LEARNING APPROXIMATE ISOMETRIC EMBEDDINGS FOR EFFICIENT TEMPLATE PLACEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

The sphere covering problem has broad applications in science however, exact solutions are computationally infeasible on curved, high-dimensional manifolds because distances must be evaluated using a position-dependent metric. We present `isobank`, a normalizing flow that learns an approximately isometric coordinate transformation, such that Euclidean distances in the latent space approximate metric-induced proper distances in the physical parameter space. This enables template placement using Euclidean distance checks in the latent space, avoiding repeated metric evaluations during placement.

We evaluate `isobank` in the context of gravitational wave template bank construction, where the sphere covering problem corresponds to the template placement problem. On a five-dimensional eccentric binary parameter space, `isobank` reaches 97% coverage at a maximum mismatch of $\mu = 0.03$ with 23% fewer templates than the state-of-the-art stochastic placement code `mbank`, which reaches 94% coverage. Coverage is measured on independent injections drawn uniformly over the masses, spins and eccentricity, using the exact waveform match rather than the metric approximation. Training the flow takes about 2.5h and placing the bank under 1h on 32 CPU cores. `isobank` will be made publicly available.

1 INTRODUCTION

Covering a manifold $\mathcal{B} \subset \mathbb{R}^D$ with a finite set of balls is a fundamental problem with broad applications across astronomy, engineering, and medicine (Elzinga & Hearn, 1972). The objective is to cover a given manifold $\mathcal{B}$ with a finite set of metric balls $b(\lambda_i) = \{\lambda \in \mathcal{B} : d_g(\lambda, \lambda_i) \leq r\}$ with radius r and distance metric d_g, such that every point $\lambda \in \mathcal{B}$ falls within at least one ball, subject to a maximum quantity of interest, *e.g.*, a measure of error or cost. The coverage or discretization of $\mathcal{B}$ naturally introduces a certain amount of error, necessitating a trade-off between the number of balls and computational cost: while a small number of metric balls $b(\lambda_i)$ will not cover $\mathcal{B}$ sufficiently to satisfy the maximum quantity of interest, a large number of $b(\lambda_i)$ will increase the computational cost, leading to the "curse of dimensionality" as the problem's dimension increases. Thus, the optimal solution will be composed of the smallest number of metric balls $b(\lambda_i)$ subject to a maximum quantity of interest. In this work, we seek an approximately optimal solution to the sphere covering problem in the context of gravitational wave (GW) template bank construction, which is commonly known as a *template placement* problem.

Modeled GW searches use the matched filtering method to cross-correlate detector data with a set of waveform models, known as *templates*, to detect expected GW signals. These templates can be generated using analytical and numerical techniques (Schmidt, 2020) and are parametrized by a continuous, high-dimensional parameter vector $\lambda \in \Lambda$ describing the physical properties of the GW source, such that the waveform manifold is defined as $\mathcal{B} := \{h(t; \lambda) \mid \lambda \in \Lambda\}$. Thus, the ith template is defined by its centroid λ_i, and the associated metric ball $b(\lambda_i)$ covers a region in $\mathcal{B}$. In this context, the "curse of dimensionality " arises from adding physical effects to maximize GW signal discovery, which increases the size of the template banks by many orders of magnitude. Moreover, finding the optimal solution is complex: the geometry of the waveform manifold $\mathcal{B}$ can be uneven, requiring more expensive distance evaluations and creating non-uniform regions that need different numbers of templates for adequate coverage.

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

To overcome these challenges, we present `isobank`: a template placement approach powered by normalizing flows. The normalizing flow learns a coordinate transformation of the waveform manifold $\mathcal{B}$ into a Euclidean latent space. In this space, the Euclidean distances approximate the proper distances[1] induced by the waveform metric at the length scales relevant to template placement. The learned map approximately flattens the metric of $\mathcal{B}$ across λ, reducing template placement to stochastic sampling with Euclidean distance checks in the latent space. This eliminates metric evaluations during placement and substantially reduces the computational cost of constructing the template bank.

2 BACKGROUND

Given a time series $s(t) = h(t;\lambda) + n(t)$, where $h(t;\lambda)$ is the unknown GW signal and $n(t)$ is the detector noise, the *matched filtering* method can "filter out " the detector noise $n(t)$ via cross-correlation with $s(t)$ and the optimal filter $K(t)$. This task is challenging because of the complexity of $n(t)$, but we can simplify the problem by approximating $n(t)$ as locally Gaussian and stationary. Thus, it can be shown that the optimal filter $K(t)$ is the template of the target GW signal $h_i \equiv h(t;\lambda_i)$ (Owen & Sathyaprakash, 1999; Allen et al., 2012).

The morphology of the template is defined, at least, by a set of fifteen $\lambda = (\phi, \theta)$ parameters. These parameters can be divided into *extrinsic* parameters ϕ, which depend on the orientation of the astronomical event with respect to an outside observer, and *intrinsic* parameters θ, which are the physical characteristics of the astronomical event that are independent of the observer's location. The geometric properties of the waveform manifold $\mathcal{B}$ are then described by its metric, which is naturally introduced by the matched filtering method. The similarity between two templates h_i and h_j is quantified in terms of the overlap $\mathcal{O}$ defined as,

$$\mathcal{O}(h_i, h_j) = (h_i|h_j) := 4\mathrm{Re}\int_{f_{\text{lower}}}^{f_{\text{upper}}} df \frac{\tilde{h}_i(f)\tilde{h}_j^*(f)}{S_n(f)}, \tag{1}$$

where $\tilde{\cdot}$ is the Fourier transform, * is the complex conjugate, the integral extends in $[f_{\text{lower}}, f_{\text{upper}}]$ Hz related to the detector's sensitivity, and $S_n(f)$ denotes the one-sided power spectral density of detector noise. It is customary to normalize the templates, so that $\mathcal{O} \in [0, 1]$, *i.e.* two identical templates yield $\mathcal{O} = 1$, while two orthogonal templates yield $\mathcal{O} = 0$. Given h_i and h_j parametrized by λ_i and $\lambda_j = \lambda_i + \Delta\lambda$, respectively, we can define the match $\mathcal{M}$ between them as the maximized overlap with respect to the extrinsic parameters ϕ such that $\mathcal{M}(\theta_i, \theta_i + \Delta\theta) = \max_{\phi,\Delta\phi} \mathcal{O}(\theta_i, \phi_i, \theta_i + \Delta\theta, \phi_i + \Delta\phi)$. In general, $\mathcal{M}$ is not symmetric and does not satisfy the triangular inequality, but it introduces a sense of "closeness" between two templates dependent on $\Delta\theta$. Trivially, we can define the *mismatch* as $\mu(\theta_i, \theta_i + \Delta\theta) = 1 - \mathcal{M}(\theta_i, \theta_i + \Delta\theta)$, which is the *quantity of interest* to limit when solving this problem. Now, we approximate the match $\mathcal{M}$ as a second-order Taylor expansion around $\Delta\theta = 0$.

$$\mathcal{M}(\theta, \theta + \Delta\theta) \approx 1 + \frac{1}{2}\frac{\partial^2 \mathcal{M}}{\partial\Delta\theta^i \partial\Delta\theta^j}\Delta\theta^i\Delta\theta^j \implies g_{ij}(\theta) := -\frac{1}{2}\left.\frac{\partial^2 \mathcal{M}}{\partial\Delta\theta^i \partial\Delta\theta^j}\right|_{\Delta\theta=0} \tag{2}$$

where $g(\theta) \equiv g_{ij}(\theta)$ is defined as the metric of the waveform manifold $\mathcal{B}$ over the intrinsic parameters θ. To evaluate the coverage of the template bank, it is customary to conduct a *simulated injection*[2] *campaign*: $h(\theta')$ waveforms are generated by sampling θ' source parameters from a prior distribution, and added to simulated detector noise[3]. To quantify the coverage, we compute the *fitting factor* (FF) for each source parameter θ' as,

$$\mathrm{FF}(\theta') = \max_{\theta\in\mathcal{B}} \mathcal{M}(\theta', \theta). \tag{3}$$

[1]Here, the proper distance refers to the geodesic distance induced by the positive definite waveform metric g, obtained by minimizing $\int_\gamma \sqrt{d\theta^\intercal g(\theta) d\theta}$ over curves γ connecting two points in $\mathcal{B}$.

[2]In GW data analysis, an injection is the result of adding a GW signal to detector noise.

[3]To generate the detector noise, we use the average power spectral density of LIGO Hanford during the third observing run (O3); `aligo_O3actual_H1.txt`.

For a chosen maximum mismatch μ, an injection is considered *recovered* if its fitting factor satisfies

$$\mathrm{FF}(\theta') \geq 1 - \mu, \tag{4}$$

and *missed* otherwise. We define the *coverage* of the template bank as the fraction of injections that are recovered.

3 Related work

A breadth of methods have been implemented in GW to solve the sphere covering problem, also known as the template placement problem. We can categorize them into four broad approaches:

- *Geometric:* it solves the *sphere covering problem* in the waveform manifold $\mathcal{B}$. Here, $g(\theta)$ is defined analytically using a second-order Taylor expansion, defining a regular lattice for placing templates in a locally flat space. This method struggles with waveform manifolds $\mathcal{B}$ that have intrinsic curvature, complex boundaries, or high dimensionality (Harry et al., 2014; Roulet et al., 2019; Phukon et al., 2025; Cabass et al., 2025).
- *Stochastic:* new templates are drawn from the parameter space Λ of interest, and are added to the template bank when they sufficiently differ from existing ones. This method evaluates the distance between templates; while it can be extended to arbitrary parameter spaces Λ, the cost per evaluation grows steeply with dimensionality and produces redundant template banks (Harry et al., 2009; Ajith et al., 2014; Kacanja et al., 2024).
- *Hybrid:* it employs geometric placement in regions of the parameter space Λ where it is most efficient, and stochastic placement where the latter provides improved coverage (Roy et al., 2017; 2019; Sharma et al., 2024).
- *Random:* new templates are drawn from a probability distribution proportional to the local metric volume element of the waveform manifold $\mathcal{B}$, reducing the computational cost of bank construction. Although random placement generally produces larger template banks than other approaches, it scales favorably to high-dimensional parameter spaces (Allen, 2022; Coogan et al., 2022; Schmidt et al., 2024).

The most widely used method in GW data analysis is stochastic template placement. However, this method is prohibitive for the large banks that high-dimensional spaces demand. `isobank` falls within the *geometric template placement* category, as the trained flow learns the coordinate transformation from the waveform manifold $\mathcal{B}$ to an approximately isometric space. In this way, the trained flow removes both obstructions at once: distances become Euclidean in the latent space, so each acceptance test reduces to a nearest-neighbor query in a k-d tree, and *no waveform or metric evaluations are performed during placement*. While similar work uses autoencoders (Cabass et al., 2025), `isobank` leverages the mathematical properties of normalizing flows for efficient template placement in higher-dimensional spaces. We compare `isobank` against `mbank` (Schmidt et al., 2024), which also uses normalizing flows but for metric volume sampling rather than learning a distance preserving representation. While `mbank` uses the flow to sample regions of high metric volume, `isobank` learns coordinates in which proper distances are approximated by Euclidean distances, enabling efficient nearest-neighbour template placement.

4 Methodology

4.1 Normalizing flows

Normalizing flows learn invertible transformations between coordinate spaces, typically parameterized by neural networks (Kobyzev et al., 2021). Consider a parameter vector $\theta \in \Lambda$, which parametrizes the waveform manifold $\mathcal{B}$, and a latent coordinate vector $z \in \mathbb{R}^D$ related by a bijective transformation $z = f_\psi(\theta)$, with Jacobian $J = \partial f_\psi / \partial \theta$. The change-of-variables formula relates probability densities according to $p_\theta(\theta) = p_z(f_\psi(\theta)) \left|\det J_\psi(\theta)\right|$. In conventional applications, this relation represents a complex probability distribution through a simple latent distribution and enables efficient sampling via the inverse transformation. Here, we instead exploit the invertibility and flexible nonlinear structure of normalizing flows to learn a coordinate transformation between the physical and latent spaces.

We adopt a RealNVP-based architecture (Dinh et al., 2017) featuring coupling layers that split the input in two: $z_a = \theta_a$, while $z_b = \theta_b \odot \exp[s_\psi(\theta_a)] + t_\psi(\theta_a)$, and θ_a parametrizes an affine transformation. Note that s_ψ and t_ψ are neural networks that determine the scaling and translation of θ_b, and $\odot$ denotes element-wise multiplication. The transformation is invertible, and its Jacobian is triangular and therefore has a tractable determinant. By cycling the masking pattern between coupling layers, every coordinate is both transformed and used as a conditioning variable.

4.2 Learning an approximately isometric embedding

This work aims to bypass repeated metric evaluations during template placement in highly curved parameter spaces. Here, the flow does not model a probability distribution or generate samples. Instead, the learned forward transformation f_ψ provides a coordinate map,

$$\mathcal{B} \subset \mathbb{R}^D \xrightarrow{f_\psi} z \in \mathbb{R}^D, \tag{5}$$

chosen such that Euclidean distances in the latent space approximate proper distances in the physical parameter space. For Euclidean latent distances to reproduce proper distances, the flat Euclidean line element must pull back to $g(\theta)$. Writing the differential as $dz = J_\psi d\theta$ requires $dz^\top dz = d\theta^\top (J_\psi^\top J_\psi)\, d\theta = d\theta^\top g(\theta)\, d\theta$, which holds for arbitrary displacements iif the network's Jacobian satisfies:

$$J_\psi^\top J_\psi = g(\theta). \tag{6}$$

Although this relation can always be satisfied locally at a single point, it cannot in general be realized by a single coordinate transformation of the same dimension over a curved region. Exact isometric embeddings can instead require a higher-dimensional Euclidean space (Nash, 1956).

Consequently, we settle for an approximate, scale-targeted embedding. We specifically optimize for distance preservation at the finite scales relevant for template placement. For two points θ_1 and θ_2, we define distance in the latent space as the Euclidean norm of their difference, $d_z(\theta_1, \theta_2) = \left\| f_\psi(\theta_1) - f_\psi(\theta_2) \right\|_2$, and compare it with their proper distance d_{proper} under the physical metric. Exact distance preservation at a given scale corresponds to $d_z/d_{\mathrm{proper}} = 1$.

4.3 Multi scale distance-matching loss

To achieve the desired performance, we need to enforce distance preservation at different scales. For this aim, the training objective is a weighted sum of four distance-matching terms[4], distinguished by the distribution from which training pairs are drawn, as summarized in Table 1.

$$\mathcal{L} = \mathcal{L}_{\mathrm{glob}} + \beta_1\, \mathcal{L}_{\mathrm{loc}} + \beta_2\, \mathcal{L}_{\mathrm{bnd}} + \beta_3\, \mathcal{L}_{\mathrm{edge}}, \tag{7}$$

Global term ($\mathcal{L}_{\mathrm{glob}}$)**:** Pairs (θ_1, θ_2) are drawn independently from the full parameter space, and we penalize the squared difference of log distances, $\mathcal{L}_{\mathrm{glob}} = \mathbb{E}\Big[\big(\log d_z - \log d_{\mathrm{proper}}\big)^2\Big]$, which is scale-invariant and therefore weights the (large) global distances on an equal footing with local ones. This term preserves the global ordering of distances and prevents the flow from collapsing distant regions of the manifold without competing with the local terms for accuracy at any particular scale.

Calibration: Before training, we determine the metric distance corresponding to the quantity of interest: the maximum mismatch μ. For sufficiently small parameter displacements, the waveform mismatch is related to the quadratic metric by $1 - \mathcal{M}(\theta, \theta + \Delta\theta) \simeq [\Delta\theta^\mathsf{T} g(\theta) \Delta\theta]/2 = d^2_{\mathrm{proper}}/2$. Thus, under the infinitesimal quadratic approximation, a target mismatch of $\mu = 0.03$ corresponds to a proper distance

$$d_{\mathrm{proper}} \simeq \sqrt{2\mu} \simeq 0.245. \tag{8}$$

At the finite separations relevant for template placement, however, the quadratic metric approximation does not accurately reproduce the true waveform mismatch. We therefore empirically calibrate the target placement distance by generating metric-guided pairs over a range of separations and evaluating their true waveform matches. We define r_{target} as the proper distance at which the mean true waveform mismatch reaches $\mu = 0.03$.

[4] $\beta_1 = 5, \beta_2 = 10, \beta_3 = 10$ are coefficients obtained empirically to balance the loss function.

Table 1: Number of pairs computed for a given scale α, and anchor point θ_1 for each loss term.

Term	*Number of Pairs*	α *scale*	θ_2 constraints
Global $\mathcal{L}_{\text{glob}}$	10^4	-	Full parameter space
Local $\mathcal{L}_{\text{loc}}$	5×10^3	$[0.1, 2.5]$	Full parameter space
Boundary $\mathcal{L}_{\text{bnd}}$	10^4	$[0.6, 1.4]$	Full parameter space
Edge $\mathcal{L}_{\text{edge}}$	10^4	$[0.6, 1.4]$	Eccentricity $e \geq 0.12$

Metric-guided local pairs: The remaining terms use pairs constructed at controlled separations relevant for template placement. Given an anchor point θ_1, we draw a direction v uniformly from the Euclidean unit sphere, as $v \sim \mathcal{U}(S^{d-1})$, and a dimensionless scale factor α. Since the physical metric is generally anisotropic, the same coordinate displacement corresponds to different metric distances depending on its direction. We therefore normalize v by its local metric norm $|v|_g$ to ensure consistent physical spacing, generating the second point as,

$$\theta_2 = \theta_1 + \alpha r_{\text{target}} \frac{v}{|v|_g} = \theta_1 + \alpha r_{\text{target}} \frac{v}{\sqrt{v^\top g(\theta_1) v}}. \tag{9}$$

This construction ensures that $\Delta\theta^\top g(\theta_1)\Delta\theta = (\alpha r_{\text{target}})^2$, so the local quadratic metric assigns the displacement the same length, αr_{target}, regardless of direction v. The resulting pair is therefore generated at approximately the desired metric scale; we then compute the actual proper distance between the two points from the metric and use it as the training target.

The metric-guided local pairs use a relative (ratio) loss with *adaptive per-pair weights* γ_i,

$$\mathcal{L}_\bullet = \frac{1}{N} \sum_{i=1}^{N} \gamma_i \, (\frac{d_{z,i}}{d_{\text{proper},i}} - 1)^2. \tag{10}$$

Since not all sampled pairs contribute equally to the final coverage quality, we reweight the loss by the current embedding error: at every gradient step, the worst fitting fraction of the batch, ranked by $|d_{z,i} - d_{\text{proper},i}|/d_{\text{proper},i}$, receives an enhanced weight while the rest keep unit weight. Early in training the errors are large and roughly uniform, so the weighting is nearly inactive; as the fit converges it concentrates effort on the genuinely hard pairs, predominantly those near the boundary of the parameter space. Moreover, normalizing Eq. 10 by d_{proper} places all sampled scales on an equal footing; an unnormalized loss $(d_z - d_{\text{proper}})^2$ would be dominated by the largest separations. In the following, we describe the different metric-guided terms of Eq. 10:

(i) *Local term* ($\mathcal{L}_{\text{loc}}$)*:* $\alpha \sim \mathcal{U}[0.1, , 2.5]$, spanning separations from $0.1, r_{\text{target}}$ to $2.5, r_{\text{target}}$; $\alpha = 1$ corresponds to the covering radius itself.

(ii) *Boundary term* ($\mathcal{L}_{\text{bnd}}$)*:* Identical to the $\mathcal{L}_{\text{loc}}$ but with $\alpha \sim \mathcal{U}[0.6,\ 1.4]$, densely sampling the window of separations that directly determines whether a point is covered by its nearest template. Its weight in Eq. 7 is twice that of the local term, reflecting its direct impact on coverage.

(iii) *Edge term* ($\mathcal{L}_{\text{edge}}$)*:* Identical to $\mathcal{L}_{\text{bnd}}$, but with anchors θ_1 drawn from a hard subregion that is underrepresented with the uniform anchor sampling. Note that for our *toy example*, the $\mathcal{L}_{edge}$ is not used since the difficulty lies in the global anisotropy covered by the $\mathcal{L}_{loc}$ and $\mathcal{L}_{bnd}$. For our *high-dimensional example*, we introduce the $\mathcal{L}_{edge}$ term and sample the high-eccentricity subregion $e \geq 0.12$, where the local geometry is most degenerate. Without this term, sparse edge regions dominate the resulting bank's coverage failures.

4.4 Training strategy

The complete training workflow consists of calibrating the target placement radius, generating metric-guided training pairs, optimizing the flow, and selecting the model. In the following, we provide details on this workflow for reproducibility.

Pair generation: We generate all training pairs before optimization, so waveform dependent computations are confined to preprocessing. We summarize in Table 1 the number of pairs for each term

Table 2: Description of the parameter spaces.

Space	*Parameters*	*D*
Toy example: aligned-spin	$M \in [30, 50]\, M_\odot$ $q \in [1, 5]$ $\chi \in [-0.99, 0.99]$ `IMRPhenomD` (Husa et al., 2016)	3
High-dimensional example: eccentric	$m_1 \in [1, 7]\, M_\odot$ $m_2 \in [1, 3]\, M_\odot$ $s_{1z} \in [-0.9, 0.9]$ $s_{2z} \in [-0.05, 0.05]$ $e \in [10^{-5}, 0.15]$ `TaylorF2Ecc` (Moore et al., 2016)	5

of the training objective defined in Eq. 7. For the global term, pairs are drawn independently from the full parameter space; since such sampling produces few pairs at placement relevant separations, the metric-guided local pairs are instead generated using Eq. 9. For each pair, we evaluate the proper distance under the metric once and store it for subsequent training.

Optimization and model description: We train the flow for 10^3 epochs using Adam (Kingma & Ba, 2014) with an initial learning rate of 5×10^{-4}, cosine annealing (Loshchilov & Hutter, 2016) to 10^{-6}, and gradient norm clipping. The model minimizes the weighted sum in Eq. 7. We use a RealNVP-type flow (Dinh et al., 2017) with nine affine coupling layers whose binary masks cycle through three patterns arranged so that every coordinate is both transformed and used as a conditioning variable within each three-layer cycle. Each coupling transformation is parametrized by a single hidden layer network with 256 neurons and a $\tanh$ activation function. The flow required approximately 2.5 hours to train on 32 CPU cores, and template placement less than one hour.

Template Placement Algorithm: With the trained flow, we perform template placement using Euclidean distances in the latent space, as summarized in Algorithm 1 in Sec. 8.1. We initialise an empty template bank $\hat{\mathcal{B}}$ and repeatedly draw batches of candidate points θ_i from a proposal distribution $q(\theta)$ over the physical parameter space. Each candidate is mapped to latent space, $z_i = f_\psi(\theta_i)$, and accepted if its nearest existing template is farther than the target placement radius,

$$\min_{z_j \in \hat{\mathcal{B}}} \|z_i - z_j\| > r_{\text{target}}.$$

Nearest-neighbour queries are performed using a k-d tree constructed from the latent coordinates of the current bank, avoiding explicit pairwise comparisons with all existing templates.

Placement is terminated when the estimated fraction of uncovered metric volume falls below a prescribed tolerance ε. The volume estimate uses the Jacobian determinant of the flow as an approximate metric-volume weight, with importance weighting to account for the proposal distribution. This criterion is only a stopping condition; coverage is assessed independently with true waveform matches. In practice we operate at a coarser effective radius than r_{target}, trading bank size against coverage (Appendix 8.1).

5 Data set

To test the effectiveness of our approach, we initially apply it to a *toy example* to learn a three-dimensional *aligned-spin* space, representative of binary black hole (BBH) systems, parameterized by the total mass M, mass ratio q, and spin parameter χ, and modeled by `IMRPhenomD` (Husa et al., 2016), as shown in Table 2.

We then apply our method to a more challenging five-dimensional *eccentric aligned-spin* parameter space covering both binary neutron star (BNS) and neutron star-black hole (NSBH) systems. The parameter space is defined by the component masses m_1 and m_2, the aligned spin projections s_{1z}

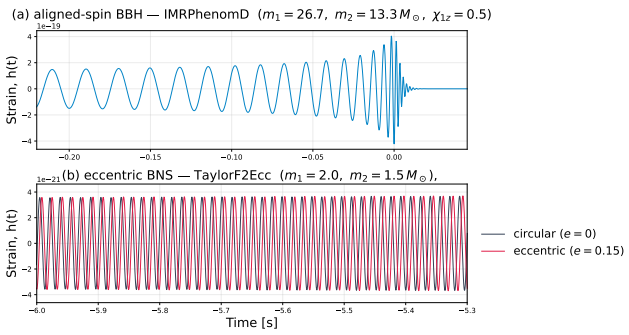


Figure 1: *(Top panel)* The aligned-spin case with the `IMRPhenomD` waveform model showing inspiral, merger and ringdown phase. *(Bottom panel)* Eccentric case with the `TaylorF2Ecc` inspiral waveform. Demonstrating the case for an inspiraling binary neutron star (BNS) signal with $e = 0$ and $e = 0.15$ to show that eccentric imprint is a slow phase drift rather than a merger feature.

and s_{2z}[5], and the eccentricity e, using the model `TaylorF2Ecc` (Moore et al., 2016). Eccentricity adds variety and structure to the waveform parameter and a strongly curved waveform manifold metric.

BNS are expected to undergo significant orbital circularization before entering the sensitive band of GW detectors, so they have low eccentricities by the time they become detectable. Following Phukon et al. (2025), we choose $e_{\max} = 0.15$ as a conservative upper bound, encompassing systems with significant residual eccentricity while remaining within the moderate eccentricity regime considered relevant for low-mass compact binaries. The computational challenge associated with eccentricity is illustrated in Lenon et al. (2021), which demonstrated the feasibility of constructing stochastic template banks for eccentric binaries for Cosmic Explorer. They found that increasing the maximum eccentricity from $e \leq 0.05$ to $e \leq 0.1$ approximately doubled the bank size to 4.5 million templates.

6 Results

6.1 Distance preservation validation

We evaluate several diagnostics of the assumption that the trained flow has learned an approximate isometry. The most direct one is the comparison of physical proper distances against latent distances for pairs of points not used in training. For a perfect isometry, all pairs should lie on the diagonal represented as a black dashed line in Fig. 2. The physical coordinates carry almost no distance information: even after optimal global rescaling, coordinate distances are essentially uncorrelated with the proper distance and would therefore appear as a scattered cloud rather than along the diagonal. On the left panel of Fig. 2, we show that the same pairs collapse onto the diagonal after the flow has learned the high dimensional problem of the eccentric space. Note that the vertical red line marks the calibrated r_{target}, below which the scale is relevant for template placement.

6.2 Flow transformation

Another test of our method's feasibility is to visualize how the flow transforms the local geometry of the parameter space. Around a set of anchor points, we construct *unit mismatch shells*, points lying at a fixed proper distance r_{target} from each anchor point, and map them through the flow. For ease of visualization, we show this in Fig. 2 only for the 3D aligned-spin space, whose three dimensions can be displayed directly. The same construction extends to the 5D eccentric space. We translate each shell so that its anchor lies at a common origin. In the physical coordinates, equal proper distance corresponds to highly anisotropic coordinate displacements, producing elongated shells. Under the learned transformation, these same points are mapped to approximately spherical shells with radii close to r_{target}, as shown in the right panel. Thus, the flow transforms the strongly anisotropic local geometry induced by the waveform metric into approximately Euclidean geometry, such that proper distances relevant for template placement are represented by ordinary Euclidean distances in the

[5]Spins are defined along the orbital angular momentum ($\hat{z}$) axis. The *toy example* parameterizes spin as a 1D scalar χ (a weighted linear combination of binary spins), while the *high-dimensional* example retains the full 2D component spin parameterization with s_{1z} and s_{2z}.

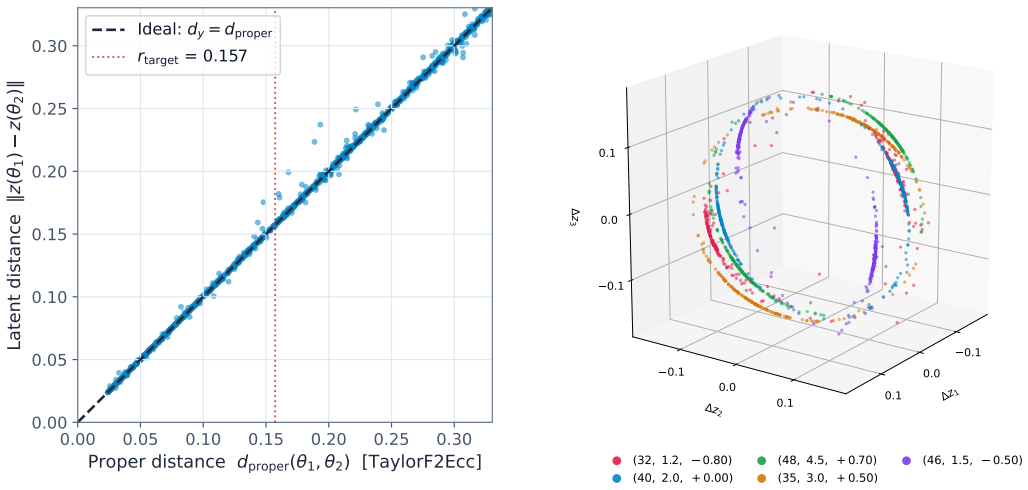


Figure 2: *Left:* Distance preservation evaluated on the eccentric 5D space on test pairs generated independently of the training set, showing proper distance against the Euclidean distance in the flow's latent space. The black dashed diagonal marks exact isometry. *Right:* Transformation of fixed proper distance mismatch shells into the flow's latent space for the 3D aligned-spin parameter space. Points lying at a fixed proper distance r_{target} from each anchor point become approximately spherical, illustrating the approximate isometry of the learned representation.

latent space. This approximate isometry allows template placement and nearest-template searches to use standard Euclidean operations without evaluating the metric.

6.3 Effectualness of the template bank

In Fig. 3, we compare the performance of `isobank` with respect to `mbank` (Schmidt et al., 2024) for the aligned-spin and eccentric spaces using the O3 PSD, with 10^4 injections in each space. Each curve shows the cumulative fraction of injections with a fitting factor below a given value. It is standard to set a threshold at FF $= 0.97$ (Owen & Sathyaprakash, 1999), so the height of each curve at this threshold is the fraction of missed signals. A curve that stays low until close to FF $= 1$ indicates good coverage, whereas a long tail towards low FF reveals regions where the bank matches signals poorly.

In the aligned-spin example, both codes perform equally well: `mbank` covers $99.80 \pm 0.04\%$ of the injections with 23,167 templates, and `isobank` covers $99.72 \pm 0.05\%$ with 23,810 templates. In the challenging eccentric example, the `mbank` bank of 2.95M templates, generated with its default settings at $\mu = 0.03$, covers $94.2 \pm 0.2\%$ of the injections and shows a long tail below FF $= 0.97$, reaching FF $= 0.61$ for the worst injection. `isobank` covers $96.6 \pm 0.2\%$ with 2.27M templates, 23% fewer, and its worst injection has FF $= 0.93$. `isobank` therefore achieves higher coverage with fewer templates in the challening eccentric space.

As we present in Table 3, `isobank` misses (does not recover) $\sim 3.36\%$ of the joint BNS and NSBH parameter space for the eccentric example, while in Phukon et al. (2025), they miss 5.68% of signals with 4.78M templates. However, this is not a complete comparison, as they use a different detector noise for their injection campaign: the PSD of the fourth observing run (O4). We leave such a comparison for future work. We also constructed an `isobank` bank for the parameter space of the O3 eccentric search of Dhurkunde & Nitz (2025) at the same minimal match of 0.95, obtaining 4.46M templates that recover 99.6% of injections at FF ≥ 0.95. However, their bank also spans a source-orientation parameter that ours does not, so this comparison is indicative only and we leave an apples-to-apples comparison for future work.

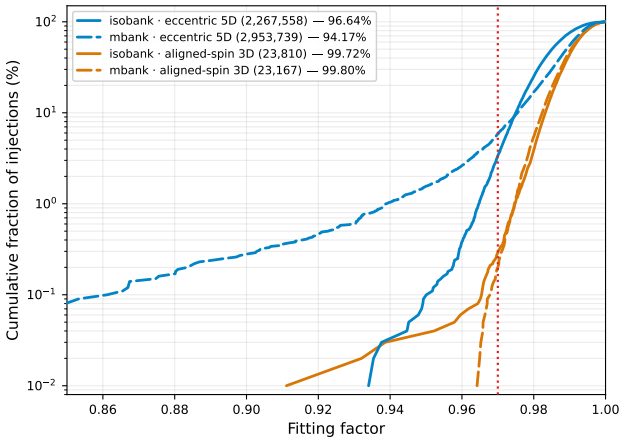


Figure 3: Cumulative distribution of fitting factors for 10^4 injections per parameter space, recovered with `isobank` (solid) and `mbank` (dashed) banks in the 3D aligned-spin (orange) and 5D eccentric (blue) spaces, using the O3 PSD. The red dotted line marks FF $= 0.97$; the height of each curve there is the fraction of missed injections. The legend gives each bank's size (templates) and coverage (%). Curves that stay low until close to FF $= 1$ indicate good coverage, whereas long tails towards low FF reveal regions where the bank matches signals poorly.

Table 3: Percentage of the missed (not recovered) injections by `isobank` that yield FF< 0.97 for different simulation campaigns.

Parameter space	Bank size	FF < 0.97 [%]		
		BNS	NSBH	All
Toy example: 3D aligned-spin (BBH)	23,810	–	–	0.28 ± 0.05
High-dimensional example: 5D eccentric	2.27M	3.90 ± 0.27	3.80 ± 0.27	3.36 ± 0.18

7 Conclusions

In this work, we address the sphere covering problem in the context of GW template bank construction or template placement. For this, we develop `isobank`, whose central idea is to learn a coordinate transformation, using normalizing flows, that approximately preserves the physical geometry of the GW waveform manifold. The flow is trained using the waveform metric at multiple distance scales, producing an approximately Euclidean latent representation in which Euclidean distances approximate proper distances on the physical manifold. Stochastic template placement can then use Euclidean nearest-neighbor queries, without repeated metric evaluations during placement.

We demonstrate that `isobank` can efficiently construct template banks in a challenging five-dimensional eccentric parameter space. At a maximum mismatch of $\mu = 0.03$, `isobank` achieves 96.6% coverage while using 23% fewer templates than the state-of-the-art template placement code `mbank`, which achieves 94.2% coverage. These results demonstrate that learning an approximately isometric representation can enable more computationally efficient placement while achieving higher coverage with fewer templates for high-dimensional waveform manifolds.

A natural extension of this work would be to apply `isobank` to higher-dimensional waveform models incorporating additional physical effects such as precession, higher-order modes, and matter effects. The computational advantage of the learned coordinate transformation is expected to become increasingly relevant as the dimensionality and complexity of the waveform manifold increase, potentially enabling more efficient searches for signals carrying new information about their astrophysical origins.

The present construction uses a flow with the same dimensionality as the physical parameter space and therefore cannot, in general, provide a global exact isometric map for a curved manifold. As noted in Schmidt et al. (2024), an interesting direction would be to investigate embeddings into a higher dimensional Euclidean space, where the additional dimensions may provide greater flexibility for representing the intrinsic geometry. Such an extension could also open the possibility of studying whether the resulting representation admits useful generative or sampling procedures.

AI DISCLOSURE SECTION

In this work, we used generative AI tools to implement methods as well as to design and provide feedback on numerical experiments. We have not used generative AI tools for translation or for qualitative and thematic data analysis, and the remaining tasks requiring disclosure are not applicable to this work. Additionally, we used generative AI tools to create and modify figures and to edit parts of the manuscript. We have reviewed all AI-assisted work: AI-generated code was verified and tested for correctness by one author. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

P. Ajith, N. Fotopoulos, S. Privitera, A. Neunzert, and A. J. Weinstein. Effectual template bank for the detection of gravitational waves from inspiralling compact binaries with generic spins. *Phys. Rev. D*, 89(8):084041, 2014. doi: 10.1103/PhysRevD.89.084041.

Bruce Allen. Performance of random template banks. *Phys. Rev. D*, 105:102003, May 2022. doi: 10.1103/PhysRevD.105.102003. URL https://link.aps.org/doi/10.1103/PhysRevD.105.102003.

Bruce Allen, Warren G. Anderson, Patrick R. Brady, Duncan A. Brown, and Jolien D. E. Creighton. FINDCHIRP: An Algorithm for detection of gravitational waves from inspiraling compact binaries. *Phys. Rev. D*, 85:122006, 2012. doi: 10.1103/PhysRevD.85.122006.

Giovanni Cabass, Digvijay Wadekar, Matias Zaldarriaga, and Zihan Zhou. Generating optimal Gravitational-Wave template banks with metric-preserving autoencoders. 11 2025.

Adam Coogan, Thomas D. P. Edwards, Horng Sheng Chia, Richard N. George, Katherine Freese, Cody Messick, Christian N. Setzer, Christoph Weniger, and Aaron Zimmerman. Efficient gravitational wave template bank generation with differentiable waveforms. *Phys. Rev. D*, 106(12): 122001, 2022. doi: 10.1103/PhysRevD.106.122001.

Rahul Dhurkunde and Alexander H. Nitz. Search for eccentric NSBH and BNS mergers in the third observing run of Advanced LIGO and Virgo. *Phys. Rev. D*, 111(10):103018, 2025. doi: 10.1103/PhysRevD.111.103018.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations (ICLR)*, 2017.

D. Jack Elzinga and Donald W. Hearn. The minimum covering sphere problem. *Management Science*, 19(1):96–104, 1972. ISSN 00251909, 15265501. URL http://www.jstor.org/stable/2629114.

I. W. Harry, B. Allen, and B. S. Sathyaprakash. Stochastic template placement algorithm for gravitational wave data analysis. *Physical Review D*, 80(10), November 2009. ISSN 1550-2368. doi: 10.1103/physrevd.80.104014. URL http://dx.doi.org/10.1103/PhysRevD.80.104014.

Ian W. Harry, Alexander H. Nitz, Duncan A. Brown, Andrew P. Lundgren, Evan Ochsner, and Drew Keppel. Investigating the effect of precession on searches for neutron-star-black-hole binaries with Advanced LIGO. *Phys. Rev. D*, 89(2):024010, 2014. doi: 10.1103/PhysRevD.89.024010.

S. Husa et al. Frequency-domain gravitational waves from nonprecessing black-hole binaries. i. new numerical waveforms and anatomy of the signal. *Phys. Rev. D*, 93:044006, Feb 2016. doi: 10.1103/PhysRevD.93.044006. URL https://link.aps.org/doi/10.1103/PhysRevD.93.044006.

Keisi Kacanja, Alexander H. Nitz, Shichao Wu, Marco Cusinato, Rahul Dhurkunde, Ian Harry, Tito Dal Canton, and Francesco Pannarale. Efficient Stochastic Template Bank Using Inner Product Inequalities. *Astrophys. J.*, 975(2):212, 2024. doi: 10.3847/1538-4357/ad7d87.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 12 2014.

Ivan Kobyzev, Simon J. D. Prince, and Marcus A. Brubaker. Normalizing Flows: An Introduction and Review of Current Methods. *IEEE Trans. Pattern Anal. Machine Intell.*, 43(11):3964–3979, 2021. doi: 10.1109/tpami.2020.2992934.

Amber K. Lenon, Duncan A. Brown, and Alexander H. Nitz. Eccentric binary neutron star search prospects for cosmic explorer. *Physical Review D*, 104(6), 2021. ISSN 2470-0029. doi: 10.1103/physrevd.104.063011. URL http://dx.doi.org/10.1103/PhysRevD.104.063011.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. 8 2016.

Blake Moore, Marc Favata, K. G. Arun, and Chandra Kant Mishra. Gravitational-wave phasing for low-eccentricity inspiralling compact binaries to 3pn order. *Phys. Rev. D*, 93:124061, Jun 2016. doi: 10.1103/PhysRevD.93.124061. URL https://link.aps.org/doi/10.1103/PhysRevD.93.124061.

John Nash. The imbedding problem for riemannian manifolds. *Annals of Mathematics*, 63(1):20–63, 1956.

Benjamin J. Owen and B. S. Sathyaprakash. Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement. *Phys. Rev. D*, 60:022002, 1999. doi: 10.1103/PhysRevD.60.022002.

Khun Sang Phukon, Patricia Schmidt, and Geraint Pratten. Geometric template bank for the detection of spinning low-mass compact binaries with moderate orbital eccentricity. *Physical Review D*, 111(4), February 2025. ISSN 2470-0029. doi: 10.1103/physrevd.111.043040. URL http://dx.doi.org/10.1103/PhysRevD.111.043040.

Javier Roulet, Liang Dai, Tejaswi Venumadhav, Barak Zackay, and Matias Zaldarriaga. Template bank for compact binary coalescence searches in gravitational wave data: A general geometric placement algorithm. *Phys. Rev. D*, 99:123022, Jun 2019. doi: 10.1103/PhysRevD.99.123022. URL https://link.aps.org/doi/10.1103/PhysRevD.99.123022.

Soumen Roy, Anand S. Sengupta, and Nilay Thakor. Hybrid geometric-random template-placement algorithm for gravitational wave searches from compact binary coalescences. *Phys. Rev. D*, 95:104045, May 2017. doi: 10.1103/PhysRevD.95.104045. URL https://link.aps.org/doi/10.1103/PhysRevD.95.104045.

Soumen Roy, Anand S. Sengupta, and Parameswaran Ajith. Effectual template banks for upcoming compact binary searches in Advanced-LIGO and Virgo data. *Phys. Rev. D*, 99(2):024048, 2019. doi: 10.1103/PhysRevD.99.024048.

Patricia Schmidt. Gravitational Waves From Binary Black Hole Mergers: Modeling and Observations. *Front. Astron. Space Sci.*, 7:28, 2020. doi: 10.3389/fspas.2020.00028.

Stefano Schmidt, Bhooshan Gadre, and Sarah Caudill. Gravitational-wave template banks for novel compact binaries. *Phys. Rev. D*, 109(4):042005, 2024. doi: 10.1103/PhysRevD.109.042005.

Abhishek Sharma, Soumen Roy, and Anand S. Sengupta. Template bank to search for exotic gravitational wave signals from astrophysical compact binaries. *Phys. Rev. D*, 109:124049, Jun 2024. doi: 10.1103/PhysRevD.109.124049. URL https://link.aps.org/doi/10.1103/PhysRevD.109.124049.

8 Appendix

8.1 Methodological details

In Section 4.4, we described the template placement procedure. Here, we provide its algorithmic form and details of the uncovered volume stopping criterion. Note that $\hat{\mathcal{B}}$ represents the template bank which is an approximation of the waveform manifold $\mathcal{B}$.

Algorithm 1 Flow-based stochastic template placement

Require: trained flow f_ψ; parameter space Θ; placement radius r_{target}; proposal density $q(\theta)$ on Θ; batch size N_b; stopping tolerance ε; maximum number of batches $N_{\max}$; $\min_\emptyset \equiv +\infty$

$\hat{\mathcal{B}} \leftarrow \emptyset$ {template bank}
for $b = 1, \ldots, N_{\max}$ **do**
  $\hat{\mathcal{B}}_0 \leftarrow \hat{\mathcal{B}}$
  Draw $\{\theta_i\}_{i=1}^{N_b} \sim q(\theta)$ and set $z_i = f_\psi(\theta_i)$; build a KD-tree from $\{f_\psi(\theta) : \theta \in \hat{\mathcal{B}}_0\}$
  $\mathcal{A} \leftarrow \emptyset$ {templates accepted in this batch}
  for θ_i **do**
    if $\min_{z_j \in \hat{\mathcal{B}}_0 \cup \mathcal{A}} \|z_i - z_j\| > r_{\text{target}}$ **then**
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{\theta_i\}$
    end if
  end for
  $\hat{\mathcal{B}} \leftarrow \hat{\mathcal{B}}_0 \cup \mathcal{A}$
  For each candidate, $V_i = |\det J_f(\theta_i)|/q(\theta_i)$ {metric-volume weight}
  $a \leftarrow \dfrac{\sum_i V_i \, \mathbf{1}\left[\min_{z_j \in \hat{\mathcal{B}}_0} \|z_i - z_j\| > r_{\text{target}}\right]}{\sum_i V_i}$ {estimated uncovered metric volume fraction}
  if $a < \varepsilon$ **then**
    break
  end if
end for
return $\hat{\mathcal{B}}$

The stopping criterion estimates the fraction of metric volume that remains uncovered by the current bank. For an exactly isometric mapping satisfying Eq. 6, the metric volume element satisfies

$$\sqrt{|g(\theta)|} = |\det J_f(\theta)|.$$

Since the learned mapping is only approximately isometric, we use $|\det J_f(\theta)|$ as an approximate metric volume weight. For candidates drawn from a proposal distribution $q(\theta)$, the corresponding importance weight is

$$w(\theta) = \frac{|\det J_f(\theta)|}{q(\theta)}.$$

The estimate is only as reliable as the effective sample size of the weights, i.e. how many candidates effectively contribute to the sum. Uniform proposals give an effective sample size of $\sim 1\%$ of the batch in the 5D eccentric space, where the metric volume concentrates in a small region of low secondary mass, so a proposal approximating $\sqrt{|g|}$ (for example log-uniform in the masses) is needed for the criterion to be meaningful. The proposal affects only the order in which regions fill and the quality of this estimate: the acceptance rule is pure latent distance, so the resulting packing is valid for any q with support covering Θ.

After each batch, the uncovered volume fraction is estimated from that batch's own candidates: each candidate contributes V_i to the denominator, and those farther than r_{target} from every template of $\hat{\mathcal{B}}_0$, the bank at the start of the batch, also contribute to the numerator. Placement terminates when this estimate falls below ε. The tolerance ε does not correspond to a coverage guarantee: it controls how close placement runs to saturation, and the resulting coverage must be verified with an explicit injection campaign.

Placement at r_{target} constrains the distance between templates, whereas coverage is set by the distance from a signal to its nearest template, which is smaller: in the saturated bank the median signal-to-template distance is $0.64, r_{\text{target}}$, so typical signals match well above the minimal match and the

bank over-covers. Operating at the intended point instead requires templates spaced further apart, so that typical signals lie near r_{target}. Reducing the bank to the size quoted in the main text raises the median signal-to-template distance to $1.07, r_{\mathrm{target}}$, the separation at which the match equals the minimal match.

8.2 Further results

8.2.1 The template bank

For the *toy example*, the 3D aligned-spin space, we draw 10^4 injections uniformly in the coordinates of the parameter space defined in Table 2. The bank of 23,810 templates covers $99.72 \pm 0.05\%$ of the injections, comparable to the $99.80 \pm 0.04\%$ achieved by a `mbank` stochastic bank of similar size (23,167 templates) on the same injections. The few uncovered injections (28 of 10^4) cluster at large mass ratio and positive spin (median $q = 4.5$, $\chi = 0.66$). Coverage falls to $99.37 \pm 0.16\%$ for $\chi > 0.5$ and to $99.37 \pm 0.16\%$ for $M < 35, M_\odot$, against 99.8% in the rest of the space. The worst-matched injection has $\mathrm{FF} = 0.911$, and only 6 injections fall below $\mathrm{FF} = 0.96$. As in the eccentric case, the template density is highest at low total mass and large mass ratio (Fig. 4). This space is low-dimensional and well described by existing placement methods, so we use it as a control: `isobank` reproduces a bank of standard quality without any tuning specific to the space.

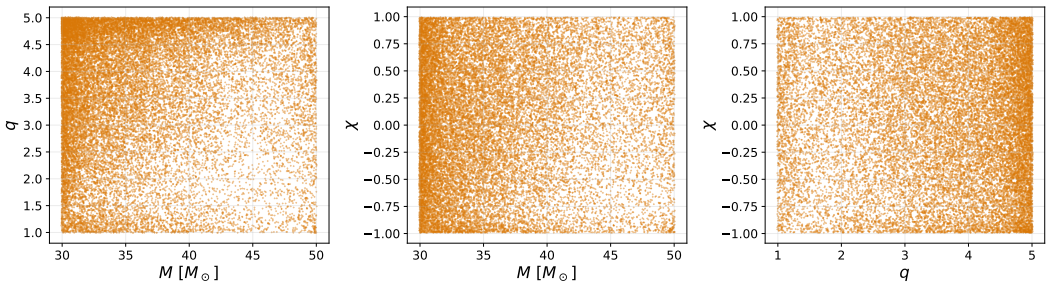


Figure 4: Projections of the 3D aligned-spin template bank (23,810 templates, all shown) onto pairs of parameters. Template density is highest at low total mass and large mass ratio.

For the *high-dimensional example*, the 5D eccentric space, we produce a simulated injection campaign (see definition in Section 2) of 10^4 waveforms drawn uniformly in the coordinates of the parameter space defined in Table 2. The resulting template bank gives an overall coverage of 96.6% with 2.27M templates. The uncovered injections are not uniformly distributed: coverage falls to $91.3 \pm 1.3\%$ for chirp masses $\mathcal{M} < 1.4, M_\odot$, to $94.5 \pm 0.4\%$ for mass ratios $q \geq 3$, and to $94.2 \pm 0.4\%$ for $\chi_1 < -0.3$, against 96.9%, 97.4% and 97.9% respectively over the rest of the space. Interestingly, the region $e \geq 0.12$ is covered at $96.5 \pm 0.4\%$, consistent with the $96.7 \pm 0.2\%$ found at lower eccentricity, so the high-eccentricity corner, which the training loss treats with a dedicated edge term, shows no loss of coverage. Fig. 5 shows the projected template density, which is strongly concentrated towards low component masses where the metric volume grows most rapidly.

8.2.2 The match

As discussed in Section 2, the match $\mathcal{M}(\theta_1, \theta_2)$ measures the noise weighted overlap between the waveforms at two points in parameter space, maximized over time and phase, and is therefore the quantity that determines template bank coverage. The target mismatch $\mu = 0.03$ corresponds to a target match $\mathcal{M} = 0.97$. To verify that the trained flow reproduces this geometry, we generate pairs of points (θ_1, θ_2) separated by a range of proper distances and compare their Euclidean separation in the latent space, $d_z(\theta_1, \theta_2) = |z(\theta_1) - z(\theta_2)|$, with the true waveform match. Fig. 6 shows that d_z is a faithful surrogate for the waveform geometry on the scale relevant to template placement: pairs separated by r_{target} have a mean match consistent with the calibration target of 0.97. Beyond this scale the relation broadens, but placement requires accurate distances only near r_{target}. Larger separations enter only through the nearest-neighbour ranking used to shortlist candidate templates when evaluating fitting factors, where the ordering of distances matters rather than their exact values.

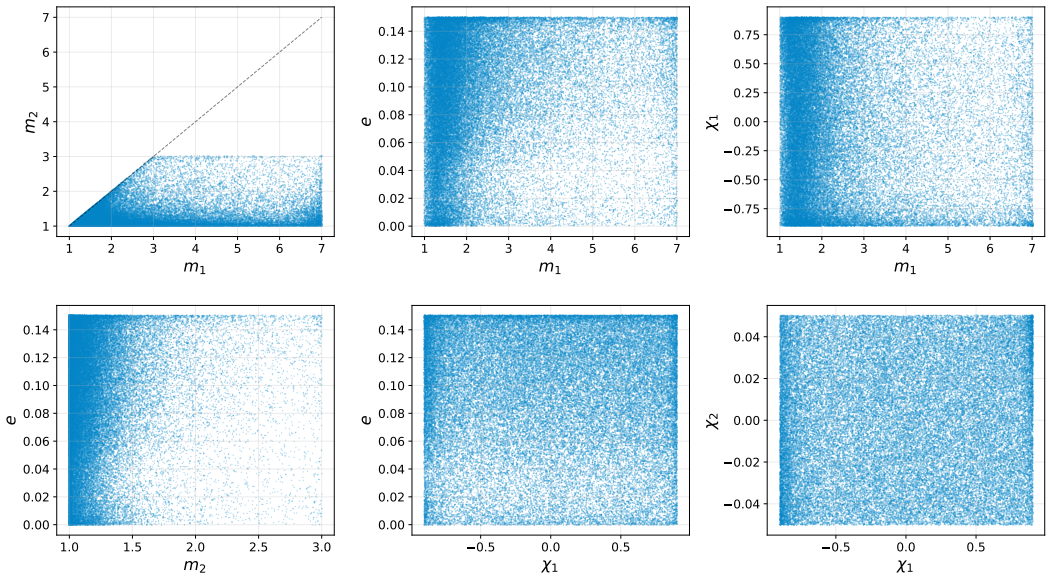


Figure 5: Projections of the 5D eccentric template bank (2,267,558 templates) onto pairs of parameters. For readability, a random subset of 60,000 templates is shown. Templates are concentrated at low component masses, where the metric volume density is largest, and are spread nearly uniformly in χ_1, χ_2 and e. The dashed line in the m_1–m_2 panel marks $m_1 = m_2$.

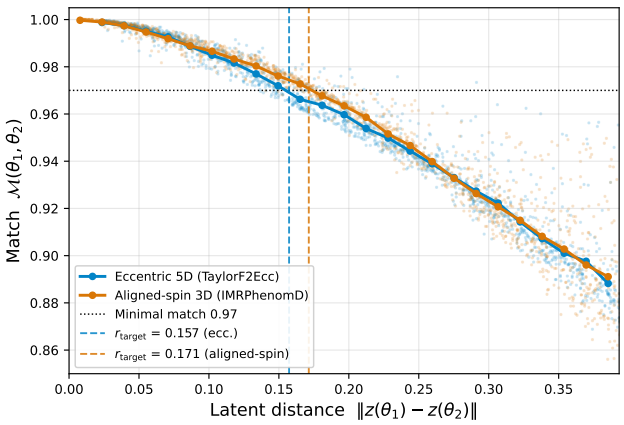


Figure 6: Individual pairs (θ_1, θ_2) shown as points: the true waveform match $\mathcal{M}(\theta_1, \theta_2)$ against their Euclidean distance in the flow's latent space, $|z(\theta_1) - z(\theta_2)|$. Solid lines with markers are means in 25 equal-width bins. The horizontal dotted line marks the minimal match $1 - \mu = 0.97$; the vertical dashed lines mark each space's calibrated placement radius r_{target} (0.157 eccentric, 0.171 aligned-spin), the radius at which the mean mismatch equals $\mu = 0.03$ and at which templates are packed in the latent space.