TopoFisher: Learning Topological Summary Statistics by Maximizing Fisher Information

Matteo Biagetti*
Area Science Park
Località Padriciano 99, Trieste
matteo.biagetti@areasciencepark.it

Mathieu Carrière
DataShape
Centre Inria d'Université Côte d'Azur
Sophia Antipolis, France
mathieu.carriere@inria.fr

Francesco Conti
DataShape
Centre Inria d'Université Côte d'Azur
Sophia Antipolis, France
contifrancesco@me.com

Enrico Maria Ferrari
Area Science Park
Località Padriciano 99, Trieste
enricomariaferrari01@gmail.com

Sven C. Heydenreich
Area Science Park
Località Padriciano 99, Trieste
svenheydenreich@gmail.com

Karthik Viswanathan
Institute of Physics
University of Amsterdam, Amsterdam
vkarthik095@gmail.com

Abstract

Persistence diagrams provide stable, interpretable summaries of geometric and topological structure, but persistence-based pipelines for parameter inference rely on hand-chosen filtrations, vectorizations, and compressors, without an objective tied to parameter uncertainty. We introduce **TopoFisher**, a differentiable persistent-homology pipeline that learns these stages by maximizing local Gaussian Fisher information from simulations near a fiducial parameter value, without posterior samples or regression targets. We give sufficient conditions under which the log-determinant Fisher loss is locally Lipschitz in the trainable parameters. On noisy spirals and Gaussian random fields, where the total Fisher information is known, learning the filtration improves on average over fixed topological pipelines and approaches unconstrained neural summaries. On weak gravitational lensing, the learned topological summaries reach $\log|F| \approx 21$, above the power spectrum (13.8), peak counts (17.1), and wavelet scattering (19.3), and close to an Information Maximising Neural Network (IMNN, 22.4) with up to $\sim 80\times$ fewer trainable parameters. Under a shift from lognormal to LPT-based simulations they lose less information than the IMNN ($\log|F| \approx 12.8$ versus 8.9), and in neural posterior estimation they reach the lowest held-out negative log-likelihood among Fisher-trained and hand-crafted summaries, with calibrated posteriors. Fisher-trained topological summaries are thus a compact front end for simulation-based inference, more robust than unconstrained neural summaries.

*Corresponding Author: matteo.biagetti@areasciencepark.it

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Parameter inference from high-dimensional data often relies on *summary statistics*: low-dimensional functions of observations that retain information about parameters of interest. This choice is especially important in simulation-based inference, where the likelihood $\log p(X|\theta)$ is intractable but samples can be generated from a simulator. Under standard regularity assumptions, the information carried by a summary statistic is quantified by its Fisher information matrix $F(\theta)$. By the Cramér–Rao bound, F^{-1} lower bounds the covariance of unbiased estimators, and maximizing $\log|F|$ minimizes the volume of the corresponding uncertainty ellipsoid [Tegmark, 1997]. Designing Fisher-informative summaries is therefore central to likelihood-free inference, from high-dimensional statistics [Giraud, 2014] to cosmology.

Topological Data Analysis (TDA) provides a useful class of such summaries through *persistence diagrams* (PDs). Built from persistent homology, PDs encode multiscale geometrical and topological features of data, such as connected components, loops, voids, and higher-dimensional cycles [Edelsbrunner et al., 2002, Carlsson, 2009]. Their stability under perturbations [Cohen-Steiner et al., 2007] and their interpretability make them well-suited for inference. In cosmology, for example, PD-based summaries have been shown to capture non-Gaussian information in galaxy and weak-lensing fields that is missed by classical two-point statistics Biagetti et al. [2021], Heydenreich et al. [2021].

However, practical PD-based pipelines require several manual choices. One must choose a filtration, which determines the ordering through which topological features are born and die; a vectorization, which maps diagrams to Euclidean spaces; and often a compressor, which reduces the final statistic to the parameter dimension. These choices strongly affect downstream inference, but are usually selected by hand rather than by a criterion tied to parameter uncertainty. Recent work on differentiable persistent homology shows that PDs and their vectorizations are almost everywhere differentiable with respect to filtration values [Carrière et al., 2021, Leygonie et al., 2022], making gradient-based tuning possible. What remains unclear is the appropriate unsupervised objective for inference. This motivates our central question: **How can we automatically tune topological summaries so that they maximize Fisher information about the parameters of interest?**

Contributions. We introduce *TopoFisher*, a framework for learning topological summary statistics for simulation-based inference. Our contributions are:

- (i) We formulate persistence-based summary learning as Fisher-information maximization. The resulting pipeline can train filtrations, vectorizations, and compressors from simulations near a fiducial parameter value, without posterior samples or supervised regression targets.
- (ii) We provide a local well-posedness result for the Gaussian-Fisher loss of the learned summary: under boundedness, Lipschitz, and non-degeneracy assumptions, the negative log-determinant Gaussian-Fisher loss is locally Lipschitz in the trainable summary parameters.
- (iii) We validate the method on controlled benchmarks with known total Fisher information. On noisy spirals and Gaussian random fields, TopoFisher recovers a large fraction of the available information, and learning the filtration improves on average over fixed topological pipelines.
- (iv) We show that TopoFisher is especially effective for weak gravitational lensing: it outperforms standard cosmological summaries in distribution, approaches the Fisher information of an end-to-end Fisher-optimized IMNN, and is more robust than the IMNN under simulator shift and in held-out posterior NLL.

Setting and scope. In simulation-based inference (SBI), high-dimensional simulator outputs are typically compressed to low-dimensional summaries before a neural density estimator approximates the posterior, likelihood, or likelihood ratio [Cranmer et al., 2020]. TopoFisher learns this summary $s_\phi(X)$ and is therefore complementary to existing SBI algorithms; when we evaluate posteriors (Section 4.3), the neural posterior estimator (NPE) is held fixed across summaries. The trainable filtrations and vectorizations could be optimized under other objectives, but the Fisher objective studied here requires a parameterized simulator, a fiducial parameter value, and simulations at finite-difference points; standard supervised benchmarks, including graph benchmarks, provide none of these and fall outside our scope.

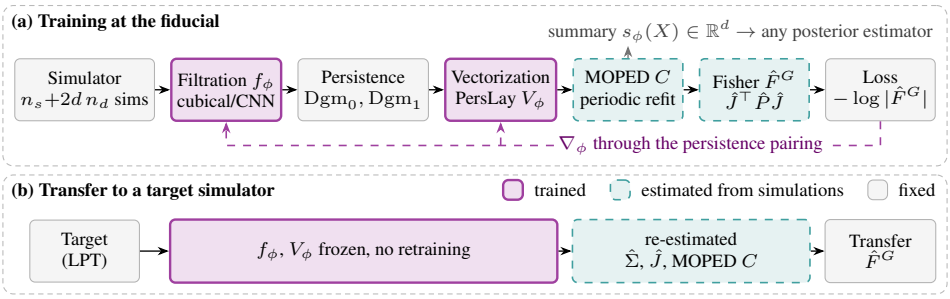


Figure 1: TopoFisher pipeline for the field benchmarks. (a) Simulations at the fiducial and at finite-difference points are filtered, converted to persistence diagrams, vectorized and compressed with MOPED; the Gaussian Fisher matrix of the summaries defines the loss, whose gradient reaches the trainable stages through the persistence pairing. On point clouds (Section 4.1) the filtration acts on an Alpha complex and the compressor is an MLP. (b) Transfer (Section 4.3): learned stages are frozen and simulation-estimated quantities are re-estimated on the target simulator.

Related work. TopoFisher connects Fisher-based summary learning with differentiable topological data analysis. In simulation-based inference, high-dimensional data are often compressed before posterior estimation [Cranmer et al., 2020]. Classical optimal-compression methods preserve Fisher information locally through linear projections or score-based summaries [Heavens et al., 2000, Alsing and Wandelt, 2018, Alsing et al., 2018], while Information Maximising Neural Networks (IMNNs) train neural summaries near a fiducial parameter value by maximizing Fisher information [Charnock et al., 2018]. TopoFisher uses the same local Fisher principle, but constrains the learned summary to factor through persistent homology, yielding a structured and interpretable alternative to black-box IMNN-style compression. In the experiments below, we therefore use an unconstrained IMNN as the main non-topological Fisher-neural reference. Amortized approaches such as BayesFlow [Radev et al., 2022] instead train a summary network jointly with a conditional density estimator, which requires simulations across the prior rather than only near a fiducial value; we compare against this setting in Appendix I.

Persistent homology provides stable multiscale topological descriptors [Edelsbrunner et al., 2002, Carlsson, 2009, Cohen-Steiner et al., 2007]. Many vectorizations have been proposed to map PDs to Euclidean space [Bubenik, 2015, Adams et al., 2017, Kusano et al., 2016, Carrière et al., 2017, Le and Yamada, 2018, Carrière et al., 2020], and recent work on differentiable persistence and learnable filtrations enables gradient-based optimization [Hofer et al., 2017, Gabrielsson et al., 2020, Hofer et al., 2020, Carrière et al., 2021, Leygonie et al., 2022, Hacquard et al., 2022], mostly in supervised or regularization-driven settings. TopoFisher instead trains topological summaries by the Fisher information they retain. von Rohrscheidt et al. [2024] use the Wasserstein distance between persistence diagrams as the discrepancy in a generalised posterior for SBI; TopoFisher instead learns the topological representation, and the two are complementary.

Persistent homology has also been applied in physics, biology, medicine, materials science, and related fields (see Hensel et al. [2021] for a review). In cosmology, PD-based summaries have been used for cosmic-web analysis, random-field morphology, and weak-lensing inference [Pranav et al., 2017, 2019, Biagetti et al., 2021, Heydenreich et al., 2021, 2022, Yip et al., 2024]. These works typically fix the filtration and vectorization by hand; TopoFisher learns them from simulations using the same Fisher criterion used to evaluate parameter constraints.

2 The TopoFisher Loss

TopoFisher is a trainable summary map (Figure 1a): data are filtered, converted into persistence diagrams, vectorized into Euclidean features, and compressed to a low-dimensional statistic. Instead of choosing these stages by hand, we optimize their trainable parameters by maximizing the Fisher information of the summary at a fiducial parameter value. This section introduces the required ingredients and then defines the TopoFisher loss.

Persistent homology in brief. A key component of TopoFisher are *persistence diagrams* (PDs), which are built from the theory of *persistent homology* (PH); see Appendix A and Edelsbrunner and Harer [2010], Oudot [2015] for more details.

Given a topological space T, the starting point of PH is to define a *filtration*, i.e., a finite sequence of growing subspaces $T_1 \subseteq \cdots \subseteq T_N \subseteq T$. The filtration depends on the data: for a point cloud, one can grow balls of increasing radius around the points (*Vietoris–Rips filtration*); for a grayscale image $f : P \to \mathbb{R}$ on a pixel grid, one can use sublevel sets $T_i = f^{-1}((-\infty, \alpha_i])$ for increasing thresholds (*cubical filtration*)[2].

PH then tracks the *topological features* (connected components, loops, cavities) that appear and disappear along the filtration, recording for each a birth time b_i and death time d_i. A *persistence diagram* summarizes this as a multiset of points $(b_i, d_i) \in \mathbb{R}^2$, with $|d_i - b_i|$ the *persistence* of the feature.

PDs enjoy two properties central to TopoFisher. They are *stable* Cohen-Steiner et al. [2007]: close filtrations yield close diagrams under appropriate metrics. And when a filtration is smoothly parametrized by trainable parameters ϕ, the resulting diagram $\mathrm{Dgm}_k(f_\phi)$ is piecewise differentiable in ϕ Carrière et al. [2021], Leygonie et al. [2022], enabling gradient-based optimization. Since the space of PDs is not Euclidean, downstream pipelines apply a *vectorization* (e.g. persistence landscapes Bubenik [2015], persistence images Adams et al. [2017], or learnable layers such as PersLay Carrière et al. [2020]) before further processing.

Fisher Information and Summary Statistics. For a parametric system with data X, a random variable taking values in a measurable space $\mathcal{X}$, let $\{p(\cdot|\theta)\}_{\theta\in\Theta}$ be a family of probability densities on $\mathcal{X}$, with $\Theta \subseteq \mathbb{R}^d$ open. The Fisher Information (FI) (Fisher [1925], Kendall and Stuart [1979]) at a fiducial parameter θ is defined as:

$$F_X(\theta) = \mathbb{E}_\theta \left[\nabla_\theta \log p(X|\theta) \nabla_\theta \log p(X|\theta)^\top \right]. \tag{1}$$

We call $F_X(\theta)$ the total Fisher information, since it quantifies the information available in the uncompressed observation X: via the Cramér–Rao bound, under standard regularity conditions [Lehmann and Casella, 1998], $(nF_X(\theta))^{-1}$ sets the matrix lower limit on the covariance of any unbiased estimator of θ based on n i.i.d. samples. Accordingly, the log-determinant $\log |F_X|$ provides a scalar measure of this total information.

In this work, we focus on complex physical systems where the likelihood $p(X|\theta)$ is often analytically intractable. This restricts our framework to a simulation-based regime, where we only assume the capacity to sample realizations $X \sim p(\cdot|\theta)$. In such cases, the high dimensionality of the data-space $\mathcal{X}$ renders direct inference or comparison prohibitive due to the curse of dimensionality [Alsing et al., 2018]. To circumvent this, we compress the data via a parametric summary mapping of the following form: $s_\phi \colon \mathcal{X} \to \mathbb{R}^M$ given by

$$s_\phi \coloneqq C_\phi \circ V_\phi \circ \mathrm{Dgm}_k \circ f_\phi, \tag{2}$$

where f_ϕ is a learnable filtration, Dgm_k[3] is the persistence diagram operator in dimension k, V_ϕ is a vectorization, and C_ϕ is a compressor to $\mathbb{R}^M$. The parameters $\phi \in \Phi \subseteq \mathbb{R}^p$ are learned and the target dimension $M \ll \dim \mathcal{X}$ is typically of the same order as the degrees of freedom.

This mapping induces a distribution $q_\phi(\cdot|\theta)$ on the summary space. The FI of the transformed system, $F_\phi(\theta)$, is defined analogously to Eq. (1) but w.r.t. the distribution q_ϕ. Crucially, the Data Processing Inequality [Cover and Thomas, 2006] implies that $\log |F_\phi(\theta_{\mathrm{fid}})| \leq \log |F_X(\theta_{\mathrm{fid}})|$, meaning that the log-determinant of the total FI represents a global information ceiling. Our goal is thus to optimize ϕ to close this gap, so that s_ϕ preserves the maximum information relevant for inferring θ.

Assumptions and Fisher estimation. To estimate the Fisher information of the learned summary, we use the standard local Gaussian approximation

$$q_\phi(\cdot|\theta) \simeq \mathcal{N}(\mu_\phi(\theta), \Sigma_\phi)$$

[2]Two standard realizations exist: the V-construction on vertices (lower-star filtration) and the T-construction on top-dimensional cubes; both are used in this work (Appendix J.2).

[3]We present the following for a single homological dimension k; the case of using multiple dimensions follows immediately, concatenating the resulting vectors prior to the compression.

near the fiducial parameter θ_{fid}, with covariance treated as locally independent of θ. Under this approximation, the Gaussian Fisher information reduces to

$$F_\phi^{\text{G}}(\theta_{\text{fid}}) = \left[J_\theta \mu_\phi(\theta_{\text{fid}})\right]^\top \Sigma_\phi^{-1} J_\theta \mu_\phi(\theta_{\text{fid}}), \tag{3}$$

where J_θ indicates the Jacobian with respect to parameters θ and it is estimated by central finite differences. In practice, we estimate Eq. (3) from a structured batch $\mathcal{B}$ of simulation data. Thus, in what follows, F_ϕ^{G} denotes the population Gaussian-Fisher matrix of the summary, while $\widehat{F}_{\phi,\mathcal{B}}^{\text{G}}$ denotes its finite-sample estimate. More details on the Fisher estimation are provided in Appendix B.

Loss and gradients. To maximize the information content of our topological summaries, we define the TopoFisher loss as:

$$\boxed{\widehat{\mathcal{L}}_{\mathcal{B}}(\phi) = -\log \left|\widehat{F}_{\phi,\mathcal{B}}^{\text{G}}(\theta_{\text{fid}})\right|,} \tag{4}$$

which we minimize via stochastic subgradient descent with respect to ϕ. This objective is computed using the Fisher matrix estimated from the batch $\mathcal{B}$, $\widehat{F}_{\phi,\mathcal{B}}^{\text{G}}$. Although persistent homology is not differentiable everywhere, we leverage the subdifferentiable framework established by Leygonie et al. [2022] and Carrière et al. [2021], which ensures well-defined gradients for optimization.

3 Stability

We identify minimal regularity conditions under which the log-determinant of the Gaussian Fisher information of Eq. (3) is locally Lipschitz in ϕ. The result applies to any pipeline whose summary s_ϕ satisfies the assumptions below, not only persistence-based ones, and is distinct from the classical regularity of $\theta \mapsto F_X^G(\theta)$ in the physical parameters [Vaart, 1998]. To our knowledge, no such analysis has been made explicit for Fisher-optimized differentiable persistence pipelines.

Assumption 3.1. *The summary map s_ϕ, the covariance Σ_ϕ, the Jacobian $J_\theta \mu_\phi(\theta_{\text{fid}})$, and the Fisher information matrix $F_\phi^G(\theta_{\text{fid}})$ satisfy the following:*

- (i) $\|s_{\phi'}(X) - s_\phi(X)\| \le L_s \|\phi' - \phi\|$ *for all $\phi, \phi' \in \Phi$ and every realization X;*
- (ii) $\sup_{\phi\in\Phi} \mathbb{E}\left[\|s_\phi(X)\|^2\right] \le M^2$*;*
- (iii) $\|\Sigma_\phi^{-1}\|_{\text{op}} \le \kappa_\Sigma$ *for all $\phi \in \Phi$;*
- (iv) $\|J_\theta \mu_{\phi'}(\theta_{\text{fid}}) - J_\theta \mu_\phi(\theta_{\text{fid}})\|_{\text{op}} \le L_J \|\phi' - \phi\|$ *for all $\phi, \phi' \in \Phi$;*
- (v) $\lambda_{\min}(F_\phi^G(\theta_{\text{fid}})) > 0$ *for all $\phi \in \Phi$,*

where $\lambda_{\min}(A)$ denotes the smallest eigenvalue of A. Under these assumptions, we obtain the following stability result, whose proof can be found in Appendix C.

Theorem 3.2. *Let $F_\phi^G(\theta_{\text{fid}})$ be the Fisher information matrix as in Eq.* (3)*, and let $\mathcal{L}\colon \Phi \to \mathbb{R}$ be defined as $\mathcal{L}(\phi) = -\log|F_\phi^G(\theta_{\text{fid}})|$. Then, under Assumptions 3.1(i)–(v), $\mathcal{L}$ is locally Lipschitz on Φ. More precisely, for every $\phi \in \Phi$ there exist $\varepsilon_0(\phi) > 0$ and $K(\phi) > 0$, depending on κ_Σ, L_s, M, L_J, $\|J_\theta \mu_\phi(\theta_{\text{fid}})\|_{\text{op}}$, and $\lambda_{\min}(F_\phi^G(\theta_{\text{fid}}))$, such that for all $\phi' \in \Phi$ with $\|\phi' - \phi\| < \varepsilon_0(\phi)$,*

$$|\mathcal{L}(\phi') - \mathcal{L}(\phi)| \le K(\phi)\, \|\phi' - \phi\|. \tag{5}$$

Proof idea. The proof proceeds in three steps. First, Assumptions 3.1 $(i) - (iii)$ yield local Lipschitz continuity of $\phi \mapsto \Sigma_\phi^{-1}$ via the resolvent identity $A^{-1} - B^{-1} = A^{-1}(B - A)\, B^{-1}$. Combined with the Lipschitz Jacobian assumption (iv), this propagates to $\phi \mapsto F_\phi^G(\theta_{\text{fid}})$. Finally, non-degeneracy (v) and Weyl's inequality keep $F_\phi^G(\theta_{\text{fid}})$ uniformly positive definite along short segments, so that a standard log-determinant perturbation bound transfers the Lipschitz property to $\mathcal{L}$.

Well-posedness of gradient-based training. Theorem 3.2 concerns the loss $\mathcal{L}$; at training time we optimize the sample-based estimator $\widehat{\mathcal{L}}_{\mathcal{B}}$, which differs from $\mathcal{L}$ by finite-sample noise on $\widehat{\Sigma}_\phi$ and on the perturbed-point means, and by the truncation bias of the central-difference Jacobian. We do not formally transfer the regularity result to $\widehat{\mathcal{L}}_{\mathcal{B}}$; the impact of these effects is monitored empirically through the diagnostics of Appendix B. The theorem applies between consecutive MOPED refits, each of which defines a new local objective (Appendix D).

4 Experiments

We evaluate TopoFisher on three benchmarks, chosen so that performance is measured against a non-topological external reference. For noisy spiral point clouds and two-dimensional Gaussian random fields the likelihood is known, so the total Fisher information F_X gives an absolute ceiling for every summary. Weak gravitational lensing has no known ceiling, but it has established, well-optimized non-topological summaries (the power spectrum, peak counts, and the wavelet scattering transform) against which a topological summary must compete. Across experiments, we compare fixed TDA summaries, TopoFisher-optimized TDA summaries, and unconstrained neural baselines. Performance is measured by $\log|F|$ and, where the ceiling is known, by $\eta_{\log} = \log|\hat{F}^G|/\log|F_X|$, the percentage of the full-data log-determinant Fisher score. Training and evaluation details are given in Appendix J.

4.1 Noisy spiral point clouds

We consider 2D point clouds $X \in \mathbb{R}^{N\times 2}$ with $N = 240$ points, sampled independently from a mixture of a uniform background on $[-h, h]^2$ (with probability $\lambda = 40/240$) and a noisy spiral $(1-\lambda)$. The spiral coordinates are defined by an angular variable $t \sim \mathcal{U}(0, 4\pi)$ and a radius $r \sim \mathcal{N}(\mu t, w^2)$. The target parameters are $\theta = (\mu, w)$, where μ controls the radial expansion and w the intrinsic width. For each fiducial configuration θ_{fid}, we generate $n_s = n_d = 10{,}000$ samples for covariance and derivative estimation for each 5 independent seeds. Since the mixture density $p(X|\theta)$ is known, the total Fisher matrix F_X admits an integral expression (see Appendix E for a derivation), which we use to compute the efficiency metric $\eta_{\log}$.

Methods. All methods share a common preprocessing step: for each point x in the cloud, they operate on the k-nearest neighbor distances $d_k(x) \in \mathbb{R}^k$ to ensure global isometry invariance. Given a point cloud X, we first construct an Alpha complex K to capture the connectivity of the points, and parametrize a scalar function f_ϕ on the vertices of the complex via two architectures:

- *TF-TDA-MLP*. The vertex values are defined as $f_\phi(x) = g_\phi(d_k(x))$, where g_ϕ is a Multi-Layer Perceptron processing each point locally.
- *TF-TDA-GNN*. Features $d_k(x)$ are propagated across the 1-skeleton of K using a Graph Neural Network to produce $f_\phi(x)$, better capturing relational structure.

To ensure a valid simplicial filtration, we extend f_ϕ to edges as $f_\phi(e_{ij}) = \|x_i - x_j\| + \max\{f_\phi(x_i), f_\phi(x_j)\}$, and to higher-dimensional simplices with a lower-star construction: $f_\phi(\sigma) = \max_{e\subset\sigma} f_\phi(e)$. The resulting persistence diagrams in homological dimensions 0 and 1 are converted into 8×8 differentiable persistence images, flattened, and compressed to $\mathbb{R}^2$ via a small MLP.

We compare these against: (i) *DTM*, a non-learnable TDA reference where f_ϕ is replaced by the fixed distance-to-measure function $g(d_k(x)) = ((1/k)\sum_j d_{k,j}(x)^2)^{1/2}$, while keeping the rest of the pipeline identical; (ii) *MLP-only* and *GNN-only*, which serve as neural references. These models use the same architectures described above to learn a 2D function of the point clouds, but aggregate the local contributions of each point via sum-pooling, thereby bypassing the persistence computation.

Results. Table 1 reports results at $\theta_{\text{fid}} = (\mu_{\text{fid}}, w_{\text{fid}}) = (0.6, 0.1)$ for $k = 100$. The TF-TDA-MLP pipeline recovers 78.2% of the total log-determinant Fisher score, exceeding the fixed-filtration DTM baseline by 8.1% and matching the unconstrained MLP-only model (78.1%) within cross-seed variance. TF-TDA-GNN underperforms DTM, while GNN-only remains competitive but below MLP-only: at $k = 100$ the local context of each point already contains enough information about the spiral geometry, and the additional relational complexity of message passing appears redundant for

this inference task. Results at four additional fiducial configurations are reported in Appendix F and reproduce the same ranking.

Beyond the quantitative match with the unconstrained model, the learned filtration is *interpretable*: visualizations show that the MLP assigns vertex values that progressively "unroll" the spiral, effectively filtering out background noise and highlighting the manifold backbone, while the black box models do not show any visible structure in the learned function (see Figures 2 and 3 in Appendix F).

Table 1: Spiral results at $\theta_{\mathrm{fid}} = (0.6, 0.1)$ and $k = 100$, averaged over 5 seeds. Constraint column reports $(\sigma(\mu), \sigma(w))$. We prefix TopoFisher-trained summaries with *TF-*; DTM is a fixed-topology ablation, and MLP/GNN-only are unconstrained non-topological references.

Method	$\log \lvert F \rvert$	(σ_μ, σ_w)	$\eta_{\log}$
Total	*24.27*	*(0.005, 0.001)*	*100%*
Fixed topological ablation: persistence without TopoFisher learning			
DTM fixed filtration	17.18 ± 0.06	$(0.014, 0.013)$	70.1%
TopoFisher summaries: our learned topological contributions			
TF-TDA-MLP	$\mathbf{18.99 \pm 0.30}$	$(\mathbf{0.013}, \mathbf{0.006})$	$\mathbf{78.2}\%$
TF-TDA-GNN	16.20 ± 0.19	$(0.016, 0.019)$	66.7%
Unconstrained neural references: non-topological summaries			
MLP-only	18.96 ± 0.58	$(0.012, 0.006)$	78.1%
GNN-only	18.26 ± 0.25	$(0.015, 0.007)$	75.2%

4.2 Gaussian random fields

We use two-dimensional Gaussian random fields (GRFs) as a controlled benchmark because their likelihood is known and the Fisher information in the full field can be computed analytically. Each sample is a zero-mean periodic field on a 64×64 grid. In Fourier space, the modes are independent Gaussian random variables with variance given by the power spectrum $P(k) = A_s \left(k/k_{\mathrm{pivot}}\right)^{-B}$, where A_s controls the overall amplitude of the field, while B controls how power is distributed across spatial scales. We choose the pivot scale $k_{\mathrm{pivot}} = \exp\langle \ln k \rangle$, where the average is over the nonzero Fourier modes, so that the Fisher correlation between A_s and B vanishes at the fiducial point. We consider fiducial parameters $\theta_{\mathrm{fid}} = (A_s, B) = (1, B_0)$ with $B_0 \in \{-2, -1, 0, 1, 2\}$, use finite-difference steps $\Delta\theta = (0.1, 0.1)$, and generate $n_s = n_d = 40{,}000$ simulations per configuration over 5 independent seeds. The standard Gaussian-field Fisher formula, derived in Appendix G, gives the same total information for all values of B_0: $\log |F_X| = 13.89$, corresponding to $\sigma(A_s) = 0.022$ and $\sigma(B) = 0.044$. This value provides an absolute information ceiling against which all learned and fixed summaries can be compared.

Methods. We compare fixed and learnable topological summaries, together with a non-topological CNN baseline. All TDA methods compute H_0 and H_1 cubical persistence on periodic fields. Fixed-filtration methods use the raw field as the cubical filtration, whereas learnable-filtration methods first transform the field with a small CNN and then compute cubical persistence on the CNN output. All resulting feature vectors are compressed to two dimensions with MOPED [Heavens et al., 2000], a locally linear Fisher-preserving Gaussian compression that is detailed in Appendix B.

Fixed topological baselines. As a topological baseline without TopoFisher optimization, we use a fixed cubical filtration with three hand-designed vectorizations: (i) persistence images [Adams et al., 2017] of size 8×8 with persistence weighting and bandwidth 1; (ii) uniformly weighted persistence silhouettes [Chazal et al., 2014] on 50 grid points; and (iii) differentiable persistence curves of birth and death values [Biagetti et al., 2021], also on 50 grid points. We refer to these as Cubical-PI, Cubical-Silhouette, and Cubical-Curves, respectively. Full definitions are given in Appendix J.3.

Learnable topological summaries. Cubical-PersLay keeps the cubical filtration fixed but learns the diagram vectorization with PersLay [Carrière et al., 2020], a permutation-invariant set network applied to persistence diagram points. We use a 16-dimensional point embedding followed by a 32-hidden-unit MLP for each homology class, with spectral normalization for Lipschitz control.

TF-CNN-PersLay additionally prepends a two-layer CNN with 8 hidden channels and 3×3 kernels, yielding a learned filtration before the same PersLay vectorization and MOPED compression.

Non-topological Fisher-neural baseline. As a black-box reference, we use an IMNN [Charnock et al., 2018]: a strided convolutional encoder followed by a dense head that directly outputs two summaries, one per parameter. The persistence and vectorization stages are replaced by the identity, and no MOPED compression is applied; the dense head is the learned compressor. The IMNN is trained end-to-end with the same local Gaussian Fisher objective used by TopoFisher, making it a direct comparison between an unconstrained Fisher-optimized neural summary and a topologically structured Fisher-optimized summary. Architectural details are provided in Appendix J.2.

Total Fisher Information baseline. The total Fisher of a power-law GRF on a fixed $N \times N$ grid does not depend on B_0 (see a derivation in Appendix G): both diagonal entries of $F(\theta)$ are functions only of the lattice mode structure and of A_s. Each of the five B_0 values therefore provides an independent benchmark with the same theoretical ceiling $\log |F_X| = 13.89$, and the natural figure of merit for a method is its mean performance across this range.

Table 2: GRF results at $N = 64$, averaged over $B_0 \in \{-2, -1, 0, 1, 2\}$ and 5 seeds per B_0. Constraint column reports $(\sigma(A_s), \sigma(B))$. The total Fisher information is $\log |F_X| = 13.89$ at every B_0. We prefix TopoFisher-trained summaries with *TF-*.

Method	$\log \lvert F \rvert$	(σ_{A_s}, σ_B)	$\eta_{\log}$
Total	*13.89*	*(0.022, 0.044)*	*100%*
Fixed topological ablations: persistence without TopoFisher learning			
Cubical-PI	11.53 ± 0.56	$(0.037, 0.093)$	83.0%
Cubical-Silhouette	10.68 ± 0.55	$(0.046, 0.129)$	76.9%
Cubical-Curves	9.22 ± 1.28	$(0.086, 0.292)$	66.4%
TopoFisher summaries: our learned topological contributions			
TF-Cubical-PersLay	11.50 ± 0.54	$(0.038, 0.093)$	82.8%
TF-CNN-PersLay	11.89 ± 0.15	$(0.035, 0.074)$	85.6%
Unconstrained Fisher-neural reference: non-topological summary			
IMNN	12.29 ± 0.09	$(0.032, 0.067)$	88.4%

Results. Table 2 reports the mean performance of each pipeline on the GRF benchmark. The best topological method is TF-CNN-PersLay, which reaches 86% of the total log-determinant Fisher score and improves over all fixed topological pipelines and over TF-Cubical-PersLay, which is level with Cubical-PI: on this benchmark the gain comes from learning the filtration, not the vectorization.

The non-topological IMNN baseline reaches 88% of the total Fisher information, comparable to the learnable topological methods. The topological pipelines, however, achieve this performance with roughly $37 - 55\times$ fewer trainable parameters: TF-Cubical-PersLay has 1,382 trainable parameters and TF-CNN-PersLay has 2,055, compared with approximately 7.5×10^4 for the GRF IMNN. This suggests that the persistence-based inductive bias provides a parameter-efficient route to high-Fisher summaries within similar constraining power.

4.3 Weak gravitational lensing

Weak gravitational lensing measures the convergence field κ, the line-of-sight projection of matter density inferred from coherent distortions of background-galaxy shapes. Its statistics constrain cosmological parameters such as $(\Omega_{\mathrm{m}}, \sigma_8)$, which control the matter density and amplitude of structure. Because κ is highly non-Gaussian on the scales of interest, the angular power spectrum C_ℓ is suboptimal and no closed-form optimal summary is known [Kilbinger, 2015, Ribli et al., 2019, Cheng et al., 2020, Allys et al., 2020]. This makes weak lensing a natural testbed for learned summary statistics. We use the `sbi_lens` lognormal and Lagrange Perturbation Theory (LPT) simulators [Lanzieri et al., 2025] to generate 512×512 convergence maps over a $10^\circ \times 10^\circ$ field of view and five tomographic redshift bins. For each of five independent seeds, we generate 20,000

maps at the fiducial cosmology and at each finite-difference point. All maps are Gaussian-smoothed with $\sigma = 2'$ before applying any summary statistic.

Evaluations. We report three evaluations in Table 3. Local Fisher measures Gaussian Fisher information on lognormal maps at the fiducial cosmology. Transfer Fisher trains learned feature extractors on lognormal maps, freezes them, and evaluates on LPT maps. For MOPED-based summaries we re-estimate the covariance, finite-difference Jacobian, precision matrix, and MOPED compressor on the LPT evaluation bank; for the IMNN the frozen two-dimensional network output is used directly and only the Fisher statistics are re-estimated. No IMNN, CNN-filtration, or PersLay weights are retrained on LPT. Finally, NPE posterior performance is measured by training a neural posterior estimator $q(\theta \mid x)$ on simulations drawn from the prior. We report two quantities. The first is $\log|F| = -\log|\hat{C}|$, where $\hat{C}$ is the covariance of $(\Omega_{\rm m}, \sigma_8)$ samples from the flow conditioned on the fiducial mock; it measures posterior tightness at the fiducial. The second is the negative log-likelihood (NLL), the mean of $-\log q(\theta_{\rm true} \mid x)$ over held-out simulations drawn from the prior; it measures posterior accuracy across the prior and, in contrast to $-\log|\hat{C}|$, penalizes both overconfidence and bias (lower is better). In all Fisher evaluations, we use an additive auto-bin approximation across the five tomographic bins, rather than a full joint tomographic Fisher matrix with cross-bin covariances. All NPE posteriors pass coverage, TARP, SBC and L-C2ST diagnostics except Cubical-PI (Appendix I).

Methods. As the total Fisher is unknown, we compare against three cosmological baselines: (i) the angular power spectrum of the log-transformed field, $\log(C_\ell)$; (ii) peak counts, a histogram of local convergence maxima sensitive to halo-like over-densities; and (iii) the wavelet scattering transform (WST) [Mallat, 2012], a state-of-the-art non-Gaussian summary built from a cascade of wavelet convolutions and pointwise moduli. We use the same learnable topological pipelines as Section 4.2, adapted to 512^2 inputs, and the same IMNN reference (CNN encoder + dense two-dimensional summary head, trained end-to-end by the local Gaussian Fisher objective).

Table 3: Weak lensing at $\sigma = 2'$ smoothing for $(\Omega_{\rm m}, \sigma_8)$, additive over 5 tomographic bins; $\pm$ is the standard deviation over 5 seeds. Constraint columns report $(\sigma(\Omega_{\rm m}), \sigma(\sigma_8))$. Transfer freezes the lognormal-trained feature extractors and evaluates Fisher information on LPT maps. MOPED/Fisher statistics are re-estimated on the LPT evaluation bank when applicable; the IMNN uses its frozen 2-D output directly. NPE: $\log|F| = -\log|\hat{C}|$, with $\hat{C}$ the covariance of flow samples conditioned on the fiducial mock; NLL is the mean negative log posterior density of the true parameters over held-out simulations drawn from the prior (nats, lower is better). Row groups separate external baselines, ablations, TopoFisher contributions, and the unconstrained Fisher-neural reference. Bolds mark the best learned summaries in each column.

	Local Fisher		**Transfer Fisher**		**NPE posterior**	
Method	$\log\|F\|$	$(\sigma_{\Omega_{\rm m}}, \sigma_{\sigma_8})$	$\log\|F\|$	$(\sigma_{\Omega_{\rm m}}, \sigma_{\sigma_8})$	$\log\|F\|$	NLL
Cosmology baselines						
$\log(C_\ell)$	13.78 ± 0.07	$(0.039, 0.059)$	13.84 ± 0.01	$(0.055, 0.089)$	15.15 ± 0.12	-5.61 ± 0.08
Peak counts	17.14 ± 0.01	$(0.026, 0.020)$	11.61 ± 0.06	$(0.119, 0.165)$	16.13 ± 0.13	-5.50 ± 0.05
Wavelet scattering	19.27 ± 0.01	$(0.013, 0.011)$	13.86 ± 0.01	$(0.056, 0.089)$	18.78 ± 0.04	-6.69 ± 0.04
Fixed topological ablations: persistence without TopoFisher learning						
Cubical-Silhouette	16.37 ± 0.01	$(0.029, 0.031)$	12.43 ± 0.01	$(0.082, 0.128)$	18.73 ± 0.10	-6.42 ± 0.07
Cubical-Curves	18.08 ± 0.01	$(0.019, 0.016)$	2.04 ± 0.48	$(1.31, 2.00)$	16.04 ± 0.06	-5.12 ± 0.04
Cubical-PI	19.17 ± 0.01	$(0.014, 0.012)$	10.84 ± 0.01	$(0.175, 0.271)$	15.92 ± 0.49	-4.59 ± 0.35
TopoFisher summaries: our learned topological contributions						
TF-Cubical-PersLay	21.18 ± 0.01	$(0.009, 0.006)$	12.27 ± 0.05	$(0.094, 0.143)$	19.72 ± 0.07	$\mathbf{-6.91 \pm 0.04}$
TF-CNN-PersLay	21.60 ± 0.08	$(0.007, 0.005)$	$\mathbf{12.82 \pm 0.04}$	$(\mathbf{0.073}, \mathbf{0.110})$	19.84 ± 0.13	-6.89 ± 0.06
Unconstrained Fisher-neural reference: black-box non-topological summary						
IMNN	$\mathbf{22.35 \pm 0.04}$	$(\mathbf{0.006}, \mathbf{0.004})$	8.92 ± 0.80	$(0.247, 0.391)$	$\mathbf{20.22 \pm 0.20}$	-6.64 ± 0.03

Results. Table 3 summarizes the local Fisher, transfer Fisher, and NPE posterior evaluations. In distribution, all topological summaries outperform the two-point baseline $\log(C_\ell)$, which reaches

only $\log|F| = 13.8$ with marginal constraints $(\sigma_{\Omega_m}, \sigma_{\sigma_8}) = (0.039, 0.059)$. This confirms that persistent homology captures non-Gaussian information in the smoothed convergence field that is inaccessible to the power spectrum. Learning the filtration and diagram vectorization gives a further substantial gain: both TopoFisher variants, TF-Cubical-PersLay and TF-CNN-PersLay, reach $\log|F| \approx 21$ with marginal constraints of $(0.009, 0.006)$ and $(0.007, 0.005)$, within $\sim 0.75 - 1.2$ nats of the unconstrained IMNN reference ($\log|F| = 22.4$) trained end-to-end with the identical Fisher objective. Crucially, this near-parity is achieved while retaining an explicitly topological representation and using up to $\sim 80\times$ fewer trainable parameters than the IMNN (see Appendix J.2 for parameter counts).

Under lognormal-to-LPT shift the IMNN drops from $\log|F| = 22.4$ to 8.9, with constraints $(0.25, 0.39)$, whereas TF-CNN-PersLay and TF-Cubical-PersLay retain 12.8 and 12.3; the power spectrum and wavelet scattering, which are not learned, transfer better than all learned summaries. Appendix I also reports fully frozen variants, in which the lognormal MOPED operator and precision matrix are kept fixed so that only the parameter response is measured on LPT; TF-CNN-PersLay retains the most information among the learned summaries in all of them. On NPE, the IMNN gives the tightest posterior at the fiducial ($\log|F| = 20.2$), with TF-CNN-PersLay just below, but the two TopoFisher summaries reach the lowest held-out NLL in Table 3 (-6.91 and -6.89, against -6.64 for the IMNN). Our results show that maximizing local Fisher information gives no guarantee of being optimal for a different simulator (LPT) or parameters different from the fiducial (SBI). TopoFisher learns topological summaries, which are physical and tend to transfer well between settings; the unconstrained IMNN learns features specific to the simulator that do not transfer. This is also evident in Fig. 5, which demonstrates that topological statistics transfer well to extended parameter spaces, while IMNN does not.

4.4 Computational and simulation cost

The topological summaries are compact (on weak lensing, $1{,}382$ and $2{,}055$ trainable parameters for TF-Cubical-PersLay and TF-CNN-PersLay, against $168{,}210$ for the IMNN), but persistent homology dominates their training time: 97% for TF-Cubical-PersLay, whose diagrams are cached, and 99.6% for TF-CNN-PersLay, whose diagrams are recomputed at every epoch, so the IMNN remains the fastest to train (Appendix K). Central finite differences need $n_s + 2d\,n_d$ simulations per fiducial, as for any local-Fisher method, and the deficit to the Fisher ceiling grows with d, from 1.7 to 4.6 nats for TF-CNN-PersLay and from 1.6 to 3.2 for the IMNN between $d = 2$ and $d = 4$ (Appendix H.3). Transfer still requires target simulations, at least for the Jacobian.

5 Conclusions

We introduced TopoFisher, which learns persistence-based summary statistics by maximizing the local Gaussian Fisher information of the final summary, turning filtration, vectorization, and compression into trainable components, and showed that its loss is locally Lipschitz in the trainable parameters under mild regularity conditions.

Learning the filtration adds information on all three benchmarks, whereas learning the vectorization alone helps on weak lensing but not on GRFs. On weak lensing, the learned topological summaries come within 0.8–1.2 nats of an unconstrained IMNN with one to two orders of magnitude fewer trainable parameters, lose less information than the IMNN under simulator shift and extended parameter spaces, and reach the lowest held-out NLL among Fisher-trained and hand-crafted summaries, comparable to an end-to-end amortized summary network. The power spectrum and wavelet scattering nevertheless transfer better than all learned summaries, and persistent homology, rather than the network, dominates the training cost.

TopoFisher is a *local* method: it optimizes Fisher information near a fiducial point under a Gaussian approximation, guarantees neither globally injective summaries nor calibrated posteriors, and its simulation cost grows with the number of parameters. The objective extends directly to several fiducials; on GRFs, one PersLay representation shared across five spectral indices stays within 0.21 nats of per-fiducial specialists. Testing this with learned filtrations, comparing with comparison-based topological SBI [von Rohrscheidt et al., 2024] on a common simulator, and studying robustness to systematic misspecification are natural next steps.

Reproducibility and impact

An public repository with code, configuration generators, and run instructions is available at `https://github.com/RitAreaSciencePark/TopoFisher`. The repository documents how to regenerate the simulation datasets, produce the YAML configs for each benchmark, train the reported pipelines, and evaluate Fisher, transfer, and SBI results. Large generated simulation datasets are not bundled with the submission because of their size.

Societal impact. This work is methodological and aimed at scientific inference from simulations. Its main positive impact is to improve robust, parameter-efficient uncertainty quantification for data-intensive sciences such as cosmology. Potential negative impacts are indirect: if applied with poorly validated simulators or outside the local regime where the Fisher approximation is reliable, the method could lead to overconfident scientific conclusions. We mitigate this risk through diagnostics, simulator-shift tests, posterior validation, and code release.

Acknowledgments and Disclosure of Funding

M.B. and E.M.F. are supported by the Ministero Università e Ricerca, project E-ARGO" (CUP J95F21002190001), with title "Fondo finalizzato al rilancio degli Investimenti delle Amministrazioni Centrali dello Stato e allo sviluppo del Paese, ex Art.1, comma 14, legge n. 160/2019". M.C. and F.C. are supported by l'Agence Nationale de la Recherche (ANR) under grants TopModel ANR-23-CE23-0014. MC is also supported by 3IA ANR-23-IACL-0001 and the PERSNET project, part of the PEPR IA programme. S.C.H. is supported by the European Union – NextGenerationEU within the project PNRR "Finanziamento di progetti presentati da giovani ricercatori" - Mission 4 Component 2 investment 1.2.

We thank Area Science Park supercomputing platform ORFEO made available for conducting the research reported in this paper and the technical support of the Laboratory of Data Engineering staff.

References

Max Tegmark. Measuring Cosmological Parameters with Galaxy Surveys. *Phys. Rev. Lett.*, 79(20): 3806–3809, November 1997. doi: 10.1103/PhysRevLett.79.3806.

Christophe Giraud. *Introduction to High-Dimensional Statistics*. Chapman and Hall/CRC, 1 edition, 2014. doi: 10.1201/b17895. URL `https://doi.org/10.1201/b17895`.

Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28:511–533, 2002.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

Matteo Biagetti, Alex Cole, and Gary Shiu. The persistence of large scale structures. Part I: Primordial non-Gaussianity. *Journal of Cosmology and Astroparticle Physics*, 2021(04):061, 2021.

Sven Heydenreich, Benjamin Brück, and Joachim Harnois-Déraps. Persistent homology in cosmic shear: Constraining parameters with topological data analysis. *Astronomy & Astrophysics*, 648: A74, 2021. doi: 10.1051/0004-6361/202039048.

Mathieu Carrière, Frédéric Chazal, Marc Glisse, Yuichi Ike, Hariprasad Kannan, and Yuhei Umeda. Optimizing persistent homology based functions. In *International Conference on Machine Learning*, pages 1294–1303. PMLR, 2021.

Jacob Leygonie, Steve Oudot, and Ulrike Tillmann. A framework for differential calculus on persistence barcodes. *Found. Comput. Math.*, 22(4):1069–1131, August 2022. ISSN 1615-3375. doi: 10.1007/s10208-021-09522-y. URL `https://doi.org/10.1007/s10208-021-09522-y`.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020. doi: 10.1073/pnas.1912789117.

Alan F Heavens, Raul Jimenez, and Ofer Lahav. Massive lossless data compression and multiple parameter estimation from galaxy spectra. *Monthly Notices of the Royal Astronomical Society*, 317 (4):965–972, 2000.

Justin Alsing and Benjamin Wandelt. Generalized massive optimal data compression. *Monthly Notices of the Royal Astronomical Society: Letters*, 476(1):L60–L64, 2018. doi: 10.1093/mnrasl/sly029.

Justin Alsing, Benjamin Wandelt, and Stephen Feeney. Massive optimal data compression and density estimation for scalable, likelihood-free inference in cosmology. *Monthly Notices of the Royal Astronomical Society*, 477(3):2874–2885, 2018.

Tom Charnock, Guilhem Lavaux, and Benjamin D Wandelt. Automatic physical inference with information maximizing neural networks. *Physical Review D*, 97(8):083004, 2018.

Stefan T. Radev, Ulf K. Mertens, Andreas Voss, Lynton Ardizzone, and Ullrich Köthe. BayesFlow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4):1452–1466, 2022.

Peter Bubenik. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 16(1):77–102, 2015.

Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8): 1–35, 2017.

Genki Kusano, Yasuaki Hiraoka, and Kenji Fukumizu. Persistence weighted gaussian kernel for topological data analysis. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2004–2013. PMLR, 2016.

Mathieu Carrière, Marco Cuturi, and Steve Oudot. Sliced Wasserstein Kernel for Persistence Diagrams. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, volume 70, pages 664–673. PMLR, July 2017.

Tam Le and Makoto Yamada. Persistence Fisher Kernel: A Riemannian Manifold Kernel for Persistence Diagrams. In Samy Bengio, Hanna Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, volume 31. Curran Associates, Inc., 2018.

Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In *International Conference on Artificial Intelligence and Statistics*, pages 2786–2796. PMLR, 2020.

Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Rickard Brüel Gabrielsson, Bradley J Nelson, Anjan Dwaraknath, and Primoz Skraba. A topology layer for machine learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1553–1563. PMLR, 2020.

Christoph Hofer, Florian Graf, Bastian Rieck, Marc Niethammer, and Roland Kwitt. Graph filtration learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4314–4323. PMLR, 2020.

Olympio Hacquard, Krishnakumar Balasubramanian, Gilles Blanchard, Clément Levrard, and Wolfgang Polonik. Topologically Penalized Regression on Manifolds. *Journal of Machine Learning Research (JMLR)*, 23(161):1–39, 2022. ISSN 1533-7928.

Julius von Rohrscheidt, Bastian Rieck, and Sebastian M. Schmon. Bayesian computation meets topology. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=0h1DtRK6dA.

Felix Hensel, Michael Moor, and Bastian Rieck. A survey of topological machine learning methods. *Frontiers in Artificial Intelligence*, 4:681108, 2021.

Pratyush Pranav, Herbert Edelsbrunner, Rien van de Weygaert, Gert Vegter, Michael Kerber, Bernard J. T. Jones, and Mathijs Wintraecken. The topology of the cosmic web in terms of persistent betti numbers. *Monthly Notices of the Royal Astronomical Society*, 465(4):4281–4310, 2017. doi: 10.1093/mnras/stw2862.

Pratyush Pranav, Rien van de Weygaert, Gert Vegter, Bernard J T Jones, Robert J Adler, Job Feldbrugge, Changbom Park, Thomas Buchert, and Michael Kerber. Topology and geometry of gaussian random fields i: on betti numbers, euler characteristic, and minkowski functionals. *Monthly Notices of the Royal Astronomical Society*, 485(3):4167–4208, 05 2019. ISSN 0035-8711. doi: 10.1093/mnras/stz541. URL https://doi.org/10.1093/mnras/stz541.

Sven Heydenreich, Benjamin Brück, Pierre Burger, Joachim Harnois-Déraps, Sandra Unruh, Tiago Castro, Klaus Dolag, and Nicolas Martinet. Persistent homology in cosmic shear. II. A tomographic analysis of DES-Y1. *A&A*, 667:A125, November 2022. doi: 10.1051/0004-6361/202243868.

Jacky H. T. Yip, Matteo Biagetti, Alex Cole, Karthik Viswanathan, and Gary Shiu. Cosmology with persistent homology: a fisher forecast. *Journal of Cosmology and Astroparticle Physics*, 2024(09): 034, 2024. doi: 10.1088/1475-7516/2024/09/034.

Herbert Edelsbrunner and John L. Harer. *Computational Topology: An Introduction*. American Mathematical Society, January 2010. ISBN 978-1-4704-6769-2.

Steve Oudot. *Persistence Theory: From Quiver Representations to Data Analysis*, volume 209 of *Mathematical Surveys and Monographs*. American Mathematical Society, 2015. ISBN 978-1-4704-3443-4.

R.A. Fisher. *Statistical methods for research workers*. Edinburgh Oliver & Boyd, 1925.

Maurice Kendall and Alan Stuart. *The advanced theory of statistics. Vol.2: Inference and relationship*. 1979.

Erich Leo Lehmann and George Casella. *Theory of point estimation*. Springer, 1998.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, July 2006. ISBN 0471241954.

A. W. van der Vaart. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998.

Frédéric Chazal, Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, and Larry Wasserman. Stochastic Convergence of Persistence Landscapes and Silhouettes. In *Proceedings of the 30th Annual Symposium on Computational Geometry (SoCG 2014)*, SOCG'14, pages 474–483. Association for Computing Machinery, 2014. ISBN 978-1-4503-2594-3. doi: 10.1145/2582112.2582128.

Martin Kilbinger. Cosmology with cosmic shear observations: A review. *Reports on Progress in Physics*, 78(8):086901, 2015.

Dezső Ribli, Bálint Ármin Pataki, José Manuel Zorrilla Matilla, Daniel Hsu, Zoltán Haiman, and István Csabai. Weak lensing cosmology with convolutional neural networks on noisy data. *MNRAS*, 490(2):1843–1860, December 2019. doi: 10.1093/mnras/stz2610.

Sihao Cheng, Yuan-Sen Ting, Brice Ménard, and Joan Bruna. A new approach to observational cosmology using the scattering transform. *Monthly Notices of the Royal Astronomical Society*, 499 (4):5902–5914, 2020.

Erwan Allys, Titouan Marchand, Jean-François Cardoso, Francisco Villaescusa-Navarro, Shirley Ho, and Stéphane Mallat. New interpretable statistics for large-scale structure analysis and generation. *Physical Review D*, 102(10):103506, 2020.

Denise Lanzieri, Justine Zeghal, T. Lucas Makinen, Alexandre Boucaud, Jean-Luc Starck, and François Lanusse. Optimal neural summarization for full-field weak lensing cosmological implicit inference. *A&A*, 697:A162, May 2025. doi: 10.1051/0004-6361/202451535.

Stéphane Mallat. Group Invariant Scattering. volume 65, pages 1331–1398, October 2012. doi: 10.1002/cpa.21413.

James R. Munkres. *Elements of Algebraic Topology*. CRC Press, 1st edition edition, 1984. ISBN 978-0-367-09141-5.

J Hartlap, P Simon, and P Schneider. Why your model parameter confidences may be too optimistic. unbiased estimation of the inverse covariance matrix. *Astronomy & Astrophysics*, 464(1):399–404, 2007.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Steven M Kay. *Fundamentals of statistical signal processing: estimation theory*, volume 1. 1997.

George Papamakarios and Iain Murray. Fast ε-free inference of simulation models with bayesian conditional density estimation. In *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, Advances in Neural Information Processing Systems, pages 1028–1036. Neural Information Processing Systems Foundation, Inc, December 2016. URL `https://nips.cc/Conferences/2016`. 30th Annual Conference on Neural Information Processing Systems, NIPS 2016 ; Conference date: 05-12-2016 Through 10-12-2016.

Jan-Matthis Lueckmann, Pedro J. Gonçalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H. Macke. Flexible statistical inference for mechanistic models of neural dynamics. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1289–1299, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

David S. Greenberg, Marcel Nonnenmacher, and Jakob H. Macke. Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning*, 2019. URL `https://api.semanticscholar.org/CorpusID:159041084`.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=HkpbnH9lx`.

Nicolas Martinet, Joachim Harnois-Déraps, Eric Jullo, and Peter Schneider. Probing dark energy with tomographic weak-lensing aperture mass statistics. *Astronomy & Astrophysics*, 646:A62, February 2021. doi: 10.1051/0004-6361/202039679.

Marika Asgari and Peter Schneider. A new data compression method and its application to cosmic shear analysis. *Astronomy & Astrophysics*, 578:A50, June 2015. doi: 10.1051/0004-6361/201424905.

Pablo Lemos, Adam Coogan, Yashar Hezaveh, and Laurence Perreault-Levasseur. Sampling-based accuracy testing of posterior estimators for general inference. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *PMLR*, pages 19256–19273, 2023.

Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. Validating bayesian inference algorithms with simulation-based calibration. *arXiv preprint arXiv:1804.06788*, 2018.

Julia Linhart, Alexandre Gramfort, and Pedro L. C. Rodrigues. L-C2ST: Local diagnostics for posterior approximations in simulation-based inference. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Mathieu Andreux, Tomás Angles, Georgios Exarchakis, Roberto Leonarduzzi, Gaspar Rochette, Louis Thiry, John Zarka, Stéphane Mallat, Joakim Andń, Eugene Belilovsky, Joan Bruna, Vincent Lostanlen, Muawiz Chaudhary, Matthew J. Hirn, Edouard Oyallon, Sixin Zhang, Carmine Cella, and Michael Eickenberg. Kymatio: Scattering transforms in python. *Journal of Machine Learning Research*, 21(60):1–6, 2020. URL `http://jmlr.org/papers/v21/19-047.html`.

A Persistent homology: preliminaries

A key component of our summary statistic, the persistence diagram, is defined through the theory of persistent homology. In this section, we recall the basics of homology groups and persistence theory.

Simplicial homology. In this section, we briefly review the fundamentals of simplicial homology with coefficients in $\mathbb{Z}/2\mathbb{Z}$, which is the setting adopted for practical computations. For a more comprehensive introduction, we refer the reader to [Munkres, 1984, Chapter 1]. The basic building blocks of simplicial (persistent) homology are *simplicial complexes*, which provide combinatorial representations of topological spaces that can be efficiently manipulated algorithmically.

Definition A.1. Given a finite set of points $X_n := \{x_1, \dots, x_n\}$ sampled from a topological space X, an *abstract simplicial complex* constructed from X_n is a collection $S(X_n)$ of subsets of X_n such that:

- if $\tau \in S(X_n)$ and $\sigma \subseteq \tau$, then $\sigma \in S(X_n)$, and
- if $\sigma, \tau \in S(X_n)$, then either $\sigma \cap \tau \in S(X_n)$ or $\sigma \cap \tau = \varnothing$.

Each element $\sigma \in S(X_n)$ is called a *simplex*, and its *dimension* is defined by $\dim(\sigma) := \mathrm{card}(\sigma) - 1$. Simplices of dimension 0 are referred to as *vertices*.

A central operator in simplicial homology is the *boundary operator*, which maps a simplex to the formal sum of its faces. Such formal sums are called *chains*, and the collection of chains forms a group, denoted by $Z_*(S(X_n))$.

Definition A.2. Given a simplex $\sigma = [x_{i_1}, \dots, x_{i_p}]$, the boundary operator ∂ is defined as

$$\partial(\sigma) := \sum_{j=1}^{p} [x_{i_1}, \dots, x_{i_{j-1}}, x_{i_{j+1}}, \dots, x_{i_p}].$$

In other words, $\partial(\sigma)$ is obtained by removing one vertex at a time from σ. This definition extends linearly to arbitrary chains.

A topological feature is represented by a *cycle*, that is, a chain whose boundary vanishes. Working over $\mathbb{Z}/2\mathbb{Z}$, this condition corresponds to requiring that each simplex in the boundary appears an even number of times, so that duplicates cancel out. Formally, a chain c is a cycle if $\partial(c) = 0$.

It follows directly that $\partial \circ \partial = 0$, meaning that the boundary of any chain is itself a cycle. However, cycles that arise as boundaries of higher-dimensional chains are considered *trivial*, and should be excluded. The set of such chains forms a subgroup denoted by $B_*(S(X_n))$.

Definition A.3. The k-th homology group is defined as the quotient

$$H_k(S(X_n)) := \frac{Z_k(S(X_n))}{B_k(S(X_n))}.$$

Equivalently, $H_k(S(X_n))$ consists of equivalence classes of k-dimensional cycles modulo boundaries. In fact, when working over $\mathbb{Z}/2\mathbb{Z}$, this group is a vector space.

Hence, homology groups are vector spaces computed on simplicial complexes built on finite data sets, and whose bases encode the different cycles of the complex which are not boundaries. Intuitively, these cycles represent topological features, whose types depend on the homology group degree: connected components ($k = 0$), loops ($k = 1$), cavities ($k = 2$), and their higher-dimensional counterparts ($k > 2$). In practice, simplicial complexes can be built on (a) point clouds with the so-called Vietoris-Rips complexes $V_\delta(X_n)$ (by creating simplices whose corresponding vertices are points that are all at distance at most δ from each other), and (b) images with the so-called cubical complexes $C(I)$ (by creating simplices whose corresponding vertices are neighboring pixels).

Persistent homology. The core objective of persistent homology is to extract *persistent* topological information from *filtrations*, i.e., sequences of growing simplicial complexes.

Definition A.4. Let $S = S(X_n)$ be a finite simplicial complex. A *filtration* of S is a family $(S_t)_t$ of subcomplexes such that $S_p \subseteq S_q$ whenever $p \leq q$. For each simplex $\sigma \in S$, define its *insertion time*, or *filtration value*, as $t(\sigma) := \inf\{t \mid \sigma \in S_t\}$.

From a topological viewpoint, the addition of a simplex σ to the filtration produces exactly one of two effects: either it gives rise to a new topological feature (for instance, inserting an edge may create a loop), hence increasing the dimension of the homology group in degree $\dim(\sigma)$ of the subcomplex, or it eliminates an existing feature of lower dimension (for example, connecting two previously disjoint components), hence decreasing the dimension of the homology group in degree $\dim(\sigma) - 1$. Using a matrix reduction algorithm that leverages the vector space structure of homology groups [Edelsbrunner and Harer, 2010, §IV.2], persistent homology associates to each feature a *critical pair* of simplices (σ_b, σ_d) corresponding to its creation and destruction, together with the associated *birth* and *death* times $t(\sigma_b)$ and $t(\sigma_d)$.[4]

Definition A.5. The resulting collection of intervals $(t(\sigma_b), t(\sigma_d))$ in homological dimension k forms a finite subset of the half-plane $\{(b, d) \in \mathbb{R}^2 \mid b < d\}$ known as the *k-persistence diagram* (PD) of the filtration $(S_t)_t$. The quantity $\frac{1}{\sqrt{2}}|t(\sigma_b) - t(\sigma_d)|$, corresponding to the distance to the diagonal $\{b = d\}$, is called the *persistence* of the feature and measures how long it remains detectable across scales.

In the case of Vietoris-Rips complexes, it is common to build filtrations $\{V_\delta(X_n)\}_{\delta>0}$ by letting their corresponding neighborhood threshold δ increase, and in the case of images, it is common to use the growing sublevel sets of the pixel value function $\{f^{-1}((-\infty, \alpha])\}_{\alpha\in\mathbb{R}}$, where an image P is represented as a function $f : P \to \mathbb{R}$ valued on the grid P of pixels.

Very importantly, PDs are *stable* [Cohen-Steiner et al., 2007]: when computed from the sublevel sets of two continuous functions f, g defined on the same space, PDs can be compared with the so-called bottleneck distance d_b (which is similar to an optimal transport metric).

Definition A.6 (bottleneck distance)**.** Let Dgm and Dgm' be two PDs. The *bottleneck distance* d_b between PDs is defined as:

$$d_b(\mathrm{Dgm}, \mathrm{Dgm}') := \inf_{P\in\mathcal{P}(\mathrm{Dgm},\mathrm{Dgm}')} c(P), \tag{6}$$

where $\mathcal{P}(\mathrm{Dgm}, \mathrm{Dgm}')$ denotes the set of *partial correspondences*, i.e., the set of subsets of $\mathrm{Dgm} \times \mathrm{Dgm}'$ s.t. the first and second projections $\pi_1 : (p, p') \mapsto p$ and $\pi_2 : (p, p') \mapsto p'$ are injective, and where the *cost* of a partial correspondence P is defined as:

$$c(P) := \max\{\max_{(p,p')\in P} \|p - p'\|_\infty, \max_{p\notin\mathrm{im}(\pi_1)} \|p - \pi_\Delta(p)\|_\infty, \max_{p'\notin\mathrm{im}(\pi_2)} \|p' - \pi_\Delta(p')\|_\infty\}. \tag{7}$$

Then, the main result of PH states that:

$$d_b(\mathrm{Dgm}_k(f), \mathrm{Dgm}_k(g)) \leq \|f - g\|_\infty.$$

Vectorizations and Differentiability. Finally, PDs are often represented by vectors to make amenable to standard computations (addition, inner product, etc).

Definition A.7. A *vectorization* of PDs is a continuous function $\varphi : \mathcal{D} \to \mathcal{H}$, where $\mathcal{D}$ is the space of PDs, and $\mathcal{H}$ is a Hilbert space.

Common representations include, e.g., the persistence landscape Bubenik [2015], the persistence image Adams et al. [2017], and neural network-based vectorizations such as PersLay Carrière et al. [2020]. Finally, a recent property of PDs is their *differentiability*: as every PD point is associated to a critical pair, the map $f \mapsto \mathrm{Dgm}_k(f)$ is piecewise smooth in f. Indeed, as the critical pairs only depend on the filtration order, the combinatorial pairing is locally constant, and PDs can be built locally by picking the insertion times at the critical pair indices in the filtration. This gather-based strategy (described initially in Carrière et al. [2021]) can be used in a gradient descent optimization scheme based on backpropagation by computing the critical pairs once in the forward pass and reading the filtration values off the input tensor through differentiable indexing on the backward pass.

B More details on Fisher estimation

We estimate the Gaussian Fisher loss $F_\phi^{\mathrm{G}}(\theta_{\mathrm{fid}})$ at fiducial parameter with structured batches $\mathcal{B}$. A batch $\mathcal{B}$ includes realizations at θ_{fid} to estimate Σ_ϕ via sample covariance, to which we apply the Hartlap

[4] Some features may never be destroyed; in that case, their death time is set to $+\infty$.

correction [Hartlap et al., 2007] for unbiasedness of the precision matrix, as well as realizations at infinitesimal perturbations around $\theta_{\rm fid}$ to compute $J_\theta \mu_\phi(\theta_{\rm fid})$ via central finite differences. These empirical estimates, denoted as $\widehat{\Sigma}_{\phi,\mathcal{B}}$ and $\widehat{J}_{\theta,\mathcal{B}}$ respectively, are combined to evaluate the Fisher matrix through the quadratic form:

$$\widehat{F}^{\rm G}_{\phi,\mathcal{B}}(\theta_{\rm fid}) = \widehat{J}^{\top}_{\theta,\mathcal{B}} \widehat{P}_{\phi,\mathcal{B}} \widehat{J}_{\theta,\mathcal{B}}, \tag{8}$$

where $\widehat{P} = \frac{n_s - M - 2}{n_s - 1} \widehat{\Sigma}^{-1}$ with the condition $n_s > M + 2$ represents the Hartlap-corrected precision matrix.

The validity of this estimator relies on two key assumptions: the local linearity of the summary response and its Gaussianity. Since finite differences are only valid in a locally linear regime, we check this assumption by repeating the derivative estimate at two step sizes and verifying that the derivatives, and the resulting Fisher matrices, are stable.

Furthermore, we validate the Gaussian approximation of the final summaries with Kolmogorov–Smirnov (KS) tests. Specifically, we apply a 1D KS test to each component of the compressed summary against a Gaussian with matching mean and variance; at a significance level of $\alpha = 0.05$, we find the summaries to be consistent with the Gaussian assumption.

MOPED Compression. To compress the output of the vectorization map V_ϕ into a reduced space $\mathbb{R}^M$, one elegant choice is MOPED compression [Heavens et al., 2000]. This is a linear operator defined by a matrix $C_\phi \in \mathbb{R}^{d \times \dim(V_\phi)}$ given by $C_\phi = [J_\theta \tilde{\mu}_\phi(\theta_{\rm fid})]^\top \tilde{\Sigma}_\phi^{-1}(\theta_{\rm fid})$, where $\tilde{\mu}_\phi(\theta)$ and $\tilde{\Sigma}_\phi(\theta)$ are the mean and covariance of the vectorization output evaluated at θ. In our implementation, $J_\theta \tilde{\mu}_\phi(\theta_{\rm fid})$ and $\tilde{\Sigma}_\phi(\theta_{\rm fid})$ are estimated using a portion of the batch $\mathcal{B}$ and we evaluate the final Fisher information on the remaining part. In practice, the Jacobian $J_\theta \tilde{\mu}_\phi(\theta_{\rm fid})$ is estimated via finite differences, the inverse covariance estimated from $\tilde{\Sigma}_\phi(\theta_{\rm fid})$ is corrected using the Hartlap factor. Under the Gaussian assumption of the vectorization output, MOPED is provably lossless, ensuring the Fisher information of the resulting d-dimensional summaries ($M = d$) matches the information content of the full vector.

C Proof of the Stability Theorem

The proof proceeds in three steps: we first establish local Lipschitz continuity of $\phi \mapsto \Sigma_\phi^{-1}$ (Proposition C.1), then of $\phi \mapsto F_\phi(\theta_{\rm fid})$ (Proposition C.2), and finally of $\mathcal{L} = -\log|\det F|$. The last step relies on a standard perturbation bound for the log-determinant of a positive-definite matrix, combined with Weyl's inequality to ensure $F_\phi(\theta_{\rm fid}) + t\,\Delta F$ stays uniformly away from singular along the segment $t \in [0,1]$. For the sake of brevity, throughout this Appendix we write $J_\phi \coloneqq J_\theta \mu_\phi(\theta_{\rm fid})$ and $F_\phi \coloneqq F^G_\phi$.

Proposition C.1. *Let $s_\phi \colon \mathcal{X} \to \mathbb{R}^d$ satisfy Assumptions 3.1(i)–(iii). Let $\varepsilon \coloneqq \|\phi' - \phi\|$ and $\delta_\Sigma \coloneqq 4 L_s M \varepsilon$. If $\kappa_\Sigma \delta_\Sigma < 1$, then*

$$\|\Sigma_{\phi'}^{-1} - \Sigma_\phi^{-1}\|_{\rm op} \le \frac{\kappa_\Sigma^2\, \delta_\Sigma}{1 - \kappa_\Sigma\, \delta_\Sigma}. \tag{9}$$

In particular, $\phi \mapsto \Sigma_\phi^{-1}$ is locally Lipschitz.

Proof. We first bound $\|\Sigma_\phi - \Sigma_{\phi'}\|_{\rm op}$. For ease of notation, given $\phi, \phi' \in \Phi$, we write $v = s_\phi(X)$, $v' = s_{\phi'}(X)$, $\mu = \mu_\phi(\theta_{\rm fid})$, $\mu' = \mu_{\phi'}(\theta_{\rm fid})$. Since $\Sigma_\phi = \mathbb{E}\left[vv^\top\right] - \mu\mu^\top$, it holds that

$$\Sigma_\phi - \Sigma_{\phi'} = \left(\mathbb{E}\left[vv^\top\right] - \mathbb{E}\left[v'(v')^\top\right]\right) - \left(\mu\mu^\top - \mu'(\mu')^\top\right).$$

Let us focus on the right-hand side of the equation. For the first term, we have the identity

$$vv^\top - v'(v')^\top = (v - v')v^\top + v'(v - v')^\top.$$

In expectation, taking operator norms and using the triangle inequality,

$$\left\|\mathbb{E}\left[vv^\top\right] - \mathbb{E}\left[v'(v')^\top\right]\right\|_{\rm op} \le \mathbb{E}\left[\|v - v'\|\,\|v\|\right] + \mathbb{E}\left[\|v'\|\,\|v - v'\|\right].$$

By (i), $\|v - v'\| \le L_s\varepsilon$ pointwise, so this factors out of both expectations. By Jensen's inequality and (ii), $\mathbb{E}[\|v\|] \le (\mathbb{E}[\|v\|^2])^{1/2} \le M$, and similarly $\mathbb{E}[\|v'\|] \le M$. Therefore

$$\left\|\mathbb{E}\left[vv^\top\right] - \mathbb{E}\left[v'(v')^\top\right]\right\|_{\mathrm{op}} \le 2L_sM\varepsilon. \tag{10}$$

For the second term, the same identity gives

$$\mu\mu^\top - \mu'(\mu')^\top = (\mu - \mu')\mu^\top + \mu'(\mu - \mu')^\top,$$

so that

$$\|\mu\mu^\top - \mu'(\mu')^\top\|_{\mathrm{op}} \le \|\mu - \mu'\|\,\|\mu\| + \|\mu'\|\,\|\mu - \mu'\|.$$

By the same reasoning as for the first term,

$$\|\mu\mu^\top - \mu'(\mu')^\top\|_{\mathrm{op}} \le 2L_sM\varepsilon. \tag{11}$$

By (10) and (11), using the triangle inequality, we get that

$$\|\Sigma_\phi - \Sigma_{\phi'}\|_{\mathrm{op}} \le 4L_sM\varepsilon =: \delta_\Sigma. \tag{12}$$

Since $\kappa_\Sigma\delta_\Sigma < 1$ by hypothesis, the matrix $\Sigma_{\phi'}$ is invertible and the resolvent identity $\Sigma_{\phi'}^{-1} - \Sigma_\phi^{-1} = -\Sigma_{\phi'}^{-1}(\Sigma_{\phi'} - \Sigma_\phi)\Sigma_\phi^{-1}$, combined with a Neumann series bound on $\|\Sigma_{\phi'}^{-1}\|_{\mathrm{op}}$ (see e.g. Horn and Johnson [2012], Corollary 5.6.16), yields the statement. □

Proposition C.2. *Let $F_\phi(\theta_{\mathrm{fid}}) = J_\phi^\top\Sigma_\phi^{-1}J_\phi$ be the Fisher information matrix as in Equation* (3)*. Under Assumptions 3.1(i)–(iv), it holds that $\phi \mapsto F_\phi(\theta_{\mathrm{fid}})$ is locally Lipschitz on Φ. More precisely, for all $\phi, \phi' \in \Phi$ with $\varepsilon := \|\phi' - \phi\|$ satisfying $\kappa_\Sigma\delta_\Sigma < 1$, where $\delta_\Sigma = 4L_sM\varepsilon$, it holds that*

$$\|F_{\phi'}(\theta_{\mathrm{fid}}) - F_\phi(\theta_{\mathrm{fid}})\|_{\mathrm{op}} \le \kappa_\Sigma L_J\varepsilon\,(2\|J_\phi\|_{\mathrm{op}} + L_J\varepsilon) + \frac{\kappa_\Sigma^2\delta_\Sigma}{1 - \kappa_\Sigma\delta_\Sigma}\left(\|J_\phi\|_{\mathrm{op}} + L_J\varepsilon\right)^2.$$

Proof. For ease of notation, let $\varepsilon := \|\phi' - \phi\|$, $P := \Sigma_\phi^{-1}$, $P' := \Sigma_{\phi'}^{-1}$, $J := J_\phi$, $J' := J_{\phi'}$, $\Delta J := J' - J$, and $\Delta P := P' - P$. We have that

$$\begin{aligned} F_{\phi'}(\theta_{\mathrm{fid}}) - F_\phi(\theta_{\mathrm{fid}}) &= {J'}^\top P' J' - J^\top P J \\ &= (J + \Delta J)^\top (P + \Delta P)(J + \Delta J) - J^\top P J \\ &= (J + \Delta J)^\top (PJ + P\Delta J + \Delta P J + \Delta P \Delta J) - J^\top P J. \end{aligned}$$

Hence, we have that

$$\begin{aligned} F_{\phi'}(\theta_{\mathrm{fid}}) - F_\phi(\theta_{\mathrm{fid}}) =& J^\top P\,\Delta J + J^\top \Delta P\, J + J^\top \Delta P\,\Delta J + (\Delta J)^\top P\, J \\ &+ (\Delta J)^\top P\,\Delta J + (\Delta J)^\top \Delta P\, J + (\Delta J)^\top \Delta P\,\Delta J. \end{aligned}$$

We recall that $\|P\|_{\mathrm{op}} \le \kappa_\Sigma$ by hypothesis (iii), $\|\Delta J\|_{\mathrm{op}} \le L_J\varepsilon$ by hypothesis (iv), and $\|\Delta P\|_{\mathrm{op}} \le \kappa_\Sigma^2\delta_\Sigma/(1 - \kappa_\Sigma\delta_\Sigma)$ by Proposition C.1. Hence, we get that

$$\|F_{\phi'}(\theta_{\mathrm{fid}}) - F_\phi(\theta_{\mathrm{fid}})\|_{\mathrm{op}} \le \kappa_\Sigma L_J\varepsilon\,(2\|J\|_{\mathrm{op}} + L_J\varepsilon) + \frac{\kappa_\Sigma^2\delta_\Sigma}{1 - \kappa_\Sigma\delta_\Sigma}\left(\|J\|_{\mathrm{op}} + L_J\varepsilon\right)^2$$

whenever $\kappa_\Sigma\delta_\Sigma < 1$. In particular, $\phi \mapsto F_\phi(\theta_{\mathrm{fid}})$ is locally Lipschitz. □

We are now able to prove the main result.

Proof of Theorem 3.2. Let $\Delta F := F_{\phi'}(\theta_{\mathrm{fid}}) - F_\phi(\theta_{\mathrm{fid}})$ and define $g\colon [0,1] \to \mathbb{R}$ by $g(t) := \log\det(F_\phi(\theta_{\mathrm{fid}}) + t\Delta F)$. By Proposition C.2, whenever $4\kappa_\Sigma L_sM\varepsilon < 1$ there exists a constant $C_F > 0$ (depending on κ_Σ, L_s, M, L_J, and $\|J_\phi\|_{\mathrm{op}}$) such that $\|\Delta F\|_{\mathrm{op}} \le C_F\varepsilon$. Set

$$\varepsilon_0(\phi) := \min\left\{\frac{1}{4\kappa_\Sigma L_sM},\ \frac{\lambda_{\min}(F_\phi(\theta_{\mathrm{fid}}))}{C_F}\right\},$$

so that for $\varepsilon < \varepsilon_0(\phi)$ both Proposition C.2 applies and $\|\Delta F\|_{\mathrm{op}} < \lambda_{\min}(F_\phi(\theta_{\mathrm{fid}}))$. By Weyl's inequality for symmetric matrices, $\lambda_{\min}(F_\phi(\theta_{\mathrm{fid}}) + t\Delta F) \ge \lambda_{\min}(F_\phi(\theta_{\mathrm{fid}})) - t\|\Delta F\|_{\mathrm{op}} > 0$ for all $t \in [0,1]$, so $F_\phi(\theta_{\mathrm{fid}}) + t\Delta F$ is positive definite along the segment and g is differentiable with

$$g'(t) = \mathrm{tr}\left((F_\phi(\theta_{\mathrm{fid}}) + t\Delta F)^{-1}\Delta F\right).$$

By the mean value theorem, $|\mathcal{L}(\phi') - \mathcal{L}(\phi)| = |g(1) - g(0)| \le \sup_{t\in[0,1]} |g'(t)|$. Since $|\mathrm{tr}(C)| \le p\,\|C\|_{\mathrm{op}}$ for any $C \in \mathbb{R}^{p\times p}$ and

$$\|(F_\phi(\theta_{\mathrm{fid}}) + t\Delta F)^{-1}\Delta F\|_{\mathrm{op}} \le \frac{\|\Delta F\|_{\mathrm{op}}}{\lambda_{\min}(F_\phi(\theta_{\mathrm{fid}})) - \|\Delta F\|_{\mathrm{op}}},$$

we obtain

$$|\mathcal{L}(\phi') - \mathcal{L}(\phi)| \le \frac{p\,\|\Delta F\|_{\mathrm{op}}}{\lambda_{\min}(F_\phi(\theta_{\mathrm{fid}})) - \|\Delta F\|_{\mathrm{op}}}. \tag{13}$$

Combining (13) with the bound $\|\Delta F\|_{\mathrm{op}} \le C_F\varepsilon$ from Proposition C.2 yields (5). □

D Stability assumptions for TDA pipelines

In this appendix we verify that the pipelines used in Section 4 satisfy Assumptions 3.1(i)–(v) of Theorem 3.2. Recall that the summary map takes the form

$$s_\phi \;=\; C_\phi \circ V_\phi \circ \mathrm{Dgm}_k \circ f_\phi,$$

as in Eq. (2). For non-topological pipelines (IMNN, MLP-only, GNN-only, $\log(C_\ell)$+MOPED, peak counts, wavelet scattering), the persistence and diagram-vectorization stages collapse to the identity. For pipelines with no trainable parameters (Cubical-PI, Cubical-Silhouette, Cubical-Curves, $\log(C_\ell)$+MOPED, peak counts, wavelet scattering, DTM), s_ϕ does not depend on ϕ, so Assumptions (i) and (iv) hold trivially with $L_s = 0$ and $L_J = 0$. We focus the verification on the components that introduce ϕ-dependence, and discuss (ii), (iii) and (v) for all pipelines, learnable or not. We present the verification for a single homological dimension k; concatenating multiple dimensions, or multiple tomographic bins as in Section 4.3, preserves each assumption since each is closed under direct sum of summaries.

Assumption 3.1(i): Lipschitz continuity of s_ϕ in ϕ. The map $\phi \mapsto s_\phi(X)$ is a composition of stages, each Lipschitz in its input. We discuss the contribution of each stage in turn.

Filtrations. The fixed filtrations used in this work (the cubical filtration, the angular power spectrum used in the lensing baseline, and the DTM vertex function on the spiral) do not depend on ϕ and contribute a constant of zero to the Lipschitz constant. The learnable filtrations are neural networks: the per-point MLP g_ϕ and the GNN of Section 4.1, the small CNN prepended to the cubical filtration in TF-CNN-PersLay (Appendix J.2), and the strided CNN encoders used in the IMNN and in the non-TDA baselines MLP-only and GNN-only. Each of these is a finite composition of affine maps with Lipschitz activations (LeakyReLU, GELU); on a compact parameter set Φ and a bounded data domain K, the gradient $\nabla_\phi f_\phi(x)$ is uniformly bounded over $\Phi \times K$, so $\phi \mapsto f_\phi(x)$ is Lipschitz with a constant independent of x. The lower-star extension $f_\phi(e_{ij}) = \|x_i - x_j\| + \max\{f_\phi(x_i), f_\phi(x_j)\}$ used on the Alpha complex (Section 4.1), and analogously the extension to higher-dimensional simplices, is the sum of a ϕ-independent term and a max of ϕ-Lipschitz scalars, hence Lipschitz with the same constant as on vertices.

Persistent homology. The map $f \mapsto \mathrm{Dgm}_k(f)$ is 1-Lipschitz with respect to the bottleneck distance on diagrams and the sup-norm on filtration values, by the bottleneck stability theorem of persistent homology Cohen-Steiner et al. [2007]. For non-topological pipelines this stage is the identity.

Vectorizations. The fixed vectorizations are Lipschitz with respect to standard distances on diagrams: persistence images are 1-Lipschitz in the 1-Wasserstein distance Adams et al. [2017]; persistence silhouettes with uniform weighting are Lipschitz in the ∞-Wasserstein distance on bounded support, as means of 1-Lipschitz tent functions Bubenik [2015]; the differentiable persistence curves of Section J.3 are sums of soft-sigmoid cumulative distribution functions, hence Lipschitz with constant $O(\tau^{-1})$ on bounded support (where τ is the soft-sigmoid temperature reported in Appendix J.3). The learnable vectorization PersLay applies a permutation-invariant set network with spectral normalization on every linear layer (Appendix J.3), which constrains each layer's operator norm to at most one and yields a Lipschitz set network in W_1 on diagrams, with a constant uniformly bounded in ϕ. The identity vectorization used by the IMNN, $\log(C_\ell)$, peak counts, and wavelet scattering is trivially 1-Lipschitz. The non-topological baselines MLP-only and GNN-only end with sum-pooling over N points, which is Lipschitz with constant $\sqrt{N}$ by Cauchy–Schwarz.

Compressors. The MOPED compressor is a fixed linear map at each gradient step. Its operator norm is bounded by the empirical operator norm of the C_ϕ estimate (defined in Appendix B), which is in turn controlled by the Hartlap correction and the condition-number monitoring described in Section 2. The periodic refit of MOPED every 50 steps (Appendix J.4) is treated as an external procedure and does not enter the within-step Lipschitz analysis. The MLP heads used as compressors on the spiral and in the pretrained TF-CNN-PersLay-MLP variant are Lipschitz on compact Φ by the same neural network argument used for filtrations.

By composition, s_ϕ is Lipschitz in ϕ with constant L_s equal to the product of the stage-wise constants.

Assumption 3.1(ii): uniform second moment bound. All simulators used in this work produce data with bounded support or with bounded second moment: the spiral is supported on a square (Section 4.1); the GRF is a centered Gaussian field on a periodic grid with finite total variance (Section 4.2); the κ maps from `sbi_lens` have finite second moment after Gaussian smoothing (Section 4.3). Together with compactness of Φ and the Lipschitz bound on s_ϕ from (i), this yields

$$M^2 \;\coloneqq\; \sup_{\phi\in\Phi} \mathbb{E}_{X\sim p(\cdot|\theta_{\mathrm{fid}})}\big[\|s_\phi(X)\|^2\big] \;<\; \infty.$$

At the level of individual stages: bounded data yield bounded filtration values, hence persistence diagrams supported in a bounded region of $\mathbb{R}^2$ Cohen-Steiner et al. [2007]; the fixed vectorizations of Section J.3 (PI, silhouette, persistence curves) all map bounded diagrams to bounded vectors, and PersLay does so as well since spectral normalization keeps its outputs bounded under bounded inputs. The compressor preserves boundedness as the composition of either a bounded linear map (MOPED) or a Lipschitz MLP on compact Φ.

Assumption 3.1(iii): bound on $\|\Sigma_\phi^{-1}\|_{\mathrm{op}}$. This assumption is verified empirically rather than by a structural argument on the pipeline. We rely on three diagnostics, run for every reported configuration: (a) the sample covariance is computed on a structured batch sized so that $n_s \gg M$ (Sections 4.1–4.3) and corrected by the Hartlap factor Hartlap et al. [2007] for unbiasedness of the precision matrix; (b) zero-variance features of the vectorization output are removed before inversion; (c) Gaussianity of the final summary is checked by per-component Kolmogorov–Smirnov tests at $\alpha = 0.05$ (Appendix B). In all reported runs, Σ_ϕ is invertible with bounded $\|\Sigma_\phi^{-1}\|_{\mathrm{op}}$.

Assumption 3.1(iv): Lipschitz continuity of the Jacobian in ϕ. The Jacobian $J_\theta\mu_\phi(\theta_{\mathrm{fid}})$ is estimated by centered finite differences with step sizes $\Delta = (\Delta_1, \ldots, \Delta_d)$ specified in Appendix J.6. Under Assumption 3.1(i), Jensen's inequality gives $\|\mu_{\phi'}(\theta) - \mu_\phi(\theta)\| \le L_s\|\phi' - \phi\|$ for every θ in the stencil. Applying the triangle inequality column-wise to the finite-difference Jacobian and using $\|A\|_{\mathrm{op}} \le \|A\|_F$ then yields

$$L_J \;\le\; \frac{2\sqrt{d}\,L_s}{\Delta_{\min}}, \qquad \Delta_{\min} \coloneqq \min_{i=1,\ldots,d} \Delta_i.$$

Assumption 3.1(v): non-degeneracy of $F_\phi(\theta_{\mathrm{fid}})$. Since Σ_ϕ^{-1} is positive definite by (iii), positive definiteness of $F_\phi(\theta_{\mathrm{fid}}) = J_\theta\mu_\phi(\theta_{\mathrm{fid}})^\top \Sigma_\phi^{-1} J_\theta\mu_\phi(\theta_{\mathrm{fid}})$ is equivalent to $J_\theta\mu_\phi(\theta_{\mathrm{fid}})$ having full column rank, i.e. the summary being sensitive to all d physical parameters at the fiducial. This is verified empirically in all reported configurations: the marginal uncertainties $\sigma(\theta_i) = (F_\phi^{-1})_{ii}^{1/2}$ reported in Sections 4.1–4.3 are finite and of the expected order of magnitude, and the condition number of F_ϕ is monitored throughout training (see (iii) above and the diagnostics in Appendix B).

E Noisy Spiral Total Fisher

This appendix derives the total Fisher information matrix for the 2D noisy spiral generative model. Points in $\mathbb{R}^2$ are sampled i.i.d. from a known two-component mixture, with a background component p_B and a spiral component p_S. Given a mixing weight $\lambda \in (0, 1)$, the point cloud $\mathcal{X} = \{x_k\}_{k=1}^N \subseteq \mathbb{R}^2$ is obtained by independently sampling each point from

$$p(x;\theta) = \lambda p_B(x) + (1-\lambda)p_S(x;\theta), \tag{14}$$

where $\theta = (\mu, w)^\top$ are the target parameters: μ controls the radial expansion of the spiral and w its width. The background density p_B does not depend on θ and is uniform on a square $\mathcal{S} = [-h, h]^2$ of area $4h^2$. Since the spiral component p_S is most naturally expressed in polar coordinates (ρ, Φ), with $x = \rho\cos\Phi$ and $y = \rho\sin\Phi$, we work in this system throughout. The Jacobian determinant of the change of variables is $|J| = \rho$, and in these coordinates the square $\mathcal{S}$ is described by

$$\mathcal{S}' = \left\{ (\rho, \Phi) : 0 \le \Phi < 2\pi,\ 0 \le \rho \le \frac{h}{\max(|\cos\Phi|, |\sin\Phi|)} \right\}. \tag{15}$$

The background density therefore reads

$$p_B(\rho, \Phi) = \frac{\rho}{4h^2}\, \mathbb{I}_{\mathcal{S}'}(\rho, \Phi), \tag{16}$$

where $\mathbb{I}_A$ denotes the indicator function of the set A. Let us now focus on the spiral component p_S. The spiral is sampled hierarchically as follows:

1. an angular coordinate t is drawn uniformly on $[0, 4\pi)$, so that the spiral completes two full turns;
2. given t, the radial coordinate is perturbed by Gaussian noise, $\rho \mid t \sim \mathcal{N}(\mu t, w^2)$;
3. the sampled point is mapped to Cartesian coordinates via $(x, y) = (\rho\cos t, \rho\sin t)$.

Since t and $t + 2\pi$ map to the same point in $\mathbb{R}^2$, to obtain an injective map we reparametrize the generative process on $\Phi \in [0, 2\pi)$ by marginalizing over the winding. Setting $\Phi \coloneqq t \bmod 2\pi$, the two preimages $t \in \{\Phi, \Phi + 2\pi\}$ are equally likely conditionally on Φ, and the hierarchical sampling becomes $\Phi \sim \mathcal{U}(0, 2\pi)$, and $\rho \mid \Phi \sim \frac{1}{2}\mathcal{N}(\mu\Phi, w^2) + \frac{1}{2}\mathcal{N}(\mu(\Phi + 2\pi), w^2)$ [5]. The spiral density therefore reads

$$p_S(\rho, \Phi \mid \theta) = \frac{1}{2\pi} \cdot \frac{1}{2\sqrt{2\pi}\, w} \left[\exp\left(-\frac{(\rho - \mu\Phi)^2}{2w^2} \right) + \exp\left(-\frac{(\rho - \mu(\Phi + 2\pi))^2}{2w^2} \right) \right] \mathbb{I}_{\mathcal{S}''}(\rho, \Phi), \tag{17}$$

supported on $\mathcal{S}'' = \{(\rho, \Phi) \colon 0 \le \Phi < 2\pi,\ \rho \ge 0\}$. Since the sample is i.i.d. from $p(\cdot; \theta)$, the total Fisher information matrix in polar coordinates reads Kay [1997]

$$F_X[i, j](\theta) = N \iint \frac{\partial_{\theta_i} p(\rho, \Phi; \theta)\, \partial_{\theta_j} p(\rho, \Phi; \theta)}{p(\rho, \Phi; \theta)}\, d\rho\, d\Phi. \tag{18}$$

Since only p_S depends on θ, we have $\partial_\theta p = (1 - \lambda)\, \partial_\theta p_S$, and (18) specializes to

$$F_X[i, j](\theta) = N \iint \frac{(1 - \lambda)^2\, \partial_{\theta_i} p_S\, \partial_{\theta_j} p_S}{\lambda\, p_B + (1 - \lambda)\, p_S}\, d\rho\, d\Phi, \tag{19}$$

where the dependence on (ρ, Φ, θ) has been suppressed for readability. To lighten the notation, we introduce $\zeta_1 = \frac{\rho - \mu\Phi}{w}$, $\zeta_2 = \frac{\rho - \mu(\Phi + 2\pi)}{w}$, and $g_\ell = \frac{1}{\sqrt{2\pi}\, w} e^{-\zeta_\ell^2/2}$ for $\ell = 1, 2$. A direct computation then yields

$$\partial_w\, p_S = \frac{1}{2\pi} \cdot \frac{1}{2} \left[g_1\, \frac{\zeta_1^2 - 1}{w} + g_2\, \frac{\zeta_2^2 - 1}{w} \right], \tag{20}$$

$$\partial_\mu\, p_S = \frac{1}{2\pi} \cdot \frac{1}{2} \left[g_1\, \frac{\zeta_1\, \Phi}{w} + g_2\, \frac{\zeta_2\, (\Phi + 2\pi)}{w} \right]. \tag{21}$$

The total Fisher information matrix $F_X(\theta)$ is then obtained by substituting (20)–(21) into (19). The resulting integral admits no known closed form and is evaluated numerically.

F Additional Noisy Spiral Results

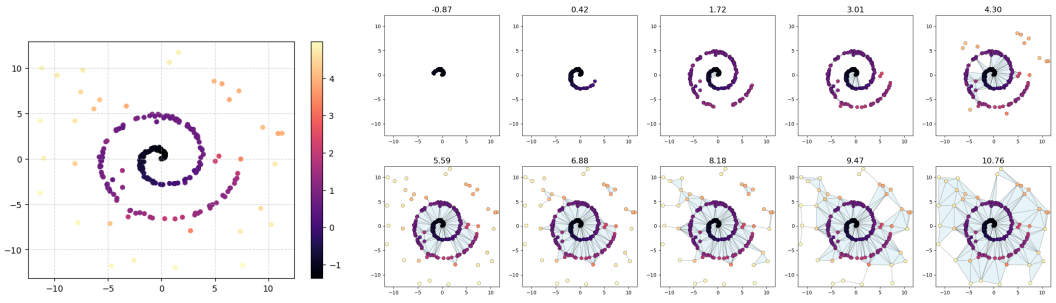
We report the results for the spiral experiment for other four fiducial parameter values in Table 4. In Figure 2 we plot the vertex values and the corresponing filtration learned by TF-TDA-MLP method.

To highlight the interpretability of the TF-TDA-MLP, we compare its learned filtration with the vertex mapping produced by the MLP-only baseline. Since the latter maps each point x to a two-dimensional statistic $f_\phi(x) \in \mathbb{R}^2$ before sum-pooling, we visualize both components in Fig. 3.

[5] The Gaussian conditional $\rho|\Phi$ places a small amount of mass on the region $\rho < 0$. For the fiducial $\mu = 0.6$, $w = 0.1$, this mass, concentrated near $\Phi \to 0^+$, is bounded by 10^{-2} and we neglect it in the following.

Table 4: Complete spiral results for all fiducial configurations $(\mu_{\text{fid}}, w_{\text{fid}})$, averaged over 5 seeds. Entries are mean $\pm$ standard deviation of $\log|F|$.

Method	$(0.6, 0.1)$	$(0.8, 0.1)$	$(0.7, 0.2)$	$(0.5, 0.2)$	$(0.7, 0.3)$
Total	*24.27*	*24.24*	*21.36*	*21.41*	*19.64*
Fixed topological ablation: persistence without TopoFisher learning					
DTM fixed filtration	17.18 ± 0.06	16.34 ± 0.25	15.83 ± 0.09	16.65 ± 0.06	15.16 ± 0.05
TopoFisher summaries: our learned topological contributions					
TF-TDA-MLP	$\mathbf{18.99 \pm 0.30}$	17.47 ± 0.63	$\mathbf{17.50 \pm 0.28}$	18.32 ± 0.37	$\mathbf{16.66 \pm 0.07}$
TF-TDA-GNN	16.20 ± 0.19	15.57 ± 0.37	15.14 ± 0.14	16.63 ± 0.43	14.72 ± 0.11
Unconstrained neural references: non-topological summaries					
MLP-only	18.96 ± 0.58	$\mathbf{17.88 \pm 0.49}$	17.48 ± 0.11	$\mathbf{18.40 \pm 0.21}$	16.54 ± 0.19
GNN-only	18.26 ± 0.25	17.45 ± 0.12	16.78 ± 0.16	17.68 ± 0.06	15.87 ± 0.08



(a) Learned vertex values $f_\phi(x)$. (b) Resulting filtration.

Figure 2: Visualization of the TF-TDA-MLP pipeline. (a) The model assigns filtration values to vertices to "unroll" the spiral geometry. (b) The corresponding filtration effectively prioritizes the manifold while filtering background noise.

In contrast to the topological approach, the purely neural baseline lacks a clear geometric interpretation. The network appears to concentrate the information on a small subset of points that assume extreme values to drive the sum-pooling operation, but their spatial distribution does not correspond to identifiable features of the manifold.

G Total Fisher information for the GRF benchmark

Let $x \in \mathbb{R}^{N\times N}$ be a zero-mean isotropic and homogeneous Gaussian random field on a periodic grid, and let $\tilde{x}_{\mathbf{k}}$ denote its discrete Fourier coefficients. Under the simulator used in Section 4.2, the field is fully specified by the isotropic power spectrum

$$P(k;\theta) = A_s \left(\frac{k}{k_{\text{pivot}}}\right)^{-B}, \qquad \theta = (A_s, B), \tag{22}$$

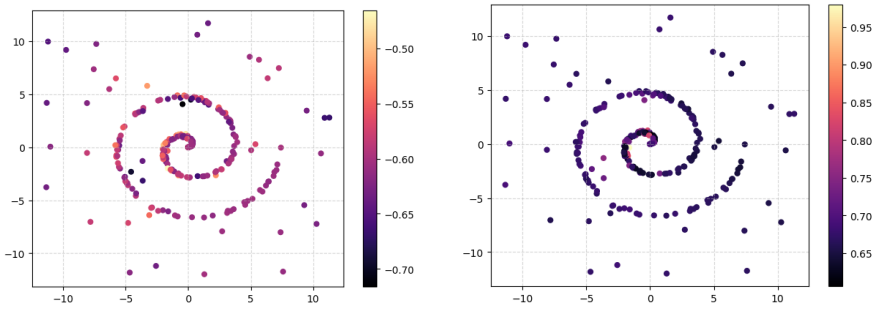
with $k = \|\mathbf{k}\|$ and k_{pivot} the geometric mean of the nonzero Fourier radii. Since the field is Gaussian with zero mean, the exact data likelihood is multivariate Gaussian with diagonal covariance $\Sigma(\theta)$ in Fourier space. The Fisher matrix is therefore

$$F_{ij}(\theta) = \frac{1}{2}\,\mathrm{Tr}\left[\Sigma^{-1}(\theta)\,\partial_{\theta_i}\Sigma(\theta)\,\Sigma^{-1}(\theta)\,\partial_{\theta_j}\Sigma(\theta)\right]. \tag{23}$$

Because $\Sigma(\theta)$ is diagonal in the Fourier basis, Eq. (23) reduces to a sum over modes:

$$F_{ij}(\theta) = \frac{1}{2}\sum_{\mathbf{k}\neq 0} \frac{\partial_{\theta_i} P(k;\theta)\,\partial_{\theta_j} P(k;\theta)}{P(k;\theta)^2} = \frac{1}{2}\sum_{\mathbf{k}\neq 0} \partial_{\theta_i}\log P(k;\theta)\,\partial_{\theta_j}\log P(k;\theta). \tag{24}$$

Here the sum runs over all nonzero discrete Fourier modes of the $N \times N$ lattice; the prefactor $1/2$ accounts for the reality condition $\tilde{x}_{-\mathbf{k}} = \tilde{x}^*_{\mathbf{k}}$. Equivalently, one may sum only over an independent half-plane of modes and drop the factor $1/2$.



(a) Component $f_\phi(x)^1$. (b) Component $f_\phi(x)^2$.

Figure 3: Vertex-wise outputs for the two components learned by the MLP-only baseline.

For the power-law spectrum used here,

$$\partial_{A_s} \log P(k;\theta) = \frac{1}{A_s}, \qquad \partial_B \log P(k;\theta) = -\log\left(\frac{k}{k_{\text{pivot}}}\right). \tag{25}$$

Substituting into Eq. (24) gives

$$F_{A_s A_s} = \frac{1}{2A_s^2} \sum_{\mathbf{k}\neq 0} 1 = \frac{N_{\text{nz}}}{2A_s^2}, \tag{26}$$

$$F_{A_s B} = -\frac{1}{2A_s} \sum_{\mathbf{k}\neq 0} \log\left(\frac{k}{k_{\text{pivot}}}\right), \tag{27}$$

$$F_{BB} = \frac{1}{2} \sum_{\mathbf{k}\neq 0} \log^2\left(\frac{k}{k_{\text{pivot}}}\right), \tag{28}$$

where $N_{\text{nz}} = N^2 - 1$ is the number of nonzero Fourier modes.

The choice of pivot scale

$$k_{\text{pivot}} = \exp\left(\frac{1}{N_{\text{nz}}} \sum_{\mathbf{k}\neq 0} \log k\right) \tag{29}$$

makes the cross term vanish exactly:

$$\sum_{\mathbf{k}\neq 0} \log\left(\frac{k}{k_{\text{pivot}}}\right) = 0 \qquad \Longrightarrow \qquad F_{A_s B} = 0. \tag{30}$$

Thus the total Fisher matrix is diagonal,

$$F(\theta) = \begin{pmatrix} \dfrac{N_{\text{nz}}}{2A_s^2} & 0 \\ 0 & \dfrac{1}{2}\sum_{\mathbf{k}\neq 0} \log^2\left(\dfrac{k}{k_{\text{pivot}}}\right) \end{pmatrix}. \tag{31}$$

Two useful consequences follow immediately. First, the bound is independent of the fiducial slope B_0, since $\partial_B \log P$ does not depend on B. Second, the overall box-size normalization cancels in k/k_{pivot}, so only the discrete lattice of mode radii matters.

For our benchmark, $N = 64$, hence $N_{\text{nz}} = 64^2 - 1 = 4095$ and

$$F_{A_s A_s} = \frac{4095}{2} = 2047.5 \qquad \Longrightarrow \qquad \sigma(A_s) = F_{A_s A_s}^{-1/2} = 0.0221. \tag{32}$$

The second diagonal entry is obtained by evaluating the lattice sum in Eq. (28) numerically on the nonzero 64×64 Fourier grid, yielding

$$F_{BB} \approx 5.16 \times 10^2, \qquad \sigma(B) = F_{BB}^{-1/2} \approx 0.044. \tag{33}$$

Finally,

$$\log |F|_{\text{tot}} = \log\big(F_{A_s A_s} F_{BB}\big) \approx 13.89, \tag{34}$$

which is the denominator of efficiency used throughout Section 4.2.

H Further GRF results

H.1 per-B results

We report the per-spectral-index results for the GRF benchmark of Section 4.2. All entries are computed at $N = 64$ and averaged over 5 independent seeds for each fiducial value $B \in \{-2, -1, 0, 1, 2\}$. The theoretical optimum is $\log |F_X| = 13.89$ for every value of B.

Table 5: GRF results for all spectral indices $B \in \{-2, -1, 0, 1, 2\}$, $N = 64$, averaged over 5 seeds. Entries are mean $\pm$ standard deviation of $\log |F|$. The theoretical optimum is $\log |F_X| = 13.89$ for every B.

Method	$B = -2$	$B = -1$	$B = 0$	$B = 1$	$B = 2$
Total			*13.89*		
Fixed topological ablations: persistence without TopoFisher learning					
Cubical-PI	10.573 ± 0.008	11.313 ± 0.006	11.997 ± 0.005	12.085 ± 0.005	11.682 ± 0.008
Cubical-Silhouette	9.895 ± 0.011	10.597 ± 0.011	11.304 ± 0.005	11.256 ± 0.004	10.369 ± 0.009
Cubical-Curves	7.073 ± 0.016	8.685 ± 0.006	10.372 ± 0.005	10.486 ± 0.007	9.467 ± 0.008
TopoFisher summaries: our learned topological contributions					
TF-Cubical-PersLay	10.561 ± 0.008	11.296 ± 0.004	11.973 ± 0.006	12.046 ± 0.003	11.627 ± 0.005
TF-CNN-PersLay	11.685 ± 0.051	11.868 ± 0.015	12.070 ± 0.024	12.033 ± 0.042	11.788 ± 0.036
Unconstrained Fisher-neural reference: non-topological summary					
IMNN	$\mathbf{12.139 \pm 0.037}$	$\mathbf{12.273 \pm 0.028}$	$\mathbf{12.343 \pm 0.040}$	$\mathbf{12.344 \pm 0.045}$	$\mathbf{12.327 \pm 0.052}$

Table 6: GRF marginal uncertainties across spectral indices, averaged over 5 seeds. Each cell reports $\sigma(A_s)/\sigma(B)$. The theoretical values are $\sigma(A_s) = 0.022$ and $\sigma(B) = 0.044$ for every B.

Method	$B = -2$	$B = -1$	$B = 0$	$B = 1$	$B = 2$
Total			0.022/0.044		
Fixed topological ablations: persistence without TopoFisher learning					
Cubical-PI	0.0436/0.1445	0.0349/0.1049	0.0327/0.0759	0.0345/0.0690	0.0405/0.0723
Cubical-Silhouette	0.0529/0.1967	0.0381/0.1445	0.0349/0.1006	0.0411/0.0916	0.0625/0.1104
Cubical-Curves	0.1845/0.7144	0.0644/0.3410	0.0379/0.1480	0.0524/0.1190	0.0929/0.1376
TopoFisher summaries: our learned topological contributions					
TF-Cubical-PersLay	0.0433/0.1452	0.0349/0.1054	0.0330/0.0761	0.0352/0.0689	0.0420/0.0713
TF-CNN-PersLay	0.0358/0.0814	0.0352/0.0754	0.0338/0.0708	0.0345/0.0706	0.0376/0.0735
Unconstrained Fisher-neural reference: non-topological summary					
IMNN	0.0322/0.0693	0.0320/0.0657	0.0322/0.0636	0.0321/0.0639	0.0320/0.0636

H.2 Training across multiple fiducials

The TopoFisher loss of Eq. (4) is local to one fiducial value. To test whether one representation can serve a range of parameter values, we train a single PersLay vectorizer, shared across the $K = 5$ GRF fiducials $B_0 \in \{-2, -1, 0, 1, 2\}$, on top of the fixed periodic cubical filtration, with either the averaged or the worst-case objective,

$$\mathcal{L}_{\mathrm{avg}}(\phi) = -\frac{1}{K}\sum_{k=1}^{K} \log\big|\hat{F}^{G}_{\phi,k}\big|, \qquad \mathcal{L}_{\min}(\phi) = -\min_{k=1,\dots,K} \log\big|\hat{F}^{G}_{\phi,k}\big|,$$

where $\hat{F}^{G}_{\phi,k}$ is the Gaussian Fisher estimate at the k-th fiducial. We compare against specialists of the same architecture trained separately at each B_0. At every fiducial, the covariance, finite-difference Jacobian and MOPED operator are estimated separately for the shared and the specialist models, so the experiment tests whether a single nonlinear representation remains near-optimal across the range, not whether a single compressor does. We report the regret, the specialist's $\log |\hat{F}|$ minus the shared model's, computed on a common evaluation bank; positive values favour the specialist.

Table 7: Multi-fiducial training on GRFs. Regret (nats) of a single PersLay vectorizer shared across $B_0 \in \{-2, \dots, 2\}$ relative to per-fiducial specialists, $\log|\hat{F}|_{\rm spec} - \log|\hat{F}|_{\rm shared}$; positive values favour the specialist. Covariance, Jacobian and MOPED are estimated separately at each fiducial for both models. The last row is an untrained shared vectorizer.

Shared model	$B_0 = -2$	$B_0 = -1$	$B_0 = 0$	$B_0 = 1$	$B_0 = 2$
Trained, averaged objective	$+0.034$	-0.015	-0.011	-0.027	$+0.211$
Trained, worst-case objective	$+0.015$	$+0.012$	$+0.049$	$+0.015$	$+0.311$
Untrained	$+0.044$	$+0.018$	-0.009	-0.042	$+0.194$

Table 7 shows that the shared representation is within 0.05 nats of the specialists at $B_0 = -2, \dots, 1$ under both objectives. The largest regret occurs at the edge of the range, $B_0 = 2$: 0.21 nats for the averaged and 0.31 nats for the worst-case objective, corresponding to Fisher ellipses about 11% and 17% larger in area. An untrained shared vectorizer (random weights, with only the input normalization fitted) shows comparable regret, at most 0.19 nats. On this benchmark the result therefore indicates that a single cubical-persistence representation serves the whole range of spectral indices, rather than that the trained weights generalize across fiducials; this is consistent with the small gain from training PersLay on a fixed cubical filtration. Each training fiducial requires its own covariance and finite-difference simulations, so the simulation cost grows linearly with K. Testing the multi-fiducial objective with a learned filtration, where training does add information, is left for future work.

H.3 Scaling with the number of parameters

To test how TopoFisher behaves as the number of inferred parameters grows, we extend the power-law spectrum of Section 4.2 with a running of the spectral index and a running of the running,

$$P(k;\theta) = A_s \left(\frac{k}{k_{\rm pivot}}\right)^{-B - \frac{\alpha}{2}\ln(k/k_{\rm pivot}) - \frac{\beta}{6}\ln^2(k/k_{\rm pivot})},$$

and infer $d = 2$, 3 and 4 parameters, (A_s, B), (A_s, B, α) and (A_s, B, α, β), on 64×64 fields. The field remains Gaussian with a diagonal covariance in Fourier space, so Eq. (24) still gives the total Fisher information exactly and provides an analytic ceiling at every d. We compare TF-CNN-PersLay, compressed with MOPED to d dimensions, with the IMNN, whose dense head outputs d summaries; both are trained with the local Fisher objective and evaluated with exact spectral norms. The number of finite-difference simulations grows as $2d\,n_d$.

Table 8 reports the results. The ceiling increases with d, and so does the deficit of both summaries relative to it. Per inferred parameter, the deficit of the IMNN stays close to 0.8 nats, whereas that of TF-CNN-PersLay grows from 0.85 to 1.16 nats; at $d = 4$, the uncertainty volume of TF-CNN-PersLay is about 10 times the full-data limit, against about 5 times for the IMNN. In this benchmark the topological summary therefore becomes relatively less efficient as parameters are added, on top of the linear growth in simulation cost common to local-Fisher methods. We did not investigate the cause; richer vectorizations or several filtrations are natural directions to explore.

Table 8: Scaling with the number of inferred parameters on GRFs with a running spectral index ($N = 64$). The ceiling is the exact total $\log|F_X|$; the deficit is the ceiling minus the summary's $\log|F|$. Mean $\pm$ standard deviation over 3 seeds for TF-CNN-PersLay and 5 seeds for the IMNN.

			$\log\lvert F\rvert$		**Deficit (nats)**	
d	**Parameters**	**Ceiling**	TF-CNN-PersLay	IMNN	TF-CNN-PersLay	IMNN
2	(A_s, B)	13.89	12.19 ± 0.08	12.32 ± 0.01	1.70	1.57
3	(A_s, B, α)	18.61	15.72 ± 0.10	16.22 ± 0.03	2.88	2.38
4	(A_s, B, α, β)	21.23	16.59 ± 0.14	18.00 ± 0.01	4.64	3.23

I Further Lensing results

This appendix provides the per-bin and transfer-learning details for the weak lensing experiment of Section 4.3. All results use the $\sigma = 2'$ smoothed 512×512 convergence maps and the parameter pair

$(\Omega_{\rm m}, \sigma_8)$. For each tomographic bin and each method, we report the mean and standard deviation across 5 independent seeds. The per-bin quantities in Tables 9 are useful for diagnosing how information changes with source redshift, but the Survey-level constraints reported in the main text are obtained by adding the Fisher matrices across bins, and then computing $\log|F_{\rm tot}|$ and $\sqrt{(F_{\rm tot}^{-1})_{ii}}$.

Per-bin behavior. The per-bin log-determinants in Table 9 show three main trends. First, the $\log(C_\ell)$ baseline is comparatively flat across bins, with $\log|F| \simeq 10.2$–10.6. Second, fixed topological summaries recover substantially more information than the power spectrum in the higher-redshift bins, where the lensing signal is stronger. Third, both TopoFisher summaries improve over all fixed topological vectorizations in every bin and increase monotonically with source redshift, TF-Cubical-PersLay from $\log|F| = 14.66$ to 18.59 and TF-CNN-PersLay from 14.86 to 19.15. The IMNN is the strongest in every bin, from 14.87 in bin 0 to 20.06 in bin 4.

Table 9: Weak lensing per-bin results for $(\Omega_{\rm m}, \sigma_8)$ at $\sigma = 2'$ smoothing. Each cell reports mean $\pm$ standard deviation of $\log|F|$ across 5 seeds. Bin index increases with effective source redshift. Transfer rows are trained on lognormal maps and evaluated on LPT maps without retraining.

Method	bin 0	bin 1	bin 2	bin 3	bin 4
External cosmology baseline					
$\log(C_\ell)$+MOPED	10.198 ± 0.086	10.574 ± 0.094	10.583 ± 0.080	10.499 ± 0.072	10.279 ± 0.069
Fixed topological ablations: persistence without TopoFisher learning					
Cubical-PI	12.672 ± 0.017	15.165 ± 0.025	16.055 ± 0.019	16.364 ± 0.020	16.346 ± 0.025
Cubical-Curves	8.935 ± 0.249	13.448 ± 0.074	14.912 ± 0.015	15.464 ± 0.014	15.296 ± 0.017
Cubical-Silhouette	13.048 ± 0.019	13.053 ± 0.020	12.673 ± 0.019	12.681 ± 0.016	13.259 ± 0.015
TopoFisher summaries: our learned topological contributions					
TF-Cubical-PersLay	14.660 ± 0.007	16.947 ± 0.006	17.855 ± 0.011	18.350 ± 0.014	18.590 ± 0.018
TF-CNN-PersLay	14.860 ± 0.091	17.497 ± 0.153	18.429 ± 0.173	18.783 ± 0.200	19.145 ± 0.179
Unconstrained Fisher-neural reference: non-topological summary					
IMNN	$\mathbf{14.871 \pm 0.057}$	$\mathbf{17.663 \pm 0.045}$	$\mathbf{18.925 \pm 0.044}$	$\mathbf{19.704 \pm 0.040}$	$\mathbf{20.056 \pm 0.073}$
Transfer Fisher: lognormal-trained summaries evaluated on LPT maps					
TF-Cubical-PersLay	-0.23 ± 0.31	5.35 ± 0.25	7.03 ± 0.52	8.94 ± 0.48	10.69 ± 0.40
TF-CNN-PersLay	$\mathbf{1.353 \pm 0.766}$	$\mathbf{5.887 \pm 0.436}$	$\mathbf{7.988 \pm 0.428}$	$\mathbf{9.747 \pm 0.179}$	$\mathbf{11.331 \pm 0.188}$
IMNN	-5.040 ± 0.965	-1.923 ± 0.765	-2.099 ± 0.683	-2.042 ± 1.779	0.580 ± 1.420

Table 10: Weak lensing per-bin marginal uncertainties for in-distribution lognormal simulations. Each cell reports $\sigma(\Omega_{\rm m})/\sigma(\sigma_8)$ averaged over 5 seeds. These are per-bin constraints only; survey-level constraints are obtained by adding Fisher matrices across bins.

Method	bin 0	bin 1	bin 2	bin 3	bin 4
External cosmology baseline					
$\log(C_\ell)$	0.1242/0.1749	0.0975/0.1401	0.0866/0.1333	0.0788/0.1311	0.0712/0.1296
Fixed topological ablations: persistence without TopoFisher learning					
Cubical-PI	0.0848/0.0987	0.0481/0.0461	0.0344/0.0305	0.0299/0.0246	0.0246/0.0178
Cubical-Curves	0.4600/0.4873	0.0904/0.0826	0.0496/0.0421	0.0365/0.0291	0.0306/0.0234
Cubical-Silhouette	0.0837/0.0884	0.0783/0.0801	0.0778/0.0853	0.0676/0.0789	0.0442/0.0529
TopoFisher summaries: our learned topological contributions					
TF-Cubical-PersLay	0.0489/0.0505	0.0285/0.0237	0.0227/0.0172	0.0195/0.0134	0.0167/0.0087
TF-CNN-PersLay	0.0414/0.0419	0.0215/0.0173	0.0174/0.0126	0.0161/0.0106	0.0133/0.0070
Unconstrained Fisher-neural reference: non-topological summary					
IMNN	0.0443/0.0450	0.0215/0.0172	0.0155/0.0111	0.0120/0.0077	0.0098/0.0052

I.1 Transfer to LPT simulations

The transfer experiment trains all learned stages on the lognormal `sbi_lens` simulator and evaluates them without retraining on LPT-based maps at the same fiducial cosmology, resolution, and smoothing scale. In the protocol of Table 3, the linear apparatus is re-estimated on LPT: the covariance (hence the precision matrix), the finite-difference Jacobian, and the MOPED operator. For the IMNN, which has no MOPED stage, only the covariance and Jacobian are re-estimated.

All learned summaries lose information under the shift: about 8.8–8.9 nats for the two TopoFisher summaries and 13.4 nats for the IMNN, so that the topological summaries end 3.4–3.9 nats above it. Learning a pixel-space filtration does not by itself make a summary simulator-specific: TF-CNN-PersLay transfers slightly better than TF-Cubical-PersLay, and markedly better once the apparatus is frozen (Table 11). The power spectrum loses no information (13.84 on LPT versus 13.78 on lognormal), consistent with the two simulators agreeing more closely in their two-point statistics than in their higher-order structure; summaries that extract more non-Gaussian information are correspondingly more exposed to the shift. Under transfer, the Fisher ellipses of all summaries become more degenerate along the $\Omega_{\rm m}$–σ_8 direction, so marginal uncertainties degrade more than $\log|F|$ alone suggests. For the TopoFisher summaries the per-bin loss is largest in the lowest-redshift bin and decreases with source redshift, whereas the IMNN loses a comparable amount in every bin (Table 9).

Fully frozen variants. To separate transfer of the representation from target-domain recalibration, we progressively freeze the linear apparatus at its lognormal values. We label each variant by whether covariance, Jacobian and MOPED are re-estimated on LPT (T) or kept from lognormal (F): (a) $(\mathrm{T},\mathrm{T},\mathrm{T})$ is the protocol above; (b) $(\mathrm{T},\mathrm{T},\mathrm{F})$ keeps the lognormal MOPED operator; (c) $(\mathrm{F},\mathrm{T},\mathrm{F})$ also keeps the lognormal precision matrix, so that only the parameter response of the frozen summary, its Jacobian, is measured on LPT. For the IMNN, (b) coincides with (a). Variants (b) and (c) evaluate LPT data with a lognormal noise model: they diagnose whether the learned parameter response survives the change of simulator, are not Fisher informations of the summary on LPT, and can exceed (a). They are therefore comparable only across the learned summaries, and not with the fixed baselines of Table 3, for which (b) and (c) were not computed.

Table 11 reports the results. TF-CNN-PersLay retains the most information among the learned summaries in every variant; in (c) it keeps $\log|F| = 14.01$, compared with 8.93 for TF-Cubical-PersLay and 8.10 for the IMNN. Its representation therefore transfers without target-domain recalibration; for this summary, re-estimating the covariance on LPT even lowers $\log|F|$ (from 14.01 in (c) to 12.82 in (a)), because the LPT covariance is broader along its response directions. TF-Cubical-PersLay instead relies on the recalibration: with the lognormal MOPED operator alone (b) it falls below the IMNN, and in (c) it is only marginally above it.

Table 11: Fully frozen transfer on weak lensing (survey level, mean $\pm$ standard deviation over 5 seeds). Each variant states whether covariance, Jacobian and MOPED are re-estimated on LPT (T) or kept from lognormal (F); (a) is the transfer protocol of Table 3. Local $\log|F|$ is the in-distribution value on lognormal maps. Variants (b) and (c) apply a lognormal noise model to LPT data and are comparable only across learned summaries. The IMNN has no MOPED stage, so (b) coincides with (a).

	Local	**(a)** $(\mathrm{T},\mathrm{T},\mathrm{T})$		**(b)** $(\mathrm{T},\mathrm{T},\mathrm{F})$		**(c)** $(\mathrm{F},\mathrm{T},\mathrm{F})$	
Method	$\log\|F\|$	$\log\|F\|$	$(\sigma_{\Omega_{\rm m}},\sigma_{\sigma_8})$	$\log\|F\|$	$(\sigma_{\Omega_{\rm m}},\sigma_{\sigma_8})$	$\log\|F\|$	$(\sigma_{\Omega_{\rm m}},\sigma_{\sigma_8})$
IMNN	22.35 ± 0.04	8.92 ± 0.80	$(0.247, 0.391)$	–	–	8.10 ± 0.63	$(0.267, 0.432)$
TF-Cubical-PersLay	21.18 ± 0.01	12.27 ± 0.05	$(0.094, 0.143)$	7.72 ± 0.07	$(0.367, 0.575)$	8.93 ± 0.12	$(0.235, 0.402)$
TF-CNN-PersLay	21.60 ± 0.08	12.82 ± 0.04	$(0.073, 0.110)$	11.06 ± 0.58	$(0.145, 0.203)$	14.01 ± 0.66	$(0.074, 0.103)$

I.2 Simulation-based inference studies

We perform simulation-based inference (also known as likelihood-free inference) using Neural Posterior Estimation [NPE; Papamakarios and Murray, 2016, Lueckmann et al., 2017, Greenberg et al., 2019]. Rather than approximating the intractable likelihood $p(d \mid \theta)$, NPE directly targets the posterior by training a conditional density estimator $q_\varphi(\theta \mid d)$, parameterised by neural-network weights φ, on simulated (θ, d) pairs drawn from the joint $p(\theta, d) = p(\theta)\, p(d \mid \theta)$.

We adopt a conditional normalizing flow as the density estimator, specifically a conditional RealNVP [Dinh et al., 2017] composed of a stack of affine coupling layers. In each coupling layer, the scale and shift parameters of the affine transformation are produced by a small MLP conditioner that takes as input both the masked half of the parameter vector and the conditioning summary statistic d.

Training proceeds by minimising the forward Kullback–Leibler divergence between the true posterior and the variational approximation,

$$\mathcal{L}(\varphi) \;=\; \mathbb{E}_{p(d)}\big[D_{\mathrm{KL}}\big(p(\theta \mid d) \,\big\|\, q_\varphi(\theta \mid d)\big)\big]\,, \tag{35}$$

which, up to a φ-independent constant, is equivalent to maximising the expected conditional log-likelihood of the flow on samples from the joint,

$$\varphi^\star \;=\; \arg\max_{\varphi}\; \mathbb{E}_{p(\theta,d)}[\log q_\varphi(\theta \mid d)]\,. \tag{36}$$

We estimate this expectation by Monte Carlo over a training set of simulator draws. At inference time we condition the trained flow on the observed summary d_{obs} and draw samples directly from $q_{\varphi^\star}(\theta \mid d_{\mathrm{obs}})$.

We note that we perform our SBI using only auto-bins (i.e. correlations within the same bin). In principle, we could improve performance by including cross-correlations between different redshift bins [compare Martinet et al., 2021]. Likewise, in a realistic cosmological analysis we would compute the cross-correlation between all bin pairs in the power spectrum analysis. This is beyond the scope of demonstrating the feasibility of our TopoFisher pipeline, and uniformly disregarding cross-correlations between bins ensures a fair comparison between summary statistics. We reserve a more realistic lensing optimization that includes cross-correlations between bins for future work. We also note that the NPE $\log|F|$ of the power spectrum and of Cubical-Silhouette exceed their local Gaussian Fisher values (15.15 versus 13.78 and 18.73 versus 16.37). The local values are computed after MOPED compression, which is lossless in theory but loses information when the derivatives and covariance are noisy [Asgari and Schneider, 2015].

Table 12: NPE posterior quality on weak lensing. $\log|F|$ and $(\sigma_{\Omega_c}, \sigma_{\sigma_8})$ are computed from flow samples conditioned on the fiducial mock; Global NLL is averaged over held-out simulations drawn from the prior, fiducial NLL over independent realizations at the fiducial cosmology (nats, lower is better). Mean $\pm$ standard deviation over seeds.

Summary	n_{train}	**Seeds**	$\log\lvert F\rvert$	$(\sigma_{\Omega_c}, \sigma_{\sigma_8})$	**NLL (global)**	**NLL (fiducial)**
Cosmology baselines						
$\log(C_\ell)$	20k	5	15.15 ± 0.12	$(0.057, 0.039)$	-5.61 ± 0.08	-5.66 ± 0.09
Peak counts	100k	5	16.13 ± 0.13	$(0.042, 0.030)$	-5.50 ± 0.05	-5.29 ± 0.08
Wavelet scattering	100k	5	18.78 ± 0.04	$(0.022, 0.018)$	-6.69 ± 0.04	-6.48 ± 0.16
Fixed topological ablations: persistence without TopoFisher learning						
Cubical-Silhouette	100k	5	18.73 ± 0.10	$(0.018, 0.013)$	-6.42 ± 0.07	-6.45 ± 0.07
Cubical-Curves	20k	5	16.04 ± 0.06	$(0.034, 0.029)$	-5.12 ± 0.04	-5.10 ± 0.10
Cubical-PI	100k	5	15.92 ± 0.49	$(0.028, 0.027)$	-4.59 ± 0.35	-5.05 ± 0.34
TopoFisher summaries: our learned topological contributions						
TF-Cubical-PersLay	100k	5	19.72 ± 0.07	$(0.014, 0.009)$	-6.91 ± 0.04	-6.96 ± 0.08
TF-CNN-PersLay	100k	5	19.84 ± 0.13	$(0.012, 0.008)$	-6.89 ± 0.06	-7.15 ± 0.13
Neural references						
IMNN	20k	5	$\mathbf{20.22 \pm 0.20}$	$(0.011, 0.007)$	-6.64 ± 0.03	$\mathbf{-7.35 \pm 0.10}$
BayesFlow (end-to-end)	100k	2	19.89 ± 0.35	$(0.011, 0.007)$	$\mathbf{-6.97 \pm 0.04}$	-7.05 ± 0.18

We show the inference constraints on our two-parameter scenario in Figure 4, and for the extended 6-parameter inference in Figure 5. The Ω_{m} in the main text is $\Omega_{\mathrm{m}} = \Omega_{\mathrm{c}} + \Omega_{\mathrm{b}}$; the total matter density is the sum of the dark matter and baryonic matter densities. The prior over cosmological parameters used during SBI training is

Table 13: Posterior calibration diagnostics for the NPE posteriors of Table 12. Coverage and TARP are evaluated at the 68% level (nominal 0.683); SBC reports the minimum p-value of the rank-uniformity test over parameters; L-C2ST reports the test p-value. HDR ratio is the area of the 68% highest-density region relative to the Gaussian ellipse with the same covariance; values well below one indicate a non-Gaussian posterior, for which the covariance-based $\log|F|$ is conservative. Mean $\pm$ standard deviation over 5 seeds.

Summary	**Coverage**$_{68}$	**TARP**$_{68}$	**SBC min.** p	**L-C2ST** p	**HDR ratio**
$\log(C_\ell)$	0.689 ± 0.018	0.680 ± 0.018	0.09 ± 0.09	0.41 ± 0.22	0.40 ± 0.03
Peak counts	0.672 ± 0.022	0.673 ± 0.010	0.39 ± 0.30	0.78 ± 0.10	0.92 ± 0.01
Wavelet scattering	0.680 ± 0.013	0.686 ± 0.019	0.31 ± 0.23	0.78 ± 0.18	0.96 ± 0.01
Cubical-Silhouette	0.667 ± 0.025	0.688 ± 0.008	0.24 ± 0.23	0.67 ± 0.22	0.99 ± 0.01
Cubical-Curves	0.693 ± 0.013	0.689 ± 0.017	0.44 ± 0.37	0.13 ± 0.08	0.97 ± 0.01
Cubical-PI	0.722 ± 0.032	0.696 ± 0.017	0.03 ± 0.06	0.07 ± 0.03	0.99 ± 0.01
TF-Cubical-PersLay	0.697 ± 0.011	0.693 ± 0.019	0.24 ± 0.13	0.20 ± 0.15	1.00 ± 0.01
TF-CNN-PersLay	0.687 ± 0.008	0.682 ± 0.023	0.34 ± 0.28	0.40 ± 0.29	1.00 ± 0.01
IMNN	0.691 ± 0.009	0.692 ± 0.009	0.14 ± 0.14	0.73 ± 0.08	0.99 ± 0.01

$$
\begin{aligned}
\Omega_c &\sim \mathcal{N}_{>0}(0.2664,\, 0.2^2),\\
\Omega_b &\sim \mathcal{N}(0.0492,\, 0.006^2),\\
\sigma_8 &\sim \mathcal{N}(0.831,\, 0.14^2),\\
h_0 &\sim \mathcal{N}(0.6727,\, 0.063^2),\\
n_s &\sim \mathcal{N}(0.9645,\, 0.08^2),\\
w_0 &\sim \mathcal{N}_{[-2.0,\,-0.3]}(-1.0,\, 0.9^2),
\end{aligned}
$$

where $\mathcal{N}_{>0}$ denotes a half-normal (truncated at zero) and $\mathcal{N}_{[a,b]}$ a normal truncated to $[a, b]$. The reference (fiducial) cosmology is the truth value of each parameter: $(\Omega_c, \Omega_b, \sigma_8, h_0, n_s, w_0) = (0.2664,\, 0.0492,\, 0.831,\, 0.6727,\, 0.9645,\, -1.0)$. We note that in the 6-parameter inference there are visible projection effects on Ω_c in most models, induced by the degeneracy between Ω_c and w_0 (in the bottom-left panel of Fig. 5). Since w_0 is only marginally constrained and has an asymmetric prior, it shifts the mean of the posterior for correlated parameters. As Ω_c and σ_8 are highly correlated as well, this also induces a bias on σ_8. Curiously, both PersLay statistics seem to be most robust against that. This behavior is consistent across multiple random seeds.

NPE results. Table 12 reports the NPE results. At the fiducial mock, the IMNN gives the tightest posterior ($\log|F| = 20.22$) and the lowest fiducial NLL (-7.35); BayesFlow and the two TopoFisher summaries follow within 0.5 nats in $\log|F|$. Averaged over the prior, however, the TopoFisher summaries reach the lowest global NLL among the Fisher-trained and hand-crafted summaries (-6.91 and -6.89), ahead of wavelet scattering (-6.69) and the IMNN (-6.64): a summary optimized at the fiducial can yield a tight posterior there while generalizing less well across the prior. The IMNN, $\log(C_\ell)$ and Cubical-Curves posteriors were trained on 20k rather than 100k simulations. For TF-Cubical-PersLay, doubling the training set from 50k to 100k changes neither $\log|F|$ nor the NLL, so the estimator has converged for that summary; we have not verified this for the 20k rows.

Amortized summary network. As an amortized reference, we train BayesFlow [Radev et al., 2022], in which a summary network acting on the convergence maps is trained jointly with the conditional flow by minimizing the NLL over the prior, on the same 100k simulations. Over two seeds it reaches a global NLL of -6.97 ± 0.04, slightly below the TopoFisher summaries (-6.91 ± 0.04), with a comparable $\log|F|$ (19.89 ± 0.35). A Fisher-trained topological summary is therefore competitive with, but does not surpass, an end-to-end amortized summary network. The two differ in what they need: BayesFlow learns its summary from simulations across the prior, whereas TopoFisher learns it from simulations near the fiducial and uses prior simulations only for the downstream posterior estimator.

Posterior calibration. Reference posterior samples are unavailable for weak lensing, so we use diagnostics that require only samples from the joint distribution (Table 13): the empirical coverage

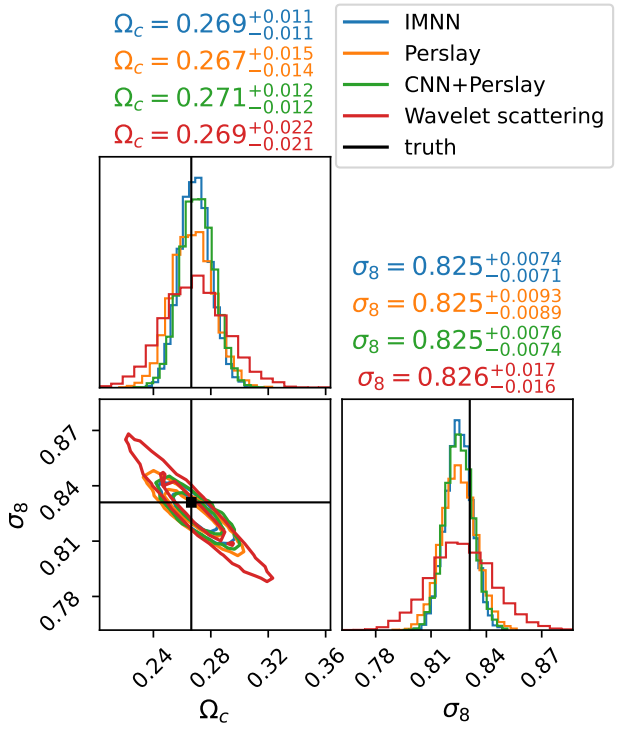


Figure 4: Comparison of SBI constraints from different summary statistics. The contours show the 1- and 2-σ uncertainties of the different models.

of 68% credible regions over held-out simulations, TARP [Lemos et al., 2023], simulation-based calibration [SBC; Talts et al., 2018], and the local classifier two-sample test L-C2ST [Linhart et al., 2023]. All posteriors except Cubical-PI pass SBC and L-C2ST at the 5% level (seed-averaged p-values of at least 0.09 and 0.13), with coverage and TARP between 0.667 and 0.697 against a nominal 0.683. The Cubical-PI posterior fails SBC ($p = 0.03$), is marginal under L-C2ST ($p = 0.07$) and over-covers (0.722), consistent with its large seed scatter in Table 12. The variance rescaling f that calibrates the 68% coverage is within 14% of one for every summary and above one for all but peak counts, so the raw posteriors are, if anything, slightly under-confident, and the NPE $\log|F|$ of Table 3 is not inflated by over-confidence. HDR ratios of at least 0.92 indicate near-Gaussian posteriors for all summaries except the power spectrum (0.40), whose strongly non-Gaussian posterior makes its covariance-based $\log|F|$ conservative.

J Architectures and training hyperparameters

This appendix describes the architectures, simulation data, compression schemes, and training hyperparameters used in the reported experiments. The notation here follows the paper, while Table 14 also gives the corresponding configuration names used in the released code. The lensing configuration files are generated over one two-parameter cosmological pair, the `Om_s8` pair, five tomographic bins, and five independent data seeds; the main weak-lensing tables report unless stated otherwise.

J.1 Common pipeline structure and code mapping

Every method is implemented as the composition of Eq. (2). For topological methods, the filtration f_ϕ produces a scalar field or filtration values, persistent homology is computed in dimensions H_0 and H_1, and the vectorization V_ϕ maps the resulting persistence diagrams to Euclidean features. For non-topological baselines, the persistence and diagram-vectorization stages are replaced by the identity.

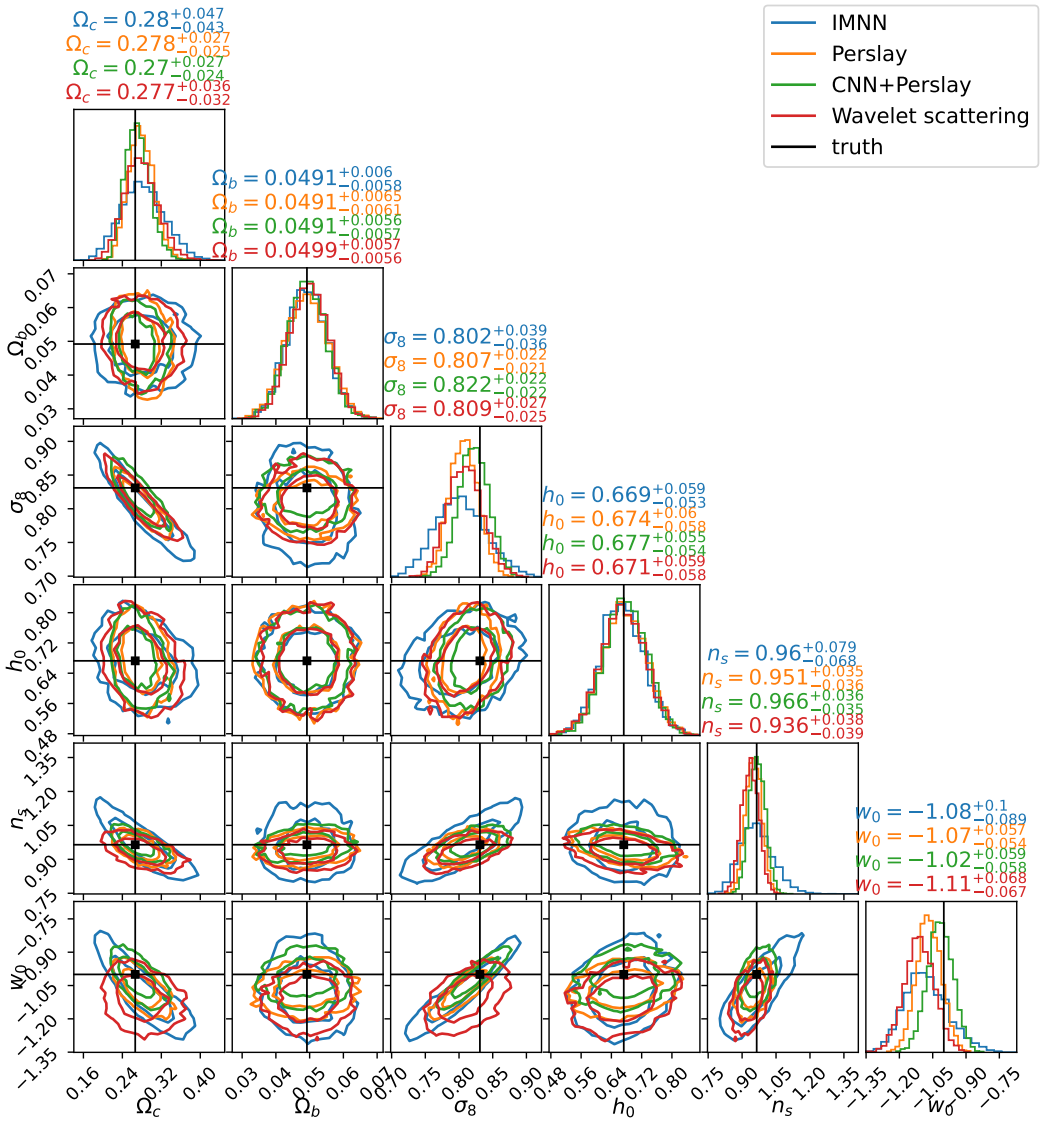


Figure 5: Comparison of SBI constraints from different summary statistics over an extended set of cosmological parameters. The contours show the 1- and 2-σ uncertainties of the different models. The equation of state of dark energy w_0 is particularly interesting to cosmologists.

Table 14: Mapping of all the methods used in the analysis. "Trainable stage" refers to parameters updated by the TopoFisher loss. MOPED is fitted from simulation summaries and carries no optimizer-updated parameters.

Method	Filtration / summary	Vectorization	Trainable stage
$\log(C_\ell)$	power spectrum	identity	none
Peak counts	peak-count histogram	identity	none
Wavelet scattering	WST coefficients	identity	none
Cubical-PI	fixed cubical PH	persistence image	none
Cubical-Silhouette	fixed cubical PH	persistence silhouette	none
Cubical-Curves	fixed cubical PH	differentiable curves	none
TF-Cubical-PersLay	fixed cubical PH	PersLay	vectorization
TF-CNN-PersLay	learned CNN filtration + PH	PersLay	filtration and vectorization
IMNN	strided CNN encoder + dense summary head	identity	CNN encoder and dense head

J.2 Filtrations and fixed scientific summaries

Fixed cubical persistence. The fixed topological pipelines use a periodic cubical filtration of the input field and compute persistence diagrams in homology dimensions H_0 and H_1. In the code this is the `cubical` filtration with `homology_dimensions: [0,1]`, `periodic: true`, and a GUDHI backend. The same filtration is used by Cubical-PI, Cubical-Silhouette, Cubical-Curves, and TF-Cubical-PersLay. For the large lensing runs, persistence features for some fixed-vectorization baselines are cached on disk when useful for throughput.

Power spectrum baseline. The $\log(C_\ell)$ baseline computes the radially averaged auto-power spectrum of each tomographic convergence map. We use 30 logarithmically spaced multipole bins, exclude the DC mode with $k_{\min} = 1$, set the field size to 512, and apply a logarithmic transform to the binned power spectrum before MOPED compression. This baseline is deterministic and has no trainable parameters.

Peak-count baseline. Peak counts are computed by identifying pixels that are strictly greater than their eight nearest neighbours. We discard a one-pixel boundary, bin peak heights into 20 uniformly spaced bins over $\kappa \in [-0.05, 0.20]$, and apply the transform $n_i \mapsto \log(1 + n_i)$ to Gaussianize the count vector before MOPED compression.

Wavelet scattering baseline. The wavelet scattering transform (WST) baseline uses the second-order scattering transform [Mallat, 2012] as implemented in KYMATIO [Andreux et al., 2020]. For each 512×512 map we use $J = 5$ dyadic scales, $L = 4$ orientations, and maximum scattering order 2. The feature vector contains the spatial mean S_0, 20 first-order coefficients S_1, and 160 second-order coefficients S_2, for a total of 181 features. We apply $\log(1 + \cdot)$ to the non-negative S_1 and S_2 coefficients and leave S_0 unchanged.

Learnable CNN filtration for TF-CNN-PersLay. The TF-CNN-PersLay pipeline prepends a full-resolution CNN before cubical persistence: a two-layer CNN with 8 hidden channels, 3×3 kernels, and circular padding, whose output defines the filtration of a periodic cubical complex. The CNN and PersLay stages are both optimized by the TopoFisher loss, with a configuration that differs between the two field benchmarks (Tables 15 and 16). On GRFs, the CNN is initialized at the identity (zero-initialized output projection), so that training starts from the fixed cubical filtration of TF-Cubical-PersLay; its output is not standardized, since per-map standardization removes most of the information on the amplitude A_s; and it is kept frozen for the first 500 epochs while PersLay and the MOPED statistics stabilize. On weak lensing, the CNN uses initialization scale 0.1, its output is standardized per map, and it is trained jointly with PersLay from the first epoch, in a single stage with cosine warm restarts; persistence diagrams are computed with the `T` construction in sub-batches of 100 maps.[6]

IMNN baseline. The IMNN baseline is an unconstrained non-topological Fisher-neural reference following the principle of Information Maximising Neural Networks [Charnock et al., 2018]. It consists of a strided convolutional encoder followed by a dense head that directly outputs $n_{\text{summaries}} = d$

[6]We thank an anonymous referee for suggesting the controlled ablations that led us to this training configuration.

summaries, where d is the number of parameters in the local Fisher analysis. In all two-parameter experiments reported here, $d = 2$. The persistence and vectorization stages are replaced by the identity, and the compression stage is also the identity: the dense head itself is the learned compressor.

For the GRF benchmark, the encoder has channel widths $[16, 32, 16]$, kernel sizes $[5, 3, 3]$, strides $[2, 2, 2]$, and circular padding. The 64×64 input is reduced to an 8×8 spatial representation and flattened before a dense head with hidden width 64 and output dimension 2.

For the weak-lensing benchmark, the encoder has channel widths $[16, 32, 32, 32, 16]$, kernel sizes $[7, 5, 3, 3, 3]$, strides $[4, 2, 2, 2, 2]$, and circular padding. The 512×512 input is reduced to an 8×8 spatial representation and flattened before a dense head with hidden width 128 and output dimension 2.

J.3 Diagram vectorizations

Identity. The identity vectorization is used by non-topological summaries such as IMNN, $\log(C_\ell)$, peak counts, and WST: the output of the filtration is directly passed to the compressor.

Persistence images. Cubical-PI uses persistence images [Adams et al., 2017] with an 8×8 grid, bandwidth 1.0, and persistence weighting. We compute features separately for H_0 and H_1 and concatenate the two outputs.

Persistence silhouettes. Cubical-Silhouette uses uniformly weighted persistence silhouettes with 50 grid points per homology dimension and a 99-th percentile support clip. The H_0 and H_1 silhouette vectors are concatenated before compression.

Differentiable persistence curves. Cubical-Curves uses differentiable cumulative curves of diagram coordinates. For the lensing runs, the vectorization uses birth and death curves (`curves: "B,D"`), 50 grid points, a 99-th percentile support clip, minimum persistence 0, and soft-sigmoid temperature 2. For the GRF runs, where diagram values have a different scale, we use the same curve family with 50 grid points and the GRF-specific temperature reported in Table 16.

PersLay. PersLay [Carrière et al., 2020] is the learnable diagram vectorization used by TF-Cubical-PersLay and TF-CNN-PersLay. It is applied separately to the H_0 and H_1 diagrams. The main configuration uses a 16-dimensional point embedding, a hidden dimension of 32, no post-pooling MLP (`post_pool_dim: 0`), and spectral normalization on the linear layers.

J.4 Compression and Fisher estimation

Unless stated otherwise, all field-based GRF and lensing pipelines use MOPED compression (Appendix B) as the final linear map before Fisher estimation, with output dimension $d = 2$ matching the parameter dimension. The covariance inverse entering the MOPED operator is Hartlap-corrected as described in Appendix B. The IMNN is the only exception: its dense head outputs a two-dimensional summary directly, so the compression stage is the identity.

Periodic MOPED refitting. For fixed pipelines, MOPED is fitted once from the relevant simulator dataset and held constant. For pipelines with learnable vectorizations or filtrations, the feature distribution drifts during training, so the MOPED operator must be periodically re-estimated.In the main lensing TF-Cubical-PersLay runs, we refit MOPED every 50 optimizer epochs; on GRFs, TF-Cubical-PersLay re-estimates MOPED within each mini-batch. In the full-resolution TF-CNN-PersLay runs, both MOPED and the PersLay input normalization are refreshed every 50 epochs using at most $1{,}000$ samples; this cap avoids repeatedly processing the full $17{,}500$-sample training split at 512^2 resolution. These refits update only the linear compression and normalization statistics; the optimizer-updated parameters remain those listed in Table 14.

Transfer-Fisher protocol. For transfer-Fisher evaluations on LPT maps, the learned nonlinear stages (CNNs, PersLay weights, and/or IMNN weights) are loaded from the corresponding lognormal checkpoint and kept fixed. For MOPED-based methods, the empirical covariance, finite-difference Jacobians, precision matrix, and MOPED operator are then re-estimated on the LPT evaluation

dataset, exactly as in any Fisher-information measurement. For the IMNN, there is no MOPED operator to refit; we evaluate the frozen two-dimensional network output and re-estimate only the Fisher statistics on LPT maps. Transfer results therefore test whether the learned feature extractor transfers; they do not retrain or fine-tune learned filtrations, vectorizations, or neural summaries on LPT maps.

J.5 Simulation dataset and finite differences

Noisy spiral datasets. For the spiral benchmark, each point cloud contains $N = 240$ points, with target parameters $\theta = (\mu, w)$. We use the finite-difference simulator datasets described in Section 4.1: $n_s = n_d = 10{,}000$ samples per configuration and five independent seeds. All point-cloud pipelines operate on k-nearest-neighbour distance features with $k = 100$ for the main table.

GRF datasets. For the GRF benchmark, each sample is a zero-mean periodic 64×64 field with power spectrum $P(k) = A_s(k/k_{\mathrm{pivot}})^{-B}$. We use $\theta_{\mathrm{fid}} = (A_s, B) = (1, B_0)$ with $B_0 \in \{-2, -1, 0, 1, 2\}$, finite-difference steps $\Delta\theta = (0.1, 0.1)$, and $n_s = n_d = 40{,}000$ simulations per configuration over five seeds. The same train/validation/test split fractions as above are used.

Lensing dataset. Weak-lensing convergence maps are generated with the `sbi_lens` lognormal simulator [Lanzieri et al., 2025] at 512×512 resolution over a $10^\circ \times 10^\circ$ field of view and five tomographic bins. Maps are smoothed with a Gaussian kernel of width $\sigma = 2'$ before any summary statistic is computed. For every configuration we use $n_s = 20{,}000$ fiducial simulations for covariance estimation and $n_d = 20{,}000$ simulations at each finite-difference point for derivative estimation. The training/validation/test split fractions are $0.875/0.0625/0.0625$.

For the LPT transfer experiment we use a separate LPT dataset at the same resolution, smoothing scale, tomographic-bin definition, and `Om_s8` finite-difference grid. Transfer configurations contain no training block: checkpoints are supplied at runtime, learned weights are frozen, and Fisher quantities are evaluated on the LPT dataset.

J.6 Training protocol and hyperparameters

Training uses structured minibatches containing fiducial simulations for covariance estimation and paired finite-difference simulations for derivative estimation. At each optimizer step, the loss is the negative log-determinant Gaussian Fisher loss of Eq. (4), optionally augmented by skewness and excess-kurtosis penalties on the compressed summary. Validation is performed on a held-out simulator dataset, and early stopping uses the validation Fisher loss.

Unless stated otherwise, optimization uses Adam with weight decay 10^{-4}, ReduceLROnPlateau scheduling with multiplicative factor 0.5 and minimum learning rate 10^{-6}, validation every 10 epochs, and five independent data seeds.

All lensing runs use train/validation/test fractions $0.875/0.0625/0.0625$. The fixed baselines and TF-Cubical-PersLay use the plateau scheduler (patience 20 validation checks, factor 0.5); TF-CNN-PersLay and the IMNN use cosine schedules. The TF-CNN-PersLay configuration was selected among four pilot configurations on two (bin, seed) cells.

The GRF differentiable-curve vectorization uses birth, death, and persistence curves on 50 grid points with a 99.9-th percentile support clip and temperature 0.1. The lensing curve vectorization instead uses birth and death curves, a 99-th percentile support clip, and temperature 2, matching the larger dynamic range of 512^2 convergence maps.

Hardware and caching. GRF runs at 64^2 resolution are CPU-friendly. Cubical persistence is computed on CPU in all pipelines, with sub-batching inside the persistence layer; GPUs are used for the learned network stages. Fixed-vectorization lensing baselines use cached persistence features when available. End-to-end timing and persistence-backend details are reported in Appendix K.

Table 15: Main lensing training hyperparameters. Fixed baselines have no optimizer-updated parameters; their entries indicate the estimator and split configuration used by the common training/evaluation driver. In that driver (fixed baselines, TF-CNN-PersLay, IMNN) an epoch is one randomly drawn mini-batch and the patience counts validation checks, made every 10 epochs; TF-Cubical-PersLay is trained on cached diagrams, where an epoch is a full pass over the training split and the patience is counted in epochs. Here λ_s and λ_k are the skewness and excess-kurtosis penalties.

Pipeline	**Epochs**	**LR**	**Batch**	**Patience**	**Clip**	(λ_s, λ_k)	**Extra settings**
Fixed MOPED baselines	2000	10^{-3}	50	100	1.0	$(0.25, 0.75)$	No trainable weights; MOPED fit/evaluation only.
TF-Cubical-PersLay	500	10^{-3}	50	100	1.0	$(0, 0)$	Diagrams computed once per bin and cached (essential classes dropped); weight decay 10^{-4}; plateau LR schedule; MOPED refit every 50 epochs; best validation checkpoint (12.5% split, checked every 10 epochs); evaluation in eval mode.
TF-CNN-PersLay	2500	6×10^{-4}	256	200	0.5	$(0.25, 0.75)^{\dagger}$	†Annealed to 0 over the first 200 epochs. Single stage; cosine LR with warm restarts ($T_0 = 400$); no warm-up; CNN trained from the first epoch (no freezing); CNN output standardized per map; loss clamp 20; NaN recovery; MOPED refit and PersLay renormalization every 50 epochs on 1000 samples; evaluation in eval mode with exact spectral norms.
IMNN	4000	5×10^{-4}	100	400	1.0	$(0.10, 0.00)$	Cosine LR schedule; identity compression; dense head outputs two summaries directly.

K Compute resources and running times

All experiments were run on the ORFEO cluster at Area Science Park with CPU-only nodes and single-GPU jobs. CPU nodes used AMD EPYC-class processors with 256 GB–1 TB RAM; GPU jobs used one NVIDIA A100 40 GB or one NVIDIA H100 80 GB accelerator. CPU persistence jobs used up to 32 cores per task (`gudhi` parallelism), while GPU jobs requested a single accelerator. Wall-clock times below are medians measured from scheduler logs. For lensing, one job means one tomographic bin and one random seed; for GRFs, one job means one spectral index B_0 and one random seed.

Per batch of eight 512^2 maps, cubical persistence takes about 3.1 s on CPU (GUDHI, forward only), against 0.95 ms for a PersLay forward pass and 0.44 ms for the IMNN on GPU. For a fixed summary, the GRF Fisher estimate is stable to about 0.1 nats between 10^3 and 4×10^4 simulations, with seed scatter below 0.05 nats from 1–2.5×10^3 simulations and below 0.02 nats from 10^4.

Table 16: GRF training hyperparameters. These settings are shared across $B_0 \in \{-2, -1, 0, 1, 2\}$ and five seeds. For the fixed baselines, TF-CNN-PersLay and the IMNN an epoch is one randomly drawn mini-batch and the patience counts validation checks, made every 10 epochs; TF-Cubical-PersLay is trained on cached diagrams, where an epoch is a full pass over the training split and the patience is counted in epochs.

Pipeline family	**Epochs**	**LR**	**Batch**	**Patience**	(λ_s, λ_k)	**Extra settings**
Fixed topological baselines	2000	10^{-3}	500	100	$(0.05, 0.20)$	MOPED compression; no trainable filtration/vectorization.
TF-Cubical-PersLay	500	10^{-3}	500	100	$(0, 0)$	Trained on cached periodic cubical diagrams; MOPED re-estimated within each mini-batch; 10% validation split checked every 5 epochs; plateau LR schedule; gradient clip 1.0; best validation checkpoint; evaluation in eval mode.
TF-CNN-PersLay	2000	10^{-3}	500	100	$(0.05, 0.20)$	CNN output not standardized; periodic persistence; CNN initialized at the identity (zero-initialized output projection) and frozen for the first 500 epochs; LR warm-up 50 epochs; plateau LR schedule; gradient clip 1.0; loss clamp 20; MOPED refit every 50 epochs; evaluation in eval mode with exact spectral norms.
IMNN	4000	5×10^{-4}	200	400	$(0.10, 0.00)$	Cosine LR schedule; identity compression; dense head outputs two summaries directly.

Table 17: Representative wall-clock times for the reported pipelines. Lensing jobs are one tomographic bin $\times$ one seed; GRF jobs are one B_0 value $\times$ one seed. Times are median scheduler wall-clock times. For lensing TF-Cubical-PersLay the persistence diagrams are computed once per bank on CPU and PersLay is then trained on the cached diagrams on a GPU (two jobs); TF-CNN-PersLay recomputes the persistence at every step (on CPU), and its jobs include the final evaluation.

Benchmark	**Pipeline**	**Resource request**	**Median time/job**
Lensing	$\log(C_\ell)$	4 CPU, 32 GB	13 min
	Peak counts	4 CPU, 32 GB	22 min
	Cubical-Silhouette	32 CPU, 120 GB	63 min
	Cubical-Curves	32 CPU, 120 GB	41 min
	Cubical-PI	32 CPU, 120 GB	49 min
	Wavelet scattering	16 CPU, 120 GB	301 min
	TF-Cubical-PersLay	16 CPU, 96 GB + 1 A100/H100, 128 GB	64 + 79 min
	TF-CNN-PersLay	1 H100, 175 GB	7.1 h
	IMNN	1 A100, 200 GB	28 min
GRF	Cubical-PI/Silhouette/Curves	32 CPU, 64 GB	5–7 min
	TF-Cubical-PersLay	8 CPU, 32 GB	2.8 h
	TF-CNN-PersLay	16 CPU, 96 GB	9.8 h
	IMNN	1 A100, 48 GB	5 min

Table 18: Where the training time goes on weak lensing (one bin $\times$ one seed, from a dedicated cost benchmark). Persistent homology dominates the topological pipelines: it is computed once and cached for TF-Cubical-PersLay and recomputed at every epoch for TF-CNN-PersLay.

Pipeline	**Persistence**	**Learned stages (fwd/bwd)**	**MOPED + Fisher**
TF-Cubical-PersLay	4,973 s (97%)	148 s	2 s
TF-CNN-PersLay	11,398 s (99.6%)	47 s	–
IMNN	–	852 s	–

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the methodological contribution, the local Lipschitz stability result, and the experimental scope. The reported claims are supported by the spiral, GRF, and weak-lensing experiments, and the local nature of Fisher optimization is discussed as a limitation.

2. **Limitations**

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

   Answer: [Yes]

   Justification: The Conclusions discuss the main limitations: TopoFisher is a local summary-design method, relies on a Gaussian summary approximation, and does not by itself guarantee globally injective summaries or calibrated posteriors over a wide prior. Additional finite-sample and estimator-bias issues are discussed in Section 3 and Appendix B.

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: The stability theorem is stated with explicit boundedness, Lipschitz, and non-degeneracy assumptions in Section 3. A proof sketch is given in the main text, and the complete proof appears in Appendix C. Appendix D discusses how the assumptions apply to the pipelines used in the experiments.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper specifies the simulators, sample sizes, finite-difference schemes, baselines, architectures, compression methods, training protocol, evaluation metrics, and compute resources. Full experimental and architectural details are provided in the appendices.

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

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

   Answer: [Yes]

   Justification: The paper provides a public repository with code, configuration generators, and run instructions. Large generated simulation datasets are not bundled because of their size, but the repository and appendices document how to regenerate the simulation datasets and reproduce the reported training and evaluation runs.

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, necessary to understand the results?

   Answer: [Yes]

   Justification: The main text describes the benchmark setup and evaluation protocol, while Appendix J reports architectures, optimizers, learning rates, batch sizes, train/validation/test splits, finite-difference steps, MOPED refitting, early stopping, and stability instruments.

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: The main experimental tables report means and standard deviations across independent seeds. The text states the number of seeds and, where relevant, averages over fiducial configurations, spectral indices, or tomographic bins.

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: Appendix K reports the hardware used, including CPU and GPU resources, memory requests, and representative wall-clock times for the reported GRF and weak-lensing pipelines. Appendix J also describes caching and hardware-relevant implementation details.

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work uses simulated scientific data and standard machine-learning methodology, with no human-subject data, private data, or high-risk deployment. The submission is anonymized and follows the NeurIPS ethics guidelines.

10. **Broader impacts**

    Answer: [Yes]

    Justification: The Reproducibility and impact section briefly discusses positive impacts for scientific uncertainty quantification and negative impacts from misuse with poorly validated simulators or outside the local Fisher regime.

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., pre-trained language models, image generators, or scraped datasets?

    Answer: [N/A]

    Justification: The paper does not release high-risk models, scraped datasets, generative media systems, or assets with substantial misuse potential.

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: Existing simulators and software packages are cited, including PowerBox, `sbi_lens`, GUDHI, Kymatio, and related libraries. All package versions and licences used are listed in the anonymized repo.

13. **New assets**

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

    Answer: [Yes]

    Justification: The paper introduces code, configuration generators, and experiment scripts, made available through the anonymized repository. The repository documents how to regenerate datasets, construct YAML configurations, train the reported pipelines, and evaluate Fisher, transfer, and SBI results. Large generated datasets are not bundled but are reproducible from the documented simulators.

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: [N/A]

    Justification: The paper does not involve crowdsourcing, human-subject experiments, or participant compensation.

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: [N/A]

    Justification: The paper does not involve human subjects or crowdsourced data collection, so IRB approval is not applicable.

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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method does not use LLMs as a methodological component. Any LLM use, if applicable, was limited to writing, editing, or formatting assistance and did not affect the scientific method, experiments, or results.