Riemannian Flow Matching for Brain Connectivity Matrices via Pullback Geometry

Antoine Collas*, **Ce Ju**, **Nicolas Salvy**, **Bertrand Thirion**
Inria, CEA, Université Paris-Saclay
Palaiseau, France
*antoine.collas@inria.fr

Abstract

Generating realistic brain connectivity matrices is key to analyzing population heterogeneity in brain organization, understanding disease, and augmenting data in challenging classification problems. Functional connectivity matrices lie in constrained spaces—such as the set of symmetric positive definite or correlation matrices—that can be modeled as Riemannian manifolds. However, using Riemannian tools typically requires redefining core operations (geodesics, norms, integration), making generative modeling computationally inefficient. In this work, we propose DIFFEOCFM, an approach that enables conditional flow matching (CFM) on matrix manifolds by exploiting pullback metrics induced by global diffeomorphisms on Euclidean spaces. We show that Riemannian CFM with such metrics is equivalent to applying standard CFM after data transformation. This equivalence allows efficient vector field learning, and fast sampling with standard ODE solvers. We instantiate DIFFEOCFM with two different settings: the matrix logarithm for covariance matrices and the normalized Cholesky decomposition for correlation matrices. We evaluate DIFFEOCFM on three large-scale fMRI datasets with more than 4600 scans from 2800 subjects (ADNI, ABIDE, OASIS-3) and two EEG motor imagery datasets with over 30000 trials from 26 subjects (BNCI2014-002 and BNCI2015-001). It enables fast training and achieves state-of-the-art performance, all while preserving manifold constraints.
Code: https://github.com/antoinecollas/DiffeoCFM

1 Introduction

Brain imaging connectivity and Riemannian geometry Modern neuroimaging analyses map brain functional signals toward *connectivity matrices*—covariance or correlation estimates between regions of interest or sensor channels [37]. These structured representations are used in many applications, such as motor imagery classification [4, 51, 41, 44], brain age prediction [16, 22, 54, 55], or disease diagnosis [15]. They are central to the analysis of signals from many neuroimaging modalities, such as functional magnetic resonance imaging (fMRI), electroencephalography (EEG), and magnetoencephalography (MEG). Brain connectivity matrices are *symmetric positive definite* (SPD) or lie in the open *elliptope* of full-rank correlation matrices, and thus belong to smooth matrix manifolds—$\mathbb{S}_d^{++}$ and its submanifold Corr_d—defined respectively by

$$\mathbb{S}_d^{++} = \left\{ \mathbf{\Sigma} \in \mathbb{R}^{d \times d} \,\middle|\, \mathbf{\Sigma}^\top = \mathbf{\Sigma}, \mathbf{\Sigma} \succ 0 \right\} \text{ and } \mathrm{Corr}_d = \left\{ \mathbf{\Sigma} \in \mathbb{S}_d^{++} \,\middle|\, \mathrm{diag}(\mathbf{\Sigma}) = \mathbf{1} \right\}. \tag{1}$$

Several Riemannian metrics have been proposed to analyse these data such as the affine-invariant metric [65, 7], the log-Euclidean metric [3], or the Euclidean-Cholesky metric [68]. They enable the definition of Riemannian operations such as geodesics, exponential maps, and parallel transport, which extend standard Euclidean operations to the manifold. Building on these manifolds, a wide

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

range of machine learning algorithms have been designed for classification [4], regression [16, 22, 55], or dimension reduction [27].

Deep generative models on manifolds Deep generative modeling has rapidly advanced with the success of generative adversarial networks (GANs)[28], variational autoencoders (VAEs)[39], autoregressive models such as PixelRNN [70], normalizing flows [21, 40], large-scale autoregressive models [9], and more recently, diffusion models [31, 66] and flow matching [49, 47, 2, 48]. These last two methods learn to synthesize data by estimating continuous-time stochastic (diffusion) or deterministic (flow) dynamics that interpolate between a source and a target distribution. Diffusion models do so by reversing a noise injection process, while flow matching aligns a learned time-dependent vector field to the velocity field of a straight-line path. Both paradigms offer good scalability and state-of-the-art results on diverse domains, from natural images [23], to speech [43], and protein structure generation [32]. More recent efforts have extended generative modeling to Riemannian manifolds. Riemannian score-based generative modeling [17] and Riemannian flow matching [12] provide general formulations for sampling on manifolds. The latter learns time-dependent vector fields on the tangent bundle that match the velocity of geodesic paths between source and target distributions. These approaches offer principled tools for generating data with geometric constraints, by treating the data domain as a manifold. These novel frameworks have already been applied to several applications such as materials discovery [56], robotics [20], or climate science [12].

Contributions In this work, we address a novel and challenging issue in neuroimaging: generating realistic brain connectivity data from actual human neuroimaging data. This challenge stems from the unique structure of brain connectivity data, which is represented by SPDs or correlation matrices that lie on non-Euclidean manifolds. Moreover, neuroimaging datasets typically have limited sample sizes, making realistic data generation particularly valuable. We propose a novel Riemannian flow matching method, DIFFEOCFM, based on pullback geometry, defined by a global diffeomorphism $\phi : \mathcal{M} \to E$, where E is Euclidean space. This method is an efficient framework that *guarantees manifold-constrained outputs by construction while avoiding computationally expensive operations specific to SPD or correlation manifolds.* We instantiate DIFFEOCFM with two diffeomorphisms tailored to different neuroimaging data: the matrix logarithm for SPD matrices and the normalized Cholesky decomposition for correlation matrices. Finally, we evaluate DIFFEOCFM on three large-scale fMRI datasets (ADNI, ABIDE, and OASIS-3; over 4600 scans from 2800 subjects) and two EEG motor imagery datasets (BNCI2014-002 and BNCI2015-001; 30000 trials from 26 subjects), demonstrating that DIFFEOCFM is capable of generating realistic, neurophysiologically meaningful samples, as validated by multiple statistical metrics.

Notations $\mathcal{M}$ is a smooth manifold with tangent space $T_x\mathcal{M}$ and Riemannian norm $\|\cdot\|_x$. We write γ for geodesics, and $\dot{\gamma}(t) \triangleq \frac{d}{dt}\gamma(t)$ for curve's speed. Let $\phi : \mathcal{M} \to E$ be a global diffeomorphism to Euclidean space E, with differential $\mathrm{D}\,\phi(x) : T_x\mathcal{M} \mapsto E$ and inverse $(\mathrm{D}\,\phi(x))^{-1}$. The pushforward of a distribution p on $\mathcal{M}$ is $\phi_{\#}p$, defined via $\int f\, d(\phi_{\#}p) = \int (f \circ \phi)\, dp$ for any f continuous on E. We denote $x \mid y \sim p(\cdot \mid y)$ for conditional sampling with label $y \in \mathcal{Y}$. Let $\mathbb{S}_d$ be the space of symmetric $d \times d$ matrices, $\mathbb{S}_d^{++}$ the SPD cone, and Corr_d the set of correlation matrices. Let LT_d^1 be the set of lower-triangular matrices with unit diagonal. We define $\mathrm{vec}_{\mathrm{sl}} : \mathbb{S}_d \mapsto \mathbb{R}^{d(d-1)/2}$ as the operator extracting strictly lower-triangular entries, and $\mathrm{vec}_{\mathrm{lt}}$ for the full lower-triangular part (diagonal included), with $\sqrt{2}$ scaling off-diagonal terms.

2 Background

Pullback Manifolds with Euclidean Spaces Let $\mathcal{M}$ be a smooth manifold, E a Euclidean space, and $\phi : \mathcal{M} \to E$ a global diffeomorphism (a smooth bijection with a smooth inverse). The diffeomorphism ϕ induces a Riemannian metric on $\mathcal{M}$ by pulling back the Euclidean metric g_E on E

$$(\phi^* g_E)_x(\xi, \eta) \triangleq g_E\left(\mathrm{D}\,\phi(x)[\xi], \mathrm{D}\,\phi(x)[\eta]\right), \quad \xi, \eta \in T_x\mathcal{M}\,. \tag{2}$$

This metric induces a Riemannian norm on the tangent space $T_x\mathcal{M}$ at each point $x \in \mathcal{M}$: $\|\xi\|_x = \sqrt{(\phi^* g_E)_x(\xi, \xi)}$. The pair $(\mathcal{M}, \phi^* g_E)$ is then called a *pullback manifold*, and $\phi^* g_E$ is the *pullback metric* of g_E by ϕ. The pullback manifold is geodesically complete and admits globally unique

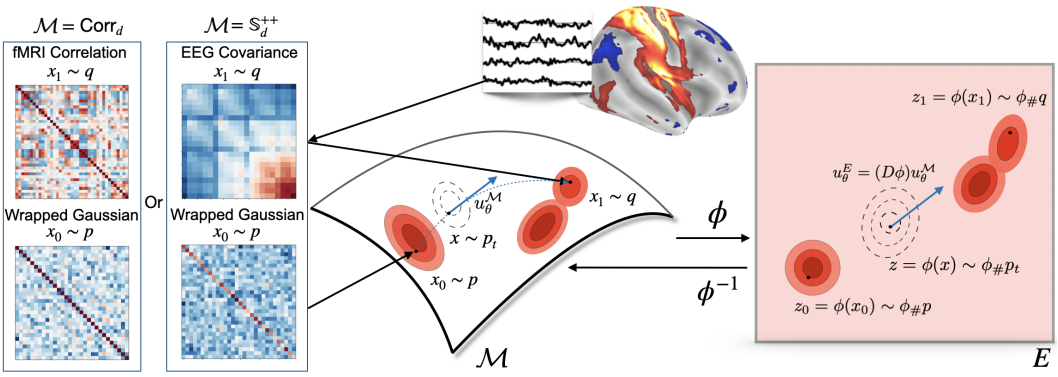


Figure 1: **Overview of DIFFEOCFM.** DIFFEOCFM is a principled framework for *deep generative modeling* on matrix manifolds. It reformulates Riemannian conditional flow matching (CFM) on a pullback manifold $(\mathcal{M}, \phi^* g_E)$ as *conventional CFM in Euclidean space* E, via a global diffeomorphism $\phi \colon \mathcal{M} \to E$. The reformulation preserves geometry in two ways: *(i)* the learned Euclidean vector field u_θ^E satisfies (7), ensuring that training u_θ^E is equivalent to training $u_\theta^{\mathcal{M}}$; *(ii)* the flow trajectories obey $\phi(x(t)) = z(t)$, so that integrating in E and pulling back via ϕ^{-1} yields the same samples as integrating directly on $\mathcal{M}$. This allows both training and sampling to be carried out efficiently in E, while remaining equivalent to operating on $\mathcal{M}$. On the left, fMRI correlation and EEG spatial covariance matrices lie on $\mathcal{M} = \mathrm{Corr}_d$ and $\mathcal{M} = \mathbb{S}_d^{++}$, respectively. These matrices are mapped to E through ϕ, a time-dependent vector field u_θ^E is trained in E, and integration is performed in E before mapping back via ϕ^{-1} to yield connectivity manifold constrained matrices.

geodesics [68, Chap. 7]. Moreover, many Riemannian operations reduce to simple computations in E. Given two points $x_0, x_1 \in \mathcal{M}$, the geodesic $\gamma : [0,1] \to \mathcal{M}$ connecting x_0 to x_1 and its associated riemannian distance are given by

$$\gamma(t) = \phi^{-1}\left((1-t)\phi(x_0) + t\phi(x_1)\right) \text{ and } d_{\mathcal{M}}(x_0, x_1) = \|\phi(x_0) - \phi(x_1)\|_E \tag{3}$$

i.e., the pullbacks of the Euclidean straight line and distance in E joining $\phi(x_0)$ and $\phi(x_1)$. The Fréchet mean [29] of a set of points $\{x^{(n)}\}_{n=1}^N \subset \mathcal{M}$ with respect to the Riemannian distance is

$$\bar{x} \triangleq \operatorname*{arg\,min}_{x \in \mathcal{M}} \sum_{n=1}^{N} d_{\mathcal{M}}(x, x^{(n)})^2 = \phi^{-1}\left(\frac{1}{N}\sum_{n=1}^{N} \phi(x^{(n)})\right). \tag{4}$$

Riemannian CFM CFM [49, 47, 2, 48] was recently extended to Riemannian manifolds [12], providing a principled framework for learning time-dependent vector fields that transport samples between probability distributions defined on such spaces. Given a manifold $\mathcal{M}$, a vector field $u_\theta^{\mathcal{M}} : [0,1] \times \mathcal{M} \times \mathcal{Y} \to T\mathcal{M}$ is trained to match the velocity of geodesics connecting samples from source and target distributions. The Riemannian CFM loss is defined as

$$\mathcal{L}(\theta) \triangleq \mathbb{E}_{t,\, y,\, x_0|y,\, x_1|y} \left\| u_\theta^{\mathcal{M}}(t, \gamma(t), y) - \dot{\gamma}(t) \right\|_{\gamma(t)}^2, \tag{5}$$

where $y \in \mathcal{Y}$ is a condition variable (such as a disease status), and $x_0|y \sim p(\cdot|y)$, $x_1|y \sim q(\cdot|y)$ and $t \sim \mathcal{U}([0,1])$. Hence, this loss is computationally intensive, as it requires evaluating geodesics $\gamma(t)$ between x_0 and x_1, their derivatives $\dot{\gamma}(t)$, and Riemannian norms $\|\cdot\|_{\gamma(t)}$. Once trained, new samples on $\mathcal{M}$ are generated by solving the Riemannian ODE

$$\dot{x}(t) = u_\theta^{\mathcal{M}}(t, x(t), y), \quad x(0) = x_0 \sim p(\cdot \mid y) \tag{6}$$

and returning $x(1)$ as a sample from the learned approximation of $q(\cdot \mid y)$.

3 DIFFEOCFM: Conditional Flow Matching on Pullback Manifolds

Pullback manifolds provide a natural setting for using Riemannian CFM [12] in practical generative modeling tasks. Indeed, when $\mathcal{M}$ is equipped with a pullback metric $\phi^* g_E$ induced by a global

diffeomorphism $\phi : \mathcal{M} \to E$, both training and sampling can be performed entirely in the Euclidean space E. We prove that this equivalence is *exact*: the learned vector field in E corresponds to one on $\mathcal{M}$, and the ODE solutions in E map to those on $\mathcal{M}$ via ϕ. This result motivates DIFFEOCFM, a conditional flow matching framework that performs all computations in E, avoiding costly geometric operations—such as computing geodesics, Riemannian norms, or manifold integration—while guaranteeing manifold-constrained outputs. An overview of the method is shown in Figure 1.

3.1 Training and sampling with a diffeomorphism

Training Rather than learning a vector field $u_\theta^{\mathcal{M}}$ directly on the manifold $\mathcal{M}$, DIFFEOCFM trains its Euclidean counterpart u_θ^E via the pullback:

$$u_\theta^E(t, z, y) \triangleq \mathrm{D}\,\phi(\phi^{-1}(z))\left(u_\theta^{\mathcal{M}}(t, \phi^{-1}(z), y)\right). \tag{7}$$

Indeed, in this case, the loss function (5) can be expressed in terms of the Euclidean vector field u_θ^E as shown in the following proposition.

Proposition 1 (Riemannian CFM loss function on pullback manifolds)**.** *The Riemannian CFM loss* (5) *can be re-expressed in terms of the Euclidean vector field* u_θ^E (7) *as*

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\, y,\, z_0|y,\, z_1|y} \left\| u_\theta^E\left(t, (1-t)z_0 + tz_1, y\right) - (z_1 - z_0) \right\|_E^2 ,$$

where $z_0|y \sim \phi_\# p(\cdot|y)$ *and* $z_1|y \sim \phi_\# q(\cdot|y)$.

It should be noted that this new loss function is much simpler to compute than the original Riemannian CFM one, as it does not require computing geodesics, their derivatives, or Riemannian norms.

Sampling DIFFEOCFM generates new samples by solving the ODE

$$\dot{z}(t) = u_\theta^E\left(t, z(t), y\right), \quad z(0) = z_0 \sim \phi_\# p(\cdot \mid y). \tag{8}$$

Despite this simple form, the procedure is fully Riemannian: the generated trajectory corresponds exactly to a manifold-valued solution under ϕ^{-1}. Indeed, the following proposition states the equivalence of the solutions of the ODEs (6) and (8).

Proposition 2 (Equivalence of ODE solutions)**.** *The solution* $x(t)$ *to* (6) *satisfies*

$$x(t) = \phi^{-1}(z(t)) \quad \textit{for all } t \in [0, 1],$$

where $z(t)$ *is the solution of the ODE* (8) *with initial condition* $z_0 = \phi(x_0)$.

The previous result establishes that the Riemannian and ODEs define equivalent flows through the diffeomorphism ϕ. In practice, these ODEs are solved numerically using explicit Runge–Kutta integrators. The next proposition shows that the equivalence also holds at the discrete level: applying the same Runge–Kutta scheme in E or on $\mathcal{M}$ yields iterates related by ϕ.

Proposition 3 (Equivalence of Runge–Kutta iterates)**.** *Let* $\{x_\ell\}$ *be the iterates produced on* $\mathcal{M}$ *by an explicit Riemannian Runge–Kutta scheme applied to the ODE* (6)*. Then, the iterates are*

$$x_\ell = \phi^{-1}(z_\ell) \quad \textit{for all } \ell \in \mathbb{N},$$

where $\{z_\ell\}$ *are the iterates obtained by applying the same scheme (same coefficients and step size) to the ODE* (8) *with initial condition* $z_0 = \phi(x_0)$.

Overall, the training and sampling algorithms for DIFFEOCFM are summarized in Algorithm 1 and 2, respectively.

Algorithm 1: DIFFEOCFM: Training

Input: step size h; samplers $\pi_{\mathcal{Y}}$, $\phi_\# p$, $\phi_\# q$
Output: Trained parameters $\theta^\star$
Initialize θ;
while *not converged* **do**
  Sample $y \sim \pi_{\mathcal{Y}}$, $t \sim \mathcal{U}([0,1])$;
  Sample $z_0 \sim \phi_\# p(\cdot \mid y)$, $z_1 \sim \phi_\# q(\cdot \mid y)$;
  $\mathcal{L} \leftarrow \|u_\theta^E(t, (1-t)z_0 + tz_1, y) - (z_1 - z_0)\|_E^2$;
  $\theta \leftarrow$ optimizer-step($\mathcal{L}$);
end

Algorithm 2: DIFFEOCFM: Sampling

Input: label y; steps L; step size h; trained $\theta^\star$
Output: Generated sample x
Sample $z_0 \sim \phi_\# p(\cdot \mid y)$;
for $\ell = 0$ **to** $L - 1$ **do**
  $z_{\ell+1} \leftarrow$ Runge-Kutta-step($u_{\theta^\star}^E, z_\ell, y, h$);
end
$x \leftarrow \phi^{-1}(z_L)$;

3.2 Diffeomorphic Embeddings for Generative Modeling of Brain Connectivity Matrices

To generate brain connectivity matrices, we map $\mathbb{S}_d^{++}$ and Corr_d to Euclidean spaces via global diffeomorphisms: the matrix logarithm for SPD matrices and the normalized Cholesky map for correlation matrices. These maps define pullback metrics on the tangent spaces $T_{\boldsymbol{\Sigma}}\mathbb{S}_d^{++} = \mathbb{S}_d$ and $T_{\boldsymbol{\Sigma}}\mathrm{Corr}_d = \{\boldsymbol{\xi} \in \mathbb{S}_d \mid \mathrm{diag}(\boldsymbol{\xi}) = 0\}$, of respective dimensions $d(d+1)/2$ and $d(d-1)/2$. This allows DIFFEOCFM to perform efficient, geometry-aware generation for both matrix types. Note that we selected these two diffeomorphisms for their simplicity and ease of implementation. However, other choices are possible. For correlation matrices, alternative parameterizations are discussed in [68, Chapter 7]. For SPD matrices, one can use the Cholesky factor with a logarithm applied to the diagonal, leading to the log-Cholesky metric [46], which also defines a global diffeomorphism.

Covariance matrices: Log–Euclidean metric On $\mathbb{S}_d^{++}$, we define the global diffeomorphism $\phi_{\mathbb{S}_d^{++}} : \mathbb{S}_d^{++} \mapsto \mathbb{R}^{d(d+1)/2}$ by composing the matrix logarithm with the vectorization map $\mathrm{vec}_{\mathrm{lt}}$:

$$\phi_{\mathbb{S}_d^{++}}(\boldsymbol{\Sigma}) = \mathrm{vec}_{\mathrm{lt}}(\log(\boldsymbol{\Sigma})) \quad \text{and} \quad \phi^{-1}_{\mathbb{S}_d^{++}}(\boldsymbol{\eta}) = \exp(\mathrm{vec}_{\mathrm{lt}}^{-1}(\boldsymbol{\eta})), \tag{9}$$

where $\log$ and $\exp$ denote the matrix logarithm and exponential, respectively. This mapping induces the *Log–Euclidean metric* on $\mathbb{S}_d^{++}$ by pulling back the standard Euclidean inner product from $\mathbb{R}^{d(d+1)/2}$: $g_{\boldsymbol{\Sigma},\mathbb{S}_d^{++}}(\boldsymbol{\xi},\boldsymbol{\eta}) = \mathrm{tr}(\mathrm{D}\log(\boldsymbol{\Sigma})[\boldsymbol{\xi}]\,\mathrm{D}\log(\boldsymbol{\Sigma})[\boldsymbol{\eta}])$, as introduced in Arsigny et al. [3].

Correlation matrices: Euclidean–Cholesky metric On Corr_d, the matrix logarithm is no longer a diffeomorphism. More generally, defining a log-based diffeomorphism is nontrivial—for example, the Riemannian logarithm associated with the affine-invariant metric does not admit a closed-form expression in this setting (see Appendix F). Instead, we use the normalized Cholesky map, a global diffeomorphism onto LT_d^1, the space of lower-triangular matrices with unit diagonal:

$$\mathrm{nchol}(\boldsymbol{\Sigma}) = \mathrm{diag}(\mathrm{chol}(\boldsymbol{\Sigma}))^{-1}\,\mathrm{chol}(\boldsymbol{\Sigma}), \tag{10}$$

where $\mathrm{chol}(\boldsymbol{\Sigma})$ is the unique Cholesky factor with positive diagonal. Its inverse is

$$\mathrm{nchol}^{-1}(\mathbf{L}) = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{L}^\top\mathbf{D}^{-1/2}, \quad \text{with}\, \mathbf{D} = \mathrm{diag}(\mathbf{L}\mathbf{L}^\top)\,. \tag{11}$$

Then, we define the diffeomorphism $\phi_{\mathrm{Corr}_d} : \mathrm{Corr}_d \mapsto \mathbb{R}^{d(d-1)/2}$ and its inverse as

$$\phi_{\mathrm{Corr}_d}(\boldsymbol{\Sigma}) = \mathrm{vec}_{\mathrm{sl}}(\mathrm{nchol}(\boldsymbol{\Sigma})) \quad \text{and} \quad \phi^{-1}_{\mathrm{Corr}_d}(\boldsymbol{\eta}) = \mathrm{nchol}^{-1}(\mathrm{vec}_{\mathrm{sl}}^{-1}(\boldsymbol{\eta}))\,. \tag{12}$$

This map induces the *Euclidean–Cholesky metric*, a Riemannian metric obtained by pulling back the Euclidean metric from the vector space $\mathbb{R}^{d(d-1)/2}$ [68]: $g_{\boldsymbol{\Sigma},\mathrm{Corr}_d}(\boldsymbol{\xi},\boldsymbol{\eta}) = \mathrm{D}\,\phi(\boldsymbol{\Sigma})[\boldsymbol{\xi}]^\top\,\mathrm{D}\,\phi(\boldsymbol{\Sigma})[\boldsymbol{\eta}]$.

Label, source, and target distributions To train DIFFEOCFM on labeled brain connectivity data, we define class-conditional source and target distributions in the Euclidean space E induced by the diffeomorphism ϕ. Given a dataset $\{(x^{(n)}, y^{(n)})\}_{n=1}^N$ of manifold-valued matrices, we map each sample to E via $z^{(n)} = \phi(x^{(n)})$. For each class y, we fit a Gaussian distribution to the embedded samples $\{z^{(n)} : y^{(n)} = y\}$ to define the class-conditional source distribution $\phi_\# p(\cdot \mid y)$. The target distribution $\phi_\# q(\cdot \mid y)$ is defined as the empirical distribution over the same class-y samples.

3.3 Related work

Denoising Diffusion Probabilistic Models (DDPMs) [31] and CFM [47, 2] have emerged as robust, state-of-the-art generative models in Euclidean spaces. Several extensions have been proposed to handle data that lie on Riemannian manifolds. These include Riemannian Score-Based Generative Modeling [17], SPD-DDPM [45], and Riemannian CFM [12]. These methods tailor the loss function and ODE/SDE solvers to the geometry of a specific manifold. However, this geometric fidelity comes at a cost: manifold-specific operations such as Riemannian gradients, exponential/logarithm maps, or parallel transport must be implemented to compute the loss function and the integration on the manifold. For instance, SPD-DDPM requires a specialized neural architecture, an SPDNet, which is computationally expensive and significantly slower to train than Euclidean counterparts; see Figure 4 in Appendix for a comparison. References [35] and [38] explore more general settings by learning bridge matches on arbitrary manifolds or data-driven Riemannian metrics, whereas our

approach focuses on Riemannian geometries defined via known pullback diffeomorphisms. In [26], the authors leverage reparameterisation with normalising flows to learn probability densities on Lie groups (non-Euclidean spaces), and thus can be seen as an early precursor to our approach.

In contrast, CorrGAN [52] offers a pragmatic approach to generating correlation matrices by training and sampling entirely in Euclidean space using a GAN. Geometric constraints, such as positive definiteness and unit diagonal, are enforced via post-processing. While the method is simple and fast, the post-processing step can change the generated data in unwanted ways and reduce their quality.

The proposed method combines the simplicity of Euclidean training with the rigor of Riemannian geometry by using a diffeomorphism ϕ to embed structured matrices, apply standard CFM, and map samples back. To the best of our knowledge, it is the only method that enables generation of both SPD and correlation matrices within a unified framework. A detailed comparison of baseline methods—highlighting their assumptions, strengths, and limitations—is provided in Appendix Section H.

4 Empirical benchmarks

These benchmarks were designed to evaluate whether the generated data *(i.)* match the test distribution and *(ii.)* enable classifiers trained on them to generalize to real data. This evaluation uses two human brain imaging modalities: *three fMRI datasets* and *two EEG datasets*.

4.1 Metrics

The metrics used are method-agnostic; that is, they are computed solely from the generated samples. They fall into two categories: (i) *quality metrics*, which assess how well the generated data approximate the real data distribution; and (ii) *classification accuracy score (CAS) metrics*, which evaluate the usefulness of generated data for training classifiers.

Quality Metrics We assess how closely the generated samples align with the real data distribution using the α-precision and β-recall metrics introduced by [1]. We compute these metrics using the `EvaGeM` library[1]. In contrast to [1], which uses a Deep neural network, `EvaGeM` employs a One-Class SVM, providing a more stable and hyperparameter-robust estimator across datasets. These metrics quantify the fidelity (how realistic the generated samples are) and the diversity (how well they span the true data distribution). We also report the harmonic mean of the two, denoted α, β-F1.

Classification accuracy score metrics We follow the CAS protocol [64], training a classifier on generated samples and evaluating it on real test data. Specifically, we assess classification utility using a logistic regression with `liblinear` solver, balances class weights and a 5-fold cross-validation to select the inverse regularization strength C from the grid $\{10^{-4}, 10^{-3}, \ldots, 10^{4}\}$. High scores indicate that the generated data preserves task-relevant information. On fMRI datasets, the task is disease classification (control vs. patient), while for EEG, it is a two-class motor imagery problem in a brain–computer interface setting. We report ROC-AUC and F1 scores.

4.2 Datasets

The experiments include both fMRI and EEG datasets; additional details are provided in Appendix G.

fMRI datasets We use three publicly available resting-state fMRI datasets. The ABIDE dataset [53] consists of 900 subjects (one scan each), including both neurotypical and autistic individuals with a mean age of 17 years, collected across 19 international sites. The ADNI dataset [73] comprises 1,900 scans from 900 older adults (mean age 74), covering normal ageing, mild cognitive impairment, and Alzheimer's disease. The OASIS-3 dataset [42] includes 1,000 subjects and 1,800 longitudinal sessions collected over 10 years, targeting healthy ageing and neurodegenerative conditions, with a mean participant age of 71 years. We z-score the time series and then compute *correlation matrices* (Corr_d) using the OAS estimator [15]. We report mean and standard deviations computed across 10 random train-test splits with subject-level grouping, ensuring that scans from the same subject do not appear in both training and test sets.

[1] `https://github.com/nicolassalvy/EvaGeM`

EEG datasets We use two publicly available EEG motor imagery datasets from the BCI competition. The BNCI2014-002 dataset [67] includes 13 subjects performing right-hand and feet imagery, recorded with 15 channels over 1 session, with 80 trials per class. The BNCI2015-001 dataset [25] comprises 12 subjects, 13 channels over 2 or 3 sessions, and 100 trials per class for the same motor imagery tasks. To process the raw data, we compute *covariance matrices* ($\mathbb{S}_d^{++}$) using the OAS estimator [13], following standard practices [51]. We report performance on a leave-one-session-out protocol on BNCI2014-002 and on cross-session experiments on BNCI2015-001. The reported standard deviations are computed over 5 inner splits and averaged over sessions.

4.3 Baselines

We present the baselines used for comparison with DIFFEOCFM, including an oracle baseline (REAL DATA) that treats test data as if it were generated samples.

REAL DATA This oracle baseline treats the test set as if it were generated data when computing metrics. It provides an upper bound on the achievable performance, both for quality and CAS metrics, showing the best any generative model could hope to match.

DIFFEOGAUSS Given a diffeomorphic embedding $\phi : \mathcal{M} \mapsto E$, we model each class-conditional distribution $q(\cdot \mid y)$ as the push-forward of a Gaussian $\mathcal{N}(\mu_y, \Sigma_y)$ fitted to the embedded training data $z^{(n)} = \phi(x^{(n)})$. Samples are drawn in E and mapped back to $\mathcal{M}$ via ϕ^{-1}, yielding a wrapped Gaussian distribution on the manifold [62, 18].

TRIANGDDPM and TRIANGCFM These baselines apply generative models to the lower-triangular part of SPD or correlation matrices, a common heuristic in manifold modeling [52]. For fMRI, they use the strictly lower-triangular entries ($\phi : \mathrm{Corr}_d \mapsto \mathbb{R}^{d(d-1)/2}$); for EEG, all lower-triangular entries ($\phi : \mathbb{S}_d^{++} \mapsto \mathbb{R}^{d(d+1)/2}$). TRIANGDDPM uses a DDPM [2], while TRIANGCFM trains a vector field using the standard CFM loss. Since ϕ is not a diffeomorphism, generated matrices that do not lie on the manifold are projected back onto it. In particular, generated matrices are not necessarily positive definite, so we apply a projection to ensure all eigenvalues are at least $\epsilon > 0$ with $\mathbf{\Sigma}_{\text{proj}} = (1-\alpha)\mathbf{\Sigma} + \alpha\mathbf{I}$; see Appendix M for more details. These methods trade geometric fidelity for simplicity, relying on post hoc projections to enforce constraints.

RIEMCFM This method applies Riemannian CFM [12] directly on the SPD manifold using the affine-invariant metric [65]. The target conditional vector field is computed analytically from the Riemannian logarithm and exponential maps. Unlike TRIANGDDPM/TRIANGCFM, RIEMCFM preserves the intrinsic geometry of $\mathbb{S}_d^{++}$ throughout training and sampling, yielding valid SPD matrices at every time step without post-hoc corrections. However, it requires computing geodesics, their derivatives and Riemannian norms under the affine-invariant metric, making it substantially more computationally expensive than all other presented methods. The reference implementation[2] focuses exclusively on the SPD manifold under the affine-invariant metric, and does not provide a corresponding construction for correlation matrices.

DIFFEOCFM (proposed) We apply the log–Euclidean map $\phi_{\mathbb{S}_d^{++}}$ (9) for EEG and the normalized Cholesky map ϕ_{Corr_d} (12) for fMRI in Algorithm 1 and 2. These diffeomorphisms allow Euclidean training with CFM while ensuring manifold-valid samples without post-processing.

Deep learning and training/sampling setups TRIANGDDPM, TRIANGCFM and DIFFEOCFM employ a two-layer MLP with 512 hidden units, trained using AdamW [50] with a learning rate of 10^{-3} and batch size 64. Training runs for 200 epochs on fMRI and 2000 epochs on EEG. RIEMCFM has a 6-layer MLP with 512 hidden units trained with AdamW (learning rate of 10^{-4}), as recommended in [12]. These four methods use the dopri5 method from the `torchdiffeq` [11] library for time integration.
All experiments were run within 10 hours on a single Nvidia A40 GPU with a 32-cores cpu.

[2] `https://github.com/facebookresearch/riemannian-fm`

5 Results

We report both quantitative results and a neurophysiological plausibility study. Quantitative comparisons are summarized in Table 1, with additional analysis of projection effects in Table 2. Neurophysiological relevance is assessed in Figure 2, which shows class-conditional fMRI connectomes via Fréchet means and topographic maps of EEG Common Spatial Patterns (CSP) filters. To complement Table 1, we provide a visual summary of our findings in Figure 3 (Appendix I). Together, these results show that DIFFEOCFM produces realistic, class-conditional samples that preserve key features of brain connectivity. For a more detailed discussion of the baselines, please refer to Appendix H.

Table 1: **Performance of generative models on 3 fMRI and 2 EEG datasets, evaluated with quality and Classification Accuracy Score (CAS) metrics.** Quality metrics (α-precision, β-recall, and α, β-F1) assess alignment with the real distribution. CAS metrics (ROC-AUC and F1) evaluate downstream performance: a classifier is trained on generated data to predict *disease status* (fMRI) or *motor imagery class* (EEG), and tested on held-out real samples. *Real Data* rows use real samples to compare training and test distributions, serving as empirical upper bounds. The proposed method is denoted DIFFEOCFM . mean $\pm$ std are reported. **Bold** values denote the best method and any methods that are not significantly worse than it (one-sided paired Wilcoxon signed-rank test, $\alpha = 0.05$).

		Quality Metrics			CAS Metrics		Time (s.)	
Dataset	Method	α-Precision ↑	β-Recall ↑	α,β-F1 ↑	ROC-AUC ↑	F1 ↑	Training ↓	Sampling ↓
ABIDE	*Real Data*	0.80 ± 0.08	0.79 ± 0.08	0.79 ± 0.03	0.67 ± 0.06	0.59 ± 0.07	N/A	N/A
	DIFFEOGAUSS	0.56 ± 0.06	0.29 ± 0.06	0.38 ± 0.06	**0.66 ± 0.04**	**0.53 ± 0.06**	**0.07 ± 0.03**	**0.06 ± 0.00**
	TRIANGDDPM	0.04 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.53 ± 0.06	0.47 ± 0.12	33.80 ± 1.19	0.37 ± 0.05
	TRIANGCFM	0.04 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.52 ± 0.05	0.40 ± 0.18	48.78 ± 1.27	0.79 ± 0.78
	DIFFEOCFM	**0.77 ± 0.09**	**0.48 ± 0.07**	**0.59 ± 0.08**	**0.64 ± 0.06**	**0.58 ± 0.07**	32.78 ± 0.96	0.40 ± 0.04
ADNI	*Real Data*	0.91 ± 0.03	0.85 ± 0.06	0.88 ± 0.03	0.62 ± 0.05	0.62 ± 0.05	N/A	N/A
	DIFFEOGAUSS	0.02 ± 0.01	0.51 ± 0.08	0.04 ± 0.02	**0.60 ± 0.05**	0.29 ± 0.13	**0.14 ± 0.01**	**0.18 ± 0.01**
	TRIANGDDPM	0.02 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.53 ± 0.05	0.18 ± 0.11	90.03 ± 2.01	0.62 ± 0.08
	TRIANGCFM	0.02 ± 0.00	0.00 ± 0.01	0.01 ± 0.01	0.56 ± 0.04	0.34 ± 0.13	87.37 ± 2.13	0.63 ± 0.09
	DIFFEOCFM	**0.62 ± 0.11**	**0.77 ± 0.02**	**0.68 ± 0.06**	**0.63 ± 0.04**	**0.47 ± 0.10**	88.01 ± 2.90	0.69 ± 0.11
OASIS-3	*Real Data*	0.88 ± 0.04	0.87 ± 0.03	0.88 ± 0.02	0.73 ± 0.05	0.63 ± 0.06	N/A	N/A
	DIFFEOGAUSS	0.51 ± 0.04	0.30 ± 0.04	0.38 ± 0.04	**0.70 ± 0.05**	0.41 ± 0.07	**0.10 ± 0.01**	**0.13 ± 0.00**
	TRIANGDDPM	0.03 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.54 ± 0.06	0.41 ± 0.14	70.39 ± 1.99	0.50 ± 0.06
	TRIANGCFM	0.06 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.52 ± 0.06	0.41 ± 0.14	67.92 ± 2.31	0.52 ± 0.07
	DIFFEOCFM	**0.60 ± 0.05**	**0.35 ± 0.04**	**0.44 ± 0.04**	**0.67 ± 0.06**	**0.53 ± 0.07**	67.83 ± 1.83	0.57 ± 0.05
BNCI 2014-002	*Real Data*	0.70 ± 0.05	0.60 ± 0.05	0.64 ± 0.03	0.83 ± 0.01	0.75 ± 0.02	N/A	N/A
	DIFFEOGAUSS	0.46 ± 0.04	**0.77 ± 0.02**	0.57 ± 0.03	0.80 ± 0.02	**0.73 ± 0.02**	**0.06 ± 0.01**	**0.08 ± 0.01**
	TRIANGDDPM	0.43 ± 0.04	0.10 ± 0.02	0.16 ± 0.03	0.52 ± 0.03	0.20 ± 0.15	257.88 ± 0.14	0.30 ± 0.05
	TRIANGCFM	0.48 ± 0.05	0.22 ± 0.03	0.30 ± 0.03	0.55 ± 0.03	0.24 ± 0.11	251.78 ± 0.85	0.35 ± 0.09
	RIEMCFM	**0.67 ± 0.07**	0.62 ± 0.06	**0.63 ± 0.03**	**0.81 ± 0.02**	0.72 ± 0.02	1983.58 ± 0.97	5.28 ± 0.58
	DIFFEOCFM	0.62 ± 0.04	0.63 ± 0.04	0.62 ± 0.02	**0.81 ± 0.02**	**0.74 ± 0.02**	253.04 ± 0.33	0.59 ± 0.08
BNCI 2015-001	*Real Data*	0.89 ± 0.01	0.89 ± 0.01	0.89 ± 0.00	0.73 ± 0.01	0.67 ± 0.01	N/A	N/A
	DIFFEOGAUSS	0.84 ± 0.01	**0.90 ± 0.01**	0.86 ± 0.01	**0.73 ± 0.01**	**0.68 ± 0.01**	**0.07 ± 0.01**	**0.16 ± 0.01**
	TRIANGDDPM	0.73 ± 0.03	0.55 ± 0.03	0.63 ± 0.02	0.60 ± 0.02	0.59 ± 0.13	319.94 ± 5.88	0.37 ± 0.07
	TRIANGCFM	0.79 ± 0.03	0.73 ± 0.03	0.76 ± 0.02	0.61 ± 0.02	0.59 ± 0.07	313.22 ± 2.16	0.38 ± 0.08
	RIEMCFM	**0.93 ± 0.04**	0.84 ± 0.05	0.88 ± 0.01	**0.73 ± 0.01**	0.66 ± 0.02	2753.93 ± 0.31	11.02 ± 0.59
	DIFFEOCFM	**0.92 ± 0.01**	0.86 ± 0.02	**0.89 ± 0.01**	**0.73 ± 0.01**	0.65 ± 0.01	319.83 ± 0.33	1.02 ± 0.08

5.1 Quantitative Study

Quality Metrics As shown in Table 1, DIFFEOCFM consistently matches or outperforms all baseline generative models across datasets in terms of α, β-F1, establishing itself as the most robust method for aligning with the true data distribution. It also achieves the highest α-precision and β-recall across the three fMRI datasets. On EEG datasets, DIFFEOGAUSS achieves higher

β-recall but at the cost of much lower α-precision, leading to weaker overall α, β-F1 scores. In contrast, TRIANGDDPM and TRIANGCFM perform poorly on all quality metrics after projection, due to a substantial degradation in sample quality. As detailed in Table 2, projecting onto the manifold—Corr_d for fMRI and $\mathbb{S}_d^{++}$ for EEG—introduces a significant performance drop. This is because TRIANGDDPM and TRIANGCFM generate matrices that visually resemble realistic connectivity patterns but contain negative eigenvalues, making them invalid. The projection step corrects these matrices but distorts their structure, leading to sharp decreases in α, β-F1—up to -0.74 on ADNI and -0.76 on OASIS-3—rendering TRIANGCFM impractical for use. This degradation is also visible in Figure 5, where post-projection alter the structure of fMRI connectomes (see Appendix M). Compared to RIEMCFM, DIFFEOCFM delivers similar performance on the two EEG datasets, while training $8\times$ faster and sampling $10\times$ faster. Finally, on EEG datasets, DIFFEOCFM nearly matches the performance of REAL DATA in terms of α, β-F1, suggesting high sample realism.

CAS Metrics For the CAS metrics, which evaluate downstream predictive performance using ROC-AUC and F1 scores, DIFFEOCFM consistently achieves strong results, often approaching the performance of REAL DATA. It obtains the highest ROC-AUC and F1 scores across all fMRI and EEG datasets. While DIFFEOGAUSS remains competitive in terms of ROC-AUC, DIFFEOCFM substantially outperforms it on F1 scores, with absolute gains of +0.05, +0.18, and +0.12 on ABIDE, ADNI, and OASIS-3, respectively. TRIANGDDPM and TRIANGCFM perform poorly across both CAS metrics on all datasets. This underperformance reflects the effect of the projection step, which alters generated matrices in ways detrimental to downstream classification. It is important to note that the classification pipeline requires inputs to be valid SPD matrices (i.e., elements of $\mathbb{S}_d^{++}$). As a result, CAS metrics cannot be computed for TRIANGDDPM and TRIANGCFM without projection. For this reason, their performance without projection is not reported in Table 2.

Table 2: **Impact of projection onto the manifolds $\mathbb{S}_d^{++}$ and Corr_d: performance difference δ = TRIANGCFM − TRIANGCFM without projections.** Negative Δ indicate degraded sample quality after projection onto the manifold. It enforces geometric constraints but severely reduces sample fidelity.

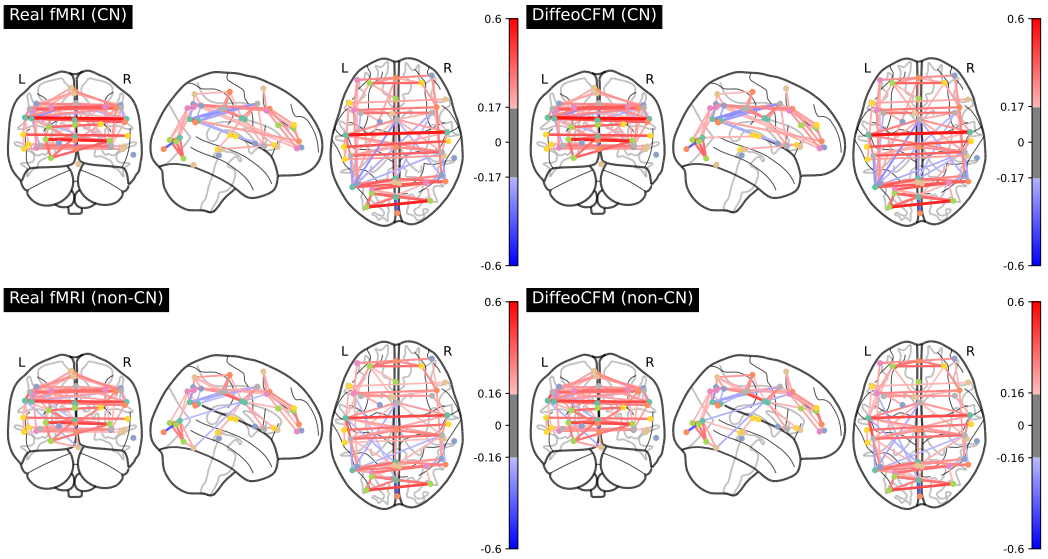
Dataset	α-precision	β-recall	α,β-F1
ABIDE	-0.34	-0.69	-0.50
ADNI	-0.63	-0.74	-0.69
OASIS-3	-0.52	-0.76	-0.64
BNCI2014-002	+0.13	-0.56	-0.19
BNCI2015-001	+0.00	-0.19	-0.09

Plotting of Generated Samples in Real Data Neighborhoods To qualitatively assess fidelity, we show Figure 10–15 (fMRI) and Figure 16–19 (EEG) in Appendix P the generated samples closest to real ones in Frobenius distance. fMRI results are grouped by control (CN) and patient (non-CN); EEG by motor imagery class. DIFFEOCFM reliably populates the neighborhood of real data, capturing class-conditional structure. We also show TRIANGCFM samples *before projection*, which appear realistic but are not SPD.

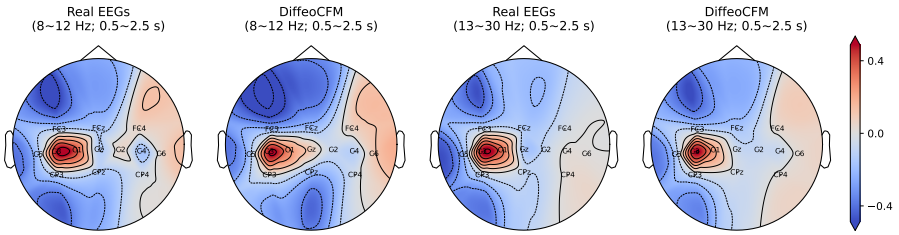
5.2 Neurophysiological Plausibility Study

fMRI Connectome Plotting Figure 2a shows group-level functional connectomes from the ADNI dataset, computed as the Fréchet mean (4) of correlation matrices conditioned on disease status. For comparison, Figure 9 in Appendix O presents the corresponding group-level connectomes derived from the ABIDE and OASIS3 datasets. The Fréchet mean is defined with respect to ϕ_{Corr_d} (12), the diffeomorphism used for the generation. For each class (CN and non-CN), we compare real connectomes (from held-out test subjects) to those generated by DIFFEOCFM. In both cases, non-CN subjects exhibit reduced connectivity across hemispheres and between frontal and posterior regions—patterns commonly associated with mild cognitive impairment and Alzheimer's disease [19].

EEG Topographic Map Figure 2b presents the group-level topographies of the first CSP filter across all subjects, derived from EEG generated by DIFFEOCFM, alongside those from real EEG recordings in the BNCI2015-001 dataset. Subject-level topographies of the first CSP filter from the same dataset are further detailed in Figure 7 (Subjects 1–6) and Figure 8 (Subjects 7–12) in Appendix N, providing a detailed comparison across individuals. We visualize CSP spatial filters in the α (8–12 Hz) and β (13–30 Hz) bands that distinguish imagined right-hand from feet movements. The filters trained on real and on generated data concentrate on the same contralateral sensorimotor regions, mirroring the close CAS scores between REAL DATA and DIFFEOCFM in Table 1. This confirms that the generative model preserves the physiologically relevant information.



(a) **Class-conditional fMRI functional connectome plotting using the Fréchet mean (ADNI).** Each panel displays class-conditional fMRI functional connectomes using the Fréchet mean (4) of correlation matrices computed with respect to the generation diffeomorphism ϕ_{Corr_d} (12). Left: real data from held-out test subjects; right: samples generated by DIFFEOCFM. The comparison illustrates both the fidelity of generated connectomes and the disease-specific connectivity structure preserved by the model.



(b) **Group-level topographic map using the first CSP's spatial filter derived from real EEGs (BNCI2015-001) and generated data by DIFFEOCFM** Each map shows the first CSP's spatial filter across 12 subjects in the α $(8-12\,\mathrm{Hz})$ and β $(13-30\,\mathrm{Hz})$ frequency bands during the first 2 seconds following stimulus onset. Filters from DIFFEOCFM at the group-level closely resemble real EEGs, preserving discriminative patterns of motor imagery classification.

Figure 2: Neurophysiological Plausibility Study of DIFFEOCFM.

6 Conclusions, Limitations, and Future Works

We introduced DIFFEOCFM, an efficient framework for generating brain connectivity matrices. By reformulating Riemannian flow matching through global diffeomorphisms, DIFFEOCFM enabled fast training and sampling while ensuring manifold-constrained outputs by construction. Applied to fMRI and EEG data, it outperformed existing baselines with neurophysiologically plausible samples.

Nonetheless, several limitations remain. First, DIFFEOCFM relies on a global diffeomorphism to Euclidean space, which exists for SPD and correlation matrices but not for all manifolds (e.g., Stiefel). Second, higher parcellation granularity makes the problem intrinsically hard: manifold dimension grows quadratically with region count, and the sample complexity grows exponentially [57]. Third, how alternative connectivity definitions (e.g., partial correlation or graphical-Lasso precision) affect generation quality remains an open question, since the choice of estimator directly defines the ground truth. Fourth, common evaluation metrics like α-precision and β-recall are geometry-agnostic and may miss neurophysiological structure.

Despite these challenges, generative modeling remains a promising direction for neuroimaging and BCI research. For instance, sharing trained generative models, rather than raw data, can facilitate multi-site collaboration with privacy guarantees [10].

Acknowledgment

This work was supported by grant ANR-22-PESN-0012 under the France 2030 program, managed by the Agence Nationale de la Recherche (ANR). CJ was supported by DATAIA Convergence Institute as part of the "Programme d'Investissement d'Avenir", (ANR-17-CONV-0003) operated by Inria. This work was performed using HPC resources from GENCI–IDRIS (Grant 2025-AD011016067).

Numerical computation was enabled by the scientific Python ecosystem: `Matplotlib` [33], `Scikit-learn` [61], `Numpy` [30], `Scipy` [72], `PyTorch` [59], `fMRIprep` [24], `Nilearn` [14], `joblib` [36], `PyRiemann` [5], `torchcfm` [69], `torchdiffeq` [11], `pandas` [58], and `moabb` [34].

References

[1] Ahmed Alaa, Boris Van Breugel, Evgeny S Saveliev, and Mihaela Van Der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International Conference on Machine Learning*, pages 290–306. PMLR, 2022.

[2] Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

[3] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 29(1):328–347, 2007. doi: 10.1137/050637996.

[4] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Multiclass brain–computer interface classification by riemannian geometry. *IEEE Transactions on Biomedical Engineering*, 59(4):920–928, 2011.

[5] Alexandre Barachant, Quentin Barthélemy, Jean-Rémi King, Alexandre Gramfort, Sylvain Chevallier, Pedro L. C. Rodrigues, Emanuele Olivetti, Vladislav Goncharenko, Gabriel Wagner vom Berg, Ghiles Reguig, Arthur Lebeurrier, Erik Bjäreholt, Maria Sayu Yamamoto, Pierre Clisson, Marie-Constance Corsi, Igor Carrara, Apolline Mellot, Bruna Junqueira Lopes, Brent Gaisford, Ammar Mian, Anton Andreev, Gregoire Cattan, and Arthur Lebeurrier. pyriemann, February 2025. URL https://doi.org/10.5281/zenodo.593816.

[6] Benjamin Blankertz, Ryota Tomioka, Steven Lemm, Motoaki Kawanabe, and Klaus-Robert Muller. Optimizing spatial filters for robust eeg single-trial analysis. *IEEE Signal processing magazine*, 25(1):41–56, 2007.

[7] Florent Bouchard, Arnaud Breloy, Antoine Collas, Alexandre Renaux, and Guillaume Ginolhac. *The Fisher–Rao Geometry of CES Distributions*, pages 37–77. Springer Nature Switzerland, Cham, 2024. ISBN 978-3-031-52116-4. doi: 10.1007/978-3-031-52116-4_2. URL https://doi.org/10.1007/978-3-031-52116-4_2.

[8] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. Jax: Autograd and xla. http://github.com/google/jax, 2018.

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[10] Richard J Chen, Judy J Wang, Drew FK Williamson, Tiffany Y Chen, Jana Lipkova, Ming Y Lu, Sharifa Sahai, and Faisal Mahmood. Algorithmic fairness in artificial intelligence for medicine and healthcare. *Nature biomedical engineering*, 7(6):719–742, 2023.

[11] Ricky T. Q. Chen. torchdiffeq, 2018. URL https://github.com/rtqichen/torchdiffeq.

[12] Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. In *The Twelfth International Conference on Learning Representations*, 2024.

[13] Yilun Chen, Ami Wiesel, Yonina C Eldar, and Alfred O Hero. Shrinkage algorithms for mmse covariance estimation. *IEEE transactions on signal processing*, 58(10):5016–5029, 2010.

[14] Nilearn contributors. nilearn. URL https://github.com/nilearn/nilearn.

[15] Kamalaker Dadi, Mehdi Rahim, Alexandre Abraham, Darya Chyzhyk, Michael Milham, Bertrand Thirion, Gaël Varoquaux, Alzheimer's Disease Neuroimaging Initiative, et al. Benchmarking functional connectome-based predictive models for resting-state fmri. *NeuroImage*, 192:115–134, 2019.

[16] Paul David and Weiqing Gu. A riemannian structure for correlation matrices. *Operators and Matrices*, 13(3):607–627, 2019.

[17] Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. *Advances in neural information processing systems*, 35:2406–2422, 2022.

[18] Thibault de Surrel, Fabien Lotte, Sylvain Chevallier, and Florian Yger. Wrapped Gaussian on the manifold of Symmetric Positive Definite Matrices. In *Forty-second International Conference on Machine Learning*, 2025.

[19] Emily L Dennis and Paul M Thompson. Functional brain connectivity using fmri in aging and alzheimer's disease. *Neuropsychology review*, 24:49–62, 2014.

[20] Haoran Ding, Noémie Jaquier, Jan Peters, and Leonel Rozo. Fast and robust visuomotor riemannian flow matching policy. *IEEE Transactions on robotics*, 2025.

[21] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

[22] Denis A Engemann, Apolline Mellot, Richard Höchenberger, Hubert Banville, David Sabbagh, Lukas Gemein, Tonio Ball, and Alexandre Gramfort. A reusable benchmark of brain-age prediction from m/eeg resting-state signals. *Neuroimage*, 262:119521, 2022.

[23] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[24] Oscar Esteban, Christopher J. Markiewicz, Ross W. Blair, Craig A. Moodie, A. Ilkay Isik, Asier Erramuzpe, James D. Kent, Mathias Goncalves, Elizabeth DuPre, Madeleine Snyder, Hiroyuki Oya, Satrajit S. Ghosh, Jessey Wright, Joke Durnez, Russell A. Poldrack, and Krzysztof J. Gorgolewski. fmriprep: a robust preprocessing pipeline for functional mri. *Nature Methods*, 16(1):111–116, 2019. doi: 10.1038/s41592-018-0235-4. URL https://doi.org/10.1038/s41592-018-0235-4.

[25] Josef Faller, Carmen Vidaurre, Teodoro Solis-Escalante, Christa Neuper, and Reinhold Scherer. Autocalibration and recurrent adaptation: Towards a plug and play online erd-bci. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 20(3):313–319, 2012.

[26] Luca Falorsi, Pim De Haan, Tim R Davidson, and Patrick Forré. Reparameterizing distributions on lie groups. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3244–3253. PMLR, 2019.

[27] P Thomas Fletcher and Sarang Joshi. Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. In *International Workshop on Mathematical Methods in Medical and Biomedical Image Analysis*, pages 87–98. Springer, 2004.

[28] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[29] Karsten Grove and Hermann Karcher. How to conjugate c 1-close group actions. *Mathematische Zeitschrift*, 132(1):11–20, 1973.

[30] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T.E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

[31] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[32] Guillaume Huguet, James Vuckovic, Kilian FATRAS, Eric Thibodeau-Laufer, Pablo Lemos, Riashat Islam, Cheng-Hao Liu, Jarrid Rector-Brooks, Tara Akhound-Sadegh, Michael M Bronstein, et al. Sequence-augmented se (3)-flow matching for conditional protein generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[33] J.D. Hunter. Matplotlib: A 2D graphics environment. *Computing in science & engineering*, 9 (3):90–95, 2007.

[34] Vinay Jayaram and Alexandre Barachant. Moabb: trustworthy algorithm benchmarking for bcis. *Journal of Neural Engineering*, 15(6):066011, sep 2018. doi: 10.1088/1741-2552/aadea0. URL https://doi.org/10.1088/1741-2552/aadea0.

[35] Jaehyeong Jo and Sung Ju Hwang. Generative modeling on manifolds through mixture of riemannian diffusion processes. *arXiv preprint arXiv:2310.07216*, 2023.

[36] Joblib Development Team. Joblib: running python functions as pipeline jobs, 2020. URL https://joblib.readthedocs.io/.

[37] Ce Ju, Reinmar J Kobler, Antoine Collas, Motoaki Kawanabe, Cuntai Guan, and Bertrand Thirion. SPD Learning for Covariance-Based Neuroimaging Analysis: Perspectives, Methods, and Challenges. *arXiv preprint arXiv:2504.18882*, 2025.

[38] Kacper Kapusniak, Peter Potaptchik, Teodora Reu, Leo Zhang, Alexander Tong, Michael Bronstein, Joey Bose, and Francesco Di Giovanni. Metric flow matching for smooth interpolations on the data manifold. *Advances in Neural Information Processing Systems*, 37:135011–135042, 2024.

[39] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.

[40] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[41] Reinmar Kobler, Jun-ichiro Hirayama, Qibin Zhao, and Motoaki Kawanabe. Spd domain-specific batch normalization to crack interpretable unsupervised domain adaptation in eeg. *Advances in Neural Information Processing Systems*, 35:6219–6235, 2022.

[42] Pamela J. LaMontagne, Tammie LS. Benzinger, John C. Morris, Sarah Keefe, Russ Hornbeck, Chengjie Xiong, Elizabeth Grant, Jason Hassenstab, Krista Moulder, Andrei G. Vlassenko, Marcus E. Raichle, Carlos Cruchaga, and Daniel Marcus. Oasis-3: Longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and alzheimer disease. *medRxiv*, 2019. doi: 10.1101/2019.12.13.19014902. URL https://www.medrxiv.org/content/early/2019/12/15/2019.12.13.19014902.

[43] Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36:14005–14034, 2023.

[44] Shanglin Li, Motoaki Kawanabe, and Reinmar Kobler. Spdim: Source-free unsupervised conditional and label shift adaptation in eeg. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Representation Learning*, volume 2025, pages 101907–101931, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/fc6cd575b8340be6ee70beaed9ca4015-Paper-Conference.pdf.

[45] Yunchen Li, Zhou Yu, Gaoqi He, Yunhang Shen, Ke Li, Xing Sun, and Shaohui Lin. Spd-ddpm: Denoising diffusion probabilistic models in the symmetric positive definite space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13709–13717, 2024.

[46] Zhenhua Lin. Riemannian geometry of symmetric positive definite matrices via cholesky decomposition. *SIAM Journal on Matrix Analysis and Applications*, 40(4):1353–1370, 2019.

[47] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

[48] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

[49] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

[50] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

[51] Fabien Lotte, Laurent Bougrain, Andrzej Cichocki, Maureen Clerc, Marco Congedo, Alain Rakotomamonjy, and Florian Yger. A review of classification algorithms for eeg-based brain–computer interfaces: a 10 year update. *Journal of neural engineering*, 15(3):031005, 2018.

[52] Gautier Marti. Corrgan: Sampling realistic financial correlation matrices using generative adversarial networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8459–8463. IEEE, 2020.

[53] Adriana Di Martino and Stewart Mostofsky. Autism brain imaging data exchange (abide), 2023. URL https://dx.doi.org/10.21227/y3v9-b041.

[54] Apolline Mellot, Antoine Collas, Pedro L. C. Rodrigues, Denis Engemann, and Alexandre Gramfort. Harmonizing and aligning M/EEG datasets with covariance-based techniques to enhance predictive regression modeling. *Imaging Neuroscience*, 1:imag–1–00040, December 2023. ISSN 2837-6056. URL https://doi.org/10.1162/imag_a_00040.

[55] Apolline Mellot, Antoine Collas, Sylvain Chevallier, Alexandre Gramfort, and Denis A Engemann. Geodesic optimization for predictive shift adaptation on eeg data. *Advances in Neural Information Processing Systems*, 37:32828–32855, 2024.

[56] Benjamin Kurt Miller, Ricky TQ Chen, Anuroop Sriram, and Brandon M Wood. Flowmm: Generating materials with riemannian flow matching. In *Forty-first International Conference on Machine Learning*, 2024.

[57] Kazusato Oko, Shunta Akiyama, and Taiji Suzuki. Diffusion models are minimax optimal distribution estimators. In *International Conference on Machine Learning*, pages 26517–26582. PMLR, 2023.

[58] The pandas development team. pandas-dev/pandas: Pandas, February 2020. URL https://doi.org/10.5281/zenodo.3509134.

[59] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035. Curran Associates, Inc., 2019.

[60] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

[61] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[62] Xavier Pennec. Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25:127–154, 2006.

[63] Gert Pfurtscheller and FH Lopes Da Silva. Event-related eeg/meg synchronization and desynchronization: basic principles. *Clinical neurophysiology*, 110(11):1842–1857, 1999.

[64] Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. *Advances in neural information processing systems*, 32, 2019.

[65] Lene Theil Skovgaard. A riemannian geometry of the multivariate normal model. *Scandinavian journal of statistics*, pages 211–223, 1984.

[66] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

[67] David Steyrl, Reinhold Scherer, Josef Faller, and Gernot R Müller-Putz. Random forests in non-invasive sensorimotor rhythm brain-computer interfaces: a practical and convenient non-linear classifier. *Biomedical Engineering/Biomedizinische Technik*, 61(1):77–86, 2016.

[68] Yann Thanwerdas. *Riemannian and stratified geometries on covariance and correlation matrices*. PhD thesis, Université Côte d'Azur, 2022.

[69] Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

[70] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

[71] Gaël Varoquaux, Alexandre Gramfort, Fabian Pedregosa, Vincent Michel, and Bertrand Thirion. Multi-subject dictionary learning to segment an atlas of brain spontaneous activity. In *Biennial International Conference on information processing in medical imaging*, pages 562–573. Springer, 2011.

[72] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, J.K. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, I. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17: 261–272, 2020.

[73] Michael W. Weiner, Dallas P. Veitch, Paul S. Aisen, Laurel A. Beckett, Nigel J. Cairns, Robert C. Green, Danielle Harvey, Clifford R. Jack Jr., William Jagust, John C. Morris, Ronald C. Petersen, Jennifer Salazar, Andrew J. Saykin, Leslie M. Shaw, Arthur W. Toga, John Q. Trojanowski, and Alzheimer's Disease Neuroimaging Initiative. The alzheimer's disease neuroimaging initiative 3: Continued innovation for clinical trial improvement. *Alzheimer's & Dementia*, 13(5):561–571, 2017. doi: https://doi.org/10.1016/j.jalz.2016.10.006. URL https://alz-journals.onlinelibrary.wiley.com/doi/abs/10.1016/j.jalz.2016.10.006.

A Pullback Manifolds with Euclidean Spaces

Section 2 briefly introduced pullback manifolds with Euclidean spaces. We here go more into details to then prove the different propositions of the paper. Let $\mathcal{M}$ be a smooth manifold, E a Euclidean space, and $\phi : \mathcal{M} \to E$ a global diffeomorphism (a smooth bijection with a smooth inverse). The diffeomorphism ϕ induces a Riemannian metric on $\mathcal{M}$ by pulling back the Euclidean metric g_E on E

$$(\phi^* g_E)_x(\xi, \eta) \triangleq g_E\left(\mathrm{D}\,\phi(x)[\xi], \mathrm{D}\,\phi(x)[\eta]\right), \quad \xi, \eta \in T_x\mathcal{M}\,. \tag{13}$$

This metric induces a Riemannian norm on the tangent space $T_x\mathcal{M}$ at each point $x \in \mathcal{M}$: $\|\xi\|_x = \sqrt{(\phi^* g_E)_x(\xi, \xi)}$. The pair $(\mathcal{M}, \phi^* g_E)$ is then called a *pullback manifold*, and $\phi^* g_E$ is the *pullback metric* of g_E by ϕ. The pullback manifold is geodesically complete and admits globally unique geodesics [68, Chap. 7]. Moreover, many Riemannian operations—distances, geodesics, exponential and logarithmic maps, parallel transport, Fréchet means—reduce to simple computations in E. Given two points $x, y \in \mathcal{M}$, the geodesic $\gamma : [0, 1] \to \mathcal{M}$ connecting x to y and its associated Riemannian distance are given by

$$\gamma(t) = \phi^{-1}\left((1-t)\phi(x) + t\phi(y)\right) \quad \text{and} \quad d_{\mathcal{M}}(x, y) = \|\phi(x) - \phi(y)\|_E \tag{14}$$

i.e., the pullbacks of the Euclidean straight line and distance in E joining $\phi(x)$ and $\phi(y)$. This structure also defines expressions for the exponential and logarithmic maps at any point $x \in \mathcal{M}$. The exponential map $\exp_x : T_x\mathcal{M} \to \mathcal{M}$ and the logarithmic map $\log_x : \mathcal{M} \to T_x\mathcal{M}$ are

$$\exp_x(\xi) = \phi^{-1}\left(\phi(x) + \mathrm{D}\,\phi(x)[\xi]\right) \quad \text{and} \quad \log_x(y) = \left(\mathrm{D}\,\phi(x)\right)^{-1}\left(\phi(y) - \phi(x)\right)\,. \tag{15}$$

When $\mathrm{D}\,\phi(x)$ is not available in closed form, it can be computed via automatic differentiation using libraries such as JAX [8] or PyTorch [60]. Its inverse, $\left(\mathrm{D}\,\phi(x)\right)^{-1}$, can be obtained from the differential of ϕ^{-1} using the identity $\left(\mathrm{D}\,\phi(x)\right)^{-1} \circ \mathrm{D}\,\phi^{-1}(\phi(x)) = \mathrm{Id}$. The parallel transport of a tangent vector $\xi \in T_x\mathcal{M}$ along the geodesic γ is the pullback of the Euclidean parallel transport from $\phi(x)$ to $\phi(y)$, which is given by

$$\mathrm{PT}_{x \to y}(\xi) = \left(\mathrm{D}\,\phi(y)\right)^{-1}\left(\mathrm{D}\,\phi(x)(\xi)\right)\,. \tag{16}$$

The Fréchet mean [29] of a set of points $\{x^{(n)}\}_{n=1}^N \subset \mathcal{M}$ with respect to the Riemannian distance is

$$\bar{x} \triangleq \operatorname*{arg\,min}_{x \in \mathcal{M}} \sum_{n=1}^{N} d_{\mathcal{M}}(x, x^{(n)})^2 = \phi^{-1}\left(\frac{1}{N}\sum_{n=1}^{N} \phi(x^{(n)})\right)\,. \tag{17}$$

B Riemannian Conditional Flow Matching

This section provides a concise overview of Flow Matching, Conditional Flow Matching, and their extension to Riemannian manifolds, complementing Section 2. For clarity, we omit conditioning on variables y such as disease status, age, or sex; the derivations extend naturally to the conditional case.

B.1 Flow Matching (Intractable Objective)

Flow Matching (FM) [49, 47, 2, 48] aims to learn a time-dependent vector field $u_\theta(t, x)$ that transports samples from a simple source (prior) distribution $p(x)$ (e.g., a standard Gaussian $\mathcal{N}(0, I)$) at $t = 0$ to a target data distribution $q(x)$ at $t = 1$. This transformation is governed by an ordinary differential equation (ODE):

$$\dot{x}(t) = u(t, x(t)), \quad x(0) \sim p_0(x) \tag{18}$$

Training Flow Matching trains a neural network $u_\theta(t, x)$ to approximate a true time-dependent vector field $u(t, x)$ that transports a source distribution $p(x)$ to a target $q(x)$ along an evolving density path $(p_t)_{t \in [0,1]}$. The objective is:

$$\mathcal{L}_{\mathrm{FM}}(\theta) \triangleq \mathbb{E}_{t \sim \mathcal{U}([0,1]),\, x \sim p_t(x)} \|u_\theta(t, x) - u(t, x)\|_2^2\,. \tag{19}$$

This formulation is generally intractable, as both $p_t(x)$ and $u(t, x)$ are unknown. Indeed, the time evolution of $p_t(x)$ is governed by the continuity equation:

$$\frac{\partial p_t(x)}{\partial t} + \mathrm{div}\left(p_t(x)\, u(t, x)\right) = 0\,, \tag{20}$$

which expresses conservation of mass along the flow.

Sampling To generate a sample x_1 from a learned model $u_\theta(t, x)$, one draws an initial sample $x_0 \sim p_0(x)$ and then solves the learned ODE from $t = 0$ to $t = 1$ using a numerical solver such as a Euler or a Runge-Kutta scheme:

$$\dot{x}(t) = u_\theta(t, x(t)) \quad \Rightarrow \quad x_1 = x_0 + \int_0^1 u_\theta(t, x(t))dt\,. \tag{21}$$

B.2 Conditional Flow Matching (CFM)

Training Conditional Flow Matching (CFM) makes the training of flow models tractable by defining explicit conditional paths and vector fields. We still consider a source distribution p and a target data distribution q. A common choice for the path connecting x_0 to x_1 is a linear interpolation:

$$x_t(x_0, x_1) = (1 - t)x_0 + tx_1$$

The corresponding target conditional vector field is $u(t, x_t|x_0, x_1) = x_1 - x_0$. The CFM training loss for a neural network $u_\theta(t, x)$ is:

$$\mathcal{L}_{\text{CFM}}(\theta) \triangleq \mathbb{E}_{t\sim\mathcal{U}([0,1]),x_0\sim p(x_0),x_1\sim q(x_1)}\|u_\theta(t, (1 - t)x_0 + tx_1) - (x_1 - x_0)\|_2^2\,. \tag{22}$$

Here, $p(x_0)$ is typically a simple noise distribution (e.g., $\mathcal{N}(0, I)$) and $q(x_1)$ is the empirical data distribution. Minimizing this CFM loss has been shown to be equivalent to minimizing the original intractable FM loss under certain conditions [48]. Other conditional path definitions can also be employed.

Sampling Sampling is performed by drawing $x_0 \sim p(x_0)$ and integrating the learned vector field $u_\theta(t, x(t))$ from $t = 0$ to $t = 1$:

$$\dot{x}(t) = u_\theta(t, x(t))$$

The solution $x(1)$ is then a sample from the learned approximation of $q(x_1)$.

B.3 Riemannian Conditional Flow Matching (RCFM)

CFM was recently extended to Riemannian manifolds [12], providing a principled framework for learning time-dependent vector fields that transport samples between probability distributions defined on such spaces.

Training Given a manifold $\mathcal{M}$, a vector field $u_\theta^{\mathcal{M}} : [0, 1] \times \mathcal{M} \to T\mathcal{M}$ (where $T\mathcal{M}$ denotes the tangent bundle of $\mathcal{M}$) is trained to match the velocity of geodesics $\gamma(t)$ connecting samples from a source distribution $p(x_0)$ on $\mathcal{M}$ and a target distribution $q(x_1)$ on $\mathcal{M}$. The Riemannian CFM loss is defined as

$$\mathcal{L}(\theta) \triangleq \mathbb{E}_{t\sim\mathcal{U}([0,1]),\, x_0\sim p(x_0),\, x_1\sim q(x_1)} \left\|u_\theta^{\mathcal{M}}(t, \gamma(t)) - \dot{\gamma}(t)\right\|^2_{\gamma(t)}\,, \tag{23}$$

where $\gamma(t)$ is the geodesic connecting x_0 to x_1 such that $\gamma(0) = x_0$ and $\gamma(1) = x_1$, and $\dot{\gamma}(t)$ is its time derivative (velocity vector) which lies in $T_{\gamma(t)}\mathcal{M}$. It should be noted that this loss extends $\mathcal{L}_{\text{CFM}}$. Indeed, for $\mathcal{M} = \mathbb{R}^d$, then we get $\mathcal{L} = \mathcal{L}_{\text{CFM}}$.

Sampling Once trained, new samples on $\mathcal{M}$ are generated by solving the Riemannian ODE

$$\dot{x}(t) = u_\theta^{\mathcal{M}}(t, x(t)), \quad x(0) = x_0 \sim p(x_0) \tag{24}$$

and returning $x(1)$ as a sample from the learned approximation of $q(x_1)$. Numerical solution of this ODE typically involves manifold operations like the exponential map.

C Proof of Proposition 1: Riemannian CFM loss function on pullback manifolds

Proposition 1 (Riemannian CFM loss function on pullback manifolds)**.** *The Riemannian CFM loss* (5) *can be re-expressed in terms of the Euclidean vector field* u_θ^E (7) *as*

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\, y,\, z_0|y,\, z_1|y} \left\|u_\theta^E\left(t, (1 - t)z_0 + tz_1, y\right) - (z_1 - z_0)\right\|^2_E\,,$$

where $z_0|y \sim \phi_\# p(\cdot|y)$ *and* $z_1|y \sim \phi_\# q(\cdot|y)$.

Proof. Fix a label y, draw $x_0 \sim p(\cdot \mid y)$ and $x_1 \sim q(\cdot \mid y)$, and set $z_0 = \phi(x_0)$, $z_1 = \phi(x_1)$. For any $t \in [0,1]$ let $z_t \triangleq (1-t)z_0 + tz_1$ and $\gamma(t) = \phi^{-1}(z_t)$.

Geodesic velocity Since z_t is a straight line in E, the chain rule gives

$$\dot{\gamma}(t) = \mathrm{D}\,\phi^{-1}(z_t)(z_1 - z_0)\,.$$

Pull-back of the vector field By definition of u_θ^E in (7),

$$u_\theta^E(t, z_t, y) = \mathrm{D}\,\phi(\gamma(t))\,\left(u_\theta^{\mathcal{M}}(t, \gamma(t), y)\right).$$

Norm preservation Because the metric on $\mathcal{M}$ is the pull-back of g_E, we have $\|\xi\|_{\gamma(t)} = \|\,\mathrm{D}\,\phi(\gamma(t))\,(\xi)\|_E$ for every $\xi \in T_{\gamma(t)}\mathcal{M}$. Applying $\mathrm{D}\,\phi(\gamma(t))$ to the difference $u_\theta^{\mathcal{M}}(t, \gamma(t), y) - \dot{\gamma}(t)$ yields

$$\begin{aligned}\mathrm{D}\,\phi(\gamma(t))\left(u_\theta^{\mathcal{M}}(t, \gamma(t), y) - \dot{\gamma}(t)\right) &= \mathrm{D}\,\phi(\gamma(t))\left(u_\theta^{\mathcal{M}}(t, \gamma(t), y)\right) - \mathrm{D}\,\phi(\gamma(t))(\dot{\gamma}(t)) \\ &= u_\theta^E(t, z_t, y) - \mathrm{D}\,\phi(\phi^{-1}(z_t))\left(\mathrm{D}\,\phi^{-1}(z_t)(z_1 - z_0)\right)\,.\end{aligned}$$

Furthermore, since $(\phi \circ \phi^{-1})(z) = z$, we have $\mathrm{D}\,\phi(\phi^{-1}(z)) \circ \mathrm{D}\,\phi^{-1}(z) = \mathrm{Id}_E$. qHence, we get

$$\mathrm{D}\,\phi(\gamma(t))\left(u_\theta^{\mathcal{M}}(t, \gamma(t), y) - \dot{\gamma}(t)\right) = u_\theta^E(t, z_t, y) - (z_1 - z_0)\,.$$

Taking squared Euclidean norms on both sides gives

$$\left\|u_\theta^{\mathcal{M}}(t, \gamma(t), y) - \dot{\gamma}(t)\right\|_{\gamma(t)}^2 = \left\|u_\theta^E(t, z_t, y) - (z_1 - z_0)\right\|_E^2.$$

Expectation Finally, averaging over $t \sim \mathcal{U}[0,1]$, $y \sim \pi_{\mathcal{Y}}$, $x_0 \sim p(\cdot \mid y)$, and $x_1 \sim q(\cdot \mid y)$ gives the desired equality of loss functions, proving Proposition 1. □

D Proof of Proposition 2: Equivalence of ODE Solutions

Proposition 2 (Equivalence of ODE solutions)**.** *The solution $x(t)$ to (6) satisfies*

$$x(t) = \phi^{-1}(z(t)) \quad \textit{for all } t \in [0,1]\,,$$

where $z(t)$ is the solution of the ODE with u_θ^E (7) and initial condition $z_0 = \phi(x_0)$.

Proof. Let $y \in \mathcal{Y}$, $x_0 \in \mathcal{M}$ and $z_0 \triangleq \phi(x_0)$. Let the Euclidean trajectory $z : [0,1] \to E$ be a solution of

$$\dot{z}(t) = u_\theta^E\left(t, z(t), y\right), \qquad z(0) = z_0.$$

Then, we define the candidate solution

$$x(t) \triangleq \phi^{-1}\left(z(t)\right) \qquad (0 \le t \le 1).$$

By the chain rule,

$$\dot{x}(t) = \mathrm{D}\,\phi^{-1}\left(z(t)\right)\left(\dot{z}(t)\right) = \mathrm{D}\,\phi^{-1}\left(z(t)\right)\left(u_\theta^E\left(t, z(t), y\right)\right).$$

For every $z = \phi\left(x\right)$, we have (7)

$$u_\theta^E\left(t, z, y\right) = \mathrm{D}\,\phi\left(x\right)\left(u_\theta^{\mathcal{M}}\left(t, x, y\right)\right).$$

Applying this with $x = x(t)$ and $z = z(t)$,

$$\dot{x}(t) = \mathrm{D}\,\phi^{-1}\left(\phi\left(x(t)\right)\right)\left(\mathrm{D}\,\phi\left(x(t)\right)\left(u_\theta^{\mathcal{M}}\left(t, x(t), y\right)\right)\right).$$

Using the identity, $\mathrm{D}\,\phi^{-1}\left(\phi(x)\right) \circ \mathrm{D}\,\phi(x) = \mathrm{Id}_{T_x\mathcal{M}}$, we get

$$\dot{x}(t) = u_\theta^{\mathcal{M}}\left(t, x(t), y\right)\,.$$

Finally, we check the initial condition,

$$x(0) = \phi^{-1}\left(z_0\right) = \phi^{-1}\left(\phi\left(x_0\right)\right) = x_0.$$

Consequently $x(t) = \phi^{-1}\left(z(t)\right)$ is a solution on $\mathcal{M}$ of (6). □

E Proof of Proposition 3: Equivalence of Runge–Kutta iterates

Proposition 3 (Equivalence of Runge–Kutta iterates)**.** *Let $\{x_\ell\}$ be the iterates produced on $\mathcal{M}$ by an explicit Riemannian Runge–Kutta scheme applied to the ODE (6). Then, the iterates are*

$$x_\ell = \phi^{-1}(z_\ell) \quad \text{for all } \ell \in \mathbb{N},$$

where $\{z_\ell\}$ are the iterates obtained by applying the same scheme (same coefficients and step size) to the ODE with vector field u_θ^E (7) and initial condition $z_0 = \phi(x_0)$.

Proof. We prove this for the fourth-order Runge-Kutta method (RK4). Similar proofs can be done for the Euler (RK1) and midpoint (RK2) schemes. The ODEs are:

$$\dot{z}(t) = u_\theta^E(t, z(t), y) \quad \text{in Euclidean space } E$$
$$\dot{x}(t) = u_\theta^{\mathcal{M}}(t, x(t), y) \quad \text{on manifold } \mathcal{M}$$

The key relationships are:

- Vector field relationship: $u_\theta^E(t, z, y) = \mathrm{D}\,\phi(\phi^{-1}(z))(u_\theta^{\mathcal{M}}(t, \phi^{-1}(z), y))$.
- Exponential map on $\mathcal{M}$: $\exp_x(\xi) = \phi^{-1}(\phi(x) + \mathrm{D}\,\phi(x)(\xi))$ for $x \in \mathcal{M}, \xi \in T_x\mathcal{M}$.
- Parallel transport on $\mathcal{M}$ from $y \in \mathcal{M}$ to $x \in \mathcal{M}$: $\mathrm{PT}_{y\to x}(\eta) = (\mathrm{D}\,\phi(x))^{-1}(\mathrm{D}\,\phi(y)(\eta))$ for $\eta \in T_y\mathcal{M}$.

We use induction. Assume $x_\ell = \phi^{-1}(z_\ell)$ for some ℓ. We show $x_{\ell+1} = \phi^{-1}(z_{\ell+1})$.

RK4 scheme in E: Given z_ℓ at time t_ℓ, and step size h:

$$\begin{aligned}
k_1^E &= u_\theta^E(t_\ell, z_\ell, y) \\
k_2^E &= u_\theta^E(t_\ell + \frac{h}{2}, z_\ell + \frac{h}{2}k_1^E, y) \\
k_3^E &= u_\theta^E(t_\ell + \frac{h}{2}, z_\ell + \frac{h}{2}k_2^E, y) \\
k_4^E &= u_\theta^E(t_\ell + h, z_\ell + hk_3^E, y) \\
z_{\ell+1} &= z_\ell + \frac{h}{6}(k_1^E + 2k_2^E + 2k_3^E + k_4^E)
\end{aligned}$$

RK4 scheme on $\mathcal{M}$: A Riemannian RK4 method involves evaluating $u_\theta^{\mathcal{M}}$ at intermediate points, transporting the resulting tangent vectors to $T_{x_\ell}\mathcal{M}$, combining them, and then using the exponential map. By hypothesis, $\phi(x_\ell) = z_\ell$.

1. **First stage ($k_1^{\mathcal{M}}$):** We have

$$k_1^{\mathcal{M}} = u_\theta^{\mathcal{M}}(t_\ell, x_\ell, y) \in T_{x_\ell}\mathcal{M}.$$

So,

$$k_1^{\mathcal{M}} = (\mathrm{D}\,\phi(x_\ell))^{-1}(u_\theta^E(t_\ell, \phi(x_\ell), y)) = (\mathrm{D}\,\phi(x_\ell))^{-1}(k_1^E).$$

The intermediate point is

$$x_A = \exp_{x_\ell}\left(\frac{h}{2}k_1^{\mathcal{M}}\right).$$

Substituting the expressions, we get

$$\begin{aligned}
x_A &= \phi^{-1}\left(\phi(x_\ell) + \mathrm{D}\,\phi(x_\ell)\left(\frac{h}{2}k_1^{\mathcal{M}}\right)\right) \\
&= \phi^{-1}\left(z_\ell + \mathrm{D}\,\phi(x_\ell)\left(\frac{h}{2}(\mathrm{D}\,\phi(x_\ell))^{-1}(k_1^E)\right)\right) \\
&= \phi^{-1}\left(z_\ell + \frac{h}{2}k_1^E\right).
\end{aligned}$$

Thus,

$$\phi(x_A) = z_\ell + \frac{h}{2}k_1^E.$$

2. **Second stage ($k_2^{\mathcal{M}}$):**

$$k_2^{\mathcal{M}} = u_\theta^{\mathcal{M}}(t_\ell + \frac{h}{2}, x_A, y) \in T_{x_A}\mathcal{M}$$

$$k_2^{\mathcal{M}} = (\mathrm{D}\,\phi(x_A))^{-1}(u_\theta^E(t_\ell + \frac{h}{2}, \phi(x_A), y)) = (\mathrm{D}\,\phi(x_A))^{-1}(k_2^E)$$

Transport $k_2^{\mathcal{M}}$ to $T_{x_\ell}\mathcal{M}$:

$$k_{2,\text{tp}}^{\mathcal{M}} = \mathrm{PT}_{x_A \to x_\ell}(k_2^{\mathcal{M}}) = (\mathrm{D}\,\phi(x_\ell))^{-1}(\mathrm{D}\,\phi(x_A)(k_2^{\mathcal{M}})) = (\mathrm{D}\,\phi(x_\ell))^{-1}(k_2^E)$$

The intermediate point

$$x_B = \exp_{x_\ell}(\frac{h}{2}k_{2,\text{tp}}^{\mathcal{M}})$$

$$\begin{aligned} x_B &= \phi^{-1}\left(\phi(x_\ell) + \mathrm{D}\,\phi(x_\ell)(\frac{h}{2}k_{2,\text{tp}}^{\mathcal{M}})\right) \\ &= \phi^{-1}\left(z_\ell + \mathrm{D}\,\phi(x_\ell)(\frac{h}{2}(\mathrm{D}\,\phi(x_\ell))^{-1}(k_2^E))\right) \\ &= \phi^{-1}\left(z_\ell + \frac{h}{2}k_2^E\right) \end{aligned}$$

Thus,

$$\phi(x_B) = z_\ell + \frac{h}{2}k_2^E$$

3. **Third stage ($k_3^{\mathcal{M}}$):**

$$k_3^{\mathcal{M}} = u_\theta^{\mathcal{M}}(t_\ell + \frac{h}{2}, x_B, y) \in T_{x_B}\mathcal{M}$$

$$k_3^{\mathcal{M}} = (\mathrm{D}\,\phi(x_B))^{-1}(u_\theta^E(t_\ell + \frac{h}{2}, \phi(x_B), y)) = (\mathrm{D}\,\phi(x_B))^{-1}(k_3^E)$$

Transport $k_3^{\mathcal{M}}$ to $T_{x_\ell}\mathcal{M}$:

$$k_{3,\text{tp}}^{\mathcal{M}} = \mathrm{PT}_{x_B \to x_\ell}(k_3^{\mathcal{M}}) = (\mathrm{D}\,\phi(x_\ell))^{-1}(\mathrm{D}\,\phi(x_B)(k_3^{\mathcal{M}})) = (\mathrm{D}\,\phi(x_\ell))^{-1}(k_3^E)$$

The intermediate point

$$x_C = \exp_{x_\ell}(hk_{3,\text{tp}}^{\mathcal{M}})$$

$$\begin{aligned} x_C &= \phi^{-1}\left(\phi(x_\ell) + \mathrm{D}\,\phi(x_\ell)(hk_{3,\text{tp}}^{\mathcal{M}})\right) \\ &= \phi^{-1}\left(z_\ell + \mathrm{D}\,\phi(x_\ell)(h(\mathrm{D}\,\phi(x_\ell))^{-1}(k_3^E))\right) \\ &= \phi^{-1}\left(z_\ell + hk_3^E\right) \end{aligned}$$

Thus,

$$\phi(x_C) = z_\ell + hk_3^E$$

4. **Fourth stage ($k_4^{\mathcal{M}}$):**

$$k_4^{\mathcal{M}} = u_\theta^{\mathcal{M}}(t_\ell + h, x_C, y) \in T_{x_C}\mathcal{M}$$

$$k_4^{\mathcal{M}} = (\mathrm{D}\,\phi(x_C))^{-1}(u_\theta^E(t_\ell + h, \phi(x_C), y)) = (\mathrm{D}\,\phi(x_C))^{-1}(k_4^E)$$

Transport $k_4^{\mathcal{M}}$ to $T_{x_\ell}\mathcal{M}$:

$$k_{4,\text{tp}}^{\mathcal{M}} = \mathrm{PT}_{x_C \to x_\ell}(k_4^{\mathcal{M}}) = (\mathrm{D}\,\phi(x_\ell))^{-1}(\mathrm{D}\,\phi(x_C)(k_4^{\mathcal{M}})) = (\mathrm{D}\,\phi(x_\ell))^{-1}(k_4^E)$$

5. **Final update on $\mathcal{M}$:** The combined tangent vector in $T_{x_\ell}\mathcal{M}$ is:

$$\begin{aligned} &\Delta x_{\text{tangent}} = \frac{h}{6}(k_1^{\mathcal{M}} + 2k_{2,\text{tp}}^{\mathcal{M}} + 2k_{3,\text{tp}}^{\mathcal{M}} + k_{4,\text{tp}}^{\mathcal{M}}) \\ &= \frac{h}{6}\left((\mathrm{D}\,\phi(x_\ell))^{-1}(k_1^E) + 2(\mathrm{D}\,\phi(x_\ell))^{-1}(k_2^E) + 2(\mathrm{D}\,\phi(x_\ell))^{-1}(k_3^E) + (\mathrm{D}\,\phi(x_\ell))^{-1}(k_4^E)\right) \\ &= (\mathrm{D}\,\phi(x_\ell))^{-1}\left(\frac{h}{6}(k_1^E + 2k_2^E + 2k_3^E + k_4^E)\right) \end{aligned}$$

Then, $x_{\ell+1} = \exp_{x_\ell}(\Delta x_{\text{tangent}})$.

$$\begin{aligned}
x_{\ell+1} &= \phi^{-1}\left(\phi(x_\ell) + \mathrm{D}\,\phi(x_\ell)(\Delta x_{\text{tangent}})\right) \\
&= \phi^{-1}\left(\phi(x_\ell) + \mathrm{D}\,\phi(x_\ell)\left((\mathrm{D}\,\phi(x_\ell))^{-1}\left(\frac{h}{6}(k_1^E + 2k_2^E + 2k_3^E + k_4^E)\right)\right)\right) \\
&= \phi^{-1}\left(z_\ell + \frac{h}{6}(k_1^E + 2k_2^E + 2k_3^E + k_4^E)\right) \\
&= \phi^{-1}(z_{\ell+1})
\end{aligned}$$

The base case $x_0 = \phi^{-1}(z_0)$ is given by the problem setup ($z_0 = \phi(x_0)$). Therefore, by induction, $x_\ell = \phi^{-1}(z_\ell)$ for all $\ell \in \mathbb{N}$ when the RK4 scheme is applied as described. □

F The affine-invariant metric for correlation matrices

The set of correlation matrices Corr_d can be viewed as a quotient manifold of symmetric positive definite matrices $\mathbb{S}_d^{++}$ by the action of positive diagonal matrices $\mathbb{D}_d^{++}$ [16, 68]:

$$\mathrm{Corr}_d = \mathbb{S}_d^{++}/\mathbb{D}_d^{++}\,, \tag{25}$$

where two matrices $\mathbf{\Sigma}, \mathbf{\Sigma}' \in \mathbb{S}_d^{++}$ are equivalent if there exists $\mathbf{D} \in \mathbb{D}_d^{++}$ such that $\mathbf{\Sigma}' = \mathbf{D}\mathbf{\Sigma}\mathbf{D}$.

This quotient structure naturally induces a Riemannian metric on Corr_d from the affine-invariant metric on $\mathbb{S}_d^{++}$, defined by

$$\langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle_{\mathbf{\Sigma}}^{\mathbb{S}_d^{++}} = \mathrm{tr}(\mathbf{\Sigma}^{-1}\boldsymbol{\xi}\mathbf{\Sigma}^{-1}\boldsymbol{\eta}). \tag{26}$$

The canonical projection (submersion) $\pi : \mathbb{S}_d^{++} \to \mathrm{Corr}_d$ is given by diagonal normalization:

$$\pi(\mathbf{\Sigma}) = \mathrm{diag}(\mathbf{\Sigma})^{-1/2}\mathbf{\Sigma}\,\mathrm{diag}(\mathbf{\Sigma})^{-1/2}. \tag{27}$$

The resulting geometry on Corr_d requires solving an optimization problem to compute the Riemannian logarithm:

$$\log_{\mathbf{A}}^{\mathrm{Corr}_d}(\mathbf{B}) = \mathrm{D}\,\pi(\mathbf{A})\left(\log_{\mathbf{A}}^{\mathbb{S}_d^{++}}(\mathbf{D}\mathbf{B}\mathbf{D})\right), \tag{28}$$

where

$$\mathbf{D} = \underset{\mathbf{D}\in\mathbb{D}_d^{++}}{\arg\min}\, d_{\mathbb{S}_d^{++}}\left(\mathbf{A}, \mathbf{D}\mathbf{B}\mathbf{D}\right)\,. \tag{29}$$

Since no closed-form solution for $\mathbf{D}$ is known, the logarithmic map on $(\mathrm{Corr}_d, \langle\cdot,\cdot\rangle^{\mathrm{Corr}_d})$ cannot be expressed analytically (contrary to $\log_{\mathbf{A}}^{\mathbb{S}_d^{++}}$).

G Datasets and Preprocessing

Resting-state fMRI datasets and preprocessing We used three publicly available resting-state fMRI datasets—ABIDE [53], ADNI [73], and OASIS-3 [42]—spanning a wide age range and neurological conditions. ABIDE includes 900 subjects (one scan each; mean age 17), both neurotypical and autistic, from 19 international sites. ADNI comprises 1900 scans from 900 older adults (mean age 74), covering normal ageing, mild cognitive impairment, and Alzheimer's disease. OASIS-3 includes 1000 participants and 1800 longitudinal sessions collected over 10 years (mean age 71), targeting healthy and pathological ageing.

Preprocessing was performed with `fMRIPrep` [24], which applies bias-field correction, skull stripping, tissue segmentation, nonlinear normalization to MNI space, motion correction, and confound estimation. Regional time series were extracted with `Nilearn` [14] using the MSDL atlas [71], followed by nuisance regression (including motion, CSF/WM signals, and global signal). Time series were z-scored and screened for minimum length, numerical anomalies (e.g., zeros or extreme values), and the conditioning of their covariance matrices.

EEG datasets Sessions with missing or non-standard protocol (e.g., 2C in BNCI2015-001) were discarded. We further filtered time series whose covariance matrices have high values (max entry $> 10^4$) or statistical outliers based on Mahalanobis distance from the group mean. This retained the top 90% most consistent trials for downstream modeling.

BNCI2014-002: The BNCI2014-002[3] dataset, provided through the BNCI Horizon 2020 initiative, includes recordings from 13 subjects engaged in motor imagery tasks. Participants were instructed to imagine movements of either their right hand or both feet for a sustained duration of 5 seconds, guided by visual cues. EEG signals were captured using an amplifier and active Ag/AgCl electrodes at a sampling rate of 512 Hz, with a total of 15 electrode channel applied to each subject. The experimental protocol comprised eight sessions per participant, each including 80 trials for hand and foot imagery, summing to 160 trials. EEG epochs were extracted from 3.0 to 8.0 seconds relative to cue onset, aligning with the motor imagery window.

BNCI2015-001: The BNCI2015-001[4] dataset, released as part of the BNCI Horizon 2020 project, contains EEG recordings from 12 individuals who performed motor imagery tasks involving either the right hand or both feet. EEG data were acquired at a 512 Hz sampling rate, preprocessed with a 0.5–100 Hz bandpass filter and a 50 Hz notch filter. Each trial spanned 5 seconds, beginning at 3.0 seconds after cue onset and ending at 8.0 seconds, corresponding to the motor imagery phase. The dataset provides EEG data from 13 electrode channels arranged in the following order: FC3, FCz, FC4, C5, C3, C1, Cz, C2, C4, C6, CP3, CPz, and CP4. For most participants (Subjects 1–8), recordings were conducted over two sessions on consecutive days. In contrast, Subjects 9–12 completed three sessions. Each session consisted of 100 trials per class, resulting in 200 trials per session per subject.

[3] https://neurotechx.github.io/moabb/generated/moabb.datasets.BNCI2014_002.html
[4] https://neurotechx.github.io/moabb/generated/moabb.datasets.BNCI2015_001.html

H Comparison of Baseline Methods: Assumptions, Strengths, and Limitations

In this section, we summarize the key assumptions, strengths, and limitations of the baseline methods discussed throughout the paper, based on our experimental findings.

- DIFFEOGAUSS: This is a simple yet practical generative approach that operates directly on SPD or correlation manifolds. While it yields competitive results in terms of the CAS metrics, it generally underperforms compared to deep generative models in terms of sample quality, see Table 1.
- TRIANGDDPM or TRIANGCFM: They are deep generative models that achieve strong performance across both CAS and quality-based metrics. However, their outputs are limited to triangular entities, requiring a post-processing projection step to obtain valid positive definite matrices. Unfortunately, this step alters the eigenvalue spectrum significantly, which degrades the quality and structure of the generated matrices (see Table 2 and Appendix M).
- SPD-DDPM: This work extends DDPM to the space of SPD manifolds under the affine-invariant Riemannian metric. However, in our experiments, we found that its implementation repeatedly relies on sampling from Gaussian distributions on SPD manifolds, as well as backpropagating through matrix operations in SPDNet. Both components are computationally intensive in theory and practice, making the method prohibitively slow. In fact, for a number of training samples above 64, we were unable to obtain results within a reasonable time frame (see Figure 4). However, in the fMRI experiments, the data have at least 1000 samples.
- RIEMANNIAN CFM: This is a foundational contribution in the family of flow matching methods, providing a general theoretical framework for generative modeling on Riemannian manifolds. However, its construction is geometry-specific and requires a dedicated implementation for each manifold. Furthermore, on SPD matrices, we observe that it is roughly $8\times$ slower to train and $10\times$ slower to sample from than DIFFEOCFM (see Table 1), due to the repeated computation of affine-invariant geodesics, their derivatives and Riemannian norms.
- DIFFEOCFM: The proposed approach introduces a novel Riemannian CFM framework using the pullback geometry, enabling direct data generation on SPD and correlation manifolds, particularly helpful in neuroscience and neuroengineering tasks. It trains and sampels efficiently (see Figure 4 and Table 1), achieves strong performance on both quality and CAS metrics (see Table 1), and consistently demonstrates neurophysiological plausibility across two distinct evaluation protocols in both fMRI and EEG (see Section 5.2).

I Summary Figure

While Table 1 provides a comprehensive numerical evaluation of all methods, a visual representation can offer clearer insights into the practical trade-offs between model performance and computational efficiency. To this end, Figure 3 plots a unified performance metric,the Average F1 Score, against both training and sampling times. This allows for a direct comparison of the performance-cost profile of each method for both fMRI and EEG data.

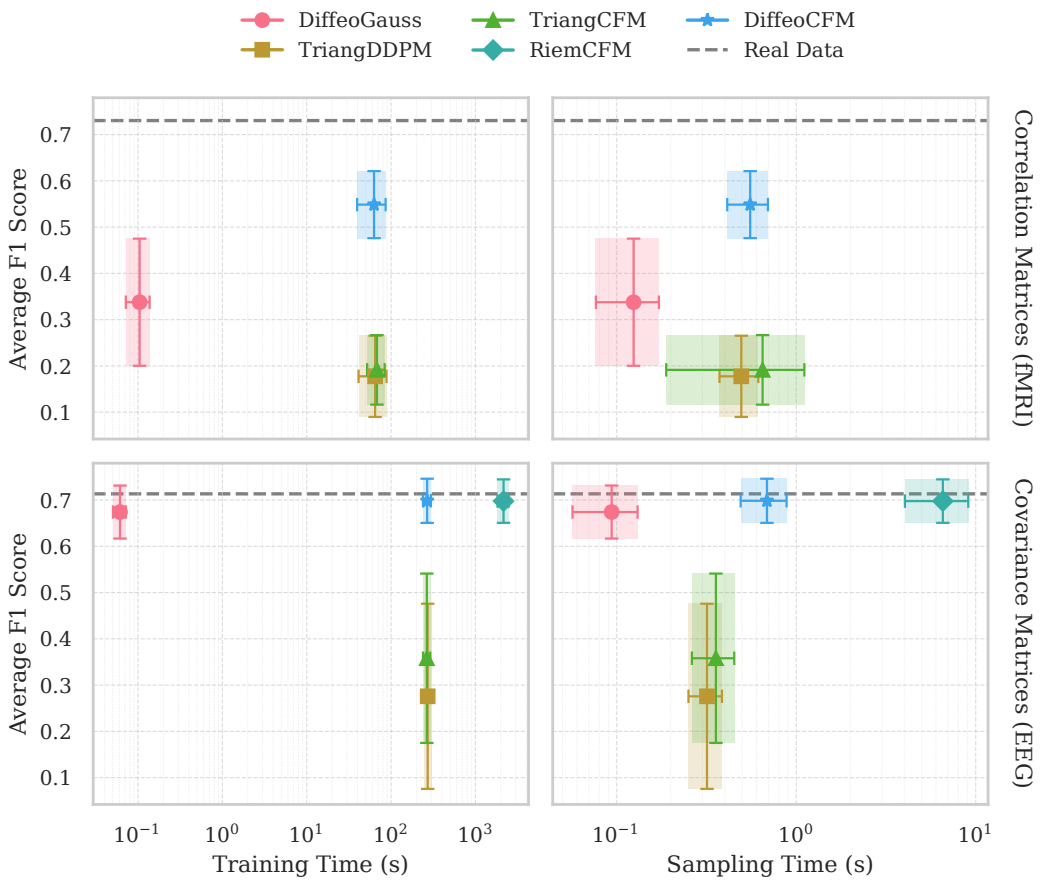


Figure 3: **Trade-off between generative performance and computational cost for fMRI (top) and EEG (bottom) data.** The figure plots the Average F1 Score against Training Time (left) and Sampling Time (right). The Average F1 Score is the mean of the quality metric (α, β-F1) and the CAS F1-score from Table 1. Each point marks the mean performance across all splits and datasets for a given modality, with error bars and shaded regions indicating the standard deviation. The dashed gray line represents the *Real Data* baseline, which serves as an empirical upper bound. Time is shown on a logarithmic scale.

J Scalability and Computational cost of SPD–DDPM [45] vs DIFFEOCFM

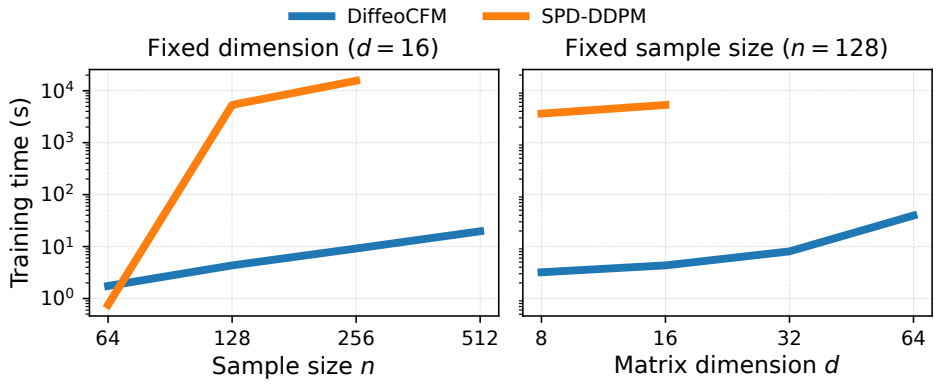


Figure 4: **Training efficiency on $\mathbb{S}_d^{++}$ (simulated data).** Log-scale training time over 1000 epochs for generating SPD matrices. We vary either the number of samples n or the matrix dimension d, keeping the other fixed. DIFFEOCFM leverages the diffeomorphism $\phi_{\mathbb{S}_d^{++}}$ (9) and is at least 1000× faster than SPD-DDPM [45]. Importantly, DIFFEOCFM *can also generate natively correlation matrices (Corr$_d$), and it performs with same training time as on $\mathbb{S}_d^{++}$.*

K Ablation Study

We conduct an ablation study to analyze the sensitivity of our proposed DIFFEOCFM model to its neural network architecture. The experiments are performed on the ABIDE dataset using the Cholesky diffeomorphism. We investigate two key architectural hyperparameters: the width (size of hidden layers) and the depth (number of hidden layers).

First, we fix the network depth to a single hidden layer and vary its width from 8 to 1024, as detailed in Table 3. Second, we fix the layer width to 512 neurons and vary the depth from one to four hidden layers, with results presented in Table 4. For both experiments, performance is evaluated using the α, β-F1 score, which is the harmonic mean of α-precision and β-recall. We also report the average training and sampling times over 10 splits.

The results from Table 3 indicate that performance generally improves with network width, peaking at 256 and 512 hidden units. A wider network of 1024 units shows a drop in performance. Regarding network depth, Table 4 clearly shows that a single hidden layer achieves the best results. Deeper models exhibit a consistent decline in performance. Based on these findings, a single hidden layer with a width of 512 was chosen for our main experiments, as it offers the best performance without unnecessary complexity.

Table 3: Ablation study on the **width** of the hidden layer for the DIFFEOCFM model on the ABIDE dataset. The network depth is fixed to one hidden layer. We report the mean and standard deviation of the α, β-F1 score, training time, and sampling time across 10 splits. **Best** values are in bold.

Hidden dim.	α, β-F1	Training time (s.)	Sampling time (s.)
8	0.47 ± 0.06	29.07 ± 1.07	0.35 ± 0.01
16	0.47 ± 0.09	**27.46 ± 1.24**	0.29 ± 0.03
32	0.54 ± 0.05	28.36 ± 1.26	**0.26 ± 0.02**
64	0.57 ± 0.07	29.69 ± 1.31	0.30 ± 0.03
128	0.60 ± 0.05	31.01 ± 1.32	0.35 ± 0.04
256	**0.63 ± 0.07**	31.39 ± 0.96	0.32 ± 0.04
512	**0.63 ± 0.06**	31.36 ± 1.14	0.31 ± 0.02
1024	0.49 ± 0.12	32.34 ± 1.17	0.36 ± 0.05

Table 4: Ablation study on the **depth** of the neural network for the DIFFEOCFM model on the ABIDE dataset. The width of all hidden layers is fixed to 512. We report the mean and standard deviation of the α, β-F1 score, training time, and sampling time across 10 splits. **Best** values are in bold.

Depth	α, β-F1	Training time (s.)	Sampling time (s.)
1	**0.62 ± 0.05**	33.06 ± 1.05	0.38 ± 0.03
2	0.52 ± 0.09	**32.93 ± 1.00**	**0.36 ± 0.02**
3	0.46 ± 0.05	34.36 ± 1.00	0.46 ± 0.04
4	0.50 ± 0.08	35.68 ± 1.01	0.48 ± 0.08

L Constraint Satisfaction ($\mathbb{S}_d^{++}$ and Corr$_d$)

Table 5 reports the fraction of generated samples that satisfy structural matrix constraints across datasets. Both DIFFEOGAUSS and DIFFEOCFM systematically generate matrices that are symmetric, positive definite, and (for correlation matrices) have unit diagonal, achieving a perfect 1.00 score on all datasets. In contrast, TRIANGCFM without projection produces matrices that are symmetric and have unit diagonal by construction but fail to ensure positive definiteness. This issue is particularly evident in fMRI datasets, where none of the generated samples satisfy the SPD constraint, and persists to a lesser extent on EEG datasets (e.g., 29% validity on BNCI2015-001). These results highlight the importance of geometry-aware methods like DIFFEOCFM, which inherently respect the manifold structure of connectivity matrices without relying on post hoc projections.

Table 5: **Fraction of generated samples satisfying matrix constraints.** We report the fraction of samples satisfying symmetry, positive definiteness, and unit diagonal constraints across datasets.

Dataset	Method	Sym.	Pos. def.	Unit diag.
ABIDE	DIFFEOGAUSS	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
	TRIANGCFM (no proj.)	**1.00 ± 0.00**	0.00 ± 0.00	**1.00 ± 0.00**
	DiffeoCFM	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
ADNI	DIFFEOGAUSS	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
	TRIANGCFM (no proj.)	**1.00 ± 0.00**	0.00 ± 0.00	**1.00 ± 0.00**
	DiffeoCFM	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
OASIS-3	DIFFEOGAUSS	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
	TRIANGCFM (no proj.)	**1.00 ± 0.00**	0.00 ± 0.00	**1.00 ± 0.00**
	DiffeoCFM	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
BNCI 2014-002	DIFFEOGAUSS	**1.00 ± 0.00**	**1.00 ± 0.00**	
	TRIANGCFM (no proj.)	**1.00 ± 0.00**	0.00 ± 0.00	
	DiffeoCFM	**1.00 ± 0.00**	**1.00 ± 0.00**	
BNCI 2015-001	DIFFEOGAUSS	**1.00 ± 0.00**	**1.00 ± 0.00**	
	TRIANGCFM (no proj.)	**1.00 ± 0.00**	0.29 ± 0.02	
	DiffeoCFM	**1.00 ± 0.00**	**1.00 ± 0.00**	

M Projection

Since TRIANGDDPM and TRIANGCFM operate in Euclidean spaces, the generated matrices are not guaranteed to be positive definite. To address this, we apply a projection to ensure all eigenvalues are at least $\epsilon > 0$. Let $\lambda_{\text{min}}(\mathbf{\Sigma})$ denote the smallest eigenvalue of $\mathbf{\Sigma}$. If $\lambda_{\text{min}}(\mathbf{\Sigma}) < \epsilon$, the matrix is projected using an affine transformation:

$$\mathbf{\Sigma}_{\text{proj}} = (1 - \alpha)\mathbf{\Sigma} + \alpha\mathbf{I} \tag{30}$$

where $\mathbf{I}$ is the identity matrix. The parameter α is chosen such that the smallest eigenvalue of $\mathbf{\Sigma}_{\text{proj}}$ becomes exactly ϵ. Given that the eigenvalues of $\mathbf{\Sigma}_{\text{proj}}$ are $\lambda_i(\mathbf{\Sigma}_{\text{proj}}) = (1 - \alpha)\lambda_i(\mathbf{\Sigma}) + \alpha$, setting

the new minimum eigenvalue to ϵ yields $\epsilon = (1-\alpha)\lambda_{\text{min}}(\mathbf{\Sigma}) + \alpha$. Solving for α, we get:

$$\alpha = \frac{\epsilon - \lambda_{\text{min}}(\mathbf{\Sigma})}{1 - \lambda_{\text{min}}(\mathbf{\Sigma})}\,. \tag{31}$$

This formulation guarantees that if $\lambda_{\text{min}}(\mathbf{\Sigma}) < \epsilon$, the new minimum eigenvalue is ϵ. Typically, ϵ is a small positive value (e.g., 10^{-8}), ensuring $\epsilon \leq 1$. In this regime, and given $\lambda_{\text{min}}(\mathbf{\Sigma}) < \epsilon$, it follows that $0 < \alpha \leq 1$, making the projection an interpolation. An important property of this projection is that if the original matrix $\mathbf{\Sigma}$ has ones on its diagonal ($\text{diag}(\mathbf{\Sigma}) = \mathbf{1}$), this property is preserved in $\mathbf{\Sigma}_{\text{proj}}$, as $\text{diag}(\mathbf{\Sigma}_{\text{proj}}) = (1-\alpha)\mathbf{1} + \alpha\mathbf{1} = \mathbf{1}$.

We randomly select 5 sample images from each generated dataset produced by the TRIANGCFM method across the 5 datasets. These images are not positive definite matrices, as they contain negative eigenvalues. We apply a projection algorithm to convert these matrices into valid correlation matrices, and both the original and projected versions are shown in Figure 5 and Figure 6. In each image, we annotate the maximum and minimum eigenvalues of the corresponding matrix. As can be seen, after the projection algorithm, all negative eigenvalues are eliminated, and the matrices become valid positive definite matrices. *Unfortunately, although this step is necessary for* TRIANGDDPM *and* TRIANGCFM*, it alters the spectrum of the matrices, which is usually crucial in the downstream classification tasks. The advantage of the proposed* DIFFEOCFM *lies in the fact that it does not require this step, and thus naturally preserves the matrix spectrum.*

N DIFFEOCFM on EEGs

N.1 Filter's Topographic Map

Common Spatial Pattern

The Common Spatial Pattern (CSP) filter is a supervised spatial filtering technique widely used in EEG-based signal analysis, particularly for motor imagery classification. It enhances class-discriminative information by projecting multi-channel EEG data onto a low-dimensional space that maximizes variance differences between two classes [6]. Formally, for a signal $x(t) \in \mathbb{R}^{N_C}$, the CSP algorithm seeks a projection matrix $W \in \mathbb{R}^{N_C \times N_C}$ such that the transformed signal $x_{CSP}(t) = W^T x(t)$ yields maximally discriminative variance patterns across classes, where N_C is the number of electrode channels. Each column vector $w_j \in \mathbb{R}^{N_C}$ $(j = 1, \cdots, N_C)$ of matrix W is called the filter of CSP.

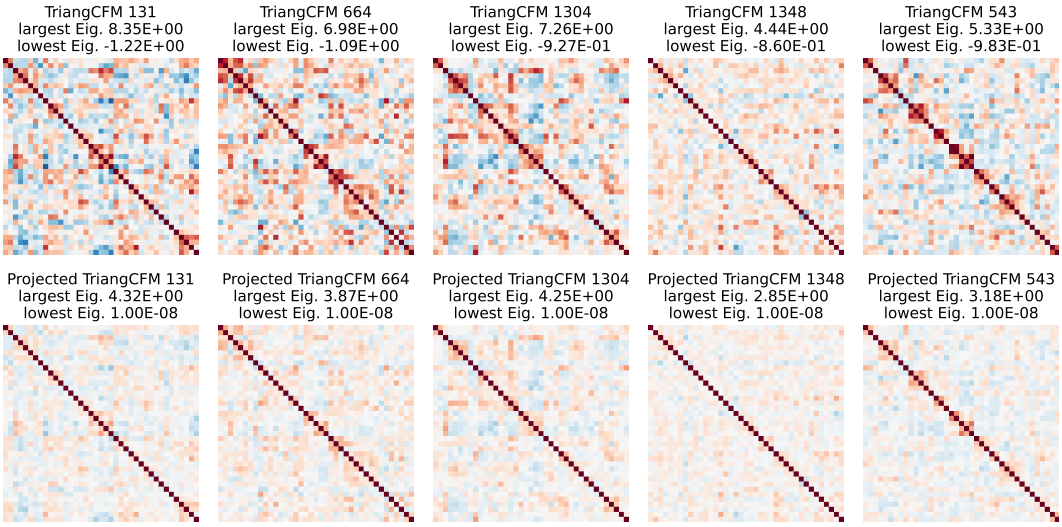
Generally, CSP is a spatial filtering method that effectively enhances the discrimination of mental states characterized by event-related de-synchronization/synchronization (ERD/ERS), which commonly occurs in the alpha (8–12 Hz) and beta (13–30 Hz) bands during motor imagery tasks [63].

Experimental Settings

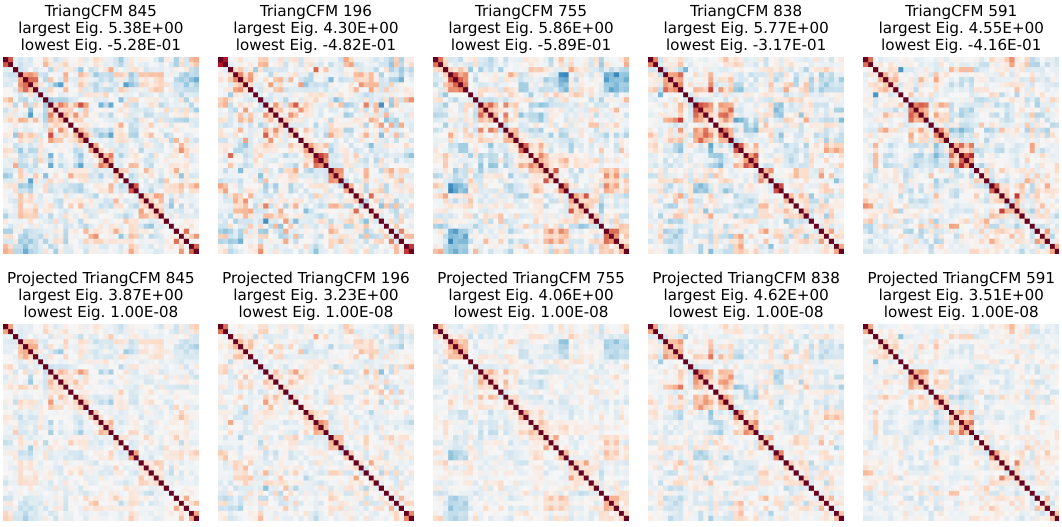
In this experiment, we aim to demonstrate whether the proposed model, DIFFEOCFM, can learn similar spatial filter patterns $\{w_j\}_{j=1}^{N_C}$ in the alpha and beta frequency bands during motor imagery, as reflected in their corresponding topographies.

Specifically, we conduct this analysis only on the BNCI2015-001 dataset, as it provides the predefined order of 13 EEG channels, allowing us to correctly map each channel to its corresponding position in the topographic montage. In contrast, the BNCI2014-002 dataset does not offer such channel ordering information.

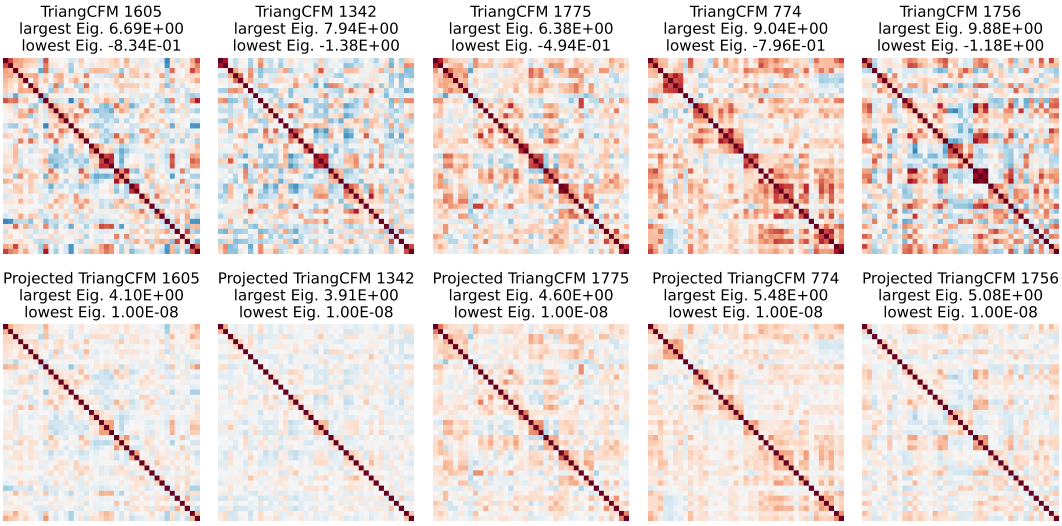
All filters were computed using data from session 1 of each subject individually, in order to remain consistent with the cross-session experimental setting. The filters for real data were computed directly using the CSP algorithm on session 1 data from each subject, focusing on the 8-12 Hz (alpha) and 13-30 Hz (beta) frequency bands during the first 2 seconds following the cue. For the generated data, filters were obtained by applying the same CSP procedure on generated datasets produced by DIFFEOCFM, with the same number of samples as in session 1. In both cases, only the spatial filter corresponding to the largest eigenvalue from CSP's generalized eigenvalue decomposition was retained for analysis.



(a) **ADNI**

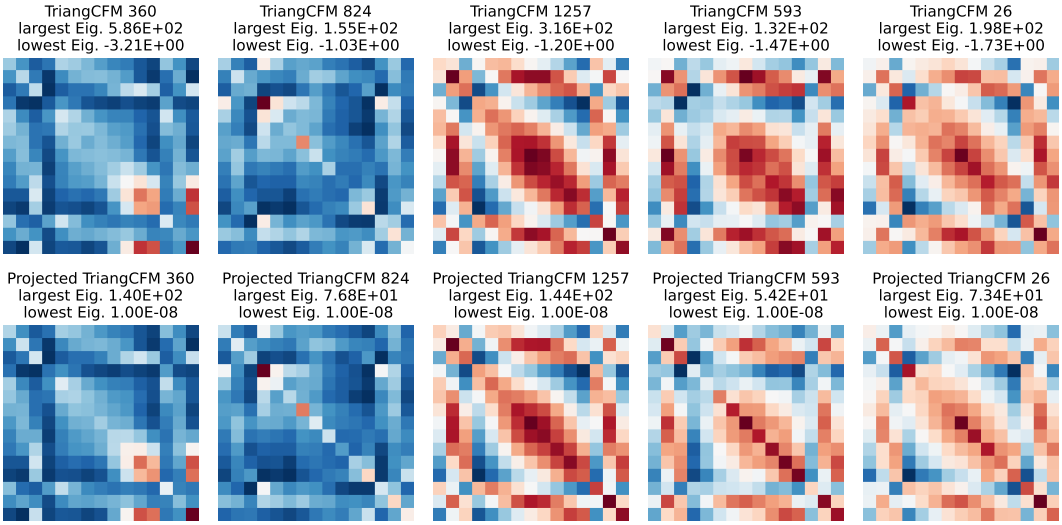


(b) **ABIDE**

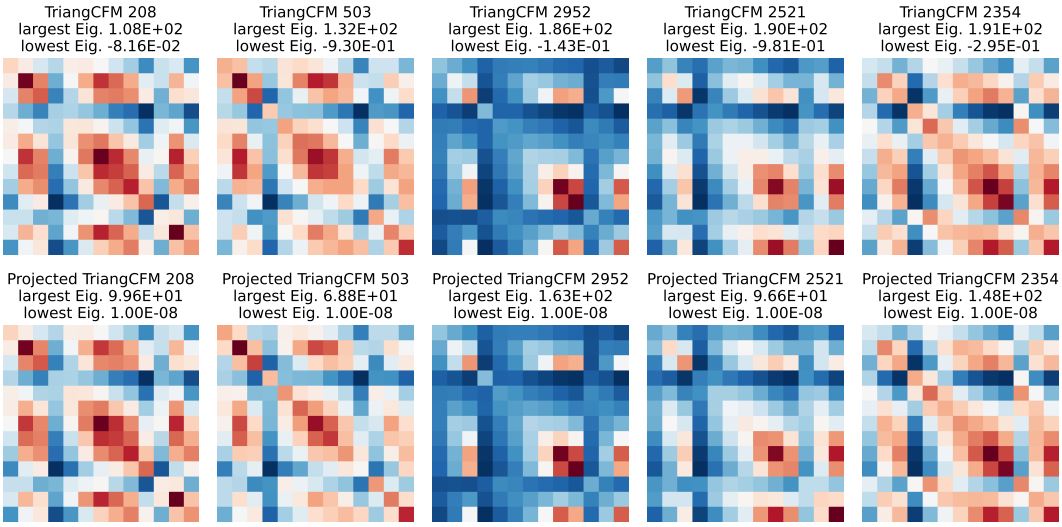


(c) **OASIS3**

Figure 5: Comparison of fMRI matrices before and after projection



(a) **BNCI2014-002**



(b) **BNCI2015-001**

Figure 6: Comparison of EEG matrices before and after projection

N.2 Results and Analysis

Figure 7 shows the topographic map using the first CSP filter for each subject in the BNCI2015-001 dataset (Subjects 1-6), while Figure 8 displays the topographic map using the first CSP's filter for each subject in the same dataset (Subjects 7-12). We can observe that, except for Subjects 5, 6, 10, and 12, the topographies for the remaining subjects appear highly similar, with a noticeable increase in signal amplitude in the C3 region. Moreover, the average topographic map of these subjects, shown in Figure 2b, also reveals a very similar filter pattern.

This results show that the model, DIFFEOCFM, used for generating the generated data has successfully captured the key characteristics and patterns of the real EEGs. This similarity in CSP's filters indicates that **the generated data closely resembles the real data in terms of spatial patterns of brain activity**, which is crucial for validating the effectiveness of the model in simulating realistic neural processes, particularly for motor imagery tasks. Additionally, it may imply that **the model has learned to preserve the underlying structure and discriminative features that are typically seen in real EEG data during motor imagery**.

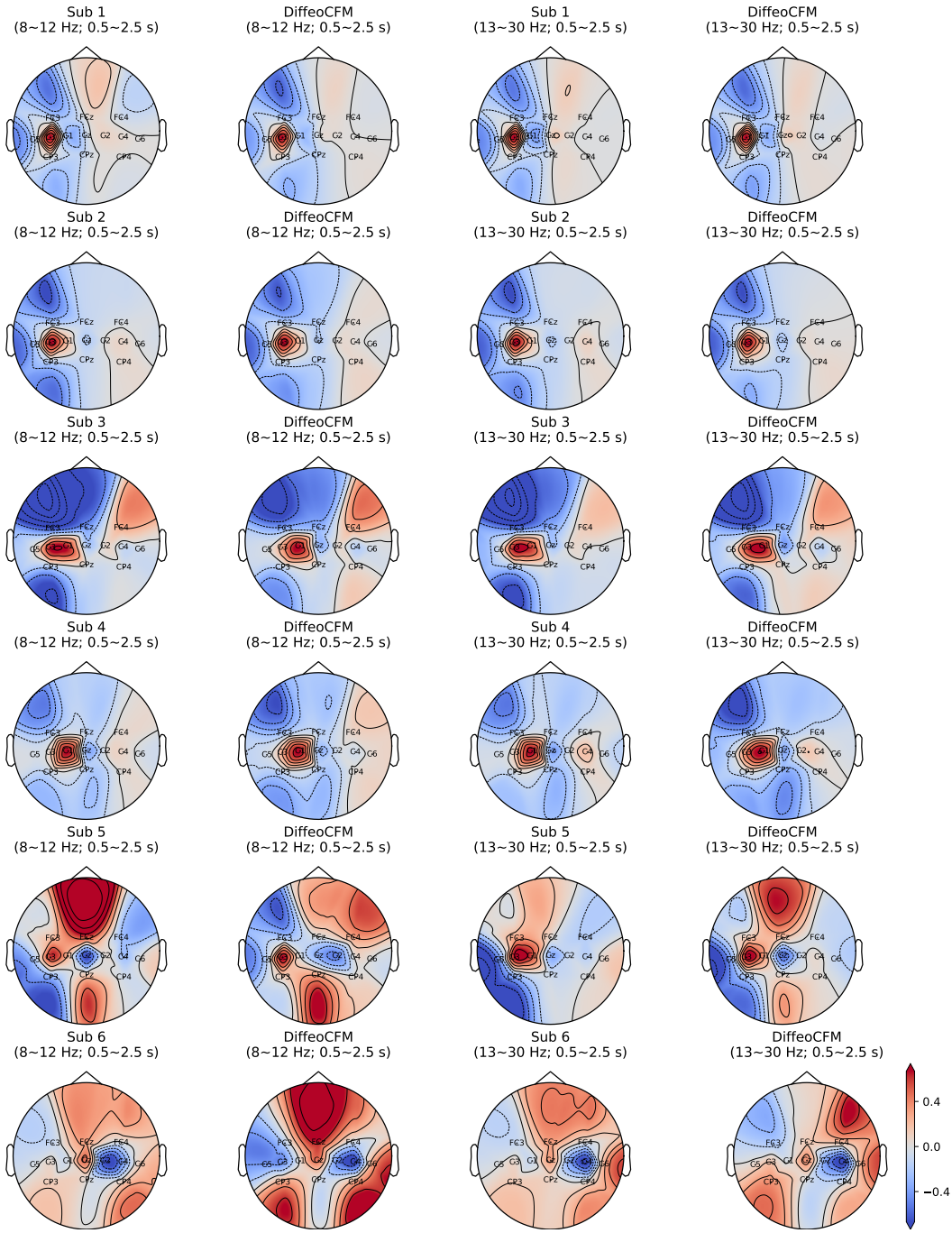


Figure 7: Subject-level topographic map using the first CSP's filter derived from real EEGs (BNCI2015-001, Sub 1-6) and generated data by DIFFEOCFM

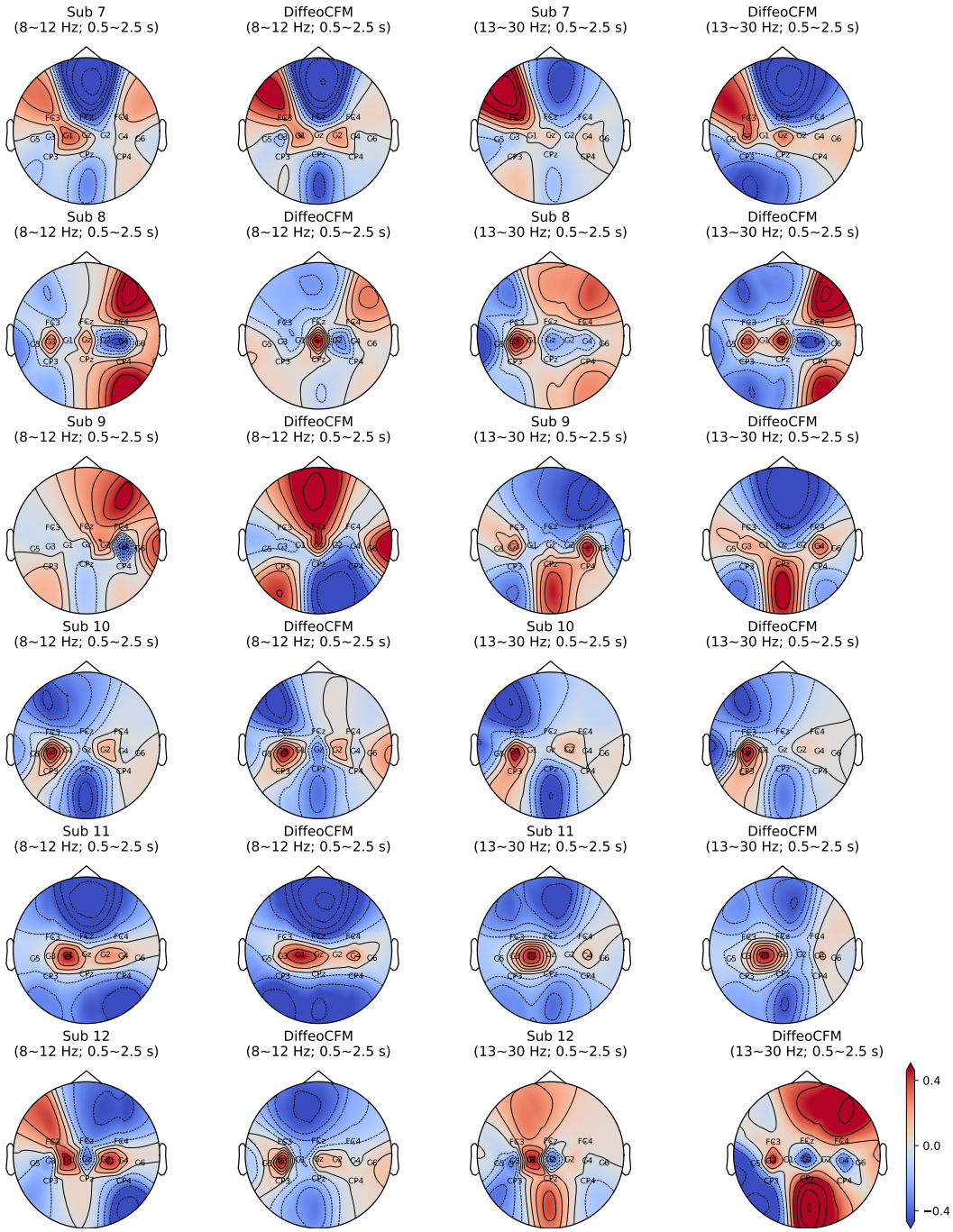


Figure 8: Subject-level topographic map using the first CSP's filter derived from real EEGs (BNCI2015-001, Sub 7-12) and generated data by DIFFEOCFM

O DIFFEOCFM on fMRI

O.1 fMRI Connectome Plotting

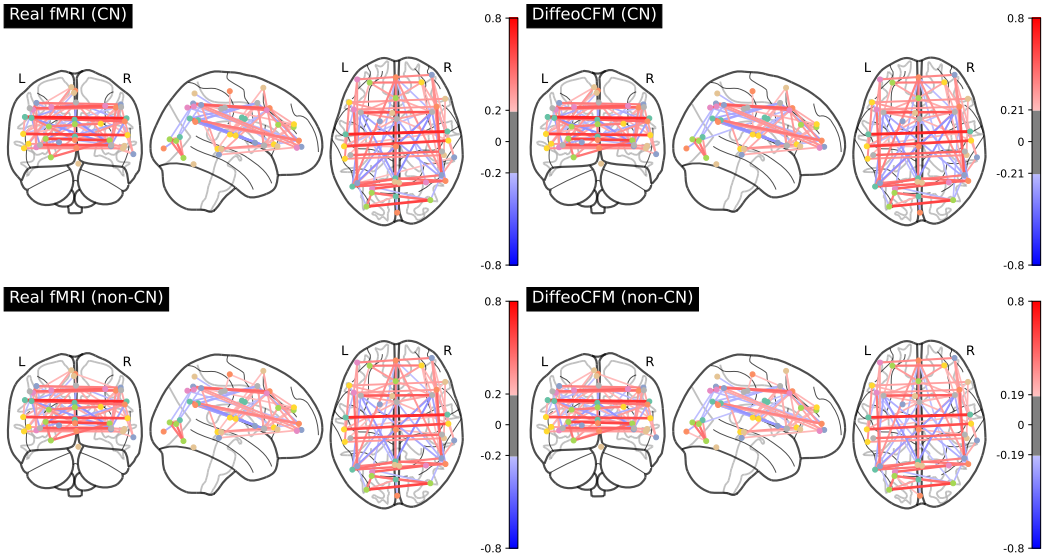
Experimental Settings

We use Nilearn [14] for neuroimaging analysis and visualization based on the MSDL brain atlas. The atlas is retrieved using the fetch_atlas_msdl function, which provides probabilistic maps of functional brain networks in the form of a 4D NIfTI image. The maps attribute extracts this image, while find_probabilistic_atlas_cut_coords computes representative 3D coordinates for each network, typically corresponding to the spatial center of the activation, to support anatomical localization. In addition, predefined region labels are extracted to annotate the corresponding brain areas. These components together enable the visualization of brain networks, including plotting the regions with spatial coordinates and labels.

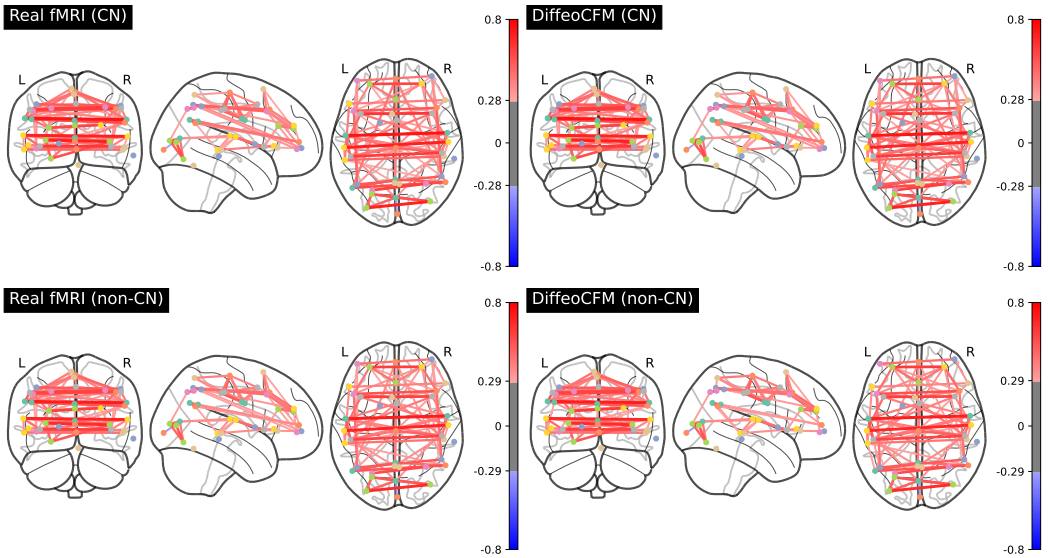
The connectome plots are grouped into two categories: "CN" and "non-CN". The adjacency_matrix used in nilearn.plotting.plot_connectome is computed as the Fréchet mean of the corresponding group of connectivity matrices in all the 5-fold experiments, and all edge thresholds in the plots are set to 90%.

Results and Analysis

Figure 9a and Figure 9b visualize group-averaged functional connectomes derived from two distinct neuroimaging datasets using the Fréchet mean. By thresholding edges to retain only the strongest 10% of connections (90% edge threshold), these plots emphasize dominant, statistically reliable interactions between brain regions while filtering out noise. The use of the Fréchet mean ensures that the group-level adjacency matrices respect the intrinsic geometric structure of covariance data, avoiding distortions caused by arithmetic averaging. **This comparison highlights both the fidelity of data generation methods by aligning generated and real data connectomes using the Fréchet mean and potential differences in functional network organization between the two cohorts.** The workflow underscores the utility of geometric statistics and stringent thresholds for interpreting brain connectivity in heterogeneous populations.



(a) **Class-conditional fMRI functional connectomes using the Fréchet mean (ABIDE).**



(b) **Class-conditional fMRI functional connectomes using the Fréchet mean (OASIS3).**

Figure 9: Group-level functional connectomes, computed as Fréchet means from two separate neuroimaging datasets, are visualized after applying a 90% edge threshold to retain only the top 10% of strongest connections. This thresholding highlights dominant and statistically robust interactions between brain regions, effectively reducing noise. Leveraging the Fréchet mean preserves the underlying geometric structure of the covariance matrices, avoiding potential distortions introduced by conventional Euclidean averaging.

P Plotting of Generated Samples in Real Data Neighborhoods

Figure 10-15 present the selected generated fMRI samples that are nearest to real data points in terms of Frobenius distance, categorized by "CN" and "non-CN" from the ADNI, ABIDE, and OASIS3 datasets.. Figure 16-19 show the selected generated EEG samples that are nearest to real samples in Frobenius distance, separately for hand and feet motor imagery tasks from the BNCI2014-002 and BNCI2015-001 datasets. Each real sample is annotated with its index in the original dataset. The first, second, and third columns correspond to samples generated by DIFFEOGAUSS, TRIANGCFM, and DIFFEOCFM, respectively. The numeric values on the generated samples indicate the Frobenius distance to the corresponding real sample. Each generated sample shown is the nearest one (in Frobenius distance) to the real sample within its class. Due to its consistently highest recall across multiple datasets, DIFFEOCFM is more likely to produce generated samples that are closer to real samples, i.e., with the smallest Frobenius distance, compared to those generated by other methods. The matrices generated by TRIANGCFM shown in the figures are presented without applying the projection step. Before applying the projection, the outputs of TRIANGCFM already exhibit strong recall scores.

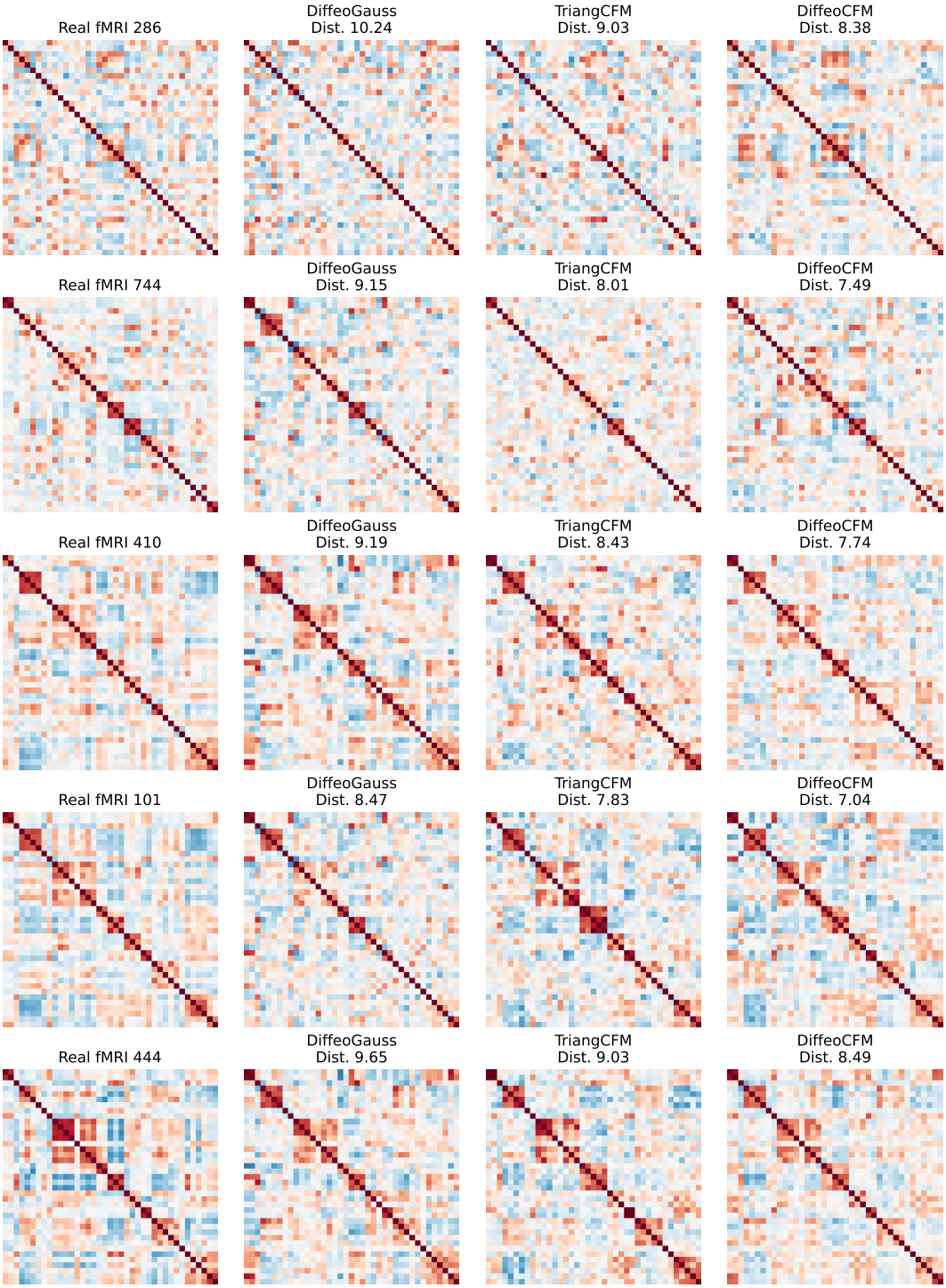


Figure 10: Nearest Generated Samples in Real Data Neighborhoods: ADNI-CN Cohort. The matrices generated by TRIANGCFM shown in the figures are presented without applying the projection step.

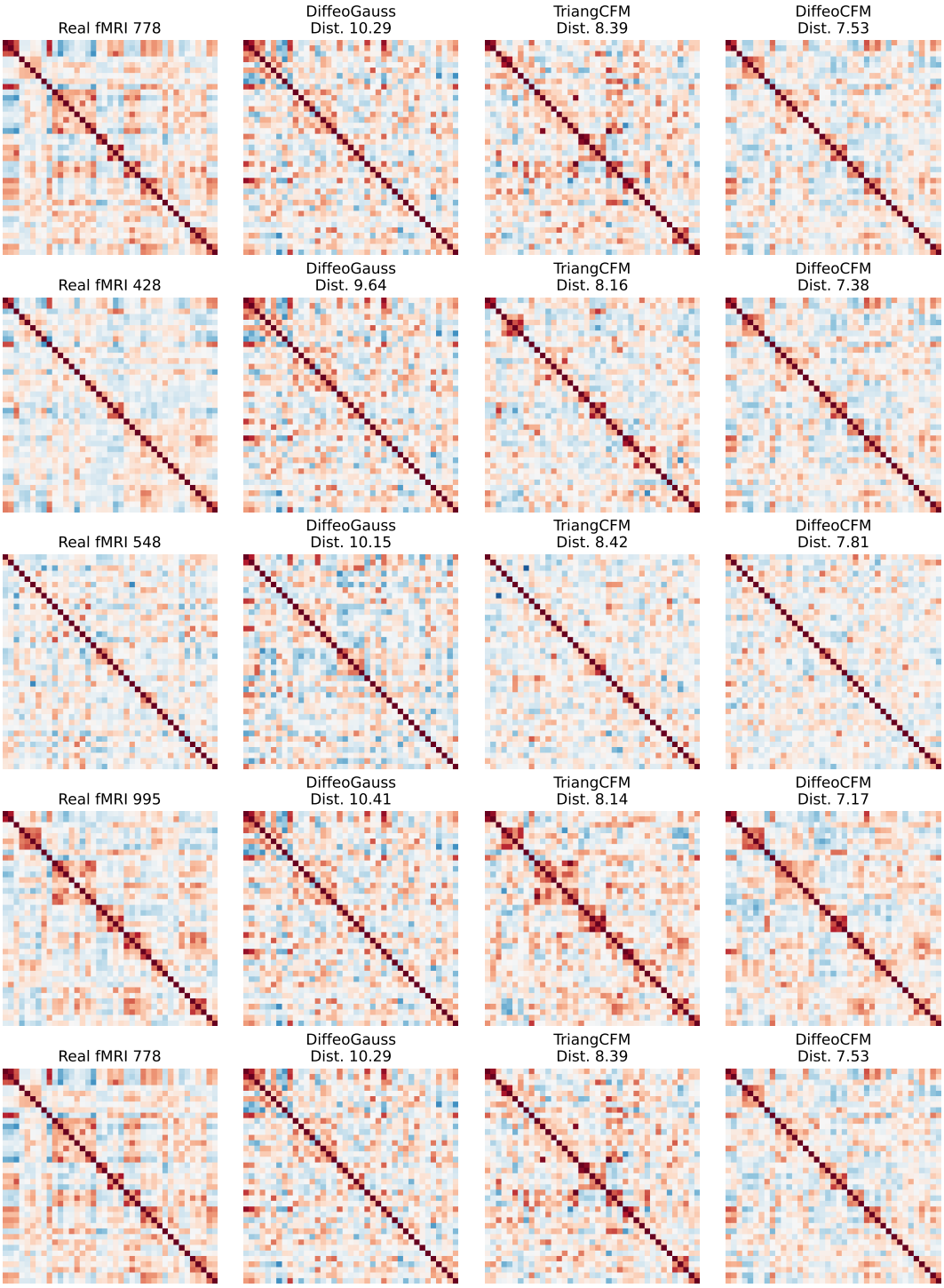


Figure 11: Nearest Generated Samples in Real Data Neighborhoods: ADNI-nonCN Cohort. The matrices generated by TRIANGCFM shown in the figures are presented without applying the projection step.

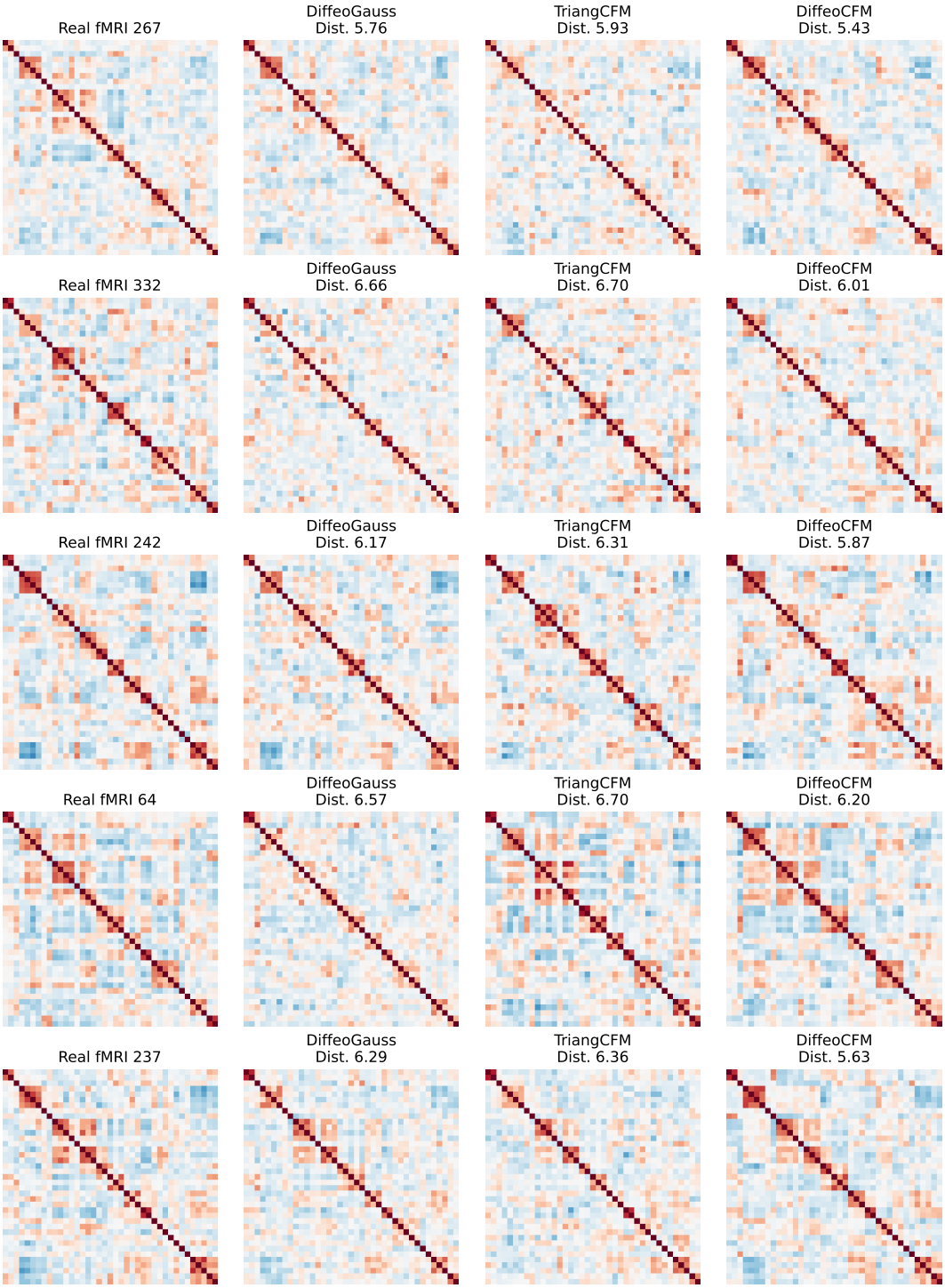


Figure 12: Nearest Generated Samples in Real Data Neighborhoods: ABIDE-CN Cohort. The matrices generated by TRIANGCFM shown in the figures are presented without applying the projection step.

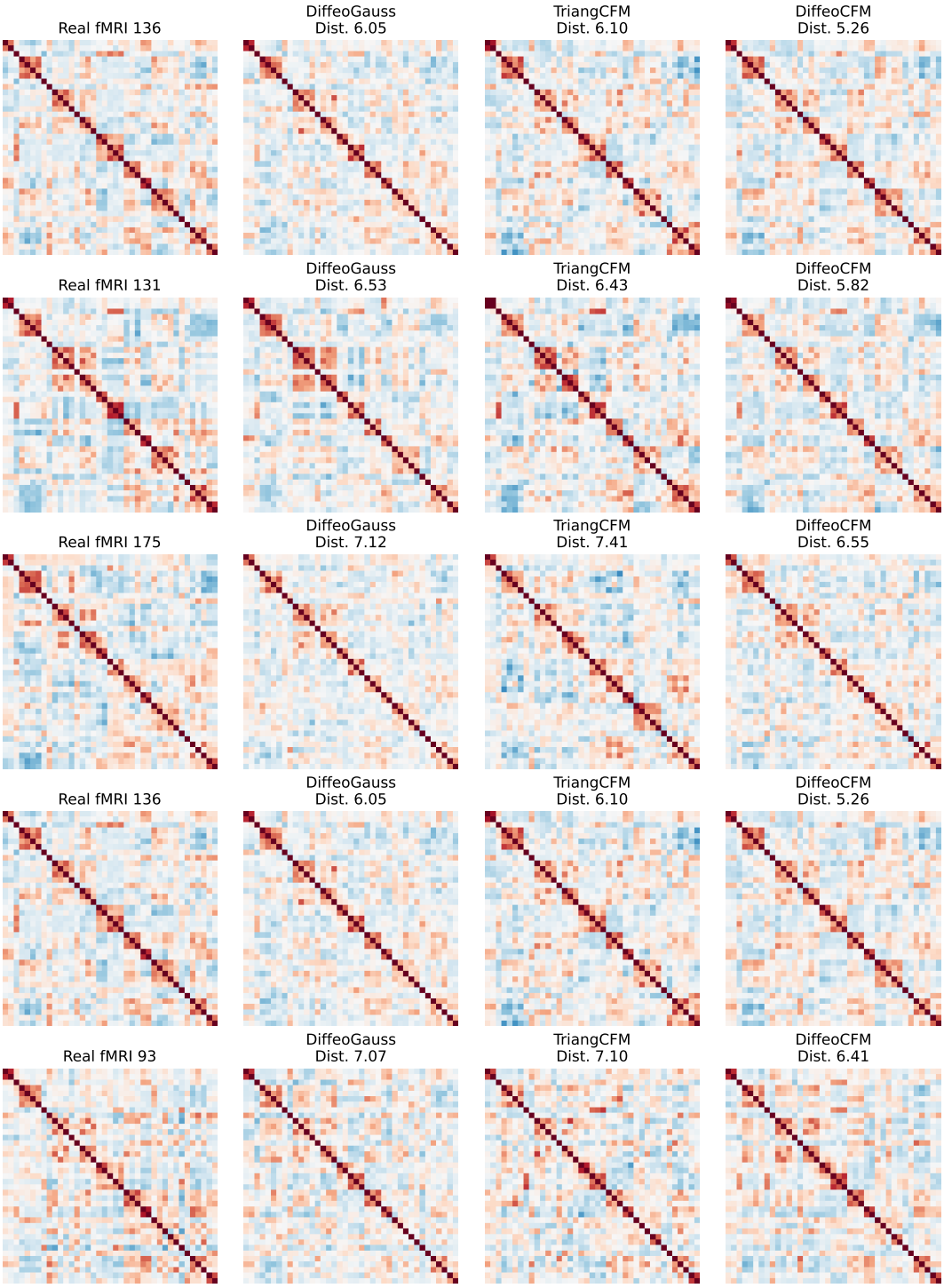


Figure 13: Nearest Generated Samples in Real Data Neighborhoods: ABIDE-nonCN Cohort. The matrices generated by TRIANGCFM shown in the figures are presented without applying the projection step.

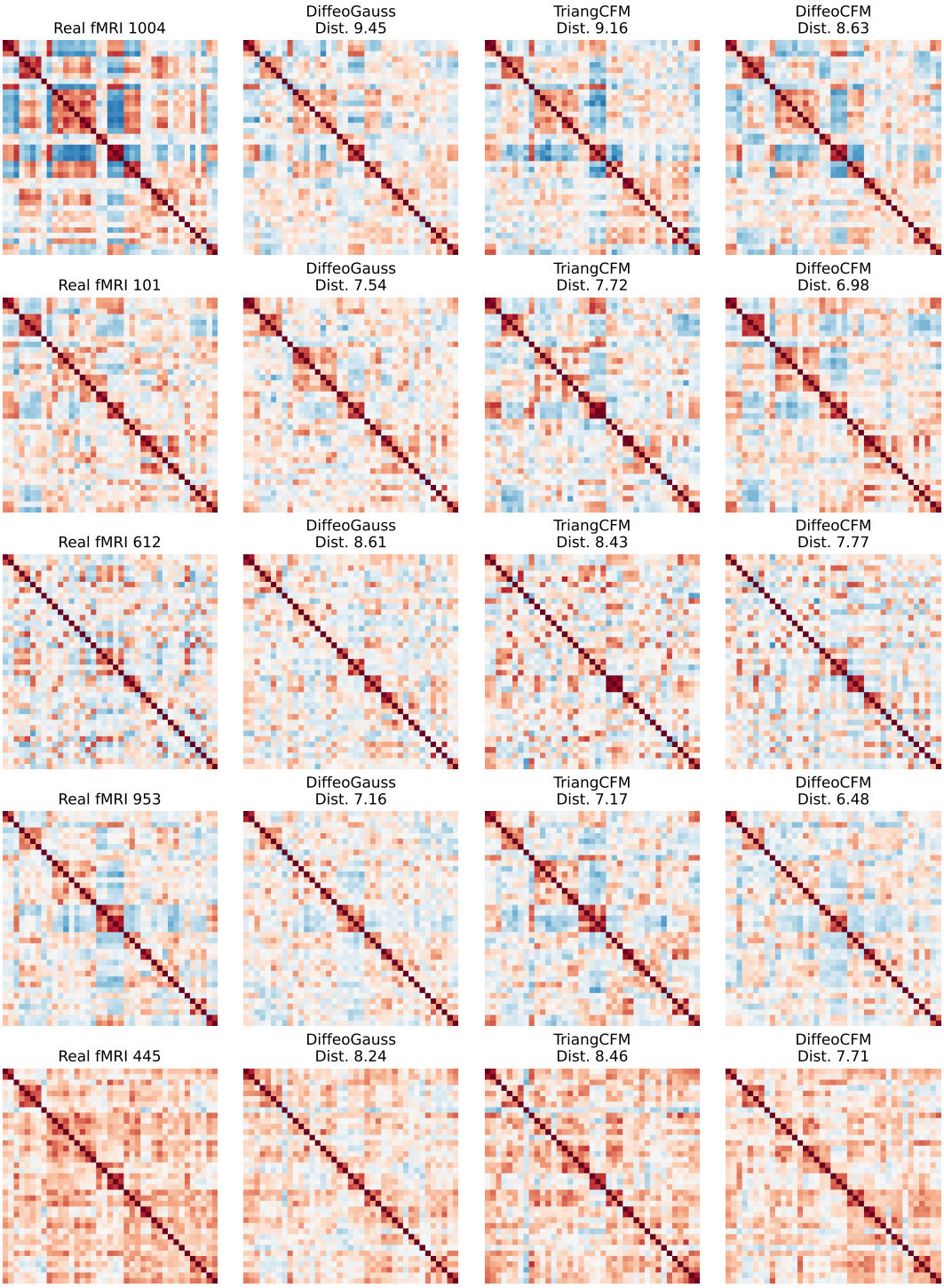


Figure 14: Nearest Generated Samples in Real Data Neighborhoods: OASIS3-CN Cohort. The matrices generated by TRIANGCFM shown in the figures are presented without applying the projection step.

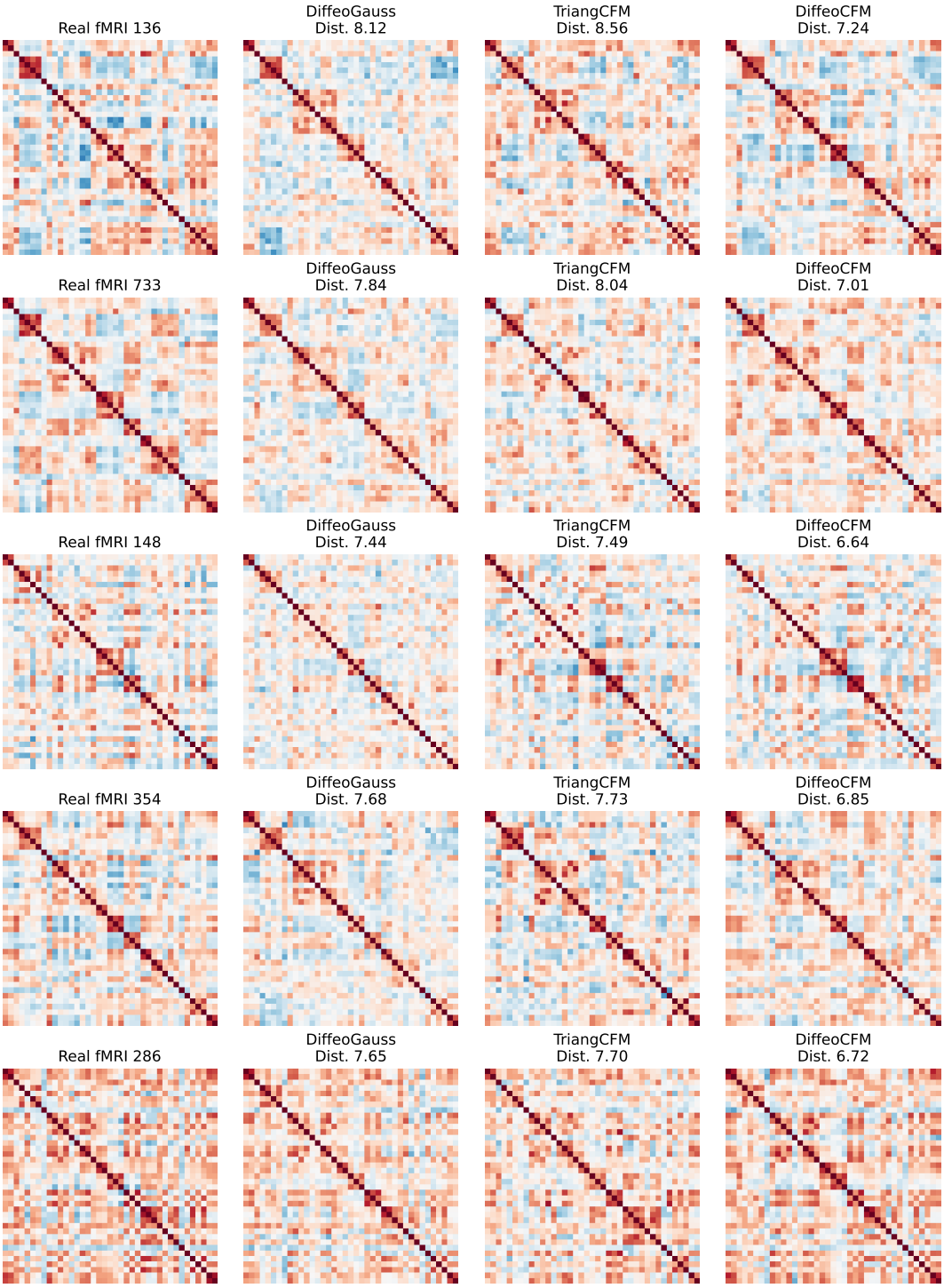


Figure 15: Nearest Generated Samples in Real Data Neighborhoods: OASIS3-nonCN Cohort. The matrices generated by TRIANGCFM shown in the figures are presented without applying the projection step.

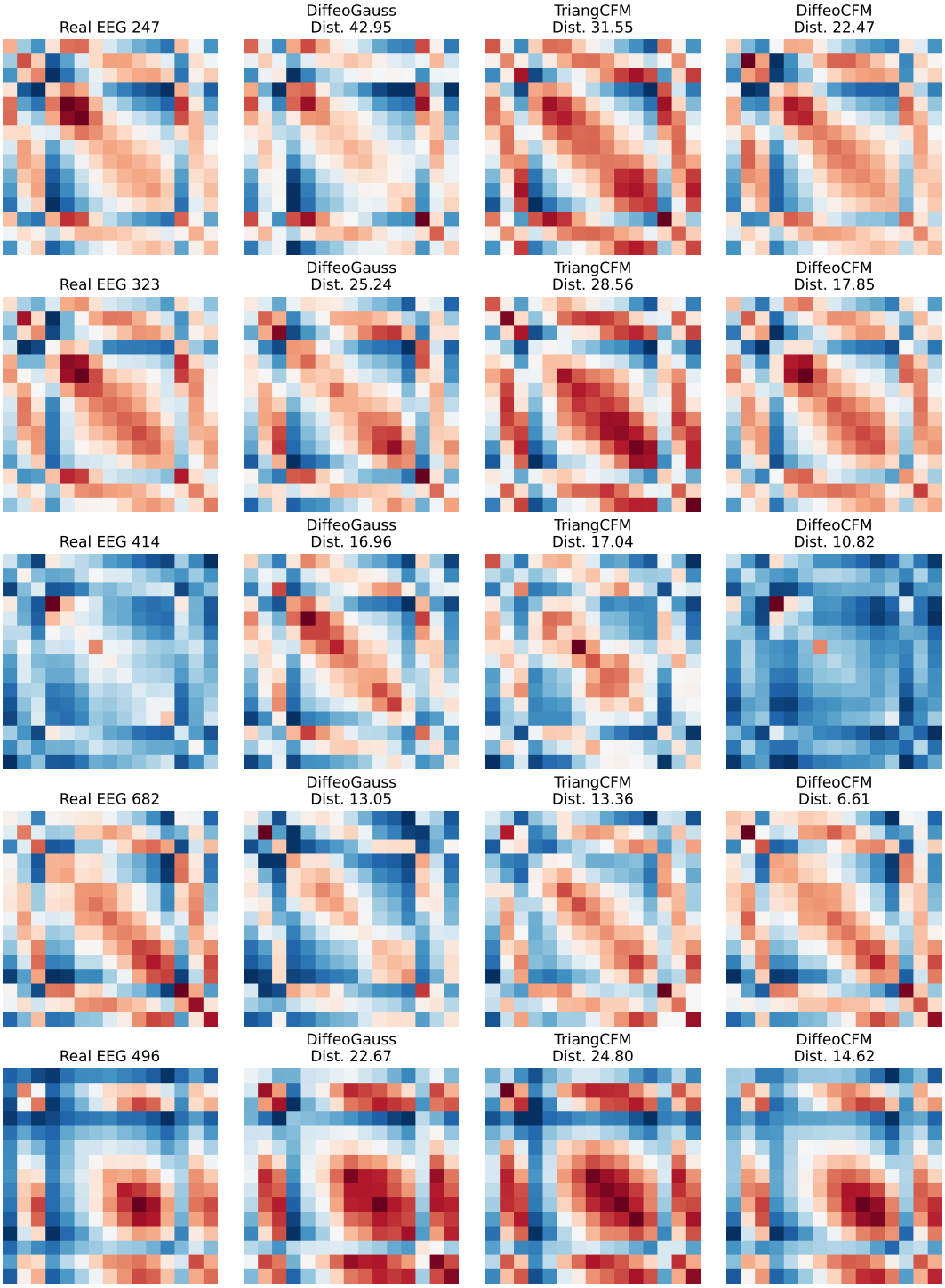


Figure 16: Nearest Generated Samples in Real Data Neighborhoods: BNCI2014-002-Right Hand.

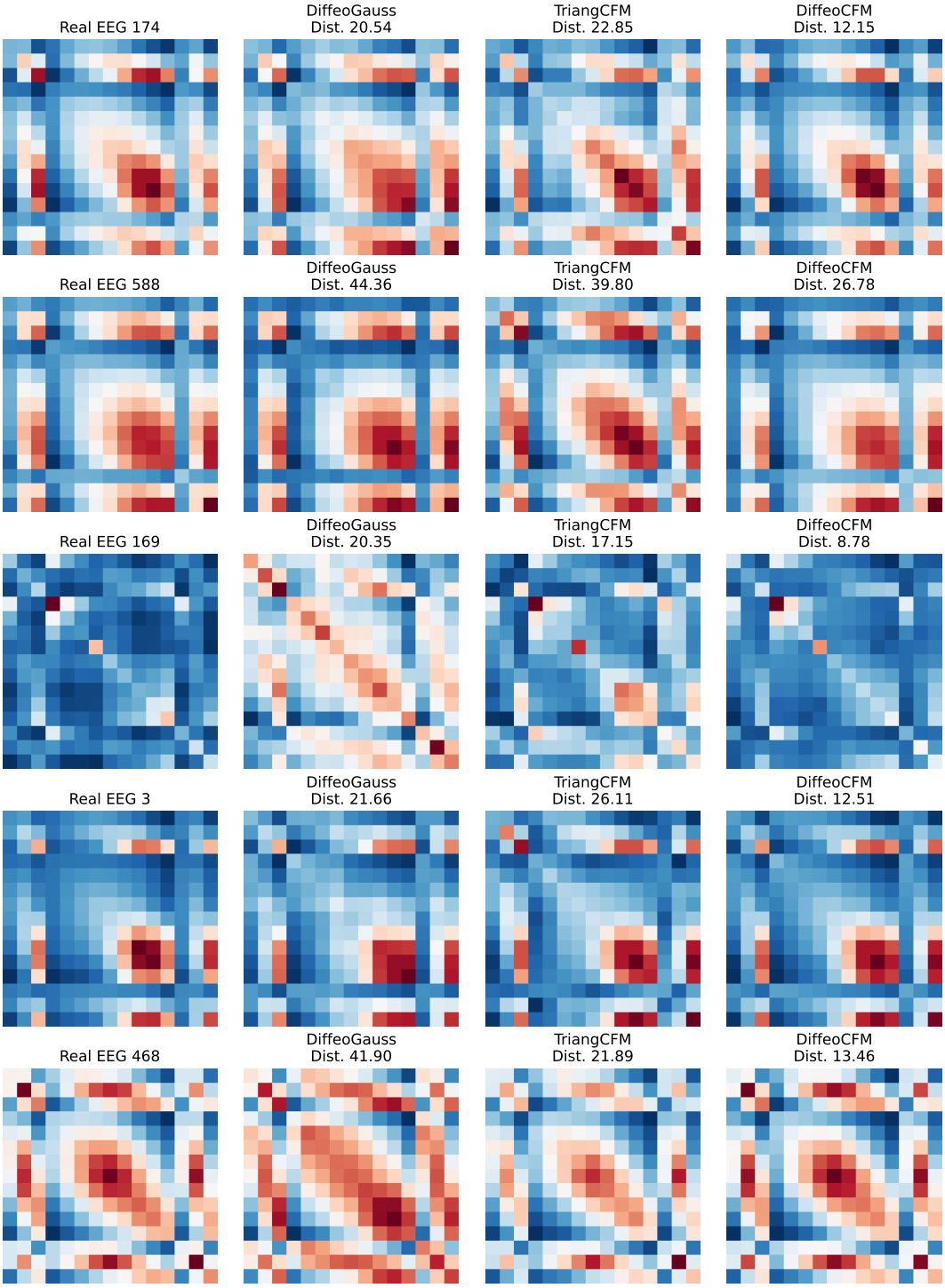


Figure 17: Nearest Generated Samples in Real Data Neighborhoods: BNCI2014-002-Feet.

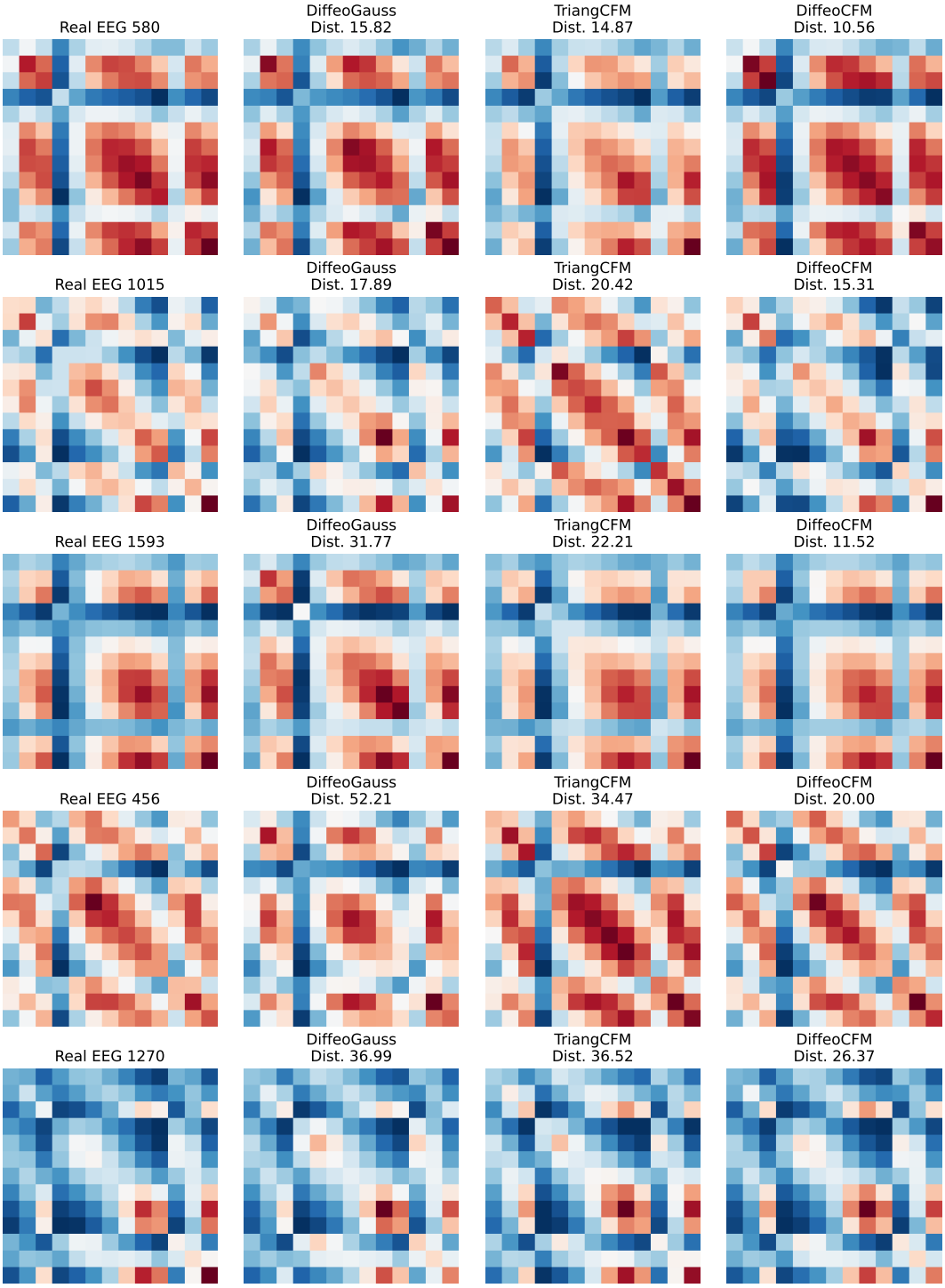


Figure 18: Nearest Generated Samples in Real Data Neighborhoods: BNCI2015-001-Right Hand.

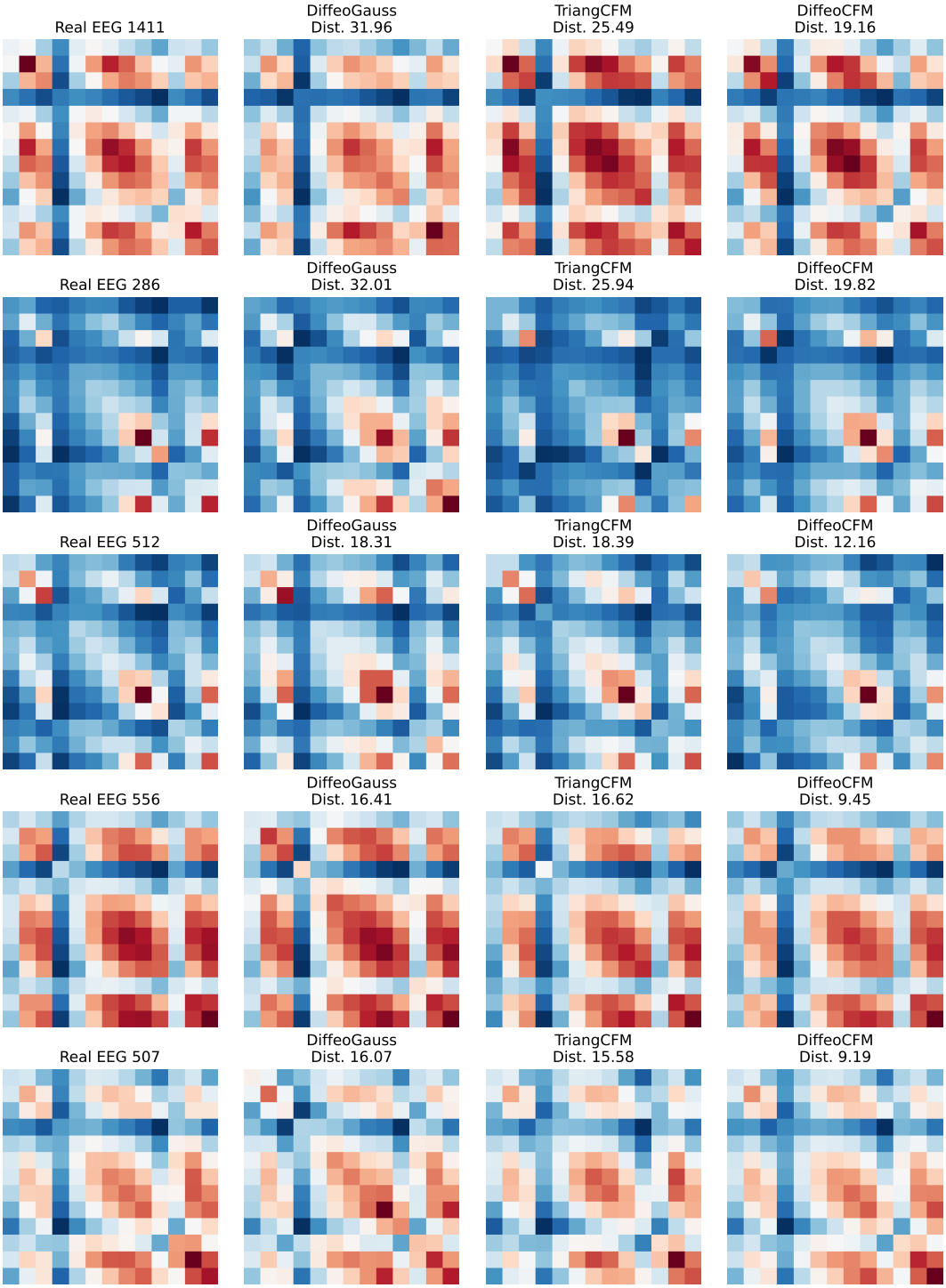


Figure 19: Nearest Generated Samples in Real Data Neighborhoods: BNCI2015-001-Feet.

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 Section 1 clearly state the core contributions—namely, conditional flow matching on pullback manifolds, its instantiation for SPD and correlation matrices, and a large-scale neuroimaging benchmark. These claims are substantiated theoretically in Section 3 and empirically in Section 4 and Section 5.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: A discussion of limitations is included in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: All propositions (i.e., Propositions 1, 2, 3) are clearly stated in Section 3. Each is accompanied by assumptions, and full proofs are deferred to the Appendix as recommended.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The datasets, preprocessing, model architecture, training procedure, and evaluation metrics are all detailed in Section 4, with full implementation and evaluation protocols described in the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

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

   Answer: [Yes]

   Justification: All datasets used in this study (ADNI, ABIDE, OASIS-3, BNCI2014-002, BNCI2015-001) are publicly available but fMRI ones (ADNI, ABIDE, OASIS-3) require registration or data use agreements. We do not redistribute these datasets, but we provide detailed instructions and references in the supplementary material to guide users on how to access and preprocess them using standard libraries (e.g., fMRIPrep, Nilearn). Anonymized code, including scripts for training, evaluation, and reproducing experimental results, is included in the supplementary material under a permissive open-source license.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All dataset splits, classifier architectures, hyperparameters, and evaluation protocols are specified in Section 4.2 and Section 4.1, with further detail provided in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Results in Table 1 are reported as mean $\pm$ std over multiple seeds/splits. The experimental setting in Section 4.2 explains variability factors, and significance trends are analyzed in Section 5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The Appendix details the compute setup used (cpu, gpu and time of execution).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: All datasets used are publicly available and widely used in the neuroscience community. No personally identifiable information is used. Anonymity and licensing are respected.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

Answer: [Yes]

Justification: We include a discussion of broader impacts of this study in Section 6.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [NA]

Justification: The models and data used in this work pose no identifiable risks for misuse. The generative models are restricted to structured brain connectivity matrices derived from neuroimaging datasets and are intended solely for research use. They do not produce personally identifiable data, and the underlying datasets (ADNI, ABIDE, OASIS-3, BNCI) are publicly released under controlled access and ethical review.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All datasets and baselines used are cited with appropriate references and comply with their respective licenses.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: No new datasets or pretrained models are introduced in the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

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

    Answer: [NA]

    Justification: No research with human subjects or crowdsourcing was conducted by the authors. All data come from existing, publicly released studies.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

Answer: [NA]

Justification: No new human subject data were collected. All datasets used had IRB approvals from their respective institutions.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: No LLMs were used in the development or core methods of this paper.

Guidelines:

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