(X) + \mathcal{E}^{(r)}_{\mathrm{freq}}(X) + \mathcal{E}^{(r)}_{\mathrm{pos}}(P)\Big). \tag{4}$$

The attention inputs are $Q_D = \mathrm{LN}_q(T_D)$ and E_S, where $E_S = T_S$ for construction and $E_S = \mathrm{LN}(T_S + H_L)$ for querying.

Cross-attention learns signal-dependent correspondences between destination and source locations:

$$(C_D, A_D) = \mathcal{A}_\theta(Q_D, E_S), \qquad A_D = \frac{1}{n_h}\sum_{a=1}^{n_h} A_D^{(a)}. \tag{5}$$

The operator includes multi-head attention, residual normalization, and a feed-forward block (Vaswani et al., 2017). It returns context $C_D \in \mathbb{R}^{N_D\times d}$ and weights averaged over n_h heads, $A_D \in \mathbb{R}^{N_D\times N_S}$. The context guides feature updates and gates, while the weights mix source waveforms. Set-based aggregation accommodates varying source counts and positions (Lee et al., 2019).

Geometry features G_D summarize source-distance statistics and source count at each destination. During field construction, they condition a context update:

$$H_0 = \mathrm{LN}_{s,c}(Q_L + C_L + \mathcal{U}_c([Q_L, C_L, G_L])), \qquad H_L = \Phi_c(H_0). \tag{6}$$

Brackets denote concatenation. The operator Φ_c refines the context through two spatial self-attention blocks followed by normalization. The resulting H_L conditions the intermediate waveform correction and enters the query-stage source tokens. Query geometry features G_Q are computed relative to P_L and condition the query gate.

The attention weights form a waveform mixture $\widetilde{X}_D$. A temporal network uses this mixture, the spline estimate, and their difference to predict a residual, scaled by a context- and geometry-conditioned gate:

$$\begin{aligned} \widetilde{X}_D &= A_D X_S, & R_D &= \mathcal{R}_\theta\Big([\overline{X}_D, \widetilde{X}_D, \widetilde{X}_D - \overline{X}_D]\Big), \\ g_D &= g_{\max}\sigma(\mathcal{G}_\theta([h_D, C_D, G_D])), & \widehat{X}_D &= \overline{X}_D + g_D \odot R_D. \end{aligned} \tag{7}$$

Here $h_D = H_L$ during construction and $h_D = Q_D$ during querying. Each destination has a scalar gate broadcast across time, with $g_{\max}$ setting its maximum value. The resulting correction $\Delta_D = g_D \odot R_D$ yields $\widehat{X}_L$ or $\widehat{X}_Q$ at the respective stage.

3.4 Dynamic-Montage Self-Supervised Learning

We jointly train field construction and query synthesis by varying electrode configurations and reconstructing held-out recorded waveforms. Each recording provides its own supervision, allowing heterogeneous montages to train the same constructor and query decoder without requiring measured waveforms at intermediate locations.

For pretraining, let $\mathcal{V}$ denote the available electrodes of a recording with waveforms X and coordinates P. We independently draw the observed, intermediate, and query cardinalities for each batch, then sample recording windows with at least $N_O + N_Q$ available electrodes. For each window draw, the location sets are sampled without replacement:

$$\begin{gathered} N_O \sim p_O, \qquad N_L \sim p_L, \qquad N_Q \sim p_Q, \\ O \sim \mathrm{Unif}\binom{\mathcal{V}}{N_O}, \quad Q \sim \mathrm{Unif}\binom{\mathcal{V} \setminus O}{N_Q}, \quad L \sim \mathrm{Unif}\binom{\mathcal{B}_X}{N_L}. \end{gathered} \tag{8}$$

Here $\binom{\mathcal{A}}{k}$ is the collection of k-element subsets of $\mathcal{A}$, and $\mathcal{B}_X$ is a pool of standard-montage locations whose electrode names are absent from the recording. Cardinalities are shared within a batch, while the location sets are resampled for each window draw. The model receives X_O and coordinates (P_O, P_L, P_Q); the held-out waveforms X_Q provide the targets. Since $O \cap Q = \varnothing$, all query channels contribute to the pretraining reconstruction loss.

All sampled configurations share the constructor parameters θ_c and query-decoder parameters θ_q. Using the two-stage mapping in Eq. (1), we optimize

$$\begin{gathered} \min_{\theta_c, \theta_q} \mathbb{E}_{(X,P,O,L,Q)\sim\Pi_{\mathrm{pre}}}\Big[\mathcal{L}^{\mathrm{pre}}_{\mathrm{rec}}(\widehat{X}_Q, X_Q) + \lambda_\Delta \mathcal{R}_\Delta + \mathcal{L}^{\mathrm{pre}}_{\mathrm{aux}}\Big], \\ \mathcal{R}_\Delta = \sum_{D\in\{L,Q\}} \frac{\|\Delta_D\|_F^2}{N_D T}, \qquad \Delta_D = g_D \odot R_D. \end{gathered} \tag{9}$$

The joint distribution Π_{pre} accounts for cardinality sampling, the resulting eligibility of recording windows, and location-set sampling. Query reconstruction gradients propagate through both $\widehat{X}_L$ and H_L, jointly shaping the intermediate field and final predictions. The correction penalty averages over the elements at each stage. Auxiliary losses encourage recovery of the observed waveforms and agreement with direct spline estimates.

We pretrain on unlabeled TUEG recordings (Obeid & Picone, 2016) and fine-tune separately on each downstream dataset. During fine-tuning, the intermediate support P_L is fixed while observed electrodes are sampled from each recording. The decoder predicts the full target montage Q, and the primary reconstruction loss uses the mask $M_j = \mathbf{1}[j \in Q \cap \mathcal{V}]\mathbf{1}[j \notin O]$ to select measured target electrodes excluded from the input. Both stages are jointly updated. Appendix B provides architecture details, sampling distributions, loss definitions, and training settings.

Table 1: Subject-held-out missing-channel NMSE (↓) across four datasets. Bold and underlined entries denote the best and second-best displayed values. PCC and reconstruction SNR are reported in Table 12. SRGDiff includes additional validation-only amplitude calibration (Appendix C.2).

	SEED			SEED-IV			CCEP-COReg				Localize-MI			
Method	2×	4×	8×	2×	4×	8×	2×	4×	8×	16×	2×	4×	8×	16×
Analytical Interpolation														
Spline	0.42100	0.84100	1.66700	0.22000	0.42600	0.89100	0.24400	0.30100	0.52100	1.15300	<u>0.28060</u>	0.32300	0.43100	0.79000
Fixed-Montage Transformer														
ESTformer	<u>0.29069</u>	<u>0.34222</u>	**0.43253**	<u>0.16784</u>	<u>0.22565</u>	**0.31598**	<u>0.21419</u>	<u>0.24946</u>	<u>0.31093</u>	<u>0.39889</u>	0.29787	<u>0.29588</u>	<u>0.32459</u>	<u>0.40724</u>
Fixed-Montage Diffusion														
STAD	0.56925	0.64959	0.78243	0.38117	0.43003	0.56927	0.44315	0.48008	0.56734	0.82385	0.39004	0.41123	0.46169	0.55549
SRGDiff	0.54407	0.65731	0.85679	0.53913	0.62843	0.73244	0.92211	0.87139	0.84781	0.96161	0.70425	0.74605	0.83013	0.86974
TopoDiff	0.82349	0.97261	0.97981	0.98478	0.97258	1.00843	0.74468	0.84035	0.93046	0.96968	0.61604	0.72578	0.86622	0.96303
Flexible-Montage Models														
ScalpINR	0.80067	0.82369	0.85612	0.77293	0.79083	0.82679	0.33489	0.35643	0.40737	0.49787	0.32138	0.33777	0.35981	0.41301
ZUNA	0.63700	0.67800	0.72300	0.61100	0.66100	0.70700	0.51400	0.51900	0.53700	0.62200	0.79600	0.78700	0.76600	0.76400
PanoMontage	**0.27537**	**0.33428**	<u>0.43732</u>	**0.16768**	**0.22387**	<u>0.31682</u>	**0.21074**	**0.24516**	**0.30019**	**0.39061**	**0.17808**	**0.20513**	**0.25047**	**0.33969**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Training and evaluation protocol. We pretrain on unlabeled TUEG recordings and fine-tune one model per downstream dataset across observation densities. The CCEP-COReg query decoder is reinitialized; the other datasets inherit the pretrained decoder. Target-domain training uses a 50,000-update ceiling, with checkpoints selected by validation missing-channel NMSE. We evaluate on SEED (Zheng & Lu, 2015), SEED-IV (Zheng et al., 2019), CCEP-COReg (Parmigiani et al., 2022), and Localize-MI (Mikulan et al., 2020) using subject-disjoint protocols. SEED and SEED-IV are evaluated at 2×, 4×, and 8×, while the two high-density datasets additionally include 16×. Observed electrode locations are sampled independently for each example, and metrics are computed only on valid target channels excluded from the input. We use missing-channel NMSE as the primary metric in the main comparison. The corresponding PCC and reconstruction SNR values appear in Table 12.

Baselines. We compare PanoMontage with spherical splines and six learned EEGSR baselines: SRGDiff (Liu et al., 2026b), STAD (Wang et al., 2025), ScalpINR (Liu et al., 2026a), ESTformer (Li et al., 2025), TopoDiff (Yao et al., 2026), and ZUNA (Warner et al., 2026). All target-trained methods use the same subject splits, normalization, observed-electrode counts, target coordinates, and evaluation metrics. Baselines use method-specific training budgets; fixed-montage models are trained separately at each factor, while ScalpINR uses joint-density training. ZUNA uses released pretrained weights with validation-only amplitude calibration. Dataset and evaluation protocols, preprocessing, and baseline implementations are detailed in Appendix A.

4.2 RECONSTRUCTION ACROSS DATASETS AND DENSITIES

Table 1 reports missing-channel NMSE across four datasets and 14 observation settings. PanoMontage achieves the lowest NMSE in 12 settings, with a maximum relative reduction of 36.5% over the strongest competing baseline. ESTformer has lower NMSE at SEED and SEED-IV 8×. PanoMontage leads at all eight settings on the two high-density datasets. These are mean-NMSE rankings; eight paired intervals favor PanoMontage, five include zero, and one favors ESTformer. The largest relative gain occurs on Localize-MI, whose test split contains one subject. Target-trained methods share subject splits and metric definitions; method-specific training and evaluation details are provided in Appendix C, with paired uncertainty estimates in Appendix D.2.

The SRGDiff row follows an adaptation of its released implementation with validation-fitted amplitude calibration. The output gain for each dataset–factor setting is fitted on validation data and frozen before test evaluation. Appendix C.2 reports the protocol and paired uncalibrated results. Table 12 gives the corresponding PCC and SNR values, with complete metrics across observation densities in Appendix D.1.

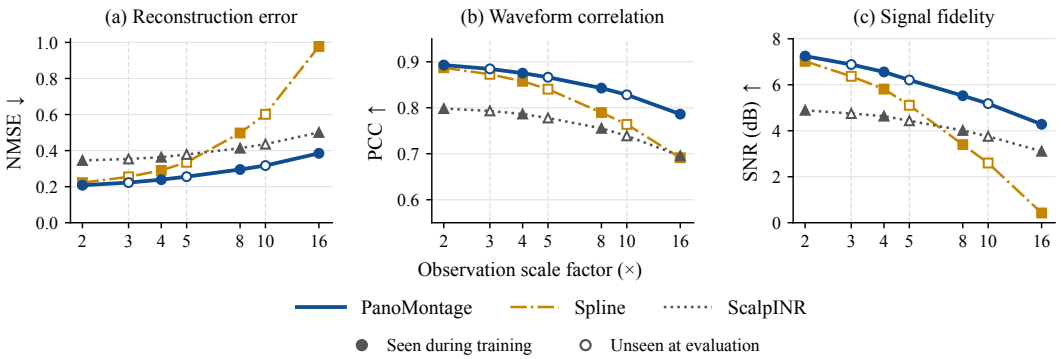

Figure 2: CCEP-COReg density generalization. Filled and hollow markers denote training and unseen factors. Panels show (a) missing-channel NMSE (↓), (b) PCC (↑), and (c) SNR (↑); lines connect measured values on logarithmic factor axes. All methods use the same 800 windows and masks per factor; values are example-weighted means.

4.3 Evaluation at Unseen Densities and Locations

Unseen observation densities. Figure 2 evaluates factors $3\times$, $5\times$, and $10\times$ unseen in training alongside $2\times$, $4\times$, $8\times$, and $16\times$. PanoMontage obtains NMSE 0.2227, 0.2554, and 0.3171 at the unseen factors, respectively, 12.5%, 23.8%, and 47.3% below Spline; it has the lowest mean NMSE at all seven factors.

Withheld electrode locations. We withhold 37 CCEP-COReg electrodes from downstream training inputs, reconstruction supervision, and validation checkpoint selection. At test time, we reconstruct them from the remaining observed electrodes and compare with their recorded waveforms.

Table 2: Mean NMSE at withheld CCEP-COReg locations. Learned methods average three training seeds.

Method	2×	4×	8×	16×
Spline	0.2189	0.2960	0.5415	1.0356
ScalpINR	0.3447	0.3684	0.4176	0.5116
PanoMontage	**0.2072**	**0.2363**	**0.2970**	**0.3951**

Table 2 uses shared test windows and masks. PanoMontage has lower mean NMSE than ScalpINR and Spline at all four factors. Table 3 reports the paired subject results: PanoMontage leads in 18 of 20 subject–factor cells; Spline leads for *sub-25* and *sub-26* at $2\times$. All five subjects favor PanoMontage at $4\times$, $8\times$, and $16\times$. Appendix D.4 details the protocol.

Table 3: Per-subject CCEP-COReg withheld-location NMSE (↓); paired masks, three-seed means for learned methods. Bold marks each factor minimum.

Subject	Method	2×	4×	8×	16×
sub-01	Spline	0.1576	0.2223	0.3802	0.7150
	ScalpINR	0.2551	0.2751	0.3188	0.4066
	PanoMontage	**0.1448**	**0.1676**	**0.2152**	**0.2969**
sub-09	Spline	0.3335	0.4280	0.8181	1.6646
	ScalpINR	0.4489	0.4792	0.5314	0.6400
	PanoMontage	**0.2974**	**0.3373**	**0.4007**	**0.5126**
sub-25	Spline	**0.0944**	0.1309	0.2508	0.5131
	ScalpINR	0.1896	0.2091	0.2589	0.3558
	PanoMontage	0.1062	**0.1266**	**0.1911**	**0.2832**
sub-26	Spline	**0.2050**	0.2761	0.4875	0.9511
	ScalpINR	0.4570	0.4738	0.5114	0.5905
	PanoMontage	0.2109	**0.2300**	**0.2896**	**0.3945**
sub-32	Spline	0.3042	0.4225	0.7708	1.3341
	ScalpINR	0.3727	0.4047	0.4673	0.5650
	PanoMontage	**0.2765**	**0.3201**	**0.3885**	**0.4886**

Dense spatial queries. With 16 observed channels, PanoMontage queries 4,096 scalp-cap coordinates. Figure 3(c) displays 30 spatially distributed waveforms at these locations on a shared scale.

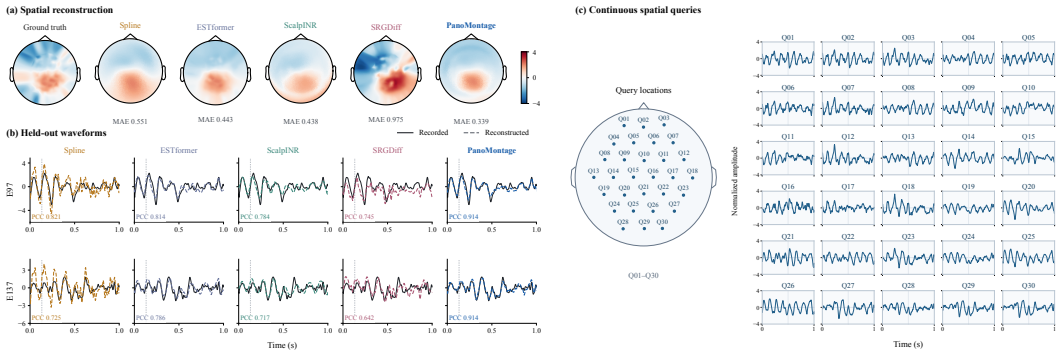

Figure 3: CCEP-COReg (16 observed, 115 held out): (a) topographies at 0.132 s, with MAE on held-out electrodes; (b) E97/E137 waveforms; (c) 30 of 4,096 spatial queries. SRGDiff is uncalibrated here; Table 1 uses validation-calibrated outputs.

Figure 3 shows a favorable example selected using the protocol in Appendix D.5. PanoMontage reaches a snapshot MAE of 0.339 versus 0.438 for the strongest baseline, with full-window PCC 0.914 at both displayed electrodes.

The withheld-location experiment evaluates standardized waveform prediction at 37 recorded electrodes excluded from downstream supervision. The dense-query experiment evaluates the model at 4,096 scalp coordinates, most of which lack reference recordings, and therefore measures query coverage rather than voltage accuracy at uninstrumented locations. Target-channel normalization statistics define the withheld-channel targets but are not provided to the model.

4.4 Role of the Intermediate Montage Field

Intermediate field versus direct reconstruction. Table 4 compares scratch-trained PanoMontage with a parameter-matched direct $O \to Q$ path that omits P_L (0.439044M versus 0.438962M parameters and 0.1795 versus 0.1941 GFLOPs at $8\times$). Direct has lower NMSE at $2\times$. PanoMontage reduces NMSE by 0.7%, 8.1%, and 15.9% at $4\times$, $8\times$, and $16\times$, respectively. Across these tested densities, the advantage of the intermediate field increases as the observations become sparser. Additional efficiency results appear in Table 20.

Table 4: Mean NMSE over three seeds on CCEP-COReg. Both models are trained from scratch; Direct is parameter-matched.

Model	$2\times$	$4\times$	$8\times$	$16\times$
PanoMontage	0.2230	**0.2490**	**0.3140**	**0.4161**
Direct	**0.2106**	0.2508	0.3415	0.4950

Waveform and context contributions. The parameter-matched comparison identifies the benefit of the combined two-stage architecture; it does not separate learned intermediate waveforms from transmitted context. Table 5 tests these paths separately on SEED-IV. Removing H_L while retaining the learned waveform, or replacing that waveform with the spline estimate while retaining H_L, increases NMSE at all three densities. These controls use one matched training seed and 1,600 paired test windows per factor. Appendix E.1 reports the full metrics and implementation details.

Table 5: SEED-IV independent field-path NMSE (↓). One training seed; 1,600 paired test windows per factor.

Variant	$2\times$	$4\times$	$8\times$
Full PanoMontage	**0.18930**	**0.24323**	**0.33843**
No transmitted H_L	0.21495	0.26880	0.35757
Spline wave + H_L	0.19507	0.28366	0.40187

4.5 Pretraining and Component Ablations

Dynamic montage sampling. On CCEP-COReg, we compare fixed configurations, random positions with fixed counts, and dynamic positions and counts under paired pretraining and fine-tuning. Dynamic sampling reduces mean NMSE relative to random positions with fixed counts by 1.1%,

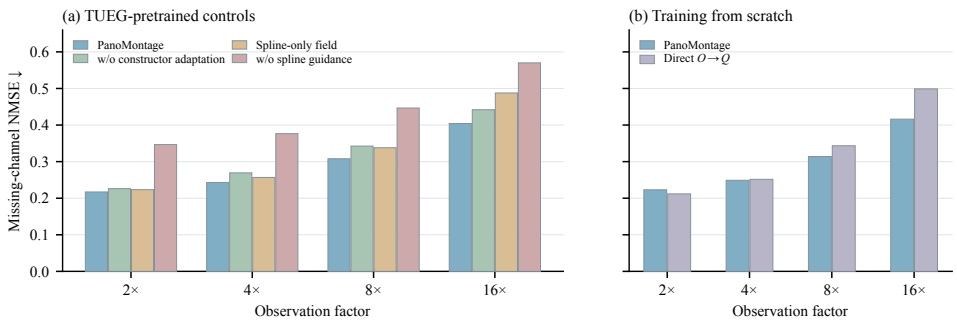

Figure 4: CCEP-COReg component ablations: three-seed mean missing-channel NMSE (↓). (a) TUEG-pretrained controls freeze the constructor, use a spline-only field, or remove spline guidance. (b) Scratch-trained PanoMontage and the Direct decoder. Standard deviations, PCC, and SNR appear in Table 17.

1.3%, and 1.0% at 8×, 10×, and 16×, with agreement across all three seeds. Relative to fully fixed configurations, the gains are 0.24%, 0.42%, and 0.47%, with two of three seeds improving at each factor.

The dynamic policy varies observed, intermediate, and query counts jointly. Fixed-count pretraining uses $(N_O, N_L, N_Q) = (8, 56, 5)$, matching the mean counts of the dynamic policy. All three variants use the same number of pretraining updates and the same downstream evaluation protocol. Table 6 shows the lowest dynamic-policy mean at all three sparse factors, although the margins against fixed montages are small and do not identify which sampled cardinality drives the gain. Appendix E.2 reports all seven densities.

Table 6: Sparse-factor CCEP-COReg NMSE across three paired seeds (↓).

Pretraining policy	8×	10×	16×
Fixed montage	0.30525	0.32802	0.39877
Random positions, fixed counts	0.30780	0.33084	0.40093
Dynamic positions and counts	**0.30451**	**0.32666**	**0.39689**

Spline guidance. Removing the internal spline estimates causes the largest increase in NMSE and variation between training seeds in Figure 4. Mean NMSE rises from 0.308 to 0.447 at 8× and from 0.404 to 0.570 at 16×, even though the learned correction remains trainable. The three-seed NMSE standard deviation also rises from 0.001 to 0.088 at 8× and from 0.003 to 0.068 at 16× (Table 17). Thus the spline estimate provides a stable starting signal for residual correction, not merely a standalone interpolation baseline.

Pretraining and adaptation. Freezing the pretrained constructor or replacing it with a spline-only field raises error at every factor. At 16×, these variants yield NMSE values of 0.442 and 0.488, compared with 0.404 after joint adaptation and 0.416 from scratch. The frozen variant still trains the query stage, so its gap from joint adaptation points to the value of updating the intermediate field; the spline-only gap shows what learned field construction adds beyond interpolation. Complete seed-level results appear in Table 17; paired dynamic-policy differences appear in Appendix E.2.

5 Conclusion

We presented PanoMontage for any-resolution EEG reconstruction across variable observed and target montages. PanoMontage constructs an intermediate latent montage before querying the requested electrodes, using differentiable spherical splines for spatial estimation and gated residuals for signal-dependent correction. Dynamic-montage self-supervised pretraining learns from heterogeneous recordings by varying electrode positions and counts. Evaluation on four subject-held-out datasets yields the lowest NMSE in 12 of 14 settings and supports generalization to unseen observation densities and locations withheld from downstream supervision. Parameter-matched comparisons show that the benefit of the intermediate field grows with input sparsity, while component ablations support the complementary roles of spline guidance and learned correction. Dynamic-montage controls further show improved reconstruction under sparse observations.

AI USE STATEMENT

AI aided language editing, selected literature searches, figure preparation, and experiment code. The authors developed the ideas, reviewed code, checked figures against the model and data, and take responsibility for this paper.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide an anonymous code repository and detail preprocessing, splits, training, and evaluation in the appendices.

REFERENCES

Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti S. Hämäläinen. MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 7:267, 2013. doi: 10.3389/fnins.2013.00267.

Janin Jäger, Alexander Klein, Martin Buhmann, and Wolfgang Skrandies. Reconstruction of electroencephalographic data using radial basis functions. *Clinical Neurophysiology*, 127(4):1978–1983, 2016. doi: 10.1016/j.clinph.2016.01.003.

Wei-Bang Jiang, Li-Ming Zhao, and Bao-Liang Lu. Large brain model for learning generic representations with tremendous EEG data in BCI. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=QzTpTRVtrP.

Zijian Kang, Weiming Zeng, Yueyang Li, Shengyu Gong, Hongjie Yan, Wai Ting Siok, and Nizhuan Wang. TGSD: Topology-guided state-space diffusion framework for EEG spatial super-resolution. *arXiv preprint arXiv:2606.03998*, 2026. URL https://arxiv.org/abs/2606.03998.

Demetres Kostas, Stéphane Aroca-Ouellette, and Frank Rudzicz. BENDR: Using transformers and a contrastive self-supervised learning task to learn from massive amounts of EEG data. *Frontiers in Human Neuroscience*, 15:653659, 2021. doi: 10.3389/fnhum.2021.653659.

Alex Lazarovich, Ofir Itzhak Shahar, Gur Elkin, and Ohad Ben-Shahar. EMAG: Differentiable 4D gaussian mixture splatting for EEG spatial super-resolution. *arXiv preprint arXiv:2605.29731*, 2026. URL https://arxiv.org/abs/2605.29731.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753. PMLR, 2019.

Dongdong Li, Zhongliang Zeng, Zhe Wang, and Hai Yang. ESTformer: Transformer utilising spatiotemporal dependencies for electroencephalogram super-resolution. *Knowledge-Based Systems*, 317:113345, 2025. doi: 10.1016/j.knosys.2025.113345.

Hongjun Liu, Leyu Zhou, Zijianghao Yang, and Chao Yao. Coordinate-queryable neural field reconstruction for EEG spatial super-resolution with unseen-electrode generation. *arXiv preprint arXiv:2606.23707*, 2026a. URL https://arxiv.org/abs/2606.23707.

Hongjun Liu, Leyu Zhou, Zijianghao Yang, and Chao Yao. Step-aware residual-guided diffusion for EEG spatial super-resolution. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=2vN2GlfV6T.

Christoph M. Michel and Micah M. Murray. Towards the utilization of EEG as a brain imaging tool. *NeuroImage*, 61(2):371–385, 2012. doi: 10.1016/j.neuroimage.2011.12.039.

Ezequiel Mikulan, Simone Russo, Sara Parmigiani, Simone Sarasso, Flavia Maria Zauli, Annalisa Rubino, Pietro Avanzini, Anna Cattani, Alberto Sorrentino, Steve Gibbs, Francesco Cardinale,

Ivana Sartori, Lino Nobili, Marcello Massimini, and Andrea Pigorini. Simultaneous human intracerebral stimulation and HD-EEG: Ground-truth for source localization methods. *Scientific Data*, 7(1):127, 2020. doi: 10.1038/s41597-020-0467-x.

Iyad Obeid and Joseph Picone. The temple university hospital EEG data corpus. *Frontiers in Neuroscience*, 10:196, 2016. doi: 10.3389/fnins.2016.00196.

Robert Oostenveld and Peter Praamstra. The five percent electrode system for high-resolution EEG and ERP measurements. *Clinical Neurophysiology*, 112(4):713–719, 2001. doi: 10.1016/S1388-2457(00)00527-7.

Sara Parmigiani, Ezequiel Mikulan, Simone Russo, Simone Sarasso, Flavia Maria Zauli, Annalisa Rubino, Anna Cattani, Matteo Fecchio, Davide Giampiccolo, Jacopo Lanzone, Piergiorgio D'Orio, Maria Del Vecchio, Pietro Avanzini, Lino Nobili, Ivana Maria Sartori, Marcello Massimini, and Andrea Pigorini. Simultaneous stereo-EEG and high-density scalp EEG recordings to study the effects of intracerebral stimulation parameters. *Brain Stimulation*, 15(3):664–675, 2022. doi: 10.1016/j.brs.2022.04.007.

F. Perrin, J. Pernier, O. Bertrand, and J. F. Echallier. Spherical splines for scalp potential and current density mapping. *Electroencephalography and Clinical Neurophysiology*, 72(2):184–187, 1989. doi: 10.1016/0013-4694(89)90180-6.

Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Ramesh Srinivasan, Paul L. Nunez, Don M. Tucker, Richard B. Silberstein, and Peter J. Cadusch. Spatial sampling and filtering of EEG with spline laplacians to estimate cortical potentials. *Brain Topography*, 8(4):355–366, 1996. doi: 10.1007/BF01186911.

Yunbo Tang, Dan Chen, Honghai Liu, Chang Cai, and Xiaoli Li. Deep EEG superresolution via correlating brain structural and functional connectivities. *IEEE Transactions on Cybernetics*, 53 (7):4410–4422, 2023. doi: 10.1109/TCYB.2022.3178370.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Guangyu Wang, Wenchao Liu, Yuhong He, Cong Xu, Lin Ma, and Haifeng Li. EEGPT: Pretrained transformer for universal and reliable representation of EEG signals. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1239.

Shuqiang Wang, Tong Zhou, Yanyan Shen, Ye Li, Guoheng Huang, and Yong Hu. Generative AI enables EEG super-resolution via spatio-temporal adaptive diffusion learning. *IEEE Transactions on Consumer Electronics*, 71(1):1034–1045, 2025. doi: 10.1109/TCE.2025.3528438.

Christopher Warner, Jonas Mago, JR Huml, Mohamed Osman, and Beren Millidge. ZUNA: Flexible EEG superresolution with position-aware diffusion autoencoders. *arXiv preprint arXiv:2602.18478*, 2026. URL https://arxiv.org/abs/2602.18478.

Laura Yao, Gengwei Zhang, Moajjem Chowdhury, Yunmei Liu, and Tianlong Chen. Geometry- and relation-aware diffusion for EEG super-resolution. *arXiv preprint arXiv:2602.02238*, 2026. URL https://arxiv.org/abs/2602.02238.

Wei-Long Zheng and Bao-Liang Lu. Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks. *IEEE Transactions on Autonomous Mental Development*, 7(3):162–175, 2015. doi: 10.1109/TAMD.2015.2431497.

Wei-Long Zheng, Wei Liu, Yifei Lu, Bao-Liang Lu, and Andrzej Cichocki. EmotionMeter: A multimodal framework for recognizing human emotions. *IEEE Transactions on Cybernetics*, 49 (3):1110–1122, 2019. doi: 10.1109/TCYB.2018.2797176.

A Datasets and Evaluation Protocols

A.1 Datasets, Splits, and Preprocessing

We pretrain on unlabeled TUEG recordings (Obeid & Picone, 2016) and evaluate on the four datasets in Table 7 (Zheng & Lu, 2015; Zheng et al., 2019; Parmigiani et al., 2022; Mikulan et al., 2020). SEED uses subjects 1–12 for training and validation and 13–15 for test; its stratified validation subset contains 1,515 of 30,312 non-overlapping training-subject windows, with 7,578 test windows. The cached train, validation, and test intervals share no raw time samples, but train and validation share trial-level normalization statistics. SEED-IV uses subjects 1, 2, 5, 6, 9–15 for training, 3 and 7 for validation, and 4 and 8 for test. Its overlapping windows stay within subject-disjoint splits. The other datasets use held-out validation subjects; Localize-MI has one test subject.

Table 7: Dataset channel counts and data partitions. Channel counts denote the channels retained per recording after quality and coordinate screening; split entries are numbers of participants unless stated otherwise.

Dataset	Channels	Train	Validation	Test
SEED	62	12	5% windows	3
SEED-IV	62	11	2	2
CCEP-COReg	125–185	26	5	5
Localize-MI	162–246	5	1	1

SEED and SEED-IV use 4-s windows with 4-s and 2-s strides, respectively; CCEP-COReg uses 1-s stimulus-locked epochs, Localize-MI uses epochs from -250 to 10 ms, and TUEG pretraining uses 4-s windows with a 2-s stride. Reconstruction is evaluated in channel-standardized waveform units. For each channel c, preprocessing computes

$$m_c = \mathrm{median}_{t\in\mathcal{T}}\, x_c(t), \quad s_c = 1.4826\,\mathrm{median}_{t\in\mathcal{T}}\,|x_c(t) - m_c| + 10^{-6},$$
$$z_c(t) = \mathrm{clip}\left(\frac{x_c(t) - m_c}{s_c}, -8, 8\right). \tag{10}$$

Here $\mathcal{T}$ spans one filtered trial for SEED/SEED-IV, one recording for TUEG, and all retained epochs and time samples of a run for CCEP-COReg/Localize-MI. Statistics are computed per channel before window sampling, including separately for each test trial/run. Target-channel statistics define evaluation targets but are never provided to the model. We add no cross-channel rereferencing. The high-density epochs retain their supplied reference; the available metadata specify baseline subtraction but not the full upstream reference.

Because spline interpolation acts on z, dense queries predict standardized waveforms, not calibrated microvolt amplitudes at uninstrumented locations. The exploratory source-localization analysis restores recorded-channel units as $\hat{x}_c = s_c\hat{z}_c + m_c$; clipping prevents exact inversion at extreme samples. A shared, observation-only physical-unit scale remains untested.

A.2 Electrode Coordinates and Observation Sampling

SEED-family coordinates follow the standard-1005 montage (Oostenveld & Praamstra, 2001), CCEP-COReg uses the HydroCel-256 template, and Localize-MI uses the provided subject-specific electrode coordinates. All are normalized to the unit sphere.

We define the scale factor as the ratio of target to observed channels. For SEED and SEED-IV, the $2\times$, $4\times$, and $8\times$ settings retain 32, 16, and 8 channels, consistent with prior EEGSR evaluations (Li et al., 2025; Wang et al., 2025; Liu et al., 2026b;a). The high-density datasets also include $16\times$: Localize-MI retains 128, 64, 32, or 16 channels on a fixed 256-coordinate target grid, and CCEP-COReg retains approximately N_{valid}/r channels at factor r. The factor sets the observed-channel count; positions are resampled per example.

Fixed pseudorandom seeds give paired methods identical evaluation windows and masks. Metrics use only measured targets excluded from the input; Appendix D.4 defines the separate location-withholding protocol. Figure 5 shows these spatial supports.

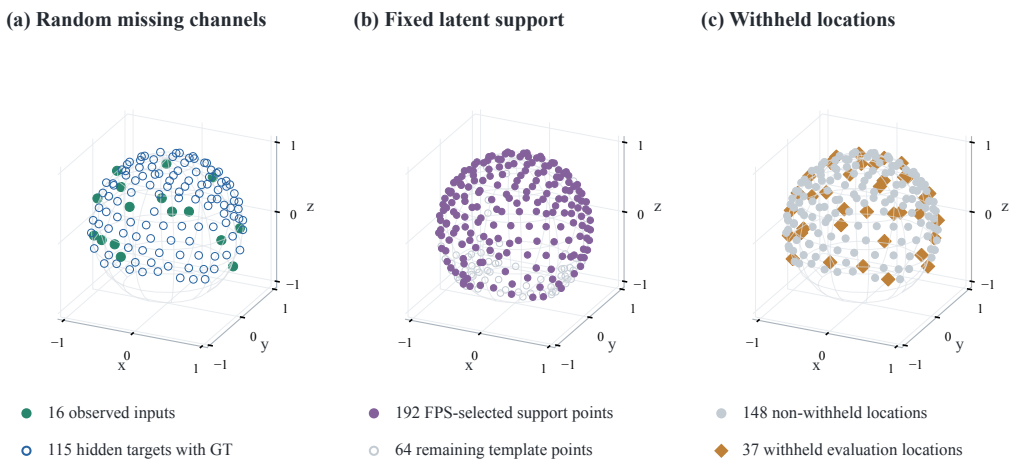

Figure 5: Three spatial protocols shown in a common three-dimensional coordinate view. (a) One CCEP-COReg $8\times$ test example has 16 observed inputs and 115 measured targets; the observation mask changes between examples. (b) The fixed latent support contains 192 farthest-point-selected locations on the GSN-HydroCel-256 template. They are computational positions, not measurements. (c) The withheld-location experiment selects 37 locations by geometry from the 185-location CCEP-COReg training union. Their waveforms are excluded from downstream inputs and supervision. Evaluation uses recorded references at the withheld locations available in each test recording. The locations may occur in TUEG pretraining. All panels use the same unit-normalized coordinates, view, and axes; the wireframe marks the unit sphere.

A.3 Evaluation Metrics and Statistical Aggregation

For evaluation batch b, M_b marks valid target channels excluded from the input, $E_b = \widehat{X}_b - X_b$, and $\mathcal{C}_b$ contains valid missing example–channel pairs. The mask is broadcast across time:

$$\begin{aligned} \mathrm{NMSE}_b &= \frac{\|M_b \odot E_b\|_F^2}{\|M_b \odot X_b\|_F^2 + \epsilon}, \\ \mathrm{SNR}_b &= 10 \log_{10} \frac{\|M_b \odot X_b\|_F^2}{\|M_b \odot E_b\|_F^2 + \epsilon}, \\ \mathrm{PCC}_b &= \frac{1}{|\mathcal{C}_b|} \sum_{(i,c)\in\mathcal{C}_b} \mathrm{corr}_t(\widehat{X}_{i,c,:}, X_{i,c,:}). \end{aligned} \tag{11}$$

Aggregation across evaluation units. The reported evaluations use task-specific aggregation units. The CCEP-COReg result in Table 1 averages metrics from 50 batches of eight examples at each factor (400 examples in total), with seed 20260716. The unseen-density results use example-weighted means (Appendix D.3); the withheld-location experiment averages within recordings, within subjects, then equally across five test subjects (Appendix D.4). Three-seed experiments report mean and sample standard deviation across training seeds.

B Model and Training Details

B.1 Spherical-Spline Reconstruction

The two stages use $S_{L\leftarrow O} = S_\alpha(P_O, P_L)$ and $S_{Q\leftarrow L} = S_\alpha(P_L, P_Q)$ in Eq. (3). With Perrin kernel matrices $\mathbf{K}_{OO}$ and $\mathbf{K}_{LO}$, the observed-to-latent interpolation matrix solves

$$\begin{bmatrix} \mathbf{K}_{OO} + \alpha \mathbf{I} & \mathbf{1}_{N_O} \\ \mathbf{1}_{N_O}^\top & 0 \end{bmatrix} \begin{bmatrix} S_{L\leftarrow O}^\top \\ c_L^\top \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{LO}^\top \\ \mathbf{1}_{N_L}^\top \end{bmatrix}, \tag{12}$$

where $c_L \in \mathbb{R}^{N_L}$ is auxiliary and $\mathbf{1}_n$ is the length-n all-one vector. Replacing (OO, LO) by (LL, QL) gives the latent-to-query system. Both FP32 solves use Perrin order $K = 32$, stiffness $m = 4$, and regularization $\alpha = 10^{-5}$.

B.2 Network Architecture

Spatiotemporal Tokenization. Coordinate encoding concatenates p, p^2, and sine/cosine features at frequencies $\{1, 2, 4, 8\}$, giving a 30-dimensional vector. Each token sums 64-dimensional temporal, spectral, and coordinate features before layer normalization. Temporal convolutions use kernels 15 and 7 followed by global average pooling; the spectral branch pools centered log-amplitude rFFT to 64 bins. Separate encoders serve the two stages; query-stage source tokens add H_L before normalization, as in Eq. (4).

Spatial Correspondence Attention. The cross-attention operator in Eq. (5) has four heads, residual normalization, and a $64 \to 256 \to 64$ GELU feed-forward block with zero dropout. The stages have separate attention, feed-forward, and normalization parameters; A_D averages the head weights.

Geometry-Conditioned Context Encoding. For each destination location, G_D contains nearest and mean source distance, squared nearest distance, and $N_S/32$. The pairs (P_O, P_L) and (P_L, P_Q) give G_L and G_Q, respectively. The refinement Φ_c in Eq. (6) has two spatial self-attention blocks followed by normalization; the query gate uses (Q_Q, C_Q, G_Q).

Spline-Anchored Residual Refinement. Each stage stacks the three waveforms in Eq. (7) into an $N_D \times 3 \times T$ tensor and uses a separate 64-hidden-channel residual network with kernels 15, 7, and 1. The gate is scalar per destination with $g_{\max} = 0.35$. Feature dimension $d = 64$ gives 439,044 parameters in total.

B.3 Training Objectives

Pretraining losses. The query reconstruction term in Eq. (9) combines waveform, correlation, and spectral losses:

$$\mathcal{L}_{\text{rec}}^{\text{pre}} = \mathcal{L}_{\text{MSE}} + 0.5\mathcal{L}_1 + 0.1\mathcal{L}_{\text{corr}} + 0.05\mathcal{L}_{\text{FFT}}. \tag{13}$$

MSE and L_1 average waveform errors over $N_Q T$ samples; $\mathcal{L}_{\text{corr}}$ averages $1 - \text{PCC}$ over query channels, with PCC computed over time. The spectral term is

$$\begin{aligned} \mathcal{L}_{\text{FFT}} &= \frac{1}{N_Q F} \left\| \log\Big(1 + |\,\text{rFFT}(\widehat{X}_Q)|\Big) - \log(1 + |\,\text{rFFT}(X_Q)|) \right\|_1, \\ F &= \lfloor T/2 \rfloor + 1. \end{aligned} \tag{14}$$

The real FFT is taken along time without temporal-mean subtraction; losses are averaged across batch windows.

The auxiliary objective contains cycle and composition terms:

$$\begin{aligned} \mathcal{L}_{\text{aux}}^{\text{pre}} &= \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}} + \lambda_{\text{comp}}\mathcal{L}_{\text{comp}}, \\ \mathcal{L}_{\text{cyc}} &= \frac{\|S_{O\leftarrow L}\widehat{X}_L - X_O\|_F^2}{N_O T}, \\ \mathcal{L}_{\text{comp}} &= \frac{\left\| S_{Q\leftarrow L}\widehat{X}_L - \text{sg}(S_{Q\leftarrow O}X_O) \right\|_F^2}{N_Q T}. \end{aligned} \tag{15}$$

Here $S_{D\leftarrow S}$ is spherical-spline interpolation and sg stops gradients. The cycle term projects $\widehat{X}_L$ back to observed locations; composition compares its pre-correction query spline estimate with the detached direct estimate from X_O. We use $\lambda_\Delta = 0.01$, $\lambda_{\text{cyc}} = 0.02$, and $\lambda_{\text{comp}} = 0.005$.

Fine-tuning losses. Fine-tuning combines waveform MSE and L_1, masked NMSE as defined in Appendix A.3, tail-normalized error, waveform correlation, log-spectral error, and temporal moments. Their weights appear in Table 8. For each example, the tail-normalized loss first computes

the same energy ratio with its denominator floored at 5% of the median positive target energy in the batch, then averages the largest 25% of these ratios. The moment term matches the temporal mean and log standard deviation of each target channel:

$$\mathcal{L}_{\mathrm{mom}} = \mathbb{E}_M\left[\frac{|\mu(\widehat{X}_Q) - \mu(X_Q)|}{\sigma(X_Q) + \epsilon} + |\log \sigma(\widehat{X}_Q) - \log \sigma(X_Q)|\right]. \tag{16}$$

The correlation loss is $1 - \mathrm{PCC}$; the spectral loss is L_1 distance between centered log-amplitude rFFT coefficients.

Table 8: Weights used for target-dataset fine-tuning. PCC denotes $1 - \mathrm{PCC}$; Tail denotes tail-normalized error. Visible is an NMSE term on observed channels. Spline-dom. penalizes target errors that exceed the spherical-spline reference for the same example.

Dataset	MSE	NMSE	Tail	L_1	PCC	FFT	Moment	Residual	Visible	Spline-dom.
SEED	0.10	0.40	0.05	0.02	4.00	0.01	0.01	0.0005	0	0
SEED-IV	0.10	0.40	0.05	0.02	4.00	0.01	0.01	0.0005	0	0
Localize-MI	0.10	0.40	0.05	0.02	4.00	0.01	0.01	0.0005	0	0
CCEP-COReg	0.20	1.00	0.20	0.05	0.20	0.05	0.02	0.0010	0.05	0.50

B.4 Dynamic-Montage Pretraining

At the start of each batch, the three cardinalities are sampled independently from discrete uniform distributions:

$$N_O \sim \mathrm{Unif}\{4, 8, 12\}, \qquad N_L \sim \mathrm{Unif}\{48, 64\}, \qquad N_Q \sim \mathrm{Unif}\{4, 6\}. \tag{17}$$

Sampling follows Eq. (8): eligible recordings have at least $N_O + N_Q$ electrodes, and $\mathcal{B}_X$ is the `standard_1020` pool excluding names present in the recording. Cardinalities are shared per batch, while O, L, and Q are resampled without replacement for each window draw.

We pretrain both modules on the full TUEG training split for 20,000 steps (batch 32, learning rate 2×10^{-4}, seed 20260702). The final checkpoint initializes the constructor and all query decoders except CCEP-COReg's, which is reset. Pretraining and fine-tuning use AdamW with weight decay 10^{-4}, gradient clipping at 1.0, and bfloat16 mixed precision; spline solves remain FP32.

B.5 Target-Domain Fine-Tuning

The intermediate support contains 192 points selected by farthest-point sampling (Qi et al., 2017) from GSN-HydroCel-256 and remains fixed. The primary loss selects measured, unobserved targets using the mask in Section 3.4; CCEP-COReg additionally supervises observed channels. All models update both stages; Table 9 specifies query-decoder initialization and other dataset settings.

One model per dataset trains across observation densities for at most 50,000 updates. Every 1,000 updates, validation missing-channel NMSE is averaged equally across training densities; its minimum selects a single checkpoint for all evaluation densities. Early stopping has 10,000-update patience and a separate 0.2%-relative-improvement anchor. Selected updates are 49k/49k/28k/27k for SEED/SEED-IV/CCEP-COReg/Localize-MI, respectively. Mechanism-control schedules are separate.

Table 9: Target-domain PanoMontage training protocols. All main models initialize the observed-to-latent field constructor from TUEG and update both modules. Projection copies observed channels to the requested output when enabled.

Dataset	Training N_O	Evaluation N_O	Query-decoder init.	Batch	LR	Projection	Max. steps
SEED	8–32	32/16/8	TUEG	64	2×10^{-5}	Yes	50k
SEED-IV	8–32	32/16/8	TUEG	32	2×10^{-5}	Yes	50k
Localize-MI	16–128	128/64/32/16	TUEG	16	2×10^{-5}	Yes	50k
CCEP-COReg	factors 2/4/8/16	$\lfloor N_{\mathrm{valid}}/r \rfloor$	reset	16	5×10^{-5}	No	50k

C Baseline Implementations and Comparison Settings

C.1 Implementation Sources and Input Adaptation

We compare spherical splines, a released-source adaptation of SRGDiff (Liu et al., 2026b), the released ZUNA checkpoint (Warner et al., 2026), and paper-specification reproductions of STAD (Wang et al., 2025), ScalpINR (Liu et al., 2026a), ESTformer (Li et al., 2025), and TopoDiff (Yao et al., 2026). SRGDiff additionally receives validation-only amplitude calibration (Appendix C.2); ESTformer uses fixed electrode slots with mask tokens, SIM/TRM depth 1/1, and dropout 0.5. TGSD is excluded because its released-source adaptation did not complete training. Target-trained methods share the splits, normalization, observed counts, coordinates, and metrics in Appendix A.

C.2 Training Budgets and Model Selection

Fixed-montage models train separately at each scale with random input masks; ScalpINR and PanoMontage train across densities. Table 10 reports the executed budgets. Minimum validation NMSE selects checkpoints, while early stopping uses a separate 0.2%-relative-improvement anchor. The update ceiling is a budget, not a convergence criterion.

Table 10: Executed target-domain optimization. Epochs are complete shuffled passes through cached training windows, retaining partial batches. SRGDiff settings in this table refer to its diffusion stage.

Method	Optimizer / LR	Batch	Ceiling	LR schedule	Validation	Patience
PanoMontage	AdamW / 2–$5 \cdot 10^{-5}$	16/32/64	50k updates	constant	1k updates	10k updates
ESTformer	Adam / 10^{-4}	32	300 epochs	cosine	5 epochs	30 epochs
STAD	Adam / $2 \cdot 10^{-4}$	32	300 epochs	constant	5 epochs	20k updates
TopoDiff	AdamW / $5 \cdot 10^{-4}$	8	300 epochs	cosine	5 epochs	20k updates
ScalpINR	AdamW / 10^{-4}	64	100k updates	5k warmup + cosine	2k updates	20k updates
SRGDiff	Adam / 10^{-4}	256	100k updates	milestones 100/150 epochs	5k updates	20k updates

Weight decay is 0.01 for TopoDiff, 10^{-4} for ScalpINR and PanoMontage, and zero for the Adam baselines. ESTformer combines temporal MAE and Fourier MSE; STAD trains an autoencoder then diffusion; ScalpINR combines missing-channel MSE, observed fidelity, and correlation; TopoDiff uses clean-waveform MSE. SRGDiff sequentially trains a VAE (waveform L_1, KL, adversarial), mapper (latent and waveform MSE), residual predictor (latent L_1), and diffusion model (smooth-L_1 noise prediction). Earlier stages are frozen after validation selection; latent normalization uses training data only.

SRGDiff validation diagnostics. On frozen factor-2$\times$ validation checkpoints, full-input VAE and mapper reconstructions have lower error than sampled diffusion outputs, whose amplitudes are enlarged. The released sampler disconnects the trained residual predictor, but these probes do not identify the source of the amplitude increase. We evaluate the sampled outputs with the validation-only calibration below.

Validation-only SRGDiff amplitude calibration. For each of 14 dataset–factor settings, we freeze the validation-selected checkpoint and fit a nonnegative gain without intercept on 400 validation windows (50 batches of eight). For missing-channel predictions p_b and targets y_b in batch b, minimizing mean batch NMSE gives

$$g = \max\left(0, \frac{\sum_b \langle p_b, y_b\rangle / \|y_b\|_2^2}{\sum_b \|p_b\|_2^2 / \|y_b\|_2^2}\right).$$

Gains and checkpoint hashes are frozen before testing on 400 windows per setting with 50 DDIM steps. Raw and calibrated predictions share a diffusion realization; observed channels are restored after scaling. The validation-fitted gain is an adaptation to the released sampler, not part of its default configuration.

The gain lowers test NMSE below the zero-prediction reference of 1 in every setting without changing PCC (Table 11). Because attenuation can also suppress uncorrelated noise, the improvement

does not identify the source of the amplitude mismatch. These results characterize the calibrated adaptation rather than the published default.

Table 11: Paired SRGDiff test NMSE before and after validation-only amplitude calibration. Each gain is fitted without test access.

Dataset	Factor	Gain	Raw	Calibrated
SEED	2×	0.3446	2.10840	0.54407
SEED	4×	0.4005	1.51456	0.65731
SEED	8×	0.2525	2.42770	0.85679
SEED-IV	2×	0.2110	6.20208	0.53913
SEED-IV	4×	0.1904	6.48319	0.62843
SEED-IV	8×	0.1802	4.99330	0.73244
CCEP-COReg	2×	0.1905	3.68785	0.92211
CCEP-COReg	4×	0.1820	4.35109	0.87139
CCEP-COReg	8×	0.2603	2.54738	0.84781
CCEP-COReg	16×	0.1343	3.67025	0.96161
Localize-MI	2×	0.4712	1.64909	0.70425
Localize-MI	4×	0.4679	1.61857	0.74605
Localize-MI	8×	0.3885	1.76589	0.83013
Localize-MI	16×	0.3742	1.46392	0.86974

C.3 ZUNA Evaluation and Amplitude Calibration

ZUNA uses the released ZUNA1.1 checkpoint with one gain and offset per dataset, fitted on missing-channel validation targets to align output amplitude with target normalization, then frozen for test evaluation. Test-fitted oracle calibration is excluded.

D Additional Reconstruction Results

D.1 Complete Reconstruction Metrics

Table 12 gives the complete NMSE, PCC, and SNR results underlying the main comparison.

Table 12: Complete missing-channel NMSE (↓), PCC (↑), and SNR (↑, dB) for the versions in Table 1. SRGDiff includes validation-only calibration; TGSD is omitted. Bold and underlining mark the best and second-best unrounded values.

		SEED			SEED-IV			CCEP-COReg				Localize-MI			
Method	Metric	2×	4×	8×	2×	4×	8×	2×	4×	8×	16×	2×	4×	8×	16×
Spline	NMSE	0.42100	0.84100	1.66700	0.22000	0.42600	0.89100	0.24400	0.30100	0.52100	1.15300	0.28060	0.32300	0.43100	0.79000
	PCC	0.792	0.711	0.613	0.878	0.804	0.707	0.889	0.863	0.798	0.689	0.848	0.828	0.790	0.705
	SNR	4.45	1.85	-0.92	6.64	3.79	0.77	6.81	5.78	3.41	0.08	5.62	5.00	3.79	1.16
ESTformer	NMSE	0.29069	0.34222	**0.43253**	0.16784	0.22565	**0.31598**	0.21419	0.24946	0.31093	0.39889	0.29787	0.29588	0.32459	0.40724
	PCC	0.836	0.801	**0.739**	0.903	0.866	0.811	0.890	0.872	0.838	**0.785**	0.842	0.836	0.811	0.751
	SNR	5.81	5.00	**3.87**	**7.84**	6.54	**5.06**	7.14	6.37	5.29	4.14	5.35	5.37	4.97	3.96
STAD	NMSE	0.56925	0.64959	0.78243	0.38117	0.43003	0.56927	0.44315	0.48008	0.56734	0.82385	0.39004	0.41123	0.46169	0.55549
	PCC	0.697	0.657	0.574	0.783	0.754	0.682	0.793	0.772	0.721	0.587	0.787	0.770	0.747	0.696
	SNR	2.56	1.99	1.15	4.23	3.69	2.47	3.70	3.35	2.59	0.89	4.18	3.97	3.43	2.62
SRGDiff	NMSE	0.54407	0.65731	0.85679	0.53913	0.62843	0.73244	0.92211	0.87139	0.84781	0.96161	0.70425	0.74605	0.83013	0.86974
	PCC	0.682	0.589	0.377	0.672	0.602	0.520	0.436	0.527	0.550	0.319	0.618	0.583	0.484	0.395
	SNR	2.71	1.86	0.68	2.69	2.03	1.36	0.37	0.61	0.73	0.17	1.57	1.30	0.82	0.61
TopoDiff	NMSE	0.82349	0.97261	0.97981	0.98478	0.97258	1.00843	0.74468	0.84035	0.93046	0.96968	0.61604	0.72578	0.86622	0.96303
	PCC	0.436	0.198	0.172	0.183	0.178	0.036	0.568	0.464	0.292	0.189	0.648	0.553	0.360	0.169
	SNR	0.86	0.12	0.09	0.07	0.12	-0.04	1.29	0.76	0.31	0.13	2.12	1.40	0.63	0.16
ScalpINR	NMSE	0.80067	0.82369	0.85612	0.77293	0.79083	0.82679	0.33489	0.35643	0.40737	0.49787	0.32138	0.33777	0.35981	0.41301
	PCC	0.408	0.381	0.343	0.439	0.421	0.386	0.807	0.794	0.761	0.701	0.796	0.786	0.768	0.730
	SNR	0.98	0.86	0.69	1.13	1.03	0.84	5.06	4.75	4.10	3.15	5.11	4.85	4.57	3.92
ZUNA	NMSE	0.63700	0.67800	0.72300	0.61100	0.66100	0.70700	0.51400	0.51900	0.53700	0.62200	0.79600	0.78700	0.76600	0.76400
	PCC	0.819	0.794	0.734	0.858	0.840	0.788	0.864	0.843	0.802	0.722	0.799	0.807	0.783	0.732
	SNR	1.98	1.70	1.42	2.14	1.80	1.51	2.95	2.92	2.81	2.13	0.99	1.04	1.16	1.17
PanoMontage	NMSE	**0.27537**	**0.33428**	0.43732	**0.16768**	**0.22387**	0.31682	**0.21074**	**0.24516**	**0.30019**	**0.39061**	**0.17808**	**0.20513**	**0.25047**	**0.33969**
	PCC	**0.845**	**0.806**	0.739	**0.905**	**0.871**	**0.816**	**0.892**	**0.874**	**0.841**	0.785	**0.899**	**0.882**	**0.853**	**0.798**
	SNR	**6.07**	**5.10**	3.83	7.84	**6.58**	5.05	**7.28**	**6.52**	**5.49**	**4.24**	**7.59**	**6.97**	**6.11**	**4.75**

D.2 Statistical Uncertainty

We estimate uncertainty from paired NMSE differences in Table 13; negative values favor PanoMontage. We bootstrap held-out subjects 10,000 times, retaining paired batches and computing their pooled mean difference; for Localize-MI, we instead resample recordings from its single test subject. Intervals use the 2.5th and 97.5th percentiles and reflect evaluation-cluster variation, not training seeds or population uncertainty. Eight lie below zero, five include zero, and SEED $8\times$ lies above zero. Comparators and paired samples match Table 1.

Table 13: Paired NMSE differences (PanoMontage minus comparator) with 95% cluster-bootstrap intervals. Localize-MI uses Spline at $2\times$ and ESTformer otherwise; its intervals resample recordings within one subject.

Dataset	Comparator	$2\times$	$4\times$	$8\times$	$16\times$	Units
SEED	ESTformer	-0.01532 [-0.01796, -0.01167]	-0.00794 [-0.00934, -0.00571]	$+0.00479$ [$+0.00197$, $+0.00700$]	–	3 subjects
SEED-IV	ESTformer	-0.00016 [-0.00377, $+0.00346$]	-0.00178 [-0.00458, $+0.00101$]	$+0.00084$ [-0.00546, $+0.00714$]	–	2 subjects
CCEP-COReg	ESTformer	-0.00346 [-0.00765, -0.00067]	-0.00431 [-0.01063, $+0.00048$]	-0.01074 [-0.01870, -0.00430]	-0.00828 [-0.01595, $+0.00021$]	5 subjects
Localize-MI	Spline/ESTformer	-0.10252 [-0.12248, -0.09028]	-0.09075 [-0.10133, -0.08191]	-0.07412 [-0.08163, -0.06703]	-0.06754 [-0.07310, -0.06252]	9 recordings

Differences by held-out subject. The main table pools paired batches rather than equally weighting subjects. All three SEED test subjects favor PanoMontage at $2\times$ and $4\times$, but ESTformer at $8\times$; the two SEED-IV subjects have opposite signs at every density. These are selected-run differences, not training-seed replicates.

D.3 Generalization to Unseen Observation Densities

PanoMontage, ScalpINR, and Spline use the same CCEP-COReg windows and masks across 100 paired eight-window batches per factor. Table 14 gives the metrics behind Figure 2. Factors $3\times$, $5\times$, and $10\times$ are unseen; the others are trained. The frozen-constructor control appears in Appendix E.1.

Table 14: Matched-window density generalization on CCEP-COReg. Entries are NMSE/PCC/SNR; † denotes unseen observation densities.

Method	$2\times$	$3\times^{\dagger}$	$4\times$	$5\times^{\dagger}$	$8\times$	$10\times^{\dagger}$	$16\times$
PanoMontage	0.2078/0.893/7.25	0.2227/0.885/6.88	0.2388/0.875/6.56	0.2554/0.866/6.21	0.2952/0.843/5.52	0.3171/0.828/5.18	0.3850/0.786/4.28
ScalpINR	0.3452/0.798/4.89	0.3536/0.793/4.75	0.3634/0.786/4.64	0.3787/0.778/4.43	0.4133/0.755/4.01	0.4359/0.739/3.76	0.5007/0.696/3.11
Spline	0.2222/0.887/7.02	0.2545/0.873/6.37	0.2903/0.858/5.81	0.3353/0.840/5.11	0.4975/0.790/3.40	0.6013/0.764/2.60	0.9776/0.691/0.42

D.4 Reconstruction at Withheld Electrode Locations

We select 37 locations (about 20% of the CCEP-COReg training electrode-name union) by geometry-only farthest-point sampling, shared across methods and seeds. Their waveforms are excluded from downstream inputs, supervision, and validation. At factor r, inputs sample $\lfloor N_{\mathrm{native}}/r \rfloor$ non-withheld electrodes. All 934 test windows per factor are scored at withheld locations with recorded references; metrics average within recordings and subjects, then equally across five test subjects. Table 15 reports three-seed NMSE and channel-wise PCC under per-channel robust normalization. Withheld-channel statistics are not model inputs, so this tests standardized waveforms, not unknown voltage scales. Coordinates may have appeared in TUEG pretraining.

Table 15: Complete withheld-location results on CCEP-COReg. Learned methods report means across three training seeds. NMSE is lower-is-better; channel-wise PCC is higher-is-better.

Method	Metric	$2\times$	$4\times$	$8\times$	$16\times$
Spline	NMSE	0.2189	0.2960	0.5415	1.0356
Spline	PCC	0.8963	0.8651	0.7972	0.6973
ScalpINR	NMSE	0.3447	0.3684	0.4176	0.5116
ScalpINR	PCC	0.8049	0.7902	0.7596	0.6947
PanoMontage	NMSE	0.2072	0.2363	0.2970	0.3951
PanoMontage	PCC	0.8990	0.8823	0.8476	0.7890

D.5 Dense Spatial Querying and Qualitative Examples

Figure 3 shows one CCEP-COReg 1-s window: 16 observed electrodes, 115 measured targets, and 30 displayed waveforms (Q01–Q30) selected by geometry from 4,096 approximately equal-area queries. PanoMontage, ScalpINR, and Spline are queried directly; ground-truth, ESTformer, and SRGDiff maps interpolate electrodes with head-boundary extrapolation. The SRGDiff snapshot is uncalibrated. Spatial MAE scores measured targets only.

From 400 fixed candidates, this favorable window met PCC ≥ 0.80, NMSE < 0.40, and lower NMSE than Spline, ESTformer, and ScalpINR. Both displayed channels met PCC ≥ 0.90, NMSE < 0.20, and over 20% lower NMSE with higher PCC than each baseline. Pairs were ranked by their smaller PCC advantage; SRGDiff was checked on finalists. The snapshot maximizes relative spatial MAE improvement among times with at least median reference variation, excluding 30 boundary samples on each side.

Figure 6 adds measured-target examples selected across the candidate-window error distribution.

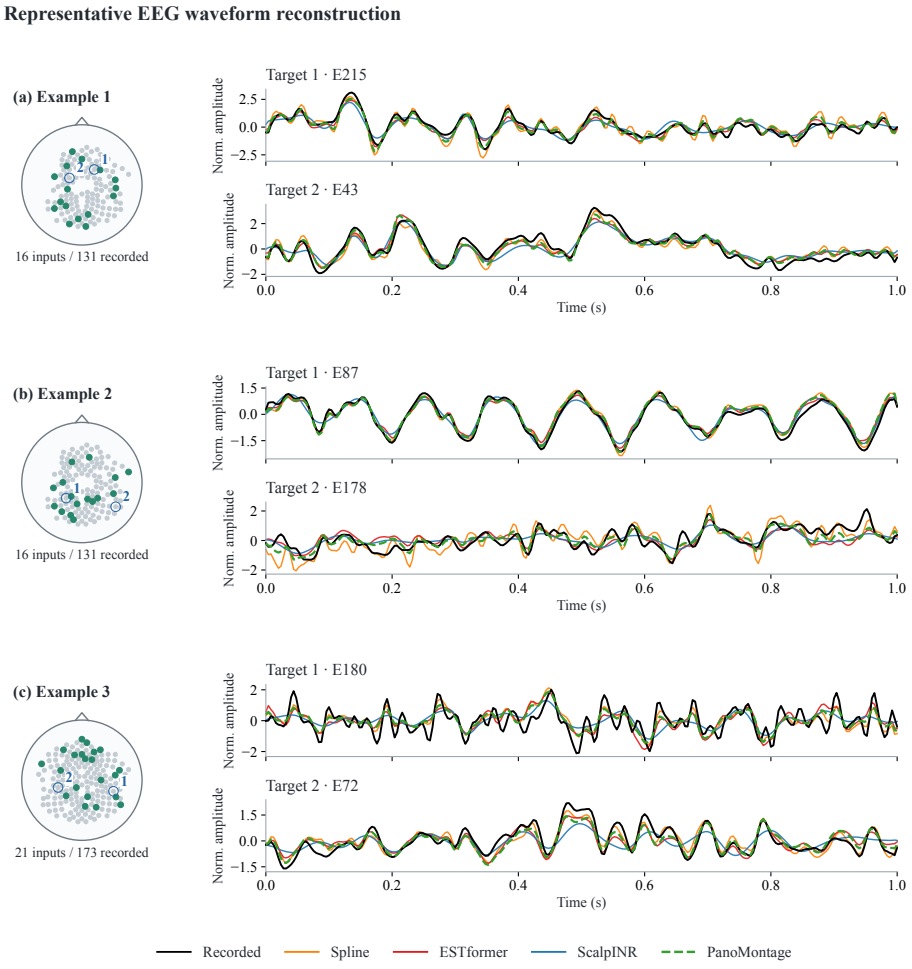

Figure 6: Representative EEG waveform reconstruction on CCEP-COReg at $8\times$. Examples 1–3 are windows nearest the 25th, 50th, and 75th percentiles of PanoMontage's hidden-channel window NMSE among 400 fixed test candidates, selected independently of any method advantage. For each window, the two displayed hidden electrodes are selected at the 25th and 75th percentile ranks of distance to the observed set, independent of waveform error. Each trace shows the complete one-second window (251 samples at 250 Hz) in robust-normalized amplitude. The scalp inset marks observed inputs in green and the two displayed targets with numbered blue rings. All methods use the same window and observation mask.

E Ablations and Further Analyses

E.1 Component Ablations and Pretraining

The three-seed CCEP-COReg mechanism study in Figure 4 is reported in full in Table 17.

Independent waveform and context controls. On SEED-IV, waveform-only decoding omits H_L from query source tokens but retains the learned intermediate waveform and both splines. The complementary control retains H_L and substitutes the first-stage spline $\overline{X}_L$ for $\widehat{X}_L$ before second-stage interpolation and waveform encoding. Its transmitted correction and regularizer become zero while context remains trainable. At identical weights, this substitution leaves H_L unchanged and preserves a nonzero context-branch gradient.

All variants share TUEG initialization, seed 20260719, and a 10k initial fine-tune continued to 50k updates. Validation occurs every 1k updates; selection uses minimum NMSE, with 10k-update patience and a 0.2% relative improvement threshold. Selected steps are 49k for Full and waveform-only, and 46k for spline-waveform plus context. Each factor has 100 paired batches of 16 windows, separate from the 400-window main test. All variants use BF16 inference; Full and Spline match batchwise across the paired evaluations.

Both paths improve all three metrics (Table 16). Learned waveform correction lowers NMSE by 3.0%, 14.3%, and 15.8% at $2\times$, $4\times$, and $8\times$ relative to spline-waveform plus context; context lowers it by 11.9%, 9.5%, and 5.4% relative to waveform-only. This single-seed test isolates each path's marginal contribution, not their interaction or seed variability.

Table 16: Independent field-path controls on SEED-IV. One matched training seed; 1,600 paired test windows per factor. Lower NMSE and higher PCC/SNR are better. Spline-waveform + context retains H_L but removes the intermediate waveform correction.

Variant	Metric	$2\times$	$4\times$	$8\times$
Full	NMSE	0.18930	0.24323	0.33843
	PCC	0.8896	0.8576	0.8027
	SNR (dB)	7.29	6.18	4.74
Learned waveform only	NMSE	0.21495	0.26880	0.35757
	PCC	0.8767	0.8455	0.7927
	SNR (dB)	6.73	5.74	4.50
Spline-waveform + context	NMSE	0.19507	0.28366	0.40187
	PCC	0.8858	0.8338	0.7599
	SNR (dB)	7.15	5.50	3.98

Initialization and mechanism controls. Frozen fixes the entire constructor and trains only the query decoder. Spline-only bypasses that constructor with $\widehat{X}_L = \overline{X}_L$ and $H_L = 0$. No spline zeros the analytical floors at both stages but retains the learned paths. Scratch PanoMontage initializes the complete two-stage model randomly; Scratch-NoSpline also zeros both floors. Direct controls use randomly initialized single-path models without latent support. CCEP-COReg comparisons use seeds 20260716, 20260720, and 20260721.

Table 17: Complete three-seed mechanism results on CCEP-COReg. Each entry is NMSE/PCC/SNR, with NMSE standard deviation over three training seeds. Values rounded to 0.000 have a nonzero standard deviation below 0.0005. All variants use the same evaluation protocol.

Variant	$2\times$	$4\times$	$8\times$	$16\times$
PanoMontage	0.217±0.001/0.891/7.07	0.243±0.000/0.875/6.54	0.308±0.001/0.837/5.40	0.404±0.003/0.776/4.12
Frozen Constructor	0.226±0.001/0.886/6.84	0.269±0.001/0.862/6.00	0.343±0.000/0.818/4.86	0.442±0.000/0.753/3.69
Spline Constructor	0.223±0.003/0.889/6.92	0.257±0.002/0.869/6.27	0.338±0.001/0.821/5.01	0.488±0.009/0.744/3.35
PanoMontage-NoSpline	0.347±0.115/0.818/5.06	0.376±0.105/0.797/4.59	0.447±0.088/0.748/3.71	0.570±0.068/0.649/2.54
PanoMontage (no pretraining)	0.223±0.002/0.888/6.89	0.249±0.001/0.871/6.38	0.314±0.001/0.833/5.31	0.416±0.001/0.770/4.00
Direct Query Decoder (no pretraining)	0.212±0.000/0.893/7.25	0.252±0.000/0.869/6.42	0.343±0.001/0.815/4.96	0.499±0.002/0.720/3.21

Removing spline guidance increases error and seed variation. Jointly adapting the learned constructor outperforms spline-only and frozen variants at every factor; pretraining improves the two-stage model throughout. Direct scratch leads at $2\times$, but two-stage scratch leads as inputs thin.

E.2 Dynamic-Montage Sampling Controls

Three paired seeds (20260716–18) compare recording-specific fixed montages, random positions with fixed counts, and random positions with dynamic counts. Fixed counts are $(N_O, N_L, N_Q) = (8, 56, 5)$; dynamic draws $N_O \in \{4, 8, 12\}$, $N_L \in \{48, 64\}$, and $N_Q \in \{4, 6\}$ have the same means. The dynamic policy varies all three supports jointly.

Within each seed, policies share initialization and TUEG window draws from recordings with at least 18 valid channels. Each trains for 20k updates (AdamW, 2×10^{-4}, weight decay 10^{-4}, batch 32). CCEP-COReg fine-tuning reinitializes the decoder and uses AdamW (5×10^{-5}), batch 16, a 50k-update ceiling, 1k-update validation, and 10k-update patience at a 0.2% improvement threshold. Minimum validation NMSE selects checkpoints; test metrics average windows, recordings, subjects, then seeds.

At $8\times$, $10\times$, and $16\times$, dynamic sampling lowers mean NMSE by 1.07%, 1.26%, and 1.01% against random positions with fixed counts (all three seeds), and by 0.24%, 0.42%, and 0.47% against fixed montages (two of three seeds). The sparse-input benefit is modest.

Table 18: Independent dynamic-montage controls on CCEP-COReg. Values are mean missing-channel NMSE over three paired training seeds; $3\times$, $5\times$, and $10\times$ are unseen during fine-tuning. Bold denotes the lowest mean.

Pretraining policy	2×	3×	4×	5×	8×	10×	16×
Fixed montage	**0.21520**	0.22963	0.24451	0.26046	0.30525	0.32802	0.39877
Random positions, fixed counts	0.21784	0.23165	0.24621	0.26214	0.30780	0.33084	0.40093
Dynamic positions and counts	0.21538	**0.22882**	**0.24407**	**0.25998**	**0.30451**	**0.32666**	**0.39689**

Table 19 gives paired seed differences. Models are selected independently on validation NMSE and share test masks; per-seed results weight the five test subjects equally. Gains over fixed montages vary by seed and do not establish a population-level effect.

Table 19: Dynamic-montage paired NMSE differences by training seed. Columns 16/17/18 denote seeds 20260716/17/18. Negative favors full dynamic sampling. RF denotes random positions with fixed counts.

	Dynamic minus fixed			Dynamic minus RF		
Factor	16	17	18	16	17	18
2×	+0.00026	+0.00191	-0.00162	-0.00305	-0.00208	-0.00224
3×	-0.00191	+0.00077	-0.00130	-0.00394	-0.00195	-0.00261
4×	-0.00201	+0.00114	-0.00046	-0.00324	-0.00142	-0.00174
5×	-0.00214	+0.00115	-0.00045	-0.00272	-0.00135	-0.00241
8×	-0.00322	+0.00156	-0.00057	-0.00359	-0.00161	-0.00469
10×	-0.00473	+0.00148	-0.00085	-0.00496	-0.00141	-0.00616
16×	-0.00567	+0.00057	-0.00051	-0.00407	-0.00060	-0.00745

E.3 Parameter-Matched Direct Reconstruction

The matched direct decoder widens tokens to 128 and residual hidden states to 100, then applies its shared-weight $O \to Q$ operator four times without P_L. Candidates were screened by parameters and profiler FLOPs, not test metrics. On a synthetic 251-sample batch (22 observations, 192 latent points, 256 targets), the selected model differs from PanoMontage by 0.019% in parameters and 1.67% in profiler FLOPs.

The measured workload in Table 20 uses a held-out CCEP-COReg $8\times$ window, batch 1, 192 latent points, and 256 targets on an A800 (20 warm-up, 100 timed iterations). PanoMontage and matched direct cost 0.1795 and 0.1941 GFLOPs, respectively; direct is 7.30% slower and uses 34.88% more peak memory. Both train from scratch for 10k steps under the same validation rule and three seeds. Thus parameter matching does not imply identical cost on this workload.

Matched direct has 5.6% lower NMSE at $2\times$; scratch PanoMontage has 0.7%, 8.1%, and 15.9% lower NMSE at $4\times$, $8\times$, and $16\times$. From $4\times$ to $16\times$, the two-stage advantage grows as observations become sparser under matched training budgets.

Table 20: Parameter-matched direct-path control with similar compute on subject-held-out CCEP-COReg. NMSE values are means over three independent training seeds. Efficiency is measured with batch size 1 on one NVIDIA A800; profiler FLOPs may omit unsupported operators. Bold NMSE denotes the best scratch-trained model at each factor. Dashes indicate measurements not repeated in this workload.

	Factor-8× efficiency				Missing-channel NMSE (↓)			
Model	Params (M)	GFLOPs	ms/example	MiB	2×	4×	8×	16×
PanoMontage (scratch)	0.439044	0.1795	17.053	86.95	0.2230	**0.2490**	**0.3140**	**0.4161**
Direct (original)	0.132	–	–	–	0.2118	0.2517	0.3434	0.4989
Direct (matched)	0.438962	0.1941	18.298	117.28	**0.2106**	0.2508	0.3415	0.4950

E.4 Sensitivity to Intermediate Support Size

A fixed CCEP-COReg checkpoint queries nested farthest-point subsets of its 192-point support ($N_L = 8, 16, 32, 64, 128, 192$), without retraining or changing query electrodes. Each factor reuses 800 windows and masks; metrics weight recordings, then subjects equally. Small supports degrade accuracy, with a plateau at 64–192 points. The minimum is at 192 for $2\times$–$8\times$ and 64 for $16\times$; larger is not always better.

Table 21: Fixed-checkpoint inference sensitivity to intermediate support size on CCEP-COReg. Values are subject-balanced missing-channel NMSE.

N_L	2×	4×	8×	16×
8	0.35016	0.36426	0.39889	0.46679
16	0.24403	0.27145	0.32371	0.41289
32	0.21563	0.24938	0.30622	0.39942
64	0.21154	0.24565	0.30188	**0.39449**
128	0.21024	0.24507	0.30221	0.39574
192	**0.20944**	**0.24434**	**0.30142**	0.39454

E.5 Computational Efficiency

We profile a preloaded SEED window (eight observations, 62 targets, 800 samples) on an idle A40 (Table 22). Each method runs in a fresh process: FP32, batch 1, three warm-up and 30 timed iterations; synchronized latency excludes loading. SRGDiff and STAD use 50 sampling steps, TopoDiff 20. PanoMontage has fewer parameters but higher latency than ESTformer; ScalpINR is also faster. Profiler FLOPs omit unsupported operators, so SRGDiff FLOPs are unreported. The A800 direct-path results in Table 20 use a separate workload.

Table 22: Current-implementation efficiency on SEED. Parameters are in millions, latency is mean ± standard deviation in ms, and peak allocated memory is in MiB. Sampling steps and workload are specified above.

Method	Params	GFLOPs	Latency	Memory
Spline	–	–	3.307±0.507	15.1
PanoMontage	0.439	0.132	19.404±0.673	132.4
ESTformer	6.793	1.546	5.097±0.409	47.9
ScalpINR	0.983	0.105	3.709±0.074	21.0
STAD	0.517	85.668	406.462±6.010	33.7
TopoDiff	22.308	36.842	91.104±1.505	112.3
SRGDiff	0.433	–	210.376±3.817	17.8