GyroSplat: Exact and Differentiable Linear Representations of Plasma Turbulence

Anonymous authors
Paper under double-blind review

Abstract

Gyrokinetic simulations model the plasma turbulence governing heat transport in magnetically confined fusion devices. Due to their 5D nature, these simulations are computationally expensive and produce massive data volumes, with a single snapshot requiring tens of gigabytes of storage. This makes directly storing intermediate timesteps impractical and motivates the use of (lossy) compression. Recent neural compression approaches achieve favorable rate–distortion trade-offs by augmenting reconstruction objectives with physics-informed losses that preserve derived quantities such as the electrostatic potential and heat flux. However, neural network based representations can be difficult to optimize and refine when additional constraints are imposed. We introduce GyroSplat, a compression method for gyrokinetic fields inspired by Gaussian splatting. GyroSplat represents each plasma field as a weighted sum of independent Gaussian–Gabor splats, yielding a representation that is linear in its coefficients. This linearity enables a staged optimization strategy in which splats are first fitted to minimize reconstruction error, then refined against the potential, and subsequently new splats are added to correct the transport. Moreover, derived quantities admit closed-form expressions in terms of the splat coefficients, enabling efficient refinement via a Gauss–Newton procedure. Experiments on gyrokinetic simulations show that GyroSplat achieves improved trade-offs at matched storage over established compression baselines on derived physical quantities while remaining competitive on field reconstruction.These results highlight Gaussian splatting as an effective and flexible approach to high-fidelity compression of scientific simulation data.

1 Introduction

Magnetic confinement fusion aims to produce energy by fusing hydrogen isotopes in a plasma that is confined by strong magnetic fields. Turbulent transport in the plasma is a central challenge for fusion energy and a major obstacle to reactor designs. Gyrokinetic simulations are the standard tool for studying such turbulence, and a single reactor design cycle may require thousands of them, run across different plasma profiles. The gyrokinetic equations evolve a 5D distribution function in time. After an initial linear growth phase, the turbulence saturates nonlinearly and settles into a statistically steady state. The main quantity of interest for practitioners is the turbulent heat flux, which transports energy out of the plasma and degrades confinement.

A major practical bottleneck of gyrokinetic simulations is their storage demand. Practitioners are mainly interested in derived quantities such as electrostatic potentials and heat flux, particle, and momentum fluxes, and are usually forced to discard the underlying 5D snapshots due to their sheer size. As a result, post-hoc analysis, such as revisiting existing diagnostics or deriving new ones, is infeasible. This is particularly limiting for complex cases involving electromagnetic effects, where practitioners may need to revisit diagnostics to refine the design space. Lossy compression offers a way to reduce storage requirements while retaining the information needed for such analyses (Choi et al., 2021; Galletti et al., 2026a). However, existing approaches typically struggle to compress field data effectively while preserving derived quantities of interest.

To tackle this problem, we introduce GyroSplat, a compression method for gyrokinetic fields inspired by Gaussian splatting. GyroSplat represents a snapshot as a weighted sum of individually parameterized Gaussian–Gabor splats (Wurster et al., 2024). Each splat describes a localized part

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

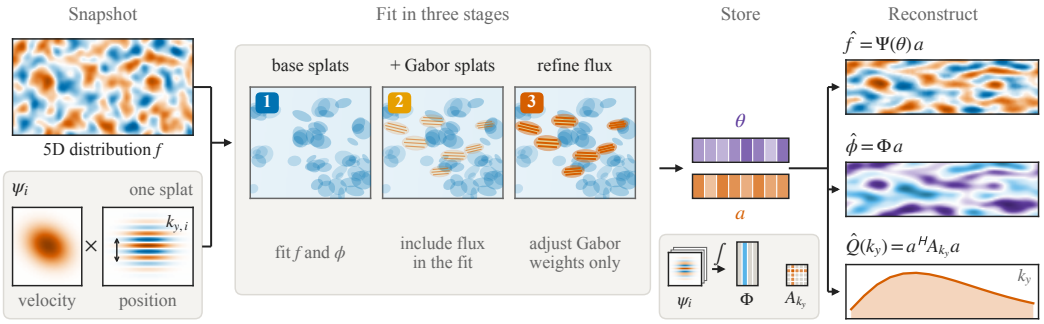


Figure 1: **GyroSplat compression in three stages.** Stage 1 fits base Gaussian splats to the distribution and potential. Stage 2 adds Gabor splats and jointly fits reconstruction and transport. Stage 3 adjusts only the Gabor amplitudes at fixed geometry. The stored geometry θ, amplitudes a and normalization information define the reconstructed snapshot. Potential and heat flux can also be evaluated from the coefficients using matrices computed from the splats and simulation geometry.

of the distribution via its position, shape and amplitude. GyroSplat is fit in three stages. First, we fit the distribution with Gaussian splats, then refine the fit for the derived potential field, and finally add frequency-modulated Gabor splats (Wurster et al., 2024) to improve heat flux reconstruction. Motivated by the governing equations we express the potential field and heat flux as linear and quadratic functions of the splat coefficients, respectively. This yields closed-form derivatives, which enable an efficient Gauss–Newton refinement targeting the quantities of interest.

We evaluate GyroSplat on electrostatic turbulence with adiabatic electrons. On this dataset we show that at fixed storage budgets, GyroSplat is competitive at the reconstruction of the 5D distribution function, while improving heat flux accuracy, compared to all other evaluated methods. Finally, GyroSplat achieves an improved reconstruction–physics trade-off compared to all competitors when including physics refinement.

Our contributions are:

- **5D splats for plasma data.** We extend Gaussian–Gabor splats to the spatial and velocity coordinates of a gyrokinetic distribution. Each snapshot is stored as a fitted set of splat positions, shapes and amplitudes.
- **Fitting for reconstruction and physical accuracy.** We fit the distribution and potential before adding splats to improve transport. With the splat geometry fixed, we refine the amplitudes using linear and quadratic expressions for the diagnostics. This coefficient refinement also applies to Fourier and wavelet representations.
- **Compression of electrostatic turbulence.** Across distinct turbulence states, GyroSplat improves the accuracy of derived quantities while retaining competitive reconstruction error. We compare accuracy and storage with physics-informed neural fields and Fourier and wavelet representations fitted to preserve physical quantities.

2 Background and Related Work

Plasma turbulence and gyrokinetics. A plasma is an ionized gas whose charged particles interact through electric and magnetic fields. In magnetic fusion devices, turbulence transports heat across the confining field, making it harder to sustain fusion temperatures. Predicting this transport requires resolving coupled particle motion and field fluctuations across spatial and temporal scales. Particles at the same position can have different velocities and contribute differently to heat transport. A kinetic description therefore retains their distribution in velocity as well as in position. Gyrokinetics averages over rapid gyromotion to reduce the kinetic description to three position and two velocity coordinates (Brizard & Hahm, 2007), making snapshots expensive to store (Galletti et al., 2026a). In the electrostatic setting considered here, fluctuations are described by the electric potential, with ions treated kinetically and electrons approximated by an adiabatic response. This model is implemented in gyrokinetic simulation codes such as GKW (Peeters et al., 2009). Compression must retain the

velocity integrals and correlations that determine fields and fluxes, including the balance between partly cancelling outward and inward transport. Compact representations also support prediction and generation. GyroSwin learns latent features to predict gyrokinetic dynamics (Paischer et al., 2025), and GyroFlow generates turbulent snapshots in latent space (Galletti et al., 2026b). Dynamical low-rank methods instead evolve factorized distributions during simulation (Einkemmer et al., 2025).

Compression in vision and graphics. Historically, image compression has relied on fixed, hand-designed transforms, such as the discrete cosine transform used in JPEG (Wallace, 1992), wavelet bases (Daubechies, 1992), or PCA (Pearson, 1901). With the rise of deep learning in the 2010s, learned codecs became increasingly popular. Most prominently, variational autoencoders (VAEs) (Kingma & Welling, 2014) learn a continuous latent representation optimized under a rate-distortion objective, while vector-quantized autoencoders (VQ-VAEs) instead map inputs to discrete codes drawn from a learned codebook (van den Oord et al., 2017). More recently, advances at the intersection of vision and graphics introduced Neural Fields (NFs), which parameterize signals as coordinate-based neural networks (Sitzmann et al., 2020). Neural radiance fields (NeRFs) (Mildenhall et al., 2020) build on this representation for 3D scene reconstruction and novel-view synthesis. However, because neural fields typically rely on a single MLP, they can be slow to train and reconstruct. To address this, several methods instead combine a small MLP with localized, grid-based feature parameters (Müller et al., 2022). This trend led to the rediscovery of Gaussian splatting (Westover, 1991; Zwicker et al., 2002), which represents a scene with a set of explicit, localized primitives. This enables fast training and real-time rendering for tasks such as 3D scene reconstruction (Kerbl et al., 2023) and image compression (Zhang et al., 2024). In contrast to neural fields, individual parameters (i.e. splats) can be added, removed, or updated without affecting the rest of the reconstruction. Neural fields lack this property as changing a parameter generally alters the output everywhere. Subsequent work has further extended Gaussian splats with Gabor functions (Gabor, 1946; Mallat & Zhang, 1993). These introduce localized oscillations within each primitive to further improve the reconstruction quality of fine details (Wurster et al., 2024; Watanabe et al., 2025).

Compression of physics simulation data. Compression for scientific data shall preserve certain quantities of interest (Seguí & Armour, 2026). While classical approaches often control errors in specified quantities of interest (Ainsworth et al., 2019; Jiao et al., 2022), physics-informed neural compressors directly optimize for them in the training stage. Examples include vector-quantized autoencoders for turbulent flow (Momenifar et al., 2022), VAPOR (Choi et al., 2021) and PINC (Galletti et al., 2026a) for plasma data, and neural fields for general relativity (Cranganore et al., 2026a). Another variant is choosing which snapshots to compress via physics-informed adaptive timestep selection (Cranganore et al., 2026b). Optimizing for reconstruction accuracy and physical consistency can cause training instabilities (Krishnapriyan et al., 2021; Wang et al., 2021), which motivates multi-loss balancing methods, such as conflict-free gradients (Liu et al., 2025).

Physics-informed compression has also been combined with Gaussian splatting, for example to fit PDE solutions (Kang et al., 2025; Rensen, 2024), while Gaussian primitives more broadly have been used to compress scalar volumes (Dyken et al., 2026) and particle simulations (Jiang et al., 2026). GyroSplat differs from these works by using Gaussian–Gabor splats to fit complex 5D distributions through a multi-stage, physics-informed procedure.

3 Compression with 5D splats

GyroSplat fits each snapshot as a weighted sum of splats in three stages. After fitting Gaussian splats to the distribution and potential, we add Gabor splats. Their oscillation enhance the representation in the frequency band used to evaluate heat transport. We then refine the amplitudes of these added splats to reduce the remaining heat flux error.

3.1 From Gaussian splats to a 5D distribution

A Gaussian splat is a smooth contribution concentrated around a chosen location. Following the 3D Gaussians used by Kerbl et al. (2023) defined the unscaled splat's envelope g as

$$g(p; m, L) = \exp\left[-\tfrac{1}{2}\|L^{-1}(p-m)\|_2^2\right], \tag{1}$$

where p is a position, m is the center, and $LL^\top$ is the covariance controlling width and orientation. Each splat i has a complex amplitude a_i which controls the magnitude and phase of its contribution to the reconstructed field. The final reconstruction is a weighted sum over all splats. Note that in contrast to scene rendering, there is no opacity value.

The gyrokinetic distribution function of interest depends on position and velocity coordinates. In field-aligned coordinates, we describe the position by $u_p = (s, x, y)$, where s is the coordinate along the magnetic field line, x is the radial coordinate crossing the nested magnetic flux surfaces of the confinement geometry, and y is the binormal coordinate, running within a flux surface perpendicular to the field line. The velocity space is spanned by $u_v = (v_\parallel, \mu)$, the velocity parallel to the field line and the magnetic moment, respectively. We factorize the two coordinate groups into $u = (u_v, u_p)$ and form a 5D envelope as

$$g_i(u) = \underbrace{g(u_v; m_i^v, L_i^v)}_{\text{velocity envelope}} \underbrace{g(u_p; m_i^p, L_i^p)}_{\text{spatial envelope}}. \tag{2}$$

The first factor selects a region of velocity space and the second selects a region of physical space. The factors L_i^v, L_i^p are lower triangular with positive diagonals, ensuring valid covariance matrices. Each splat is separable between velocity and position. Their sum can describe correlations because splats at different positions can have different velocity centers and widths, giving a velocity distribution that changes with spatial position.

To represent oscillations at frequencies which are relevant to heat transport, we multiply the Gaussian envelope by a plane wave, following Gabor representations (Gabor, 1946; Wurster et al., 2024). The resulting Gaussian–Gabor splat and the decoded distribution are

$$\psi_i(u) = g_i(u) e^{\mathrm{i} k_i^\top (u_p - m_i^p)}, \qquad \hat{f}(u) = \sum_{i=1}^{N} a_i \psi_i(u). \tag{3}$$

Here N is the number of splats, $a_i \in \mathbb{C}$ is a complex amplitude, and k_i is the spatial frequency of the oscillatory factor. We call the Gaussian with $k_i = 0$ a *base splat* and the modulated Gaussian with $k_i \neq 0$ a *Gabor splat*. The real part of a Gabor splat alternates between positive and negative values within its envelope, as illustrated in Fig. 5. Modulation shifts the spectrum toward the chosen frequency, allowing one localized splat to describe several oscillations. We modulate along the binormal direction, $k_i = (0, 0, k_{y,i})$, because it determines large parts of the turbulent transport.

An independently stored splat has five center coordinates, nine Cholesky entries, one modulation frequency and two amplitude values, giving 17 real parameters. We deploy the Gabor splats as groups, where the 14 envelope parameters are shared and stored once per group, while their individual amplitudes and frequencies remain separate. This merging of Gabor splats into groups enhances the storage capabilities by more than three times, compared to base splats. A Gabor group occupies the same location, but with the learnable amplitudes can adjust their influence on the reconstruction. We also store four normalization values per snapshot. Section 3.3 describes the parameter allocation, and Appendix A gives the storage convention.

3.2 Physical quantities from the coefficients

Linear and quadratic diagnostics. To fit a compressed distribution on certain derived diagnostics, we first need to know how these quantities depend on the stored coefficients. In reduced-order models (Berkooz et al., 1993; Rozza et al., 2008), we can write the reconstruction as $\hat{f} = \Psi c$, where c contains the coefficients and the columns of Ψ are fixed basis functions, such as Fourier modes, wavelets, or splats. Substituting this reconstruction into a linear diagnostic $d(f) = Lf$ or a quadratic diagnostic $q_j(f) = f^{\mathsf{H}} B_j f$, with B_j Hermitian, gives

$$\hat{d} = (L\Psi)c, \qquad \hat{q}_j = c^{\mathsf{H}} (\Psi^{\mathsf{H}} B_j \Psi) c. \tag{4}$$

Here H denotes conjugate transpose. With a fixed basis, the matrices $L\Psi$ and $\Psi^{\mathsf{H}} B_j \Psi$ can be computed once and reused to evaluate the diagnostics and their derivatives. Weighted integrals are linear examples. In fluid simulations, kinetic energy at fixed density is quadratic in velocity, and enstrophy is quadratic in vorticity (Kraichnan, 1967). These quantities keep their quadratic form even when the flow evolves nonlinearly, so the same coefficient calculation applies.

For Fourier and wavelet representations, the basis functions are fixed throughout fitting. Splats also have learned geometry $\theta = \{m_i^v, m_i^p, L_i^v, L_i^p, k_i\}_{i=1}^N$, which determines $\Psi(\theta)$. We hold this geometry fixed during coefficient updates. Section 4.4 compares how fitting physical quantities changes the distribution reconstructed by splats, Fourier modes and wavelets.

Gyrokinetic potential and heat flux. For electrostatic plasma data, the potential is obtained from the distribution through the linear quasineutrality map (Brizard & Hahm, 2007). Heat flux combines the drift generated by this potential with an energy-weighted velocity moment of the distribution. Both factors are linear in the distribution, making their product quadratic. Applying Eq. 4 to the potential and heat flux spectrum gives

$$\hat{\phi} = \Phi a, \qquad \hat{Q}(k_y) = a^{\mathsf{H}} A_{k_y} a. \tag{5}$$

Here $Q(k_y)$ is the heat flux at binormal Fourier mode k_y. Column i of Φ is the potential of basis function i at unit amplitude. The Hermitian matrix A_{k_y} combines the pairwise contributions of basis functions to flux at mode k_y. Appendix B derives these expressions from the discrete velocity integrals and field solve. The matrices Φ and A_{k_y} depend on the splats and simulation geometry and can be recomputed when needed. Their products and derivatives can also be evaluated without forming dense matrices. A constant coefficient accounts for normalization offsets.

3.3 A THREE-STAGE FITTING PIPELINE

Fitting follows the three stages depicted in Figure 1. We allocate approximately 75% of the parameter budget to base splats and 25% to groups of Gabor splats, which are added in stage 2. Appendix C gives initialization settings and parameter counts.

Stage 1. Fit base splats to the distribution and potential. We begin with base Gaussian splats ($k_i = 0$). First, the zonal component ($k_y = 0$) of the snapshot is normalized separately so that its larger power does not dominate the reconstruction loss. The centers of the splats are then sampled from the grid with probability proportional to the squared magnitude of the normalized distribution. Initial standard deviations are two grid cells, and amplitudes are initialized by the regularized least-squares solve in Eq. (6). For the normalized target $f_\star$ and the inital splat geometry, we first solve only for the amplitudes (Golub & Pereyra, 1973),

$$a^\star = \arg\min_a \|\Psi a - f_\star\|_2^2 + \lambda_a \|Da\|_2^2, \tag{6}$$

where D is a diagonal regularization matrix and λ_a sets its strength. Starting from these fitted amplitudes, we optimized the center, shape and amplitudes of the splats with AdamW (Loshchilov & Hutter, 2019) on a Huber loss (Huber, 1964).

After the distribution fit, we continue updating the same parameters with AdamW and MSE loss, but combined with the potential error, following PINC (Galletti et al., 2026a). We deviate from the PINC recipe by omitting the flux loss in this stage. At the end of stage 1, the base splats approximate both the distribution and potential. Appendix C gives the update schedule.

Stage 2. Add Gabor splats and fit both groups jointly. We use the reserved budget to add groups of seven Gabor splats. The splats in a group share one Gaussian envelope throughout fitting and have separate amplitudes and modulation frequencies. Group centers are spread across the 5D grid. The initial binormal modes are $m \in \{1, 6, 11, 16, 21, 26, 31\}$, corresponding to $k_y = 2\pi m$ on normalized y. These seeds span the range of nonzero modes used by the flux objective. Figure 5 illustrates how modulation shifts a splat's spectrum. We initialize the added amplitudes by fitting the residual of the base splats, then solve again with the potential residual included. During the joint fit, distribution and potential losses update the base splats, while the flux loss updates the Gabor groups. Both groups contribute to all reconstructed quantities, so changes to either can affect every metric.

Stage 3. Refine the flux at fixed geometry. The final stage adjusts the Gabor splat amplitudes while keeping the splat geometry fixed. We use damped Gauss–Newton steps on the flux residuals (Nocedal & Wright, 2006). Let a_G contain the Gabor amplitudes from Eq. 3 and set $z = (\mathrm{Re}\, a_G, \mathrm{Im}\, a_G)$. The vector $q(z)$ has the heat flux values in the fitted band, and $q_\star$ their reference values. With $J(z)$ their Jacobian, a step solves

$$\delta = \arg\min_d \|J(z)d - [q_\star - q(z)]\|_2^2 + \lambda \|d\|_2^2, \qquad z \leftarrow z + \delta. \tag{7}$$

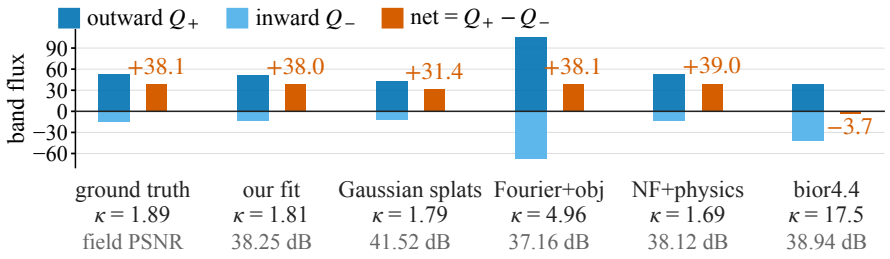


Figure 2: Analysis of transported heat flux for different methods (CR of 142×) for a single snapshot. While competitors achieve matching net transport as ground truth, they heavily over or underestimate outward or inward transport. Our metric κ captures this difference while 5D field PSNR does and relative flux error do not.

The minimization considers possible changes d to the Gabor amplitudes and selects δ. The parameter λ controls how strong large changes are penalized. We obtain the Jacobian by differentiating the quadratic flux readout with respect to the real and imaginary Gabor amplitudes. The squared flux residual is quartic, so we solve the problem iteratively, with damping to limit each update. We evaluate the final reconstruction on the distribution and potential as well as the fitted flux.

4 Experiments

In this section we elaborate on the experimental design around simulation data and storage budgets (Section 4.1), followed by the baselines we compare GyroSplat to (Section 4.2) and the metrics used for evaluation (Section 4.3). Finally, we present our results in Section 4.4.

4.1 Simulation data and compression ratios

Data collection. We reuse a subset of the dataset introduced in Paischer et al. (2025). The dataset is generated by sampling configurations around the cyclone-base-case (Dimits et al., 2000) generated with GKW (Peeters et al., 2009) and parameterised by four dimensionless parameters. These comprise the ion-temperature gradient R/L_t, density gradient R/L_n, magnetic shear $\hat{s}$, and the safety factor q. We filter this dataset to assemble a diverse set of snapshots that exhibit different characteristics through the compression lens. Specifically, we select 39 saturated-phase snapshots from 27 trajectories, with 12 streamer, 12 mixed and 15 strong-zonal states. Streamer states contain radially elongated fluctuations, strong-zonal states are dominated by the binormally averaged ($k_y = 0$) component, and mixed states have neither dominant pattern. Each snapshot contains one complex ion distribution function on a $32 \times 8 \times 16 \times 85 \times 32$ grid, ordered as $(v_\parallel, \mu, s, x, y)$, with adiabatic electrons. For more details on data generation we refer to Appendix A.

Evaluation at fixed compression ratios. We define the compression ratio (CR) as the size of the raw data divided by the size of the compressed representation. To keep the comparison across methods fair while remaining computationally tractable, we evaluate all methods at two fixed CRs, 302 and 142, and compare compression quality using the metrics from Section 4.3. These values are the maximum CRs reached by Galletti et al. (2026a)'s largest autoencoder and neural field models that fit on a single H100 GPU (80 GB) under full-batch gradient descent.

4.2 Baselines

We identify three different categories of methods that are commonly used in practice, (i) classical codecs, (ii) neural compressors and splats, and (iii) the aforementioned with physics refinement. We select representatives for each category as outlined below for a thorough comparison of compression methods.

Classical codecs. We compare to classic compression methods such as wavelet and Fourier top-K codes (Daubechies, 1992; DeVore, 1998), principal component analysis (PCA) (Eckart & Young,

1936), ZFP (Lindstrom, 2014) and SZ3 (Liang et al., 2023). These baselines compress the distribution without an additional physical objective.

Neural compressors and plain splats. For a comparison to neural compressors we include a set of Autoencoders (Galletti et al., 2026a, AE), also comprising Variational Autoencoders (Kingma & Welling, 2014, VAE), and Vector-quantized Autoencoders (van den Oord et al., 2017, VQ-VAE). They are trained on the reconstruction-only objective from PINC (Galletti et al., 2026a) on a training split that does not encapsulate the evaluation samples. In addition we include the neural fields from Galletti et al. (2026a), and Gaussian splats fitted only to the 5D distribution function.

Baselines with physical refinement. To compare physical refinement across representations, we include physics losses and Fourier and Haar codes refined against 3D potential field and heat flux. For this comparison we mainly consider neural fields as Galletti et al. (2026a) reports unstable end-to-end autoencoder training with physical losses. Appendix A gives the baseline settings and parameter-selection procedures.

4.3 Metrics

We elaborate on the metrics we use for evaluating our results as follows. Unless mentioned otherwise, we report the median across the 39 generated snapshots for each metric.

Field reconstruction. Peak signal-to-noise ratio (PSNR) measures pointwise mean squared reconstruction error (MSE) relative to the squared reference peak, in decibels, according to

$$\mathrm{PSNR} = 10 \log_{10} \frac{P_{\mathrm{ref}}}{\mathrm{MSE}}, \qquad \mathrm{MSE} = \frac{1}{M} \sum_{j=1}^{M} |\hat{v}_j - v_j|^2. \tag{8}$$

Here v is the reference array which is instantiated either by the 5D distribution function f or by the 3D potential field ϕ, $\hat{v}$ is its reconstruction, M the number of evaluated entries, and P_{ref} the squared reference peak. Higher PSNR means a smaller error. An increase of about 3 dB halves the mean squared error for the same reference. We report PSNR for the distribution f and potential ϕ, using the array layouts and peak conventions specified in Appendix A.

Heat transport. Practitioners usually investigate the heat flux spectrum $Q(k_y)$ along the k_y direction, as it provides useful insights into turbulent transport. The spectrum is derived from the 5D distribution function as a quadratic interaction between the 5D field and the 3D potential field as

$$Q(k_y) = \int \mathbf{C} \int \mathbf{v}^2 \phi \mathbf{f} \, \mathrm{d}v_{\parallel} \mathrm{d}\mu \; \mathrm{d}x \mathrm{d}s, \tag{9}$$

where $\mathbf{C} \in \mathbb{R}^{x \times s \times y}$ comprises geometric and operating parameters, $\mathbf{v} \in \mathbb{R}^{v_{\parallel} \times \mu}$ denotes particle energy, and $\phi \in \mathbb{R}^{x \times s \times y}$ is the 3D electrostatic potential field. To compare reconstructed and ground-truth heat-flux spectra, we compute the relative L1 error across binormal modes, excluding the zonal mode ($k_y = 0$),

$$\mathrm{relL1}(Q) = \frac{\sum_{k_y=1}^{31} |\hat{Q}(k_y) - Q(k_y)|}{\sum_{k_y=1}^{31} |Q(k_y)|}. \tag{10}$$

Balance of outward and inward transport. In addition to relative error on the flux spectrum, we define a new metric targeting the balance between outward and inward transport. The reason is that the flux is a sum over pointwise inward and outward transport contributions. The final flux error is measured as an aggregated difference over those, however two methods may obtain similar flux errors while exhibiting larger errors on the underlying pointwise inward and outward transport.

To that end, we first define inward and outward contributions, and then assemble them into a metric that measures their difference to ground truth snapshots. Let q denote the heat flux integrand over the simulation domain Ω, and define the outward and inward contributions as $Q_+ = \int_\Omega \max(q, 0) \, \mathrm{d}\Omega$ and $Q_- = \int_\Omega \max(-q, 0) \, \mathrm{d}\Omega$, both non-negative, such that the net flux is $Q = Q_+ - Q_-$. We then

Table 1: Compression of electrostatic turbulence. Median scores over 39 snapshots at compression ratios near $302\times$ and $142\times$. Bold marks the best column entry. GyroSplat yields the best reconstruction error on derived physics while maintaining comparable 5D field reconstruction.

method	$\mathrm{PSNR}_f \uparrow$	$\mathrm{PSNR}_\phi \uparrow$	flux relL1 $\downarrow$	$\kappa/\kappa_{\mathrm{truth}}$
CR $\approx 302\times$ *(73,728 stored reals)*				
Wavelet top-K (bior4.4) (Cohen et al., 1992)	36.40	−12.35	1.2260	7.428
Wavelet top-K (Haar) (Daubechies, 1992)	36.09	−7.95	1.0156	5.385
PCA, per snapshot (Eckart & Young, 1936)	35.08	−2.54	0.4551	1.102
ZFP (Lindstrom, 2014)	32.72	−5.61	0.9794	18.972
SZ3 (Liang et al., 2023)	33.45	−15.87	1.2873	19.873
Fourier top-K (DeVore, 1998)	35.90	23.62	0.2104	3.659
Autoencoder (Galletti et al., 2026a)	35.49	15.42	0.1308	0.864
VAE (Galletti et al., 2026a)	34.35	9.33	0.1925	0.819
VQ-VAE (van den Oord et al., 2017; Galletti et al., 2026a)	34.16	12.91	0.2176	0.793
Neural field (field objective) (Galletti et al., 2026a)	37.39	14.73	0.1729	0.901
Neural field (+ physics stage) (Galletti et al., 2026a)	37.33	26.50	0.1443	0.887
GyroSplat (field objective only)	**39.33**	7.87	0.2674	0.951
GyroSplat (ours)	37.59	**27.71**	$\mathbf{5\times10^{-7}}$	**0.985**
CR $\approx 142\times$ *(156,669 stored reals)*				
Wavelet top-K (bior4.4)	38.94	−9.82	1.1491	10.062
Wavelet top-K (Haar)	38.43	−7.13	0.9938	9.001
PCA, per snapshot	37.88	1.69	0.3513	1.065
ZFP	35.29	−3.39	0.7987	3.769
SZ3	38.42	−12.23	0.5367	4.349
Fourier top-K	37.39	27.06	0.1005	2.800
Neural field (field objective)	38.17	14.66	0.1450	0.915
Neural field (+ physics stage)	38.12	26.99	0.1041	0.894
GyroSplat (field objective only)	**41.43**	9.60	0.1608	0.961
GyroSplat (ours)	38.20	**29.24**	$\mathbf{4\times10^{-8}}$	**0.971**

Table 2: Ablations on the multi-stage optimization pipeline illustrating a transition from Gaussian splats to our GyroSplat. We report median over 39 snapshots for a compression ratio of $302\times$. Grey settings illustrate harmful choices that result in worse PSNR_ϕ and should be avoided.

configuration within each stage	$\mathrm{PSNR}_f \uparrow$	$\mathrm{PSNR}_\phi \uparrow$	flux relL1 $\downarrow$	$\kappa/\kappa_{\mathrm{truth}}$
1 Fit base splats to the distribution and potential				
Gaussian splats	39.06	8.95	0.4022	0.986
Potential objective	38.10	25.94	0.3178	0.944
Potential fit with flux loss	37.99	24.71	0.1730	0.895
2 Add Gabor splats and fit both groups jointly				
Joint fit of distribution, potential and flux	37.59	27.73	1.4×10^{-4}	0.984
3 Refine Gabor amplitudes at fixed geometry				
Gauss–Newton on all splats	37.77	−3.25	1.1×10^{-4}	1.037
No joint fit	37.67	16.45	1.9×10^{-4}	1.015
GyroSplat	37.59	27.71	5×10^{-7}	0.985

define $\kappa = (Q_+ + Q_-)/|Q|$ which measures the combined magnitude of the positive and negative contributions relative to their net. To measure the distance to ground-truth snapshots, we report $\kappa/\kappa_{\mathrm{truth}}$, dividing each reconstruction's ratio by its own snapshot's reference ratio before taking the median. We verify in Figure 2 that a divergence in in/outward transport is accurately captured by $\kappa/\kappa_{\mathrm{truth}}$ and provide further supporting examples in Figure 4 in the appendix.

4.4 Results

Improved reconstruction–physics trade-off. We report our main results in Table 1. At matched compression ratio our GyroSplat attains the highest PSNR_ϕ and the lowest error on heat flux for both compression ratios, highlighting the effectiveness of our multi-stage physics refinement. As noted in prior work there usually exists a reconstruction–physics trade-off (Seguí & Armour, 2026; Galletti et al., 2026a). We also observe this trade-off in our results. This indicates that our multi-stage optimization procedure achieves a favorable trade-off compared to neural appraoches to compression. We also show a qualitative example for inward/outward flux contributions in Figure 2 that are accurately captured by GyroSplat while being over or underestimated by competitors.

Table 3: Reconstruction and diagnostics after coefficient refinement. The rows compare how the three representations retain distribution and potential accuracy while fitting the full nonzero-mode flux spectrum. Scores are medians over 39 snapshots at the CR of 302×. Cancellation is normalized per snapshot. Bold marks the best column entry.

representation	$\text{PSNR}_f \uparrow$	$\text{PSNR}_\phi \uparrow$	flux relL1 $\downarrow$	$\kappa/\kappa_{\text{truth}}$
CR ≈302×				
Haar (+ our objective)	22.15	−11.49	$\mathbf{8.16\times10^{-8}}$	4.682
Fourier (+ our objective)	35.88	24.65	0.0128	2.928
GyroSplat (ours)	**37.59**	**27.71**	5.14×10^{-7}	**0.985**
CR ≈142×				
Haar (+ our objective)	32.15	−8.17	4.88×10^{-8}	2.246
Fourier (+ our objective)	37.38	27.73	0.0014	2.582
GyroSplat (ours)	**38.20**	**29.24**	$\mathbf{4.49\times10^{-8}}$	**0.971**

The fitting stages preserve different quantities. The ablations in Table 2 show why we fit the diagnostics in these stages. The potential objective improves the derived field without a large loss of distribution accuracy. Adding flux optimization to this stage helps the transport, but reduces accuracy of the other metrics. Introducing Gabor groups and fitting both groups jointly gives a better balance before the final amplitude polish. In the last stage, the polish reduces the flux error to near zero, while the field and potential metrics change little. It does not replace joint fitting, as the control without that step reduces the PSNR of the potential significantly. Moreover, using stage 3 to change the amplitudes of all splats, damages the potential reconstruction even further. More comparisons and their stopping points are given in Appendix C.

Physical refinement improves classical codes. Our proposed physics refinement shows strong performance when applied to GyroSplat. However, the objective itself is independent of the basis it is being applied to. To test its effect on classic codecs, we apply it to the Fourier top-K and the Wavelet top-K (Haar) codecs and report the results in Table 3. Indeed we observe that both methods experience a significant reduction in flux error, however they significantly degrade on the other reconstruction metrics. Furthermore, while flux error improves for both methods, they still introduce too much cancellation between outward and inward transport, as indicated by $\kappa/\kappa_{\text{truth}}$.

5 Limitations

Our evaluation is restricted to electrostatic turbulence with adiabatic electrons. Electromagnetic effects and kinetic electrons introduce additional fields and finer-scale structure, and whether GyroSplat's accuracy on derived quantities carries over to these regimes remains to be shown. Furthermore, storage conventions differ between baselines (Section 4.1), so compression ratios are not strictly comparable across all methods. Finally, we only consider the 3D potential field and the flux spectrum as derived quantities. The preservation of other diagnostics, and the behavior of simulations restarted from compressed snapshots, are left for future work.

6 Conclusion

Fusion reactor design requires thousands of gyrokinetic simulations for different plasma scenarios to quantify the loss of energy due to plasma turbulence. These simulations cause prohibitive storage demands forcing practitioners to discard useful data. We introduce GyroSplat as an attempt to alleviate this bottleneck. GyroSplat compresses gyrokinetic snapshots with 5D Gaussian–Gabor splats, fitting the 5D distribution function and 3D potential field before refining the splat amplitudes against the reference heat flux spectrum. On electrostatic turbulence with adiabatic electrons, it improves physical accuracy while retaining competitive 5D reconstruction at the evaluated budgets. Overall our results demonstrate that GyroSplat is a promising alternative to current compression methods, specifically due to its improved preservation of derived physics quantities.

In the future we aim to apply GyroSplat to gyrokinetic data beyond adiabatic electrons to kinetic electron and electromagnetic setups. Furthermore, we aim to investigate its performance beyond gyrokinetics to domains such as computational fluid dynamics, where derived quantities such as energy and enstrophy need to be preserved during compression.

AI-USE STATEMENT

We used generative AI in three ways. For writing assistance, it helped revise wording and improve clarity. For research ideation and execution, it helped discuss method, baselines and experiment choices, develop and debug code, analyze results, and prepare scientific figures. The authors are responsible for the final manuscript and its results.

REFERENCES

Mark Ainsworth, Ozan Tugluk, Ben Whitney, and Scott Klasky. Multilevel techniques for compression and reduction of scientific data-quantitative control of accuracy in derived quantities. *SIAM Journal on Scientific Computing*, 41(4):A2146–A2171, 2019. doi: 10.1137/18M1208885.

G Berkooz, P Holmes, and J L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25:539–575, 1993. doi: 10.1146/annurev.fl.25.010193.002543.

A. J. Brizard and T. S. Hahm. Foundations of nonlinear gyrokinetic theory. *Rev. Mod. Phys.*, 79: 421–468, Apr 2007. doi: 10.1103/RevModPhys.79.421.

Jong Choi, Michael Churchill, Qian Gong, Seung-Hoe Ku, Jaemoon Lee, Anand Rangarajan, Sanjay Ranka, Dave Pugmire, CS Chang, and Scott Klasky. Neural data compression for physics plasma simulation. In *Neural Compression Workshop @ ICLR*, 2021.

Albert Cohen, Ingrid Daubechies, and Jean-Christophe Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45(5):485–560, 1992. doi: 10.1002/cpa.3160450502.

Sandeep Cranganore, Andrei Bodnar, Arturs Berzins, and Johannes Brandstetter. Einstein fields: A neural perspective to computational general relativity. In *International Conference on Learning Representations*, volume 2026, pp. 46338–46401, 2026a.

Sandeep S. Cranganore, Andrei Bodnar, Gianluca Galletti, Fabian Paischer, and Johannes Brandstetter. ANTiC: Adaptive neural temporal in situ compressor. In *International Conference on Machine Learning (ICML)*, volume 306 of *Proceedings of Machine Learning Research*, 2026b.

Ingrid Daubechies. *Ten Lectures on Wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, 1992.

Ronald A. DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998. doi: 10.1017/S0962492900002816.

A. M. Dimits, G. Bateman, M. A. Beer, B. I. Cohen, W. Dorland, G. W. Hammett, C. Kim, J. E. Kinsey, M. Kotschenreuther, A. H. Kritz, L. L. Lao, J. Mandrekas, W. M. Nevins, S. E. Parker, A. J. Redd, D. E. Shumaker, R. Sydora, and J. Weiland. Comparisons and physics basis of tokamak transport models and turbulence simulations. *Physics of Plasmas*, 7(3):969–983, 03 2000. ISSN 1070-664X. doi: 10.1063/1.873896.

Landon Dyken, Sharmistha Chakrabarti, Nathan Debardeleben, Steve Petruzza, Qi Wu, Will Usher, and Sidharth Kumar. Efficient compression of structured and unstructured volumes via learned 3d gaussian representation, 2026.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. doi: 10.1007/BF02288367.

Lukas Einkemmer, Katharina Kormann, Jonas Kusch, Ryan G. McClarren, and Jing-Mei Qiu. A review of low-rank methods for time-dependent kinetic simulations. *Journal of Computational Physics*, 538:114191, 2025. ISSN 0021-9991. doi: 10.1016/j.jcp.2025.114191.

Dennis Gabor. Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering*, 93:429–441, 1946. doi: 10.1049/ji-3-2.1946.0074.

Gianluca Galletti, Gerald Gutenbrunner, Sandeep S. Cranganore, William Hornsby, Lorenzo Zanisi, Naomi Carey, Stanislas Pamela, Johannes Brandstetter, and Fabian Paischer. Physics-informed neural compression of high-dimensional plasma data, 2026a. arXiv:2602.04758.

Gianluca Galletti, Gerald Gutenbrunner, William Hornsby, Lorenzo Zanisi, Naomi Carey, Stanislas Pamela, Johannes Brandstetter, and Fabian Paischer. A shortcut to statistically steady-state turbulence with flow matching, 2026b. arXiv:2607.13022.

G. H. Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10(2):413–432, 1973. doi: 10.1137/0710036.

Peter J. Huber. Robust estimation of a location parameter. *Ann. Math. Statist.*, 35:73–101, 1964.

Bo Jiang, Youyuan Liu, Taolue Yang, Sheng Di, and Sian Jin. 3d gaussian splatting for scientific particle data compression and rendering, 2026.

Pu Jiao, Sheng Di, Hanqi Guo, Kai Zhao, Jiannan Tian, Dingwen Tao, Xin Liang, and Franck Cappello. Toward quantity-of-interest preserving lossy compression for scientific data. *Proc. VLDB Endow.*, 16(4):697–710, December 2022. doi: 10.14778/3574245.3574255.

Namgyu Kang, Jaemin Oh, Youngjoon Hong, and Eunbyung Park. Pig: Physics-informed gaussians as adaptive parametric mesh representations. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 73022–73046, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. doi: 10.1145/3592433.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR*, 2014.

Robert H. Kraichnan. Inertial ranges in two-dimensional turbulence. *The Physics of Fluids*, 10(7): 1417–1423, 07 1967. doi: 10.1063/1.1762301.

Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael Mahoney. Characterizing possible failure modes in physics-informed neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 26548–26560. Curran Associates, Inc., 2021.

Xin Liang, Kai Zhao, Sheng Di, Sihuan Li, Robert Underwood, Ali M. Gok, Jiannan Tian, Junjing Deng, Jon C. Calhoun, Dingwen Tao, Zizhong Chen, and Franck Cappello. Sz3: A modular framework for composing prediction-based error-bounded lossy compressors. *IEEE Transactions on Big Data*, 9(2):485–498, 2023. doi: 10.1109/TBDATA.2022.3201176.

Peter Lindstrom. Fixed-rate compressed floating-point arrays. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2674–2683, 2014. doi: 10.1109/TVCG.2014.2346458.

Qiang Liu, Mengyu Chu, and Nils Thuerey. Config: Towards conflict-free training of physics informed neural networks. In *International Conference on Learning Representations*, volume 2025, pp. 59531–59566, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019.

S.G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. doi: 10.1109/78.258082.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, pp. 405–421. Springer, 2020.

Mohammadreza Momenifar, Enmao Diao, Vahid Tarokh, and Andrew D. Bragg. A physics-informed vector quantized autoencoder for data compression of turbulent flow. In *2022 Data Compression Conference (DCC)*, pp. 01–10, 2022. doi: 10.1109/DCC52660.2022.00039.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4), July 2022. doi: 10.1145/3528223.3530127.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2nd edition, 2006.

Fabian Paischer, Gianluca Galletti, William Hornsby, Paul Setinek, Lorenzo Zanisi, Naomi Carey, Stanislas Pamela, and Johannes Brandstetter. Gyroswin: 5d surrogates for gyrokinetic plasma turbulence simulations. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 49227–49254. Curran Associates, Inc., 2025. doi: 10.52202/085713-1647.

Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 11 1901. ISSN 1941-5982. doi: 10.1080/14786440109462720.

A. G. Peeters, Y. Camenen, F. J. Casson, W. A. Hornsby, A. P. Snodin, D. Strintzi, and G. Szepesi. The nonlinear gyro-kinetic flux tube code GKW. *Computer Physics Communications*, 180(12): 2650–2672, 2009. doi: 10.1016/j.cpc.2009.07.001.

M. L. Rensen. Physics-informed Gaussian splatting: Solving partial differential equations with Gaussians. Master's thesis, Delft University of Technology, 2024. MSc thesis, Computer Graphics and Visualisation group; supervisors M. Weinmann, B. Buschmann, E. Eisemann.

G. Rozza, D. B. Phuong Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):229–275, 2008. doi: 10.1007/s11831-008-9019-9.

Aleix Seguí and Wesley Armour. A geometric lens on physics-aligned data compression. In *Proceedings of the 43rd International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2026.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 7462–7473, 2020.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6309–6318, 2017.

G.K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992. doi: 10.1109/30.125072.

Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021. doi: 10.1137/20M1318043.

Haato Watanabe, Kenji Tojo, and Nobuyuki Umetani. 3D Gabor splatting: Reconstruction of high-frequency surface texture using Gabor noise. In *Eurographics 2025 – Short Papers*. The Eurographics Association, 2025. doi: 10.2312/egs.20251040.

Lee Alan Westover. *Splatting: a parallel, feed-forward volume rendering algorithm*. The University of North Carolina at Chapel Hill, 1991.

Skylar Wurster, Ran Zhang, and Changxi Zheng. Gabor splatting for high-quality gigapixel image representations. In *ACM SIGGRAPH 2024 Posters*. Association for Computing Machinery, 2024. doi: 10.1145/3641234.3671081. URL https://doi.org/10.1145/3641234.3671081.

Xinjie Zhang, Xingtong Ge, Tongda Xu, Dailan He, Yan Wang, Hongwei Qin, Guo Lu, Jing Geng, and Jun Zhang. GaussianImage: 1000 FPS image representation and compression by 2D Gaussian splatting. In *European Conference on Computer Vision (ECCV)*, volume 15067 of *Lecture Notes in Computer Science*, pp. 327–345. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72673-6_18.

M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 07/2002-09/2002 2002.

A ELECTROSTATIC DATASET: EVALUATION PROTOCOL

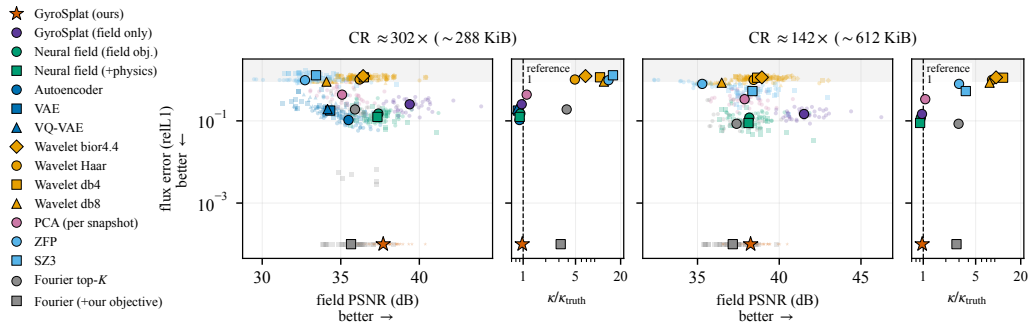


Figure 3: Electrostatic turbulence: reconstruction and transport at compression ratios near $302\times$ and $142\times$. Light points show individual snapshots and solid markers show method medians. Wide panels compare distribution PSNR and flux error. Narrow panels report $\kappa/\kappa_{\mathrm{truth}}$, with reference one. GyroSplat has median ratios 0.972 and 0.967, compared with 3.170 and 2.775 for the refined Fourier code.

This appendix gives the electrostatic snapshot selection, storage accounting, baseline configurations and diagnostic checks.

Evaluation-set construction. The electrostatic dataset consists of 39 held-out *snapshots* (single saturated-phase time slices, not full trajectories) drawn from 27 distinct nonlinear gyrokinetic flux-tube trajectories generated with the solver GKW (Peeters et al., 2009); each snapshot is the complex distribution function of the kinetic ion species, with the electrons treated adiabatically, stored as interleaved real and imaginary parts in GKW's normalization over real coordinates (the diagnostics Fourier-transform the periodic binormal axis, whose modes are the k_y of the banded metrics; §3). The classes: 12 streamer (7 trajectories), 12 mixed (11), 15 strong-zonal (11); two trajectories contribute snapshots to two classes (streamer and mixed) at different times, and at most three timesteps/snapshots come from any one trajectory.

Set construction and classification. The set was built by classifying a 309-candidate pool (the dataset's test split, earlier in/out-of-distribution probes, and 48 further held-out trajectories at several saturated timesteps) with a regime classifier fixed in advance, then balancing per class. The classifier reads each snapshot's own stored diagnostics. Normalizing the per-k_y power spectrum to unit mass, the zonal fraction is its $k_y{=}0$ share and the flux peak is the mode of the unit-normalized flux spectrum; a snapshot is strong-zonal at zonal fraction ≥ 0.60, otherwise streamer if its flux peak sits at mode ≥ 5, otherwise mixed.

Held-out dataset. None of the snapshots appears in the training set of the learned autoencoder, VAE or VQ-VAE baselines. They were also held out when developing GyroSplat's fitting stages. Every other family fits per snapshot and has no training set. Ramp-phase (pre-saturation) slices were excluded by convention: apparent "streamers" at early times are transients, and saturated streamer states exist only in a minority of the held-out trajectories, which is why the streamer class draws on fewer of them.

Aggregation across snapshots. We compute each metric on each snapshot and report the median of the 39 values. Means within a snapshot, such as the MSE used by PSNR, are a separate operation. For cancellation, each reconstructed ratio is divided by that snapshot's reference ratio before taking the median. Table 8 gives interquartile ranges, and Table 7 gives medians within the three regimes.

Evaluation metrics. All methods are evaluated on the same snapshots with the same diagnostic routines. For a reference array v and reconstruction $\hat{v}$, we use

$$\mathrm{PSNR}(v) = 10 \log_{10} \frac{\max_u |v(u)|^2}{\langle |\hat{v} - v|^2 \rangle_u}.$$

For the distribution, v contains the stored real and imaginary channels; for the potential, it is the real-space potential array. Let Q_m denote the heat flux at stored binormal mode m. We evaluate the complete nonzero spectrum,

$$\mathrm{relL1}(Q) = \frac{\sum_{m=1}^{31} |\hat{Q}_m - Q_m|}{\sum_{m=1}^{31} |Q_m|}, \qquad \kappa = \frac{Q_+ + Q_-}{|Q_+ - Q_-|}.$$

Here Q_+ and Q_- are the positive and negative contributions of the transport integrand, summed over the same nonzero modes and all remaining diagnostic coordinates. The tables report $\kappa/\kappa_{\mathrm{truth}}$, with the reference computed on that same domain.

Budgets. We compare methods near two target counts, 73,728 and 156,669 stored reals, against a raw snapshot of 22,282,240 reals. The deployed splat code also stores four normalization values, so its reported totals are 73,732 and 156,673. Other methods approach the targets through their coefficient count, latent size, matrix rank or compression tolerance. Their achieved ratios, including stored normalization for the neural fields, are reported in Table 8. A real-count ratio does not fully specify a file size: the saved codes use different precisions and require different metadata. We therefore distinguish this comparison from a comparison at equal serialized size. Table 6 includes a separate parameter-precision control for the splat model.

Storage accounting for top-K codes. A sparse transform code needs both coefficient values and a support map. The top-K entries in the tables use the value-count convention, whereas the archived files include the support map and the information needed to invert the transform. Neural compression has a different shared cost: the AE-family code is stored per snapshot, while the decoder and, where applicable, the codebook are shared. We retain these conventions in the tables and provide the complete artifact sizes separately.

Learned-latent budget. The learned latents (autoencoder, VAE, VQ-VAE) exist only at the 73,728-real budget (the PINC recipe's latent shape defines it), so the 142× block carries no learned-latent row and nothing is imported across tiers. Additional autoencoder ablations were not run, because of the limits of our computational budget.

Baselines. The autoencoder, VAE and VQ-VAE rows (van den Oord et al., 2017; Galletti et al., 2026a) use reconstruction-only configurations from PINC. We did not retrain these autoencoders with physics losses. PINC describes unstable end-to-end training and uses staged adapter fine-tuning. Its neural field has lower heat flux and potential errors than the fine-tuned autoencoders in its reported comparison, at different compression ratios. We therefore use the neural field for our physics-informed neural comparison, while retaining the absence of equally supervised autoencoder results as a limitation. These rows measure reconstruction-based compression; neural fields are per-snapshot MLP fits at 73,799 and 156,902 reals (within +0.15% of block budget) under a density objective and with an added physics stage, in the coordinate-network line (Sitzmann et al., 2020; Müller et al., 2022); wavelet rows are per-snapshot top-K (DeVore, 1998) in four bases (Haar; db4 and db8, the compactly supported orthonormal family (Daubechies, 1992); bior4.4, the CDF 9/7 biorthogonal pair (Cohen et al., 1992)); Principal component analysis (Pearson, 1901) (PCA) is per snapshot. The ZFP rows (Lindstrom, 2014) run a prior protocol of the error-bounded class (compressors that guarantee a per-value error tolerance): one 4D layout per channel pair, with the absolute tolerance searched per snapshot to the target CRs, the stream self-contained bytes. SZ3 (Liang et al., 2023), the other error-bounded row, is measured the same way; its 142× cell is the comparison's sharpest split between field and physics. Splat fits use a fixed random seed, while wavelet, PCA and ZFP rows are deterministic; learned and NF rows are single checkpoints or single fits, no seed averaging.

Baseline settings and parameter selection. We reuse the AE, VAE and VQ-VAE configurations for reconstruction-only training and retrained these model following the PINC (Galletti et al., 2026a)

recipe. Their architecture and training choices follow that work, rather than a new hyperparameter search for this comparison. We evaluate the models on the 39 held-out snapshots at their native 73,728-real latent budget. These rows do not include PINC's physics-informed autoencoder fine-tuning. The VAE is evaluated with its stochastic decoder.

The neural fields follow the PINC training procedure, with network size adjusted to the two storage tiers. They use discrete coordinate embeddings, SiLU activations and residual connections. Width and depth are 131 and 5 at $302\times$, and 158 and 7 at $142\times$, giving 73,799 and 156,902 parameters. The real and imaginary channels are standardized separately at each magnetic-moment coordinate. Reconstruction training uses 20 epochs of AdamW, learning rate 7×10^{-3}, weight decay 10^{-8} and batches of 4096 points. The learning rate follows cosine decay to 10^{-12}, while the sampled fraction of the grid increases linearly from 0.2 to 1. The physics-informed variant adds the PINC distribution, integral, spectral and monotonicity losses for 100 further epochs. This stage uses learning rate 5×10^{-4}, weight decay 10^{-12}, 20 warmup epochs and cosine decay to 10^{-8}. It selects the checkpoint by potential PSNR, evaluated every two epochs. Each neural field is fitted to its own snapshot. These settings are inherited from PINC's implementation and are fixed across the evaluation set.

For the classical codecs, parameter selection targets storage rather than a validation loss. Fourier keeps the largest coefficients of the full 5D transform. The wavelet comparison evaluates Haar, db4, db8 and bior4.4 with one decomposition level and periodic boundaries, retaining coefficients according to magnitude. PCA uses the velocity/field-line versus radial/binormal unfolding. Coefficient thresholds, PCA ranks and ZFP/SZ3 error tolerances are adjusted per snapshot to approach the requested compression ratio. The wavelet bases are reported as separate variants, rather than choosing a different basis for each snapshot. The budgets and support-index conventions are given above. The physically refined Fourier and Haar variants use the fixed objective and update schedule specified below.

Fitting seed. We keep the random seed fixed across the evaluation set so that each configuration uses a reproducible initialization. The resulting quartiles show how performance changes with the snapshot. Establishing the variation due to the fitting seed would require repeated fits of the current configuration; the present tables do not provide that estimate.

The flux integrand, $Q_\pm$, and κ. The reference diagnostics evaluate signed heat flux contributions over velocity, field-line position and Fourier indices $(v_\parallel, \mu, s, k_x, k_y)$. Summing them gives the net heat flux. We sum positive contributions into Q_+ and the magnitudes of negative contributions into Q_-, giving $Q = Q_+ - Q_-$ and $\kappa = (Q_+ + Q_-)/|Q|$. The signs refer to contributions in these diagnostic coordinates. The heat flux spectrum has already summed over most of them, so it does not reveal how much cancellation occurred within each mode.

Interpreting cancellation. We divide each reconstructed κ by the reference value of the same snapshot. Values above one indicate stronger cancellation relative to net transport; values below one indicate weaker cancellation. The two physical metrics answer different questions. A reconstruction can fit the net flux closely while retaining an inaccurate balance of positive and negative contributions, as the refined transform codes illustrate. Conversely, fitting the distribution alone can give a cancellation ratio close to the reference without recovering the potential or the flux accurately. We therefore read $\kappa/\kappa_{\mathrm{truth}}$ together with the other reconstruction metrics.

VAE sampling. We evaluate a fixed, reproducible draw for each VAE reconstruction and store its latent code. This keeps the representation used for scoring available for later checks. The table contains one draw per snapshot; it does not report an average over latent samples.

Numerical precision and saved representations. Every reported reconstruction is saved and evaluated again after loading its compressed representation. This separates the accuracy of the stored result from progress reported during fitting. Flux errors are computed from the stored nonzero-mode spectra (without zonal flow) at their recorded precision, including small nonzero residuals. Reproducing a diagnostic to numerical precision does not establish that every further reduction matters in a simulation, so we also examine distribution, potential and cancellation errors.

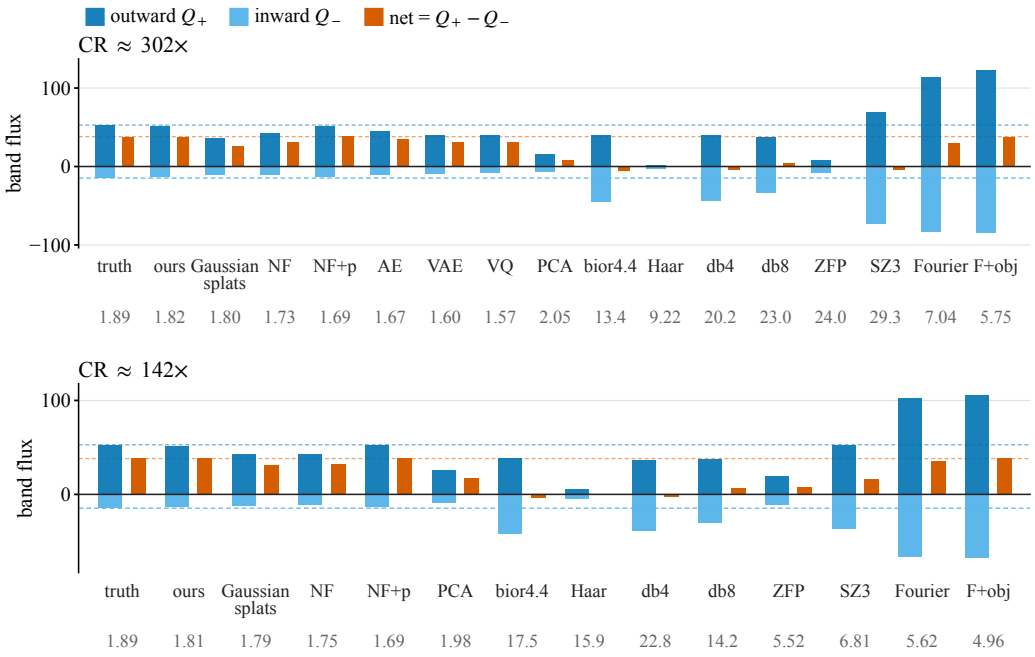


Figure 4: Positive, negative and net heat flux contributions for one streamer snapshot at both compression budgets. The dashed lines show the reference values. Annotations give the method's median $\kappa/\kappa_{\mathrm{truth}}$ over the evaluation set, rather than the ratio of the displayed snapshot. The bars illustrate how errors in opposing contributions can be hidden by their sum.

Repeated diagnostic evaluation. At fixed geometry, the assembled operators can be reused to evaluate flux and its derivatives directly from the coefficients. Full-field reconstruction still requires evaluating the splats. The flux has a constant Hessian at fixed geometry, although its squared residual loss generally does not.

Sign reversals. Relative flux error does not directly indicate whether the sign of net transport is preserved. We therefore also compare the signs of the predicted and reference sums over all nonzero modes. Each table entry counts the snapshots with opposite signs. Some wavelet codes reverse transport frequently despite good distribution reconstruction, while other methods preserve its sign but still have substantial errors in its magnitude or cancellation. These are different failures of the reconstructed diagnostics.

Wavelet variants. The additional wavelet rows test a representation choice without changing the fitting objective. All bases use one decomposition level, periodic boundaries and magnitude-based coefficient selection. Their distribution errors can be similar while their potential, flux and cancellation errors differ. Table 4 makes this dependence on the basis visible alongside the two wavelet rows in Table 1.

Table 4: The remaining wavelet top-K variants (evaluation-set medians; wrong-sign counts as in the sign-flip paragraph above; $\kappa/\kappa_{\mathrm{truth}}$ as in Table 1, reference one).

basis	CR	PSNR$_f$ ↑	PSNR$_\phi$ ↑	flux relL1 ↓	$\kappa/\kappa_{\mathrm{truth}}$	wrong sign
Haar	302.2×	36.09	−7.95	1.0156	5.385	35/39
db4	302.2×	36.46	−12.36	1.1470	11.189	39/39
db8	302.2×	34.09	−11.38	0.9054	13.540	3/39
Haar	142.2×	38.43	−7.13	0.9938	9.001	18/39
db4	142.2×	38.65	−9.23	1.1213	12.724	38/39
db8	142.2×	36.48	−8.40	0.8546	8.511	2/39

Fitting the distribution alone. The field-objective splat baseline asks what the representation can preserve without any loss on physical quantities. It uses global normalization, random centers and 1,000 Adam updates on distribution MSE, with no amplitude solve, potential fit or Gabor stage. The full budget is assigned to plain Gaussians, which need 16 real parameters because their modulation frequency is fixed at zero. This fit attains high distribution PSNR, but its potential and heat flux remain inaccurate. Its relatively close cancellation ratio does not remove those errors. The result motivates including the diagnostics in fitting rather than relying on distribution accuracy alone.

The Fourier pair. The refined Fourier row starts from the same top-K code as its reconstruction-only counterpart. Both use the nominal budgets of $73{,}728$ and $156{,}669$ real values; a complex coefficient occupies two values. We leave the selected coefficient positions unchanged and optimize their amplitudes against the physical objective. This isolates the effect of coefficient refinement within the retained Fourier subspace, without changing the storage allocation.

Coefficient refinement. Fourier and Haar coefficients are first optimized with Adam on the distribution, potential and flux losses, with weights 1, 0.5 and 0.5. Each loss is divided by its value before refinement. We use 400 steps, an initial learning rate of 3×10^{-3} and cosine decay. Up to eight damped Gauss–Newton steps then reduce the flux residual over modes 1–31.

What Fourier refinement changes. The Fourier pair shows that direct supervision of a diagnostic is useful beyond the splat representation. Changing the retained coefficients improves potential and flux accuracy while largely preserving the distribution fit. The improvement has limits: cancellation remains excessive and the full-spectrum flux residual does not reach the level of GyroSplat. We repeat the Fourier comparison at both storage tiers to avoid basing the conclusion on one code size. Refinement helps at both, but more retained coefficients permit a closer flux fit.

Refinement of a Haar code. We apply the coefficient updates to Haar to test how the reconstruction changes when a localized transform code is fitted to the same diagnostics. The flux can be matched closely, but doing so substantially reduces distribution PSNR and does not recover an accurate potential. Cancellation also remains too strong, despite improving during refinement. The control supports evaluating the whole reconstructed state after physical fitting, rather than ranking the methods by flux residual alone.

Rendering the distribution. Rendering reconstructs a snapshot on the simulation grid from the stored splat parameters. We evaluate the sum in Eq. 3 by using the separation between velocity and position in each splat. Grid coordinates are represented by their indices divided by the number of grid points along each axis, as during fitting.

For each splat, we evaluate its Gaussian envelope on the two-dimensional velocity grid and its spatial envelope on the three-dimensional position grid. The spatial factor also includes the oscillatory term of a Gabor splat. We collect these values in matrices $V \in \mathbb{R}^{N_v \times N}$ and $S \in \mathbb{C}^{N_p \times N}$, where N_v and N_p are the numbers of velocity and spatial grid points, and each column corresponds to one of the N splats. The normalized reconstruction is then

$$\widetilde{F} = V \ \mathrm{diag}(a)\, S^{\top}.$$

Here a contains the complex splat amplitudes, and $S^{\top}$ denotes the transpose. Each entry of $\widetilde{F}$ gives the reconstructed value at one combination of velocity and position.

We compute this expression with two real matrix multiplications, one for the real part and one for the imaginary part. The velocity factors are evaluated once per reconstruction and reused across spatial positions. To limit temporary memory, we evaluate the spatial factors and their matrix products in blocks. Every block includes all splats. This avoids constructing a full five-dimensional array for each individual splat.

Finally, we reshape the result into the original grid layout, with separate real and imaginary channels. We restore the stored normalization by rescaling the binormally averaged component and the remaining fluctuations separately and adding back their means. The potential and heat flux are evaluated after this step. The fitting stages differentiate through the same reconstruction to update the splat parameters.

B DIAGNOSTICS OF A LINEAR RECONSTRUCTION

We derive the potential and heat flux of a field written as a sum of fixed basis functions. Table 5 gives the dimensions for the electrostatic dataset.

From a distribution to potential and flux. Let $f \in \mathbb{C}^M$ denote the sampled distribution and $\mathcal{Q} : \mathbb{C}^M \to \mathbb{C}^P$ the discrete quasineutrality map. It integrates over velocity with the simulation weights and solves for the potential on the spatial grid. Thus $\phi = \mathcal{Q}f$. At each binormal mode, the heat flux diagnostic takes the real part of a sesquilinear pairing $\mathcal{B}_{k_y}(f, \phi)$. This pairing combines an energy-weighted velocity moment of f with the radial drift generated by ϕ, using the same quadrature as the simulation diagnostics.

Substituting the splat representation. At fixed geometry, column i of Ψ contains splat ψ_i evaluated on the grid. Substituting $\hat{f} = \Psi a$ into the diagnostic maps gives

$$\hat{\phi} = \mathcal{Q}(\Psi a) = \sum_i a_i \mathcal{Q}(\psi_i) = \Phi a,$$

$$\hat{Q}(k_y) = \mathrm{Re} \sum_{i,j} \bar{a}_i a_j \mathcal{B}_{k_y}(\psi_i, \mathcal{Q}(\psi_j)) = a^{\mathsf{H}} A_{k_y} a.$$

The potential matrix is $\Phi = [\mathcal{Q}(\psi_1) \ \cdots \ \mathcal{Q}(\psi_N)]$. Defining $C_{ij} = \mathcal{B}_{k_y}(\psi_i, \mathcal{Q}(\psi_j))$, the flux matrix is its Hermitian part $A_{k_y} = (C + C^{\mathsf{H}})/2$. The off-diagonal entries account for interactions between different splats. In the measured electrostatic splat banks, an isolated splat carries zero net heat flux, so transport comes from these pairwise terms. These expressions give the diagnostics of the reconstruction; their agreement with the reference depends on the fitted parameters.

Constructing the operators. Each unit-amplitude splat is evaluated on the grid and passed through the discrete velocity integrals and quasineutrality map. Its potential response becomes one column of Φ. Inner products of its velocity-moment and drift responses provide the entries of A_{k_y}, without rendering every pair of splats. On the electrostatic dataset's grid, those flux-side responses have $16 \cdot 85 = 1{,}360$ entries per mode, giving $\mathrm{rank}(A_{k_y}) \leq 2{,}720$. The operator can therefore be applied through its factors rather than stored as a dense matrix. It must be updated if the geometry changes.

Offsets and stored information. Fixed normalization offsets give an affine decoder $\hat{f} = b + \Psi a$. Augmenting a with a fixed coefficient of one and Ψ with the column b restores the same linear and quadratic notation. The splat parameters and normalization information are stored. Diagnostic matrices are derived from these parameters and shared simulation geometry, so storing those matrices is optional and would incur a separate caching cost.

Table 5: Shapes of the diagnostic operators for the electrostatic dataset. Grid sizes are specific to this dataset. The deployed storage tiers correspond to 4,337 and 9,216 independent 17-real splat equivalents; shared Gabor envelopes can change the number of basis columns.

object	shape	count	meaning
snapshot $\mathbf{x}$	$(32, 8, 16, 85, 32)$, $\mathbb{C}$	$M = 11{,}141{,}120$	distribution on the 5D grid
splat render ψ_i	same as $\mathbf{x}$	one per splat	unit-amplitude splat field
$\Psi(\theta)$	$M \times N$	–	decode matrix, $\hat{\mathbf{x}} = \Psi a$
a	N, $\mathbb{C}$	recipe-dependent	stored amplitudes
potential ϕ	$(16, 85, 32)$, $\mathbb{C}$	$P = 43{,}520$	spatial grid only
Φ	$P \times N$	–	column i = potential of splat i
A_{k_y}	$N \times N$, Hermitian	31 (modes 1..31)	$[A]_{ij}$ = pair (i, j) flux
splat parameters	17 reals	per splat	§3

C THE FITTING PIPELINE

Figure 1 and Section 3.3 give the encoding sequence, and Table 2 reports the main stage comparison. This appendix specifies the fitting settings and extends the comparison to the full set of ablations.

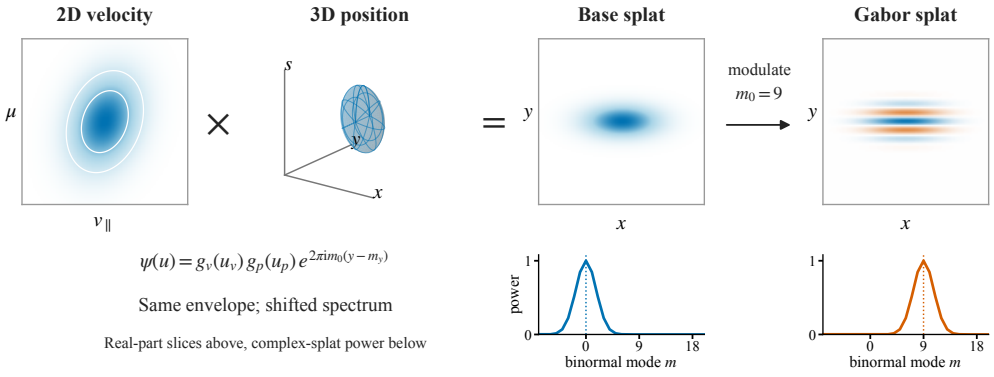


Figure 5: **Gaussian and Gabor splats in position and frequency.** The product of 2D velocity and 3D spatial envelopes forms a 5D Gaussian. The right panels show real (x, y) slices at fixed $v_\parallel$, μ and s, with blue for positive and orange for negative values. Below, the power spectra of the full complex splats show the shift from mode zero to the Gabor modulation mode $m_0 = 9$. The finite envelope gives a band of frequencies around the peak.

Table 6: Ablation of the fitting procedure on all 39 electrostatic snapshots. Colored groups identify the three fitting stages; normalization and initialization are operations within stage 1. The base-only and mixed representations are fitted under the same nominal budget, with different allocations of that budget. Alternative rows test the purpose of individual steps or compare changes to the fitting sequence; not every row changes only one setting. Actual compression ratios are shown, including the two reduced-storage controls. The flux metric covers modes 1–31.

stage / ablation	CR	$\mathrm{PSNR}_f \uparrow$	$\mathrm{PSNR}_\phi \uparrow$	flux relL1 $\downarrow$	$\kappa/\kappa_{\mathrm{truth}}$
1 Fit base splats					
floor (global norm, random placement)	302.2	39.17	7.92	0.2809	0.952
+ zonal-split normalization	302.2	38.23	13.26	0.3434	0.981
+ importance placement, exact amplitude init	302.2	39.06	8.95	0.4022	0.986
+ geometry warmup (field only)	302.2	37.94	9.00	0.3448	0.966
+ physics fine-tune (ϕ on, flux off)	302.2	38.10	25.94	0.3178	0.944
2 Add Gabor splats and fit jointly					
+ Gabor splats	302.2	37.84	17.40	0.2808	0.992
+ ϕ-aware observable co-fit	302.2	37.59	27.73	1.38×10^{-4}	0.984
3 Refine Gabor amplitudes					
+ flux Gauss–Newton GyroSplat (deployed)	302.2	37.59	27.71	5.14×10^{-7}	0.985
Alternative configurations					
co-fit → field-optimal solve (field_solve)	302.2	37.86	10.85	0.2905	1.005
polish without co-fit (gn_nocofit)	302.2	37.67	16.45	1.88×10^{-4}	1.015
no modulation, equal bytes, to fine-tune (noky)	302.2	38.15	26.20	0.3120	0.939
random placement, after field fitting (random_init)	302.2	39.02	15.86	0.2834	0.960
random placement, end-to-end (random_full)	302.2	36.28	26.67	5.57×10^{-6}	1.246
field fit without amplitude warm start (no_warmstart)	302.2	35.90	−1.85	0.7104	1.392
flux term on at the fine-tune (flux_on)	302.2	37.99	24.71	0.1730	0.895
all Gabor splats, after field and potential fitting (all_gabor)	302.3	36.85	23.32	0.1759	1.280
all Gabor splats, end-to-end (all_gabor_full)	302.3	36.86	10.22	7.53×10^{-8}	1.347
40% Gabor allocation, after amplitude initialization (frac40)	302.2	37.62	16.18	0.2701	1.023
60% Gabor allocation, after amplitude initialization (frac60)	302.3	37.03	13.01	0.2667	1.042
quantize to `fp16` (0.5× bytes) (s7_pack)	604.4	37.59	27.39	0.0016	0.985
`fp32` at matched parameter storage (N_{eq}=2168) (fp32match)	604.7	36.77	25.37	1.42×10^{-4}	1.043
base splats only, un-masked flux chain (noky_full)	302.2	35.43	−26.81	0.0357	1.739
deployed splats, un-masked flux chain (base_gnall)	302.2	37.77	−3.25	1.13×10^{-4}	1.037

The fit has the three stages defined in Section 3.3. Table 6 records intermediate operations within those stages, including normalization and initialization, and tests selected changes to the procedure. Its extra rows do not define additional fitting stages.

Budget allocation. We reserve one quarter of the parameter budget for Gabor groups. This fraction refers to storage, since sharing makes a Gabor group cheaper than storing its members independently. Each base splat uses 17 reals. Within a Gabor group, the five center coordinates and nine Cholesky entries are shared throughout fitting and stored once; each of the seven carriers has its own complex amplitude and modulation frequency. A group therefore uses $14 + 3 \cdot 7 = 35$ reals. The two presets contain 3,254 or 6,912 base splats and 526 or 1,119 Gabor groups. Including four normalization values, their totals are $73{,}732$ and $156{,}673$ reals. The expanded bases contain 6,936 and 14,745 functions.

Base centers are sampled with probability proportional to $|f_\star|^2$, with initial Gaussian standard deviations of two grid cells. We solve for the amplitudes before fitting the geometry. Gabor group centers are sampled from equal segments of the flattened grid so that the added groups are spread across the domain. Their initial standard deviations are two cells along $v_\parallel$, μ, s and x, and 1.5 cells along y. The seven initial binormal mode indices are $1, 6, 11, 16, 21, 26, 31$. These seeds cover the full range used by the flux objective. We first fit the Gabor amplitudes to the residual left by the base splats, then include the potential in the amplitude solve before joint fitting.

Fitting schedule. The field fit uses 1,000 AdamW steps with a cosine learning-rate schedule and Huber loss, followed by two updates at reduced learning rates (Loshchilov & Hutter, 2019). Amplitude solves use a ridge weight of 10^{-3}. We then run 40 epochs with distribution and potential losses and no flux term. The joint fit uses 60 epochs, extended to 120 when the ratio of potential-loss to field-loss geometry-gradient norms exceeds 50. Stage 3 applies eight damped Gauss–Newton steps to the Gabor amplitudes. In the evaluated implementation, the Jacobian is obtained by differentiating the flux readout through the reconstruction. The all-Gabor controls use blocks of 4,096 spatial points during fitting, compared with 16,384 in the other recipes.

What the stages are intended to preserve. The initial fits test how far accurate reconstruction of the distribution carries over to its diagnostics. Adding the potential objective improves the potential substantially without requiring a large change in distribution accuracy. Gabor splats and joint fitting then give the optimizer a separate group of parameters for reducing flux error. The table compares these configured fits at similar storage budgets, so differences between the base-only and mixed rows also include the change in budget allocation.

The role of the final refinement. The median flux error after the joint fit is small, but a few snapshots retain much larger residuals. For the snapshot with the largest pre-refinement flux error, the final stage reduces relative L1 from 0.186 to 7.4×10^{-6}. This is the main reason to keep the polish: it addresses fits that the median alone makes easy to overlook. It can still change the potential and cancellation, which are evaluated again afterwards.

Placement and amplitude initialization. The random_init and random_full controls replace importance placement with uniform random centers. We evaluate the change after the distribution fit and after the full recipe because a favorable early score may not lead to a better final reconstruction. Random placement is competitive at the early checkpoint but gives a poorer final balance of distribution and physical accuracy. The no_warmstart control omits the initial amplitude solve. Its poorer field and potential reconstruction shows the value of starting geometry fitting from amplitudes already adapted to the snapshot.

Replacing joint fitting with a field-only solve. The field_solve control starts from the initialized Gabor bank and fits all amplitudes to the distribution at fixed geometry. It replaces the potential-aware solve and joint optimization. This asks whether a simpler amplitude update can retain the physical accuracy of stage 2. It improves distribution PSNR relative to the joint fit, but its potential and flux remain much less accurate.

Skipping the joint fit before polishing. The gn_nocofit control keeps the Gabor splats but proceeds directly from their residual amplitude initialization to flux polishing. It omits the potential-aware amplitude solve and joint fitting. The flux error falls, but the potential is less accurate than in the deployed recipe. This shows why a small flux residual alone is not a sufficient reason to omit the preceding joint fit.

Introducing flux supervision in stage 1. The flux_on control adds the flux term while fitting the base Gaussians to the distribution and potential. It is evaluated at the end of stage 1. The added term reduces flux error, but distribution and potential PSNR decrease and cancellation moves farther from the reference. The control therefore exposes a trade-off within the base fit, which motivates giving flux its own parameter group in the later stage.

Allocating storage to the two splat groups. The frac40 and frac60 controls increase the Gabor share of the parameter budget from 0.25 to 0.40 and 0.60. At fixed storage, this leaves fewer parameters for base splats. We evaluate these rows after Gabor amplitude initialization, before the joint fit. More Gabor capacity slightly reduces flux error at that point, but distribution and potential accuracy decrease. These rows illustrate the allocation trade-off; they do not establish the best fraction after a complete fit.

Using only Gabor splats. The all_gabor control allocates the whole budget to shared Gabor groups before fitting the distribution and potential. It tests whether a separate unmodulated base group is needed in this recipe. The all_gabor_full continuation adds a potential-aware amplitude solve and Gauss–Newton updates to all amplitudes. It reaches a very small flux residual, but loses potential accuracy and retains too much cancellation. These results support keeping separate roles for the two groups when the aim is to preserve all the reported quantities.

Allowing the polish to change every amplitude. The base_gnall control lets the flux update change both base and Gabor amplitudes. It also omits the joint fit, although it retains the potential-aware amplitude solve. The poor potential reconstruction shows the risk of using this shortened fitting sequence. Because both the starting fit and the update mask differ from the deployed model, this row does not isolate the effect of the mask alone.

Representation size and parameter precision. The s7_pack control rounds the fitted parameters to `fp16` without refitting them. It tests how much of the final accuracy depends on storing those parameters at higher precision. The fp32match control instead fits a smaller representation in `fp32` at approximately the same parameter-storage cost. Keeping more splats at lower precision gives better distribution and potential PSNR, whereas the smaller `fp32` representation gives a lower flux error. The preferred choice therefore depends on which reconstruction quantities must remain accurate.

A Gaussian-only route to the physical fit. The noky control omits the known-zero modulation parameter from plain Gaussians and spends the saved storage on additional base splats. It is evaluated after fitting the distribution and potential, where it performs similarly to the corresponding base-only recipe. This tests a storage choice before Gabor splats are introduced; it is not by itself a test of their role in the complete pipeline. The noky_full continuation applies an unmasked flux polish to the Gaussian amplitudes. Its remaining flux error and poor potential reconstruction show that this particular Gaussian-only fitting sequence does not recover the balance achieved by the mixed representation.

D DISPERSION AND PER-CLASS MEDIANS

The aggregate comparison does not show whether a method works consistently across snapshots or mainly on one regime. Table 8 therefore reports the middle half of the per-snapshot scores, and Table 7 separates streamer, mixed and strong-zonal states. For GyroSplat, streamers remain more difficult to reconstruct in both the distribution and potential, although flux errors are small in every class.

Table 7: Per-class medians at CR≈302× for 12 streamer, 12 mixed and 15 strong-zonal snapshots. Each metric is computed on a snapshot before aggregation within its class. Flux error uses all nonzero binormal modes; $\kappa/\kappa_{\mathrm{truth}}$ is normalized per snapshot, with one indicating agreement. The rows include the main baselines, additional wavelet bases and physically refined transform codes.

	streamer (12)				mixed (12)				strong zonal (15)			
family	PSNR_f	PSNR_ϕ	relL1	$\kappa/\kappa_{\mathrm{truth}}$	PSNR_f	PSNR_ϕ	relL1	$\kappa/\kappa_{\mathrm{truth}}$	PSNR_f	PSNR_ϕ	relL1	$\kappa/\kappa_{\mathrm{truth}}$
GyroSplat (ours)	37.21	26.91	1.35×10^{-6}	1.037	38.06	27.04	5.69×10^{-7}	0.974	37.59	30.79	3.41×10^{-7}	0.965
GyroSplat (field obj. only)	39.59	9.72	0.3322	1.008	39.72	8.23	0.2673	0.960	38.78	5.84	0.2228	0.930
Neural field (field obj.)	37.42	18.15	0.1949	0.956	37.67	12.53	0.1620	0.921	37.32	14.93	0.1306	0.890
Neural field (+physics)	37.19	24.87	0.1218	0.937	37.65	26.09	0.1460	0.894	37.18	29.81	0.1476	0.864
Autoencoder	34.15	17.09	0.1924	0.904	35.91	12.37	0.1291	0.879	36.06	16.01	0.1003	0.842
VAE	32.45	12.61	0.2848	0.877	34.71	8.56	0.1973	0.821	34.97	8.70	0.1483	0.799
VQ-VAE	33.30	13.51	0.2483	0.860	34.41	12.08	0.2220	0.797	34.48	12.69	0.1794	0.763
Wavelet (bior4.4)	36.90	−3.89	1.2406	6.591	36.67	−13.74	1.2048	7.758	35.98	−17.28	1.2367	7.600
Wavelet (Haar)	36.39	0.84	1.0152	4.802	36.36	−9.36	1.0159	5.064	35.66	−12.20	1.0150	6.388
Wavelet (db4)	36.79	−3.85	1.1949	8.142	36.71	−13.77	1.1458	11.423	35.98	−17.81	1.1398	13.109
Wavelet (db8)	34.47	−2.14	0.9042	12.565	34.57	−12.36	0.9032	13.788	33.86	−14.91	0.9085	15.987
PCA	34.71	6.16	0.5351	1.116	35.45	−5.83	0.5851	1.130	35.08	−4.39	0.3486	1.056
ZFP	32.40	2.99	0.9505	7.800	32.81	−6.01	0.9821	20.838	32.80	−11.06	0.9763	19.717
SZ3	34.27	−11.76	1.0552	37.093	33.77	−16.57	1.3333	15.495	32.61	−16.33	1.3235	19.787
Fourier top-K	35.58	21.89	0.2574	3.833	36.32	22.56	0.2135	3.687	36.22	27.67	0.1694	3.396
Fourier (+ our objective)	35.45	22.91	0.0085	2.911	36.32	24.19	0.0122	2.953	36.22	28.68	0.0173	2.883
Haar (+ our objective)	23.40	−3.31	8.39×10^{-8}	4.834	22.89	−12.43	6.69×10^{-8}	4.115	21.12	−16.02	9.40×10^{-8}	5.465

Table 8: Variation across the 39 evaluation snapshots: median [25th, 75th percentile] for each method and budget. The cancellation ratio is normalized before aggregation. Sign flips count snapshots for which the summed nonzero-mode heat flux has the opposite sign to the reference. These intervals describe variation across snapshots at the evaluated checkpoint or fitting seed.

family	CR	$\mathrm{PSNR}_f\uparrow$	$\mathrm{PSNR}_\phi\uparrow$	flux relL1 ↓	$\kappa/\kappa_{\mathrm{truth}}$	sign flips ↓
CR ≈302×						
GyroSplat (ours)	302.2×	37.59 [37.08, 38.35]	27.71 [26.70, 30.25]	$5.14\,[0.67, 24.06]\times10^{-7}$	0.985 [0.953, 1.014]	0
GyroSplat (field obj. only)	302.2×	39.33 [38.64, 40.09]	7.87 [0.17, 12.46]	0.2674 [0.2211, 0.3281]	0.951 [0.920, 1.003]	0
Neural field (field obj.)	301.8×	37.39 [36.77, 37.89]	14.73 [11.78, 17.93]	0.1729 [0.1281, 0.1977]	0.901 [0.882, 0.949]	0
Neural field (+physics)	301.8×	37.33 [36.71, 37.84]	26.50 [25.15, 29.46]	0.1443 [0.1207, 0.1616]	0.887 [0.854, 0.934]	0
Autoencoder	302.2×	35.49 [34.86, 36.26]	15.42 [11.77, 18.39]	0.1308 [0.0995, 0.1796]	0.864 [0.842, 0.900]	0
VAE	302.2×	34.35 [33.58, 35.38]	9.33 [6.81, 13.15]	0.1925 [0.1455, 0.2692]	0.819 [0.790, 0.867]	0
VQ-VAE	302.2×	34.16 [33.42, 34.82]	12.91 [9.50, 14.52]	0.2176 [0.1821, 0.2433]	0.793 [0.762, 0.849]	0
Wavelet (bior4.4)	302.2×	36.40 [35.81, 37.00]	−12.35 [−19.60, −3.85]	1.2260 [1.1551, 1.2576]	7.428 [6.739, 9.558]	39
Wavelet (Haar)	302.2×	36.09 [35.52, 36.58]	−7.95 [−15.21, 0.84]	1.0156 [1.0103, 1.0173]	5.385 [4.599, 6.659]	35
Wavelet (db4)	302.2×	36.46 [35.82, 37.00]	−12.36 [−19.51, −3.85]	1.1470 [1.1077, 1.1784]	11.189 [8.533, 14.124]	39
Wavelet (db8)	302.2×	34.09 [33.71, 34.71]	−11.38 [−18.68, −2.14]	0.9054 [0.8958, 0.9245]	13.540 [10.687, 16.322]	3
PCA	302.7×	35.08 [34.50, 36.06]	−2.54 [−8.68, 4.97]	0.4551 [0.3474, 0.6460]	1.102 [1.023, 1.249]	0
ZFP	304.8×	32.72 [32.16, 33.13]	−5.61 [−12.59, 3.08]	0.9794 [0.9300, 0.9909]	18.972 [5.025, 26.538]	5
SZ3	300.8×	33.45 [32.54, 34.32]	−15.87 [−24.21, −9.32]	1.2873 [1.0590, 1.4424]	19.873 [11.675, 45.645]	34
Fourier top-K	302.2×	35.90 [35.34, 36.57]	23.62 [21.27, 27.06]	0.2104 [0.1693, 0.2481]	3.659 [3.325, 4.135]	0
Fourier (+ our objective)	302.2×	35.88 [35.32, 36.57]	24.65 [23.10, 27.99]	0.0128 [0.0092, 0.0175]	2.928 [2.801, 3.274]	0
Haar (+ our objective)	302.2×	22.15 [19.68, 23.83]	−11.49 [−17.79, −4.54]	$8.16\,[6.03, 10.46]\times10^{-8}$	4.682 [4.011, 5.879]	0
CR ≈142×						
GyroSplat (ours)	142.2×	38.20 [37.64, 38.90]	29.24 [28.47, 31.63]	$4.49\,[2.32, 14.15]\times10^{-8}$	0.971 [0.955, 0.993]	0
GyroSplat (field obj. only)	142.2×	41.43 [40.51, 42.02]	9.60 [4.38, 15.12]	0.1608 [0.1402, 0.1917]	0.961 [0.948, 0.983]	0
Neural field (field obj.)	142.0×	38.17 [37.65, 38.82]	14.66 [12.07, 20.01]	0.1450 [0.1104, 0.1714]	0.915 [0.899, 0.960]	0
Neural field (+physics)	142.0×	38.12 [37.42, 38.49]	26.99 [26.06, 28.94]	0.1041 [0.0874, 0.1282]	0.894 [0.862, 0.935]	0
Wavelet (bior4.4)	142.2×	38.94 [38.33, 39.58]	−9.82 [−16.55, −1.56]	1.1491 [1.0982, 1.2031]	10.062 [7.932, 12.471]	38
Wavelet (Haar)	142.2×	38.43 [37.92, 38.98]	−7.13 [−13.27, 1.26]	0.9938 [0.9686, 1.0065]	9.001 [5.276, 24.797]	18
Wavelet (db4)	142.2×	38.65 [38.23, 39.27]	−9.23 [−16.35, −0.59]	1.1213 [1.0614, 1.1469]	12.724 [9.966, 19.005]	38
Wavelet (db8)	142.2×	36.48 [35.86, 36.96]	−8.40 [−14.06, 1.71]	0.8546 [0.8290, 0.8723]	8.511 [6.903, 10.157]	2
PCA	143.4×	37.88 [37.33, 38.75]	1.69 [−1.05, 7.87]	0.3513 [0.2881, 0.4276]	1.065 [1.016, 1.139]	0
ZFP	144.9×	35.29 [34.80, 36.17]	−3.39 [−6.40, 4.69]	0.7987 [0.7534, 0.8922]	3.769 [2.210, 4.563]	1
SZ3	142.8×	38.42 [37.72, 39.38]	−12.23 [−21.36, −6.61]	0.5367 [0.4394, 0.5856]	4.349 [3.295, 5.361]	0
Fourier top-K	142.2×	37.39 [36.60, 37.78]	27.06 [24.80, 30.46]	0.1005 [0.0869, 0.1190]	2.800 [2.636, 3.158]	0
Fourier (+ our objective)	142.2×	37.38 [36.59, 37.78]	27.73 [26.32, 31.00]	0.0014 [0.0000, 0.0028]	2.582 [2.392, 2.776]	0
Haar (+ our objective)	142.2×	32.15 [30.11, 34.72]	−8.17 [−14.05, −0.26]	$4.88\,[3.63, 6.81]\times10^{-8}$	2.246 [1.554, 2.717]	0