Benchmark Summary (v1)

- Generated: 2026-02-10 07:27:00
- Dataset root: `data/benchmarks/v1/offset_noise_36`
- n_samples: 36
- fluct_ratio: 0.07
- noise_ratio: 0.01
- seed: 123

How to Read (Quickstart)

- まず見る: `Global Best (per case)`（ケースごとの最良を俯瞰）。
- 次に見る: 各ケースの `Highlights (auto)` と `key_decomp_dashboard.png`（処理の全体像を最短で把握）。
- 分解: `field_rmse`, `field_r2` は **全係数での再構成**の良さ（可逆変換はほぼ1.0になりやすい）。
- 分解: `k_req_r2_0.95`, `field_r2_topk_k64` は **圧縮としての良さ**（本資料の主指標）。
- 分解: `field_r2_k*` / `mode_r2_vs_k.png` は **prefix-trunc（係数の並び順依存）**なので、wavelet/RBF等では解釈しづらい場合がある。
- 学習: `val_rmse/val_r2` は **係数空間**（decomposer/codecでスケールが違うため手法間比較に注意）。
- 学習: `val_field_rmse/val_field_r2` は **field空間**（比較しやすい、本資料の主指標）。
- 異常検知: `status=failed` と `error`（CSV）を確認。図は `key_decomp_dashboard.png` が最短。

Metrics (Definitions)

metric	stage	definition	mask & weights	notes
`field_rmse`	decomposition	eval mask内のRMSE（真値 field vs 再構成field）。	mask = domain mask ∩ dataset mask（存在する場合）、weightsはdomain weightsがあれば使用。	小さいほど良い。
`field_r2`	decomposition	eval mask内のR^2（**全係数**で再構成）。	同上	可逆変換（DCT/FFT等）はほぼ1.0になりやすい。
`field_r2_k{1,4,16,64}`	decomposition	係数を **prefix-trunc**（先頭Kだけ残して残り0）して再構成したR^2。	同上	係数順に意味が薄い手法（wavelet/RBF/dict等）では解釈が難しい場合がある。未対応layoutは空欄。
`field_r2_topk_k{1,4,16,64}`	decomposition	係数エネルギー `mean(coeff^2)` の大きい順に **top-K** を残して再構成したR^2。	同上	順序依存が小さく、手法間比較の補助に有用。
`n_components_required` (`n_req`)	decomposition	`energy_cumsum>=0.9` に到達する最小K（係数エネルギーの累積）。	coeffエネルギー（layout依存、channelsは合算）。	offset優勢データでは小さく出やすい。Kの"必要数"の目安。`fft2` は周波数半径順で累積するため、負周波数が配列末尾にあっても過大評価されにくい。
`k_req_r2_0.95`	decomposition	`field_r2_topk` が 0.95 に到達する最小K（gridから求める）。	同上	圧縮としての主指標（小さいほど良い）。未計算の場合は空欄。
`val_rmse/val_r2`	train	cond→coeff（target_space）予測の指標（validation）。	係数空間（codec/coeff_postに依存）。	手法間比較は注意。係数のスケールが違うと同じ意味にならない。
`val_field_rmse/val_field_r2`	train	予測係数→decode→inverse_transformで復元したfieldの指標（validation）。	mask = domain mask ∩ dataset mask（存在する場合）。	手法間比較しやすい主指標。
`decomp_r2`	train table	そのdecompose(cfg)の `field_r2`（分解が十分再構成できているかの参照）。	-	train性能の比較前に、分解自体が破綻していないか確認する。

(空欄は「未計算（係数レイアウト非対応 / field_eval未対応 / 例外でスキップ）」を意味します。)

Data Generation / Problem Setting (共通)

- 各サンプルは `field = offset + fluct + noise` を満たす（offset優勢）。
- `fluct` は offset の **5-10% 程度**（v1は `fluct_ratio=0.07`）。
- `noise` は offset の **約1%**（v1は `noise_ratio=0.01`）。

cond の定義

- scalar: `cond.shape=(N,4)`、`cond[:,0]=offset`、`cond[:,1:4]=pattern weights`

- vector: `cond.shape=(N,8)`、`cond[:,0]=offset_u`、`cond[:,1]=offset_v`、`cond[:,2:5]=weights_u`、`cond[:,5:8]=weights_v`

fluct/noise のスケーリング（概念式）

```
base = sum_j w_j * pattern_j            # patternはmask内でmean=0/std=1に正規化
base *= (fluct_ratio * offset) / RMS(base, mask)
noise ~ N(0,1)
noise *= (noise_ratio * offset) / RMS(noise, mask)
field = offset + base + noise
```

patterns（先頭3つのみ使用）

- rectangle/arbitrary_mask: `sin(2πx)`, `sin(2πy)`, `cos(2π(x+y))`
- disk/annulus: `r`, `r^2`, `cos(theta)`
- sphere_grid: `sin(lat)`, `cos(lat)`, `sin(lon)`

この問題で期待される挙動（判断のコツ）

- offset優勢のため、DC（定数）を1モードで持つ手法は `K=1` から `R^2` が上がりやすい。
- fluctは低次パターン中心のため、低次数基底（DCT/低次Zernike/低次Graph Fourier等）が有利になりやすい。
- mask境界付近だけ誤差が大きい場合は、境界条件・補間・maskの扱いの不整合が疑わしい（`per_pixel_r2_map` で確認）。

mask の扱い（ドメイン別）

- rectangle/sphere_grid: 全点有効（maskなし）。
- disk/annulus: 幾何マスク（領域外は0埋め、評価は領域内のみ）。
- arbitrary_mask: 固定の不規則マスク（`domain_mask.npy`）。領域外は0埋め、評価はmask内のみ。

Cases (ドメイン別テストケース)

case	domain	field	grid	range	notes
`rectangle_scalar`	rectangle	scalar	64x64	x=[-1.0, 1.0], y=[-1.0, 1.0]	
`disk_scalar`	disk	scalar	64x64	x=[-1.0, 1.0], y=[-1.0, 1.0]	center=[0.0, 0.0], radius=1.0
`annulus_scalar`	annulus	scalar	64x64	x=[-1.0, 1.0], y=[-1.0, 1.0]	center=[0.0, 0.0], r_inner=0.35, r_outer=1.0
`arbitrary_mask_scalar`	arbitrary_mask	scalar	64x64	x=[-1.0, 1.0], y=[-1.0, 1.0]	mask=domain_mask.npy
`sphere_grid_scalar`	sphere_grid	scalar	18x36	x=[-180.0, 170.0], y=[-90.0, 90.0]	n_lat=18, n_lon=36, lon_range=[-180.0, 170.0]
`rectangle_vector`	rectangle	vector	64x64	x=[-1.0, 1.0], y=[-1.0, 1.0]	
`disk_vector`	disk	vector	64x64	x=[-1.0, 1.0], y=[-1.0, 1.0]	center=[0.0, 0.0], radius=1.0
`annulus_vector`	annulus	vector	64x64	x=[-1.0, 1.0], y=[-1.0, 1.0]	center=[0.0, 0.0], r_inner=0.35, r_outer=1.0
`arbitrary_mask_vector`	arbitrary_mask	vector	64x64	x=[-1.0, 1.0], y=[-1.0, 1.0]	mask=domain_mask.npy
`sphere_grid_vector`	sphere_grid	vector	18x36	x=[-180.0, 170.0], y=[-90.0, 90.0]	n_lat=18, n_lon=36, lon_range=[-180.0, 170.0]
`mesh_scalar`	mesh	scalar	289x1	x=[-1.0, 1.0], y=[-1.0, 1.0]	planar triangulated grid mesh (289 verts)

Methods (実行した分解手法の特徴)

method	description
`annular_zernike`	Annulus向けZernike系基底。
`autoencoder`	Autoencoder（非線形圧縮）。Torch依存（環境により無効）。
`dct2`	2D DCT（Dirichlet境界）。高速で安定。
`dict_learning`	Dictionary Learning（スパース符号化）。係数が疎になりやすい。
`disk_slepian`	Disk Slepian（帯域制限 + 空間集中）。
`fft2`	2D FFT（周期境界）。高速だがマスクは0埋めが必要。
`fft2_lowpass`	
`fourier_bessel`	Fourier-Bessel（disk分離基底）。
`fourier_jacobi`	Fourier×Jacobi（disk分離基底、Zernike一般化）。
`gappy_graph_fourier`	固定基底 + 観測maskでridge最小二乗（可変mask向け）。
`gappy_pod`	Gappy POD（観測mask下で係数推定）。
`graph_fourier`	グラフラプラシアン固有基底。mask/不規則領域に適用可能（固定mask前提になりやすい）。
`helmholtz`	Helmholtz分解（周期境界、FFT）。ベクトル場のcurl-free/div-free分離。

method	description
helmholtz_poisson	Poissonソルバ系Helmholtz（periodic/dirichlet/neumann）。
laplace_beltrami	Laplace-Beltrami固有基底（mesh）。
pod	POD（SVD/PCA）。データ駆動の低ランク分解。
pod_em	欠損対応POD（EM/ALSで欠損を推定しつつ基底学習）。可変maskに強い。
pod_joint	ベクトル場を結合してPOD（u,v相関を活用）。
pod_joint_em	欠損対応のjoint POD（可変mask + u,v相関）。
pod_svd	POD（SVD実装）。
polar_fft	極座標リサンプル + FFT/DCT（近似、disk/annulus）。
pseudo_zernike	Pseudo-Zernike（Zernike一般化、disk）。
pswf2d_tensor	PSWF（近似的な帯域制限基底、tensor版）。
rbf_expansion	RBF基底 + ridge最小二乗。任意mask/可変maskにも適用しやすい。
spherical_harmonics	球面調和関数（sphere_grid）。
spherical_slepian	球面Slepian（sphere_grid）。
wavelet2d	2D Wavelet（多重解像度）。局所構造に強い。
zernike	Zernike（disk直交基底）。

Plot Guide (What each figure means)

file	stage	what it is	what to look for
plots/key_decomp_dashboard.png	decomposition	2x2ダッシュボード（R^2 vs K / scatter / true-recon-error / per-pixel R^2）。	まずこれを見る。
plots/mode_r2_vs_k.png	decomposition	R^2 vs K（field_r2_k*と同系、prefix-trunc）。	Kでの劣化の仕方を見る（順序依存に注意）。
plots/field_scatter_true_vs_recon_*.png	decomposition	真値 vs 再構成の散布図（R^2付き）。	バイアス（傾き/切片）や外れ値を確認。
plots/per_pixel_r2_map_*.png / *_hist_*.png	decomposition	位置ごとのR^2（サンプル方向の系列で算出）。	境界/特定領域だけ弱い等の空間バイアスを検知。
plots/coeff_spectrum.png	decomposition	係数エネルギースペクトル（layoutに応じて index/degree/2D を表示）。	どのモードが支配的か、top-Kの妥当性確認。
train/plots/val_residual_hist.png	train	係数予測残差の分布（val）。	外れ値や系統誤差の有無。
train/plots/field_eval/field_scatter_true_vs_pred_*.png	train	field空間での真値 vs 予測散布図（val）。	fieldとして妥当か（係数空間より解釈しやすい）。
train/plots/field_eval/per_pixel_r2_map_*.png	train	field空間での位置ごとのR^2（val）。	予測が空間的にどこで崩れているか。

(図が存在しない場合は、(a)係数レイアウト非対応、(b)例外でスキップ、(c)そのrunで無効化、のいずれかです。)

Results

Global Best (per case)

case	best_decomp(cfg)	rmse	r2	best_train(cfg)	val_rmse	val_r2	val_field_rmse	v
rectangle_scalar	fft2	1.969e-09	1.000000	fft2	2.401e-02	0.737012	3.405e-02	0
disk_scalar	pod	1.130e-07	1.000000	pseudo_zernike	4.240e-03	0.910056	2.967e-02	0
annulus_scalar	pod_em	7.847e-03	0.989865	annular_zernike	5.552e-03	0.945274	3.134e-02	0
arbitrary_mask_scalar	pod	9.150e-08	1.000000	wavelet2d_k64	1.032e-01	0.944795	1.584e-02	0
sphere_grid_scalar	dct2	4.517e-09	1.000000	spherical_harmonics_scipy_bench	9.012e-03	0.901779	2.440e-02	0

case	best_decomp(cfg)	rmse	r2	best_train(cfg)	val_rmse	val_r2	val_field_rmse	v
rectangle_vector	helmholtz	0.000e+00	1.000000	helmholtz	1.816e-02	0.883531	2.591e-02	0
disk_vector	pod_joint_em	7.975e-03	0.991907	pseudo_zernike	4.822e-03	0.891683	3.120e-02	0
annulus_vector	pod_joint_em	8.012e-03	0.992121	polar_fft	2.342e-02	0.897195	2.565e-02	0
arbitrary_mask_vector	pod_joint_em	8.016e-03	0.993009	gappy_graph_fourier_bench	1.331e-01	0.921325	2.091e-02	0
sphere_grid_vector	dct2	3.300e-09	1.000000	spherical_harmonics_scipy_bench	9.733e-03	0.897570	2.354e-02	0
mesh_scalar	laplace_beltrami	1.592e-02	0.961517	laplace_beltrami	7.961e-03	0.826122	3.168e-02	0

Method Summary (Across cases)

各methodについて、各case内でそのmethodの成功runが複数ある場合は「代表として最良（主に r2_topk_k64 最大）」を1つ選び、その代表の指標の中央値を示します（空欄はその指標が未計算/未実行）。

Scalar cases

method	n_cases	median_r2_topk_k64	median_k_req_0.95	median_val_field_r2
gappy_graph_fourier	1	0.976360	32	0.947691
wavelet2d	2	0.879115		0.899490
spherical_harmonics	1	0.937276		0.887717
autoencoder	3	0.976731	16	0.860531
pseudo_zernike	1	0.979626	8	0.858169
graph_fourier	5	0.976361	8	0.858125
disk_slepian	1	0.968803	32	0.857758
fourier_jacobi	1	0.975130	8	0.857242
fourier_bessel	1	0.968277	8	0.857099
zernike	1	0.967842	8	0.856603
dct2	2			0.855192
laplace_beltrami	1	0.961517	8	0.852310
pswf2d_tensor	1	0.980260	32	0.851345
rbf_expansion	3	0.970092	64	0.850842
pod_em	4	0.989825	4	0.849289
dict_learning	3	0.946666	32	0.845986
pod	3	1.000000	4	0.844380
polar_fft	2			0.844325
annular_zernike	1	0.957231	8	0.836965
pod_svd	1	1.000000	4	0.836885
fft2	1			0.836885
spherical_slepian	1	0.168643		0.831449
fft2_lowpass	1			

Vector cases

method	n_cases	median_r2_topk_k64	median_k_req_0.95	median_val_field_r2
spherical_slepian	1	0.377093		0.964466
autoencoder	1	0.978326	16	0.953719
pod_joint	1	0.992514	8	0.950003
pod	1			0.946499

method	n_cases	median_r2_topk_k64	median_k_req_0.95	median_val_field_r2
helmholtz	1			0.946499
helmholtz_poisson	1			0.946499
pod_joint_em	4	0.992475	6	0.936262
dct2	2			0.933236
spherical_harmonics	1	0.955897	8	0.929476
graph_fourier	4	0.982547	8	0.927420
rbf_expansion	3	0.980485	64	0.926714
gappy_graph_fourier	3	0.983751	4	0.926611
polar_fft	2			0.907899
pseudo_zernike	1	0.984511	4	0.892964
fourier_bessel	1	0.976025	4	0.892700

rectangle_scalar

Problem setting

- domain: rectangle ()
- field: scalar
- grid: 64x64, x=[-1.0, 1.0], y=[-1.0, 1.0]
- cond: (N,4)
- mask: all-valid (no mask)

Highlights (auto)

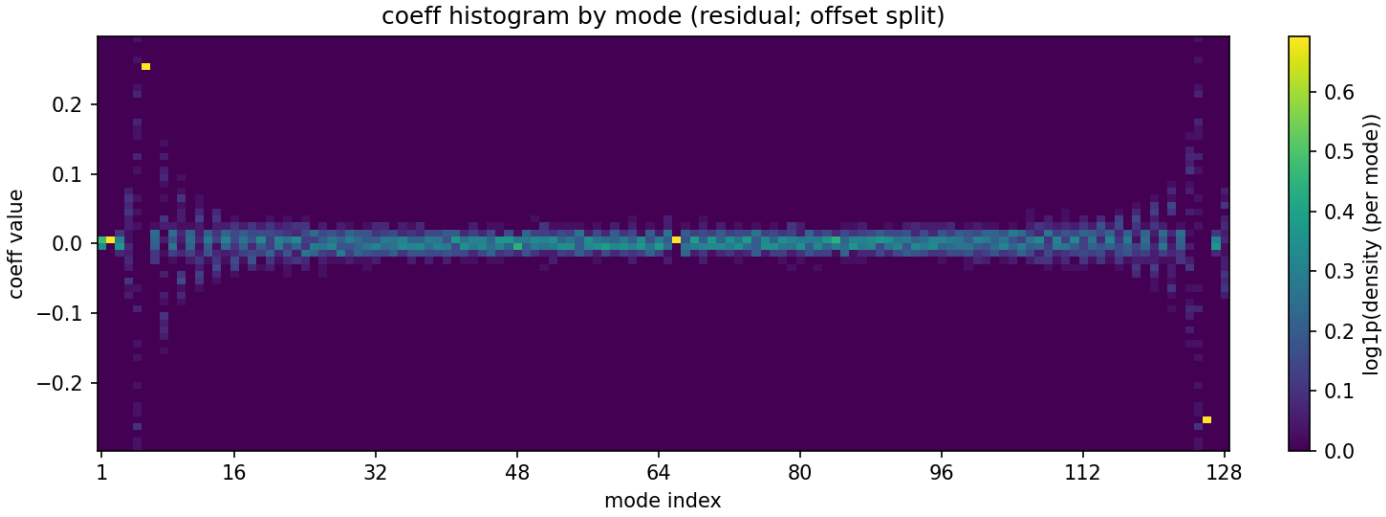
- decomposition: best full recon = fft2 (fft2) (field_rmse=1.969e-09, field_r2=1.000000)
- decomposition: best compression proxy = pod_svd (pod_svd) (k_req_r2_0.95=4, r2_topk_k64=1.000000)
- decomposition: best top-energy@64 = pod_svd (pod_svd) (r2_topk_k64=1.000000, k_req_r2_0.95=4)
- train: best coeff-space = fft2 (fft2) (ridge) (val_rmse=2.401e-02, val_r2=0.737012)
- train: best field-space = dict_learning_bench (dict_learning) (ridge) (val_field_rmse=2.772e-02, val_field_r2=0.836662)
- train: mismatch detected (best coeff-space != best field-space)

Decomposition (field reconstruction)

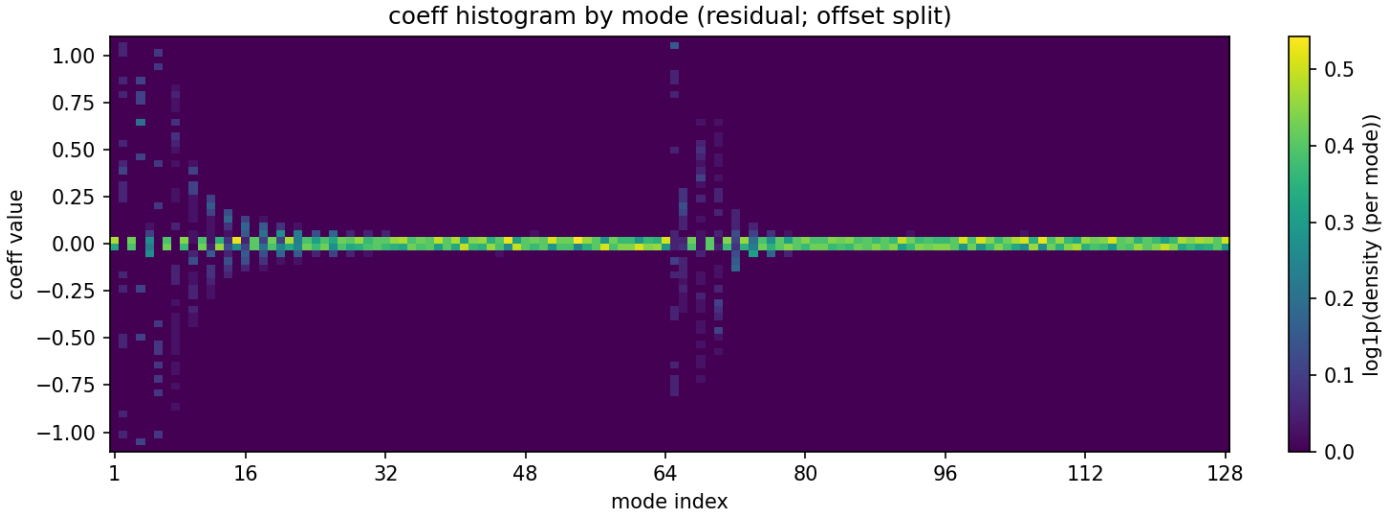
decompose(cfg)	method	status	rmse	r2	r2_k1	r2_k64	fit	n_req	k_req_r2_0.95	run
fft2	fft2	ok	1.969e-09	1.000000	-0.000000	0.979196	1.6ms	25		run
dct2	dct2	ok	5.624e-09	1.000000	-0.000000	0.969107	1.8ms	321		run
pod_svd	pod_svd	ok	8.548e-09	1.000000	0.441127	1.000000	22.1ms	3	4	run
pod	pod	ok	9.943e-08	1.000000	0.455484	1.000000	21.9ms	3	4	run
pod_em	pod_em	ok	7.836e-03	0.989747	0.455484	0.989747	238.5ms	3	4	run
pswf2d_tensor_bench	pswf2d_tensor	ok	1.145e-02	0.980260	-0.000000	0.980256	1.5ms	42	32	run
fft2_lowpass_k64	fft2_lowpass	ok	1.176e-02	0.979196	-0.000000	0.979196	2.0ms	25		run
autoencoder_bench	autoencoder	ok	1.244e-02	0.976666	0.009478	0.976666	8.188s	16	16	run
rbf_expansion_k64	rbf_expansion	ok	1.413e-02	0.970092	-11.636825	0.970092	1.8ms	56	64	run
graph_fourier_bench	graph_fourier	ok	1.480e-02	0.967036	-0.000000	0.967036	150.4ms	31	16	run
dict_learning_bench	dict_learning	ok	1.617e-02	0.959631	0.026836	0.959631	432.7ms	29	32	run
wavelet2d_k64	wavelet2d	ok	2.484e-02	0.907354	0.007147	0.907354	1.7ms	59		run

Coefficient histograms by mode (per decomposer)

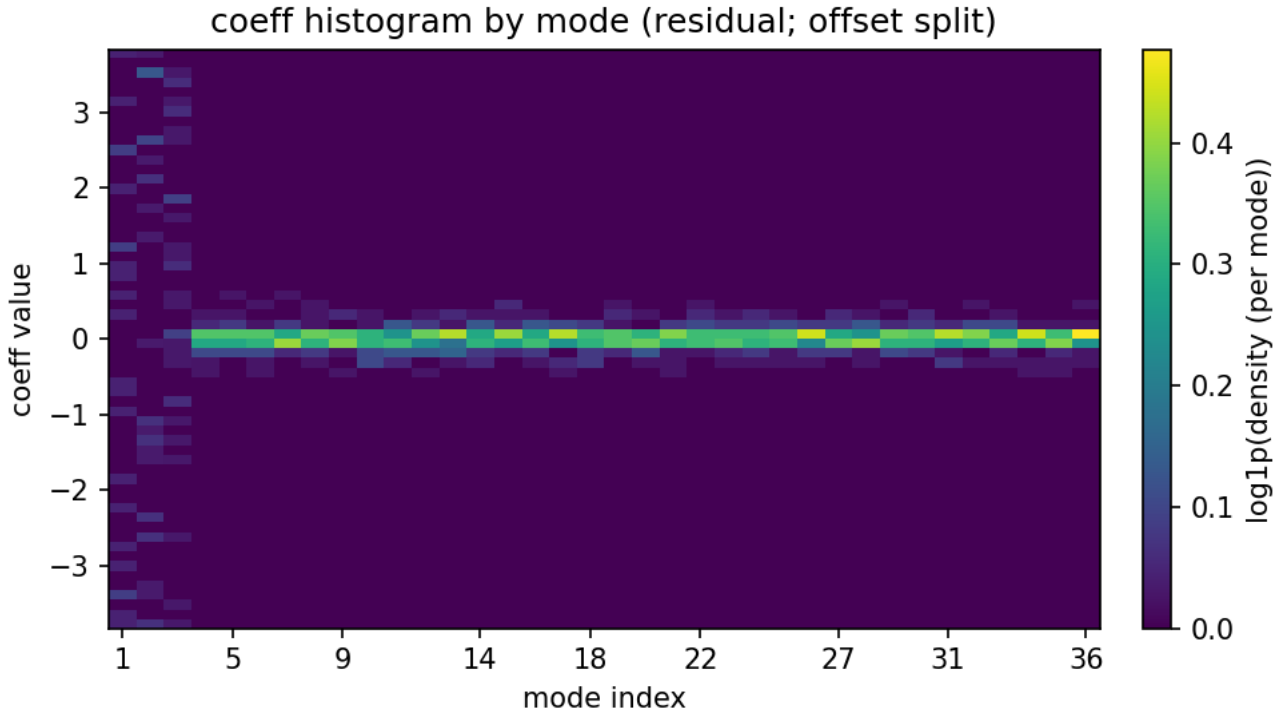
fft2 (fft2)



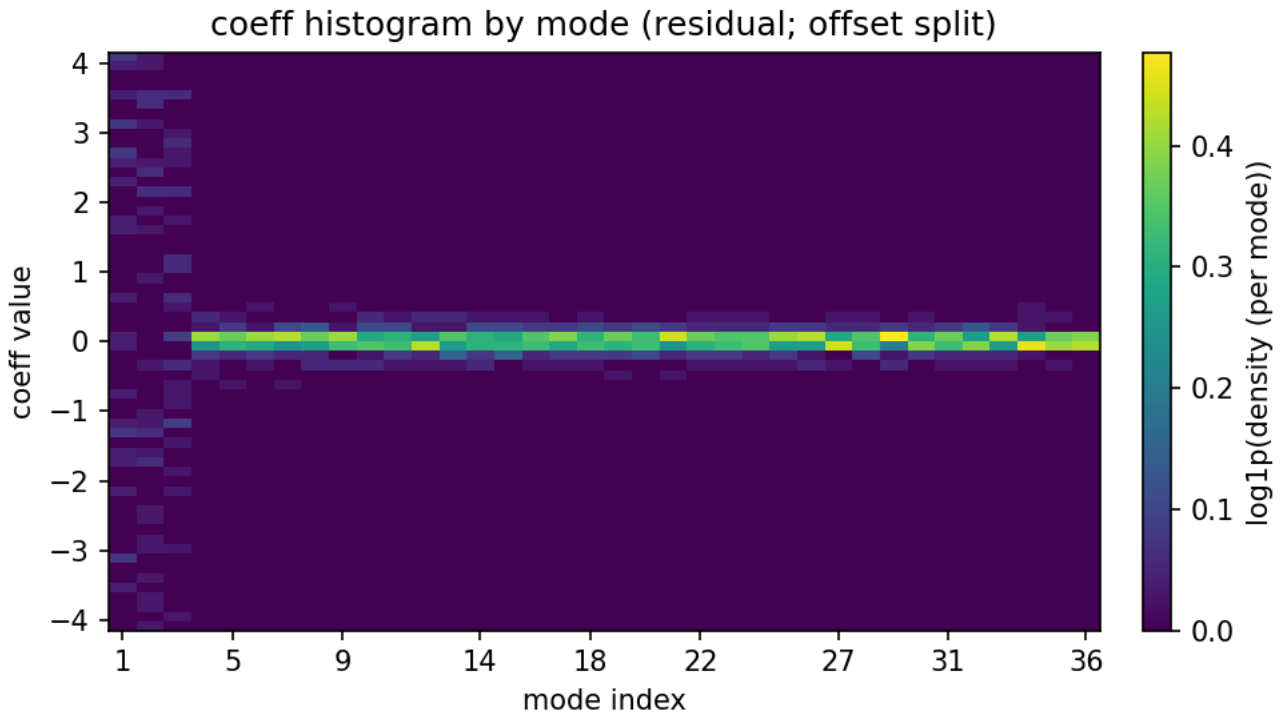
dct2 (dct2)



pod_svd (pod_svd)

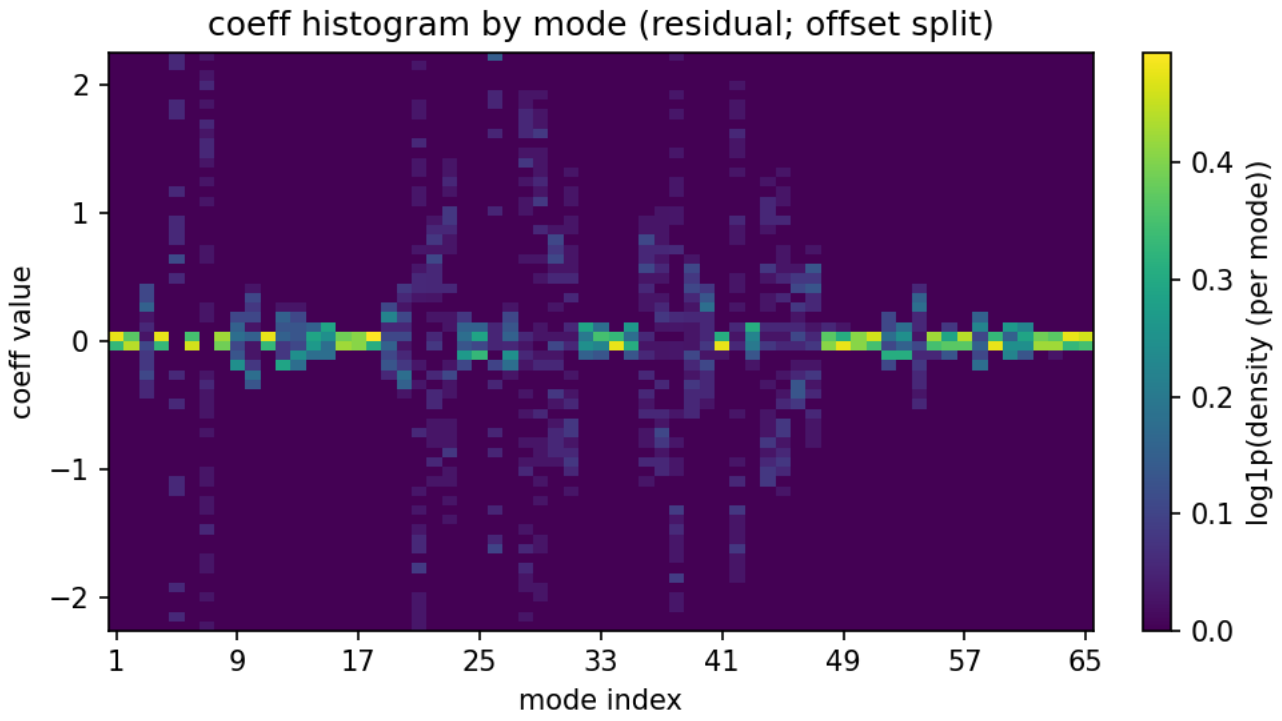


pod (pod)

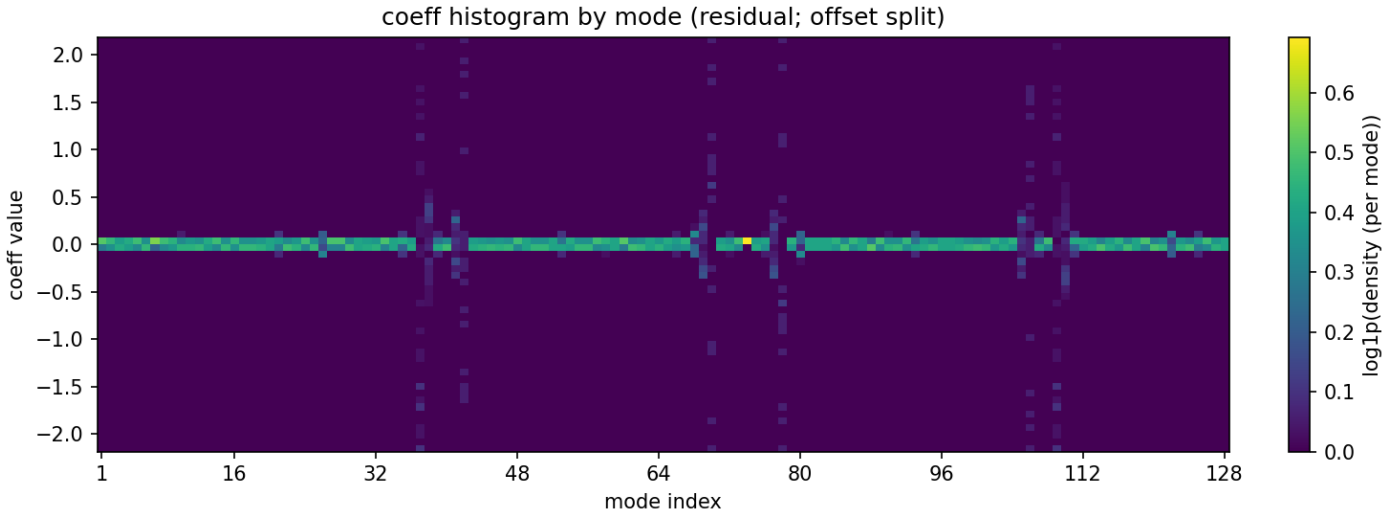


pod_em (pod_em)

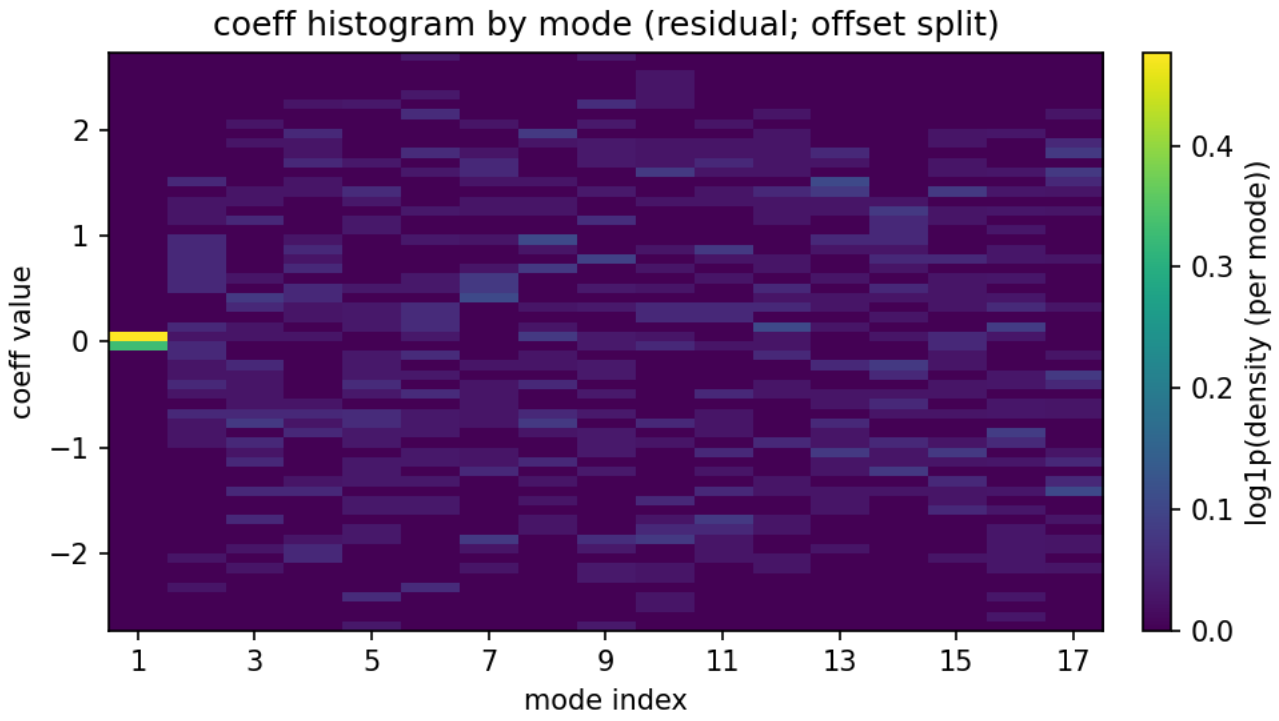
pswf2d_tensor_bench (pswf2d_tensor)



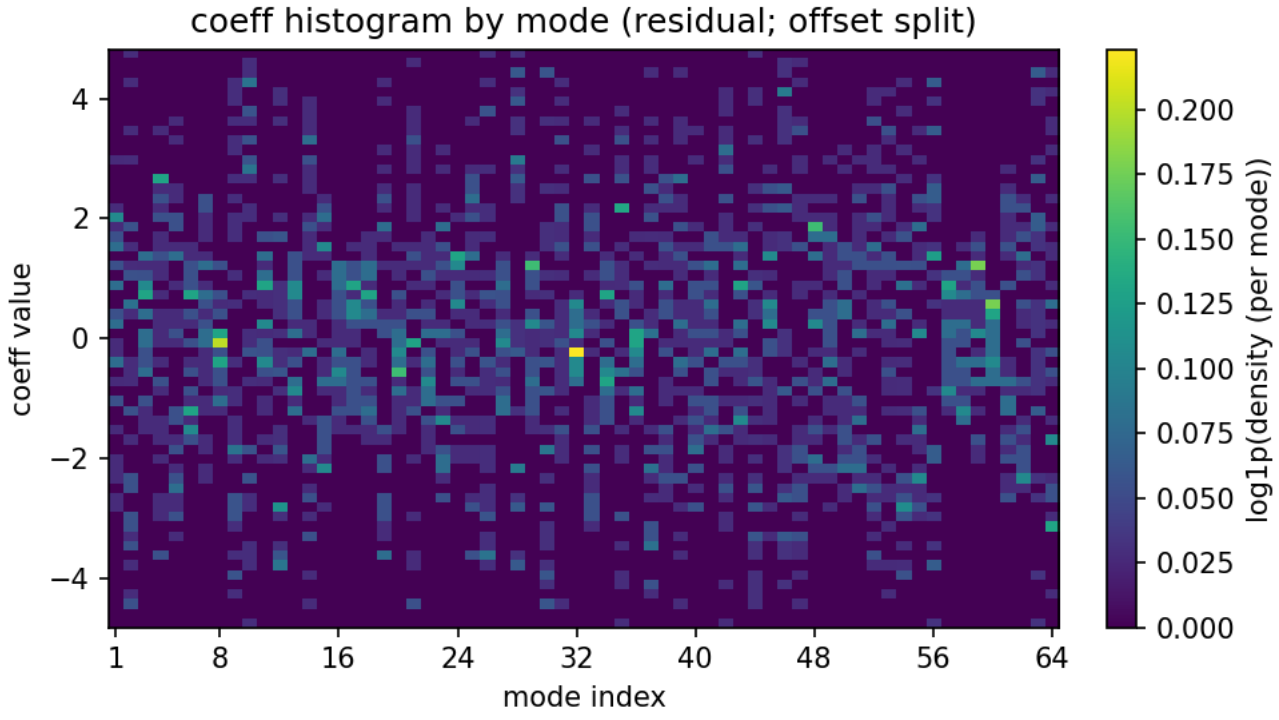
fft2_lowpass_k64 (fft2_lowpass)



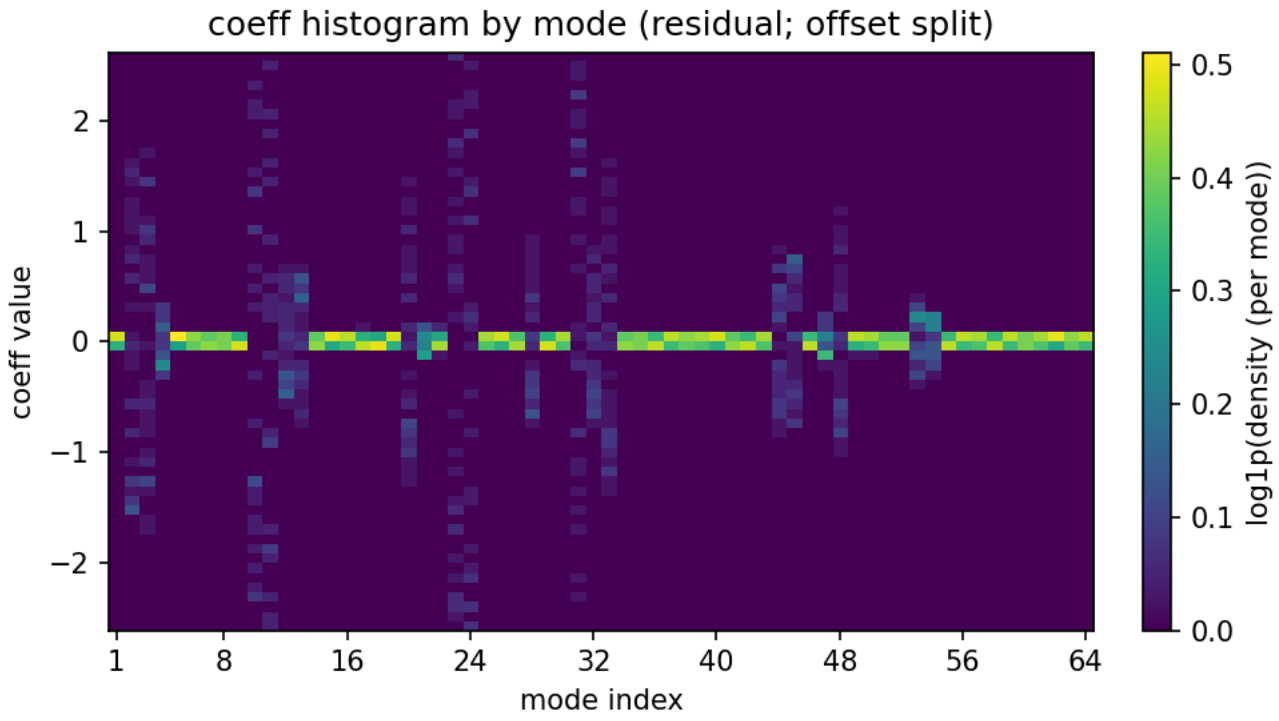
autoencoder_bench (autoencoder)



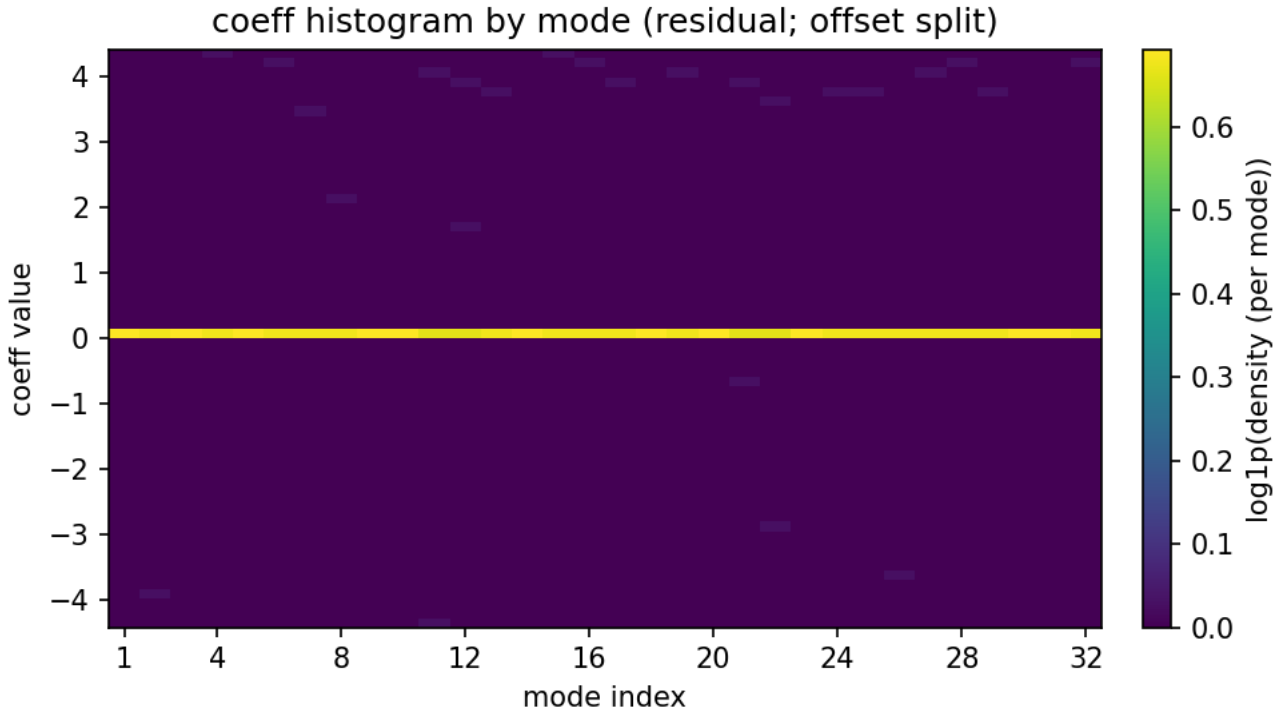
rbf_expansion_k64 (rbf_expansion)



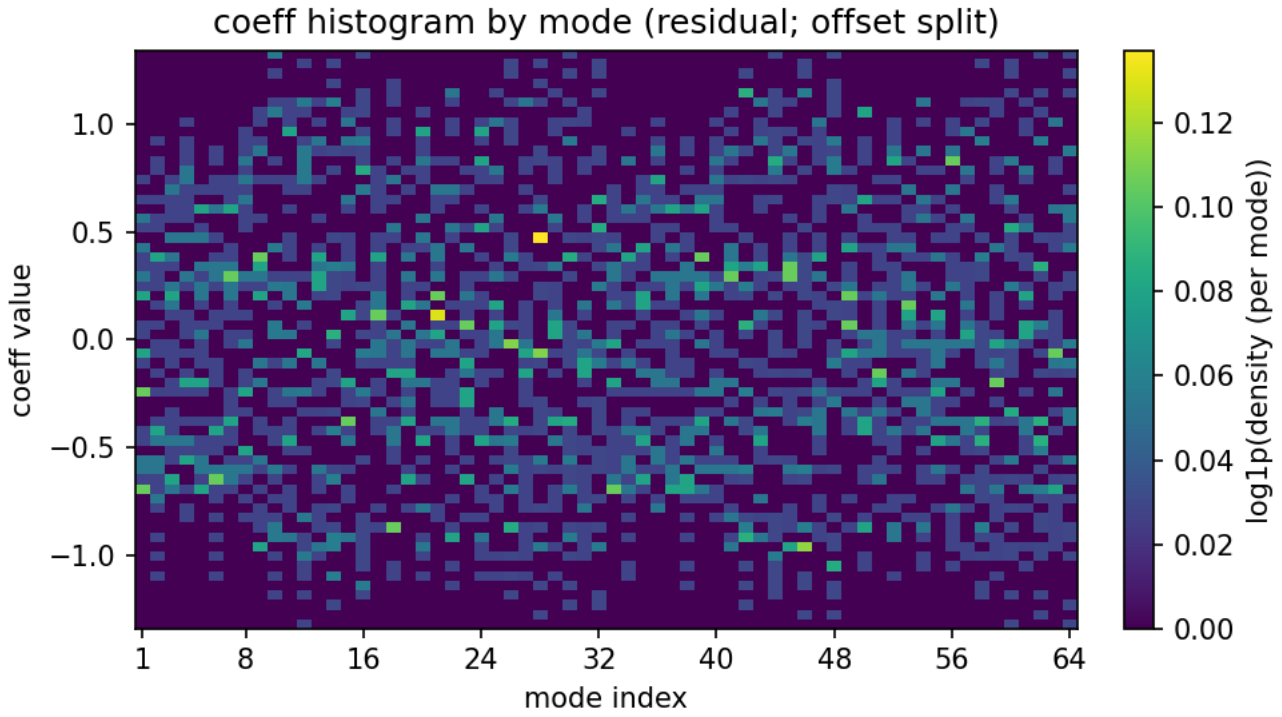
graph_fourier_bench (graph_fourier)



dict_learning_bench (dict_learning)



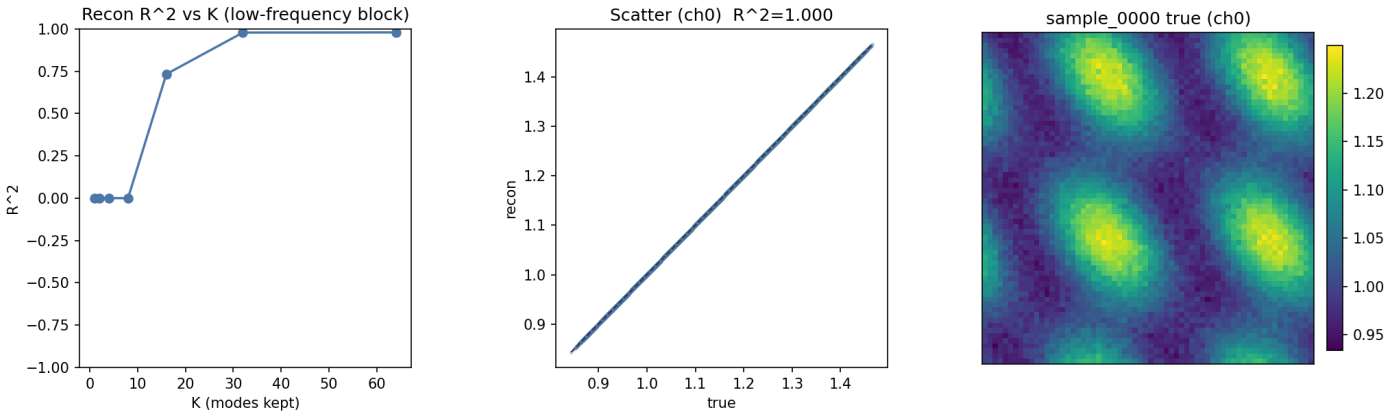
wavelet2d_k64 (wavelet2d)

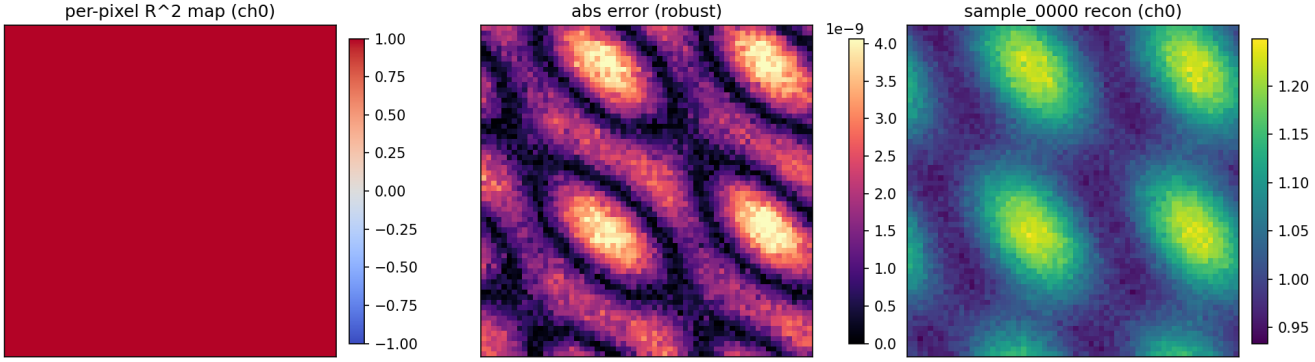


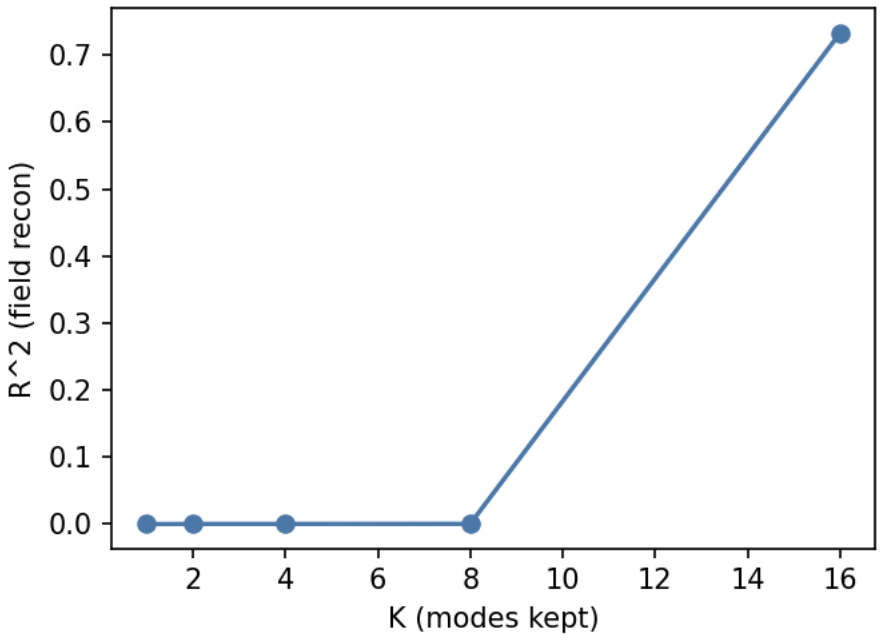
Compression leaderboard (Top-energy @K)

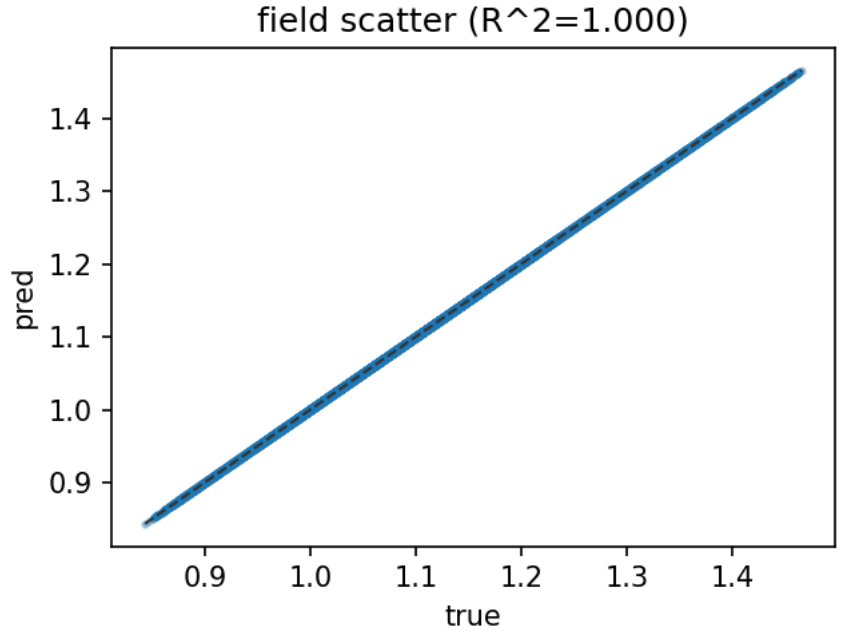
decompose(cfg)	method	r2_topk_k64	r2_topk_k16	k_req_r2_0.95	n_req
pod_svd	pod_svd	1.000000	0.989221	4	3
pod	pod	1.000000	0.989747	4	3
pod_em	pod_em	0.989747	0.989747	4	3
pswf2d_tensor_bench	pswf2d_tensor	0.980260	0.942515	32	42
autoencoder_bench	autoencoder	0.976666	0.976666	16	16

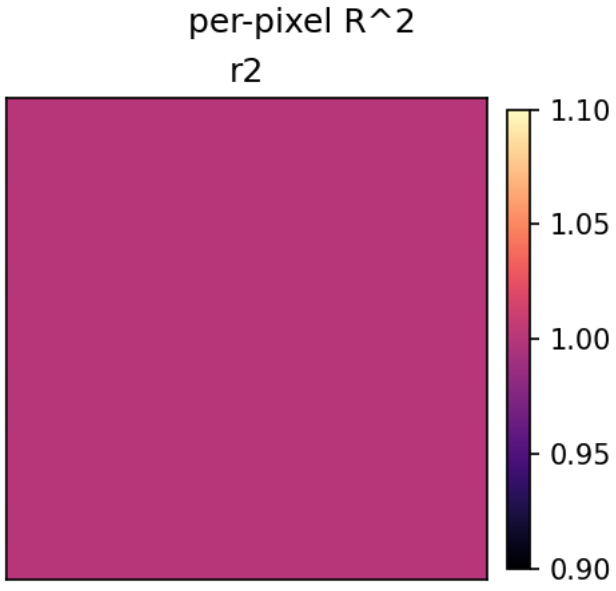
Key decomposition plots (best_rmse=fft2)











Train (cond -> coeff prediction)

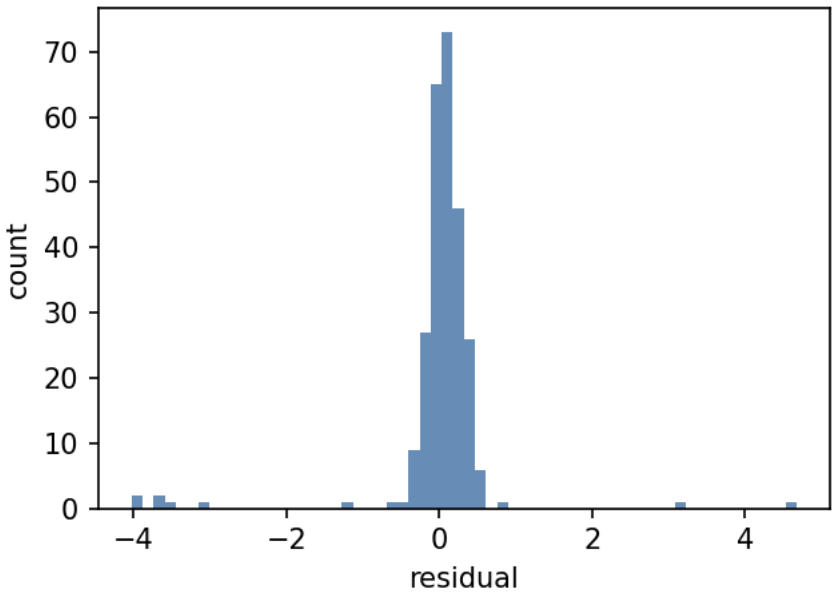
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2	fit	run

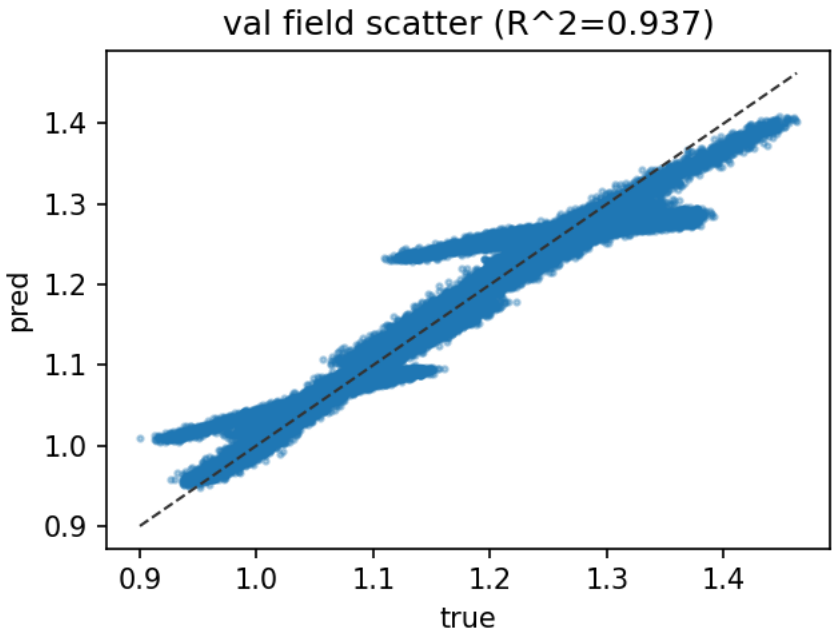
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2	fit	run
fft2	fft2	1.000000	ridge	ok	2.401e-02	0.737012	3.405e-02	0.836885	18.4ms	run
dct2	dct2	1.000000	ridge	ok	3.395e-02	0.737012	3.405e-02	0.836885	7.4ms	run
wavelet2d_k64	wavelet2d	0.907354	ridge	ok	2.464e-01	0.752962	3.115e-02	0.851131	2.1ms	run
pswf2d_tensor_bench	pswf2d_tensor	0.980260	ridge	ok	2.536e-01	0.757198	3.230e-02	0.851345	3.7ms	run
graph_fourier_bench	graph_fourier	0.967036	ridge	ok	2.540e-01	0.756454	3.210e-02	0.851247	1.7ms	run
pod	pod	1.000000	ridge	ok	3.573e-01	0.737012	3.405e-02	0.836885	5.3ms	run
pod_svd	pod_svd	1.000000	ridge	ok	3.573e-01	0.737012	3.405e-02	0.836885	3.9ms	run
autoencoder_bench	autoencoder	0.976666	ridge	ok	4.832e-01	0.784604	3.166e-02	0.856473	1.7ms	run
pod_em	pod_em	0.989747	ridge	ok	5.139e-01	0.746895	3.321e-02	0.845228	1.7ms	run
rbf_expansion_k64	rbf_expansion	0.970092	ridge	ok	5.847e-01	0.918302	3.202e-02	0.850842	4.3ms	run
dict_learning_bench	dict_learning	0.959631	ridge	ok	6.921e-01	0.068091	2.772e-02	0.836662	3.4ms	run

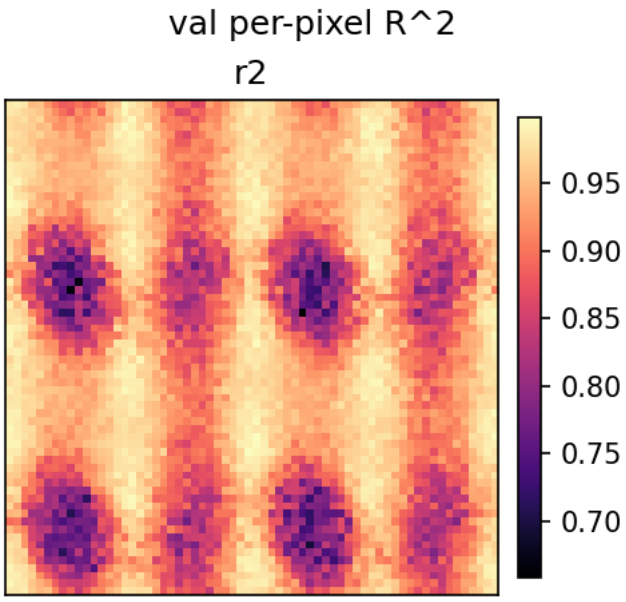
Train leaderboard (field-space)

decompose(cfg)	method	model	val_field_rmse	val_field_r2	run
dict_learning_bench	dict_learning	ridge	2.772e-02	0.836662	run
wavelet2d_k64	wavelet2d	ridge	3.115e-02	0.851131	run
autoencoder_bench	autoencoder	ridge	3.166e-02	0.856473	run
rbf_expansion_k64	rbf_expansion	ridge	3.202e-02	0.850842	run
graph_fourier_bench	graph_fourier	ridge	3.210e-02	0.851247	run

Key train plots (best_field_eval=dict_learning_bench)







disk_scalar

Problem setting

- domain: `disk` (center=[0.0, 0.0], radius=1.0)
- field: `scalar`
- grid: `64x64`, x=[-1.0, 1.0], y=[-1.0, 1.0]
- cond: `(N,4)`
- mask: geometric domain mask (outside is 0-filled; evaluation uses inside only)

Highlights (auto)

- decomposition: best full recon = `pod` (`pod`) (field_rmse=1.130e-07, field_r2=1.000000)
- decomposition: best compression proxy = `pod` (`pod`) (k_req_r2_0.95=4, r2_topk_k64=1.000000)
- decomposition: best top-energy@64 = `pod` (`pod`) (r2_topk_k64=1.000000, k_req_r2_0.95=4)
- train: best coeff-space = `pseudo_zernike` (`pseudo_zernike`) (`ridge`) (val_rmse=4.240e-03, val_r2=0.910056)
- train: best field-space = `dict_learning_bench` (`dict_learning`) (`ridge`) (val_field_rmse=2.431e-02, val_field_r2=0.845986)
- train: mismatch detected (best coeff-space != best field-space)

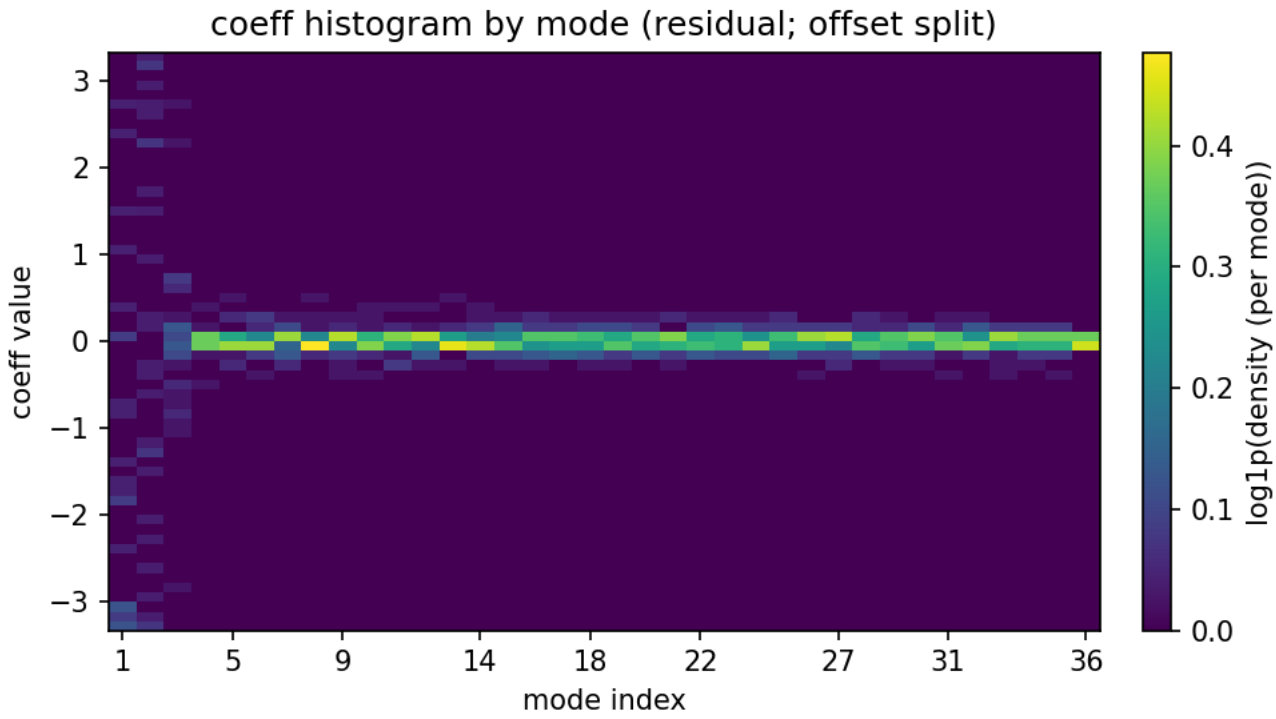
Decomposition (field reconstruction)

decompose(cfg)	method	status	rmse	r2	r2_k1	r2_k64	fit	n_req	k_req_r2_0.95	run
pod	pod	ok	1.130e-07	1.000000	0.538669	1.000000	10.8ms	2	4	run
pod_em	pod_em	ok	7.761e-03	0.989785	0.538669	0.989785	625.8ms	2	4	run

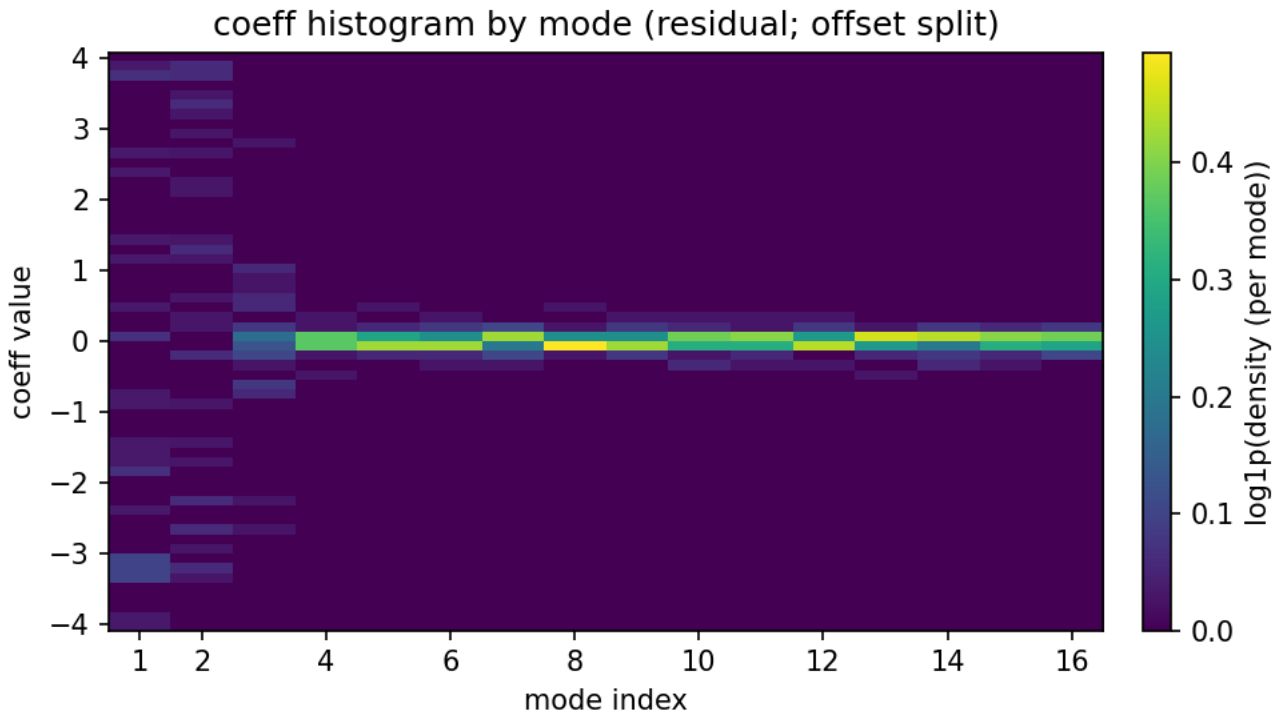
decompose(cfg)	method	status	rmse	r2	r2_k1	r2_k64	fit	n_req	k_req_r2_0.95	run
polar_fft	polar_fft	ok	9.365e-03	0.986563			2.5ms	34		run
pseudo_zernike	pseudo_zernike	ok	1.153e-02	0.979626	-0.000072	0.979626	2.6ms	2	8	run
autoencoder_bench	autoencoder	ok	1.229e-02	0.976731	0.006146	0.976731	7.420s	16	17	run
graph_fourier_bench	graph_fourier	ok	1.264e-02	0.975489	-0.000000	0.975489	136.3ms	6	8	run
fourier_jacobi	fourier_jacobi	ok	1.269e-02	0.975130	-0.000082	0.975130	2.9ms	6	8	run
disk_slepian_bench	disk_slepian	ok	1.418e-02	0.968803	0.364664	0.968803	53.0ms	33	32	run
fourier_bessel_neumann	fourier_bessel	ok	1.434e-02	0.968277	-0.000022	0.968277	2.2ms	5	8	run
zernike	zernike	ok	1.438e-02	0.967842	-0.000112	0.967842	2.4ms	3	8	run
dict_learning_bench	dict_learning	ok	1.844e-02	0.946666	-0.035671	0.946666	591.2ms	29		run

Coefficient histograms by mode (per decomposer)

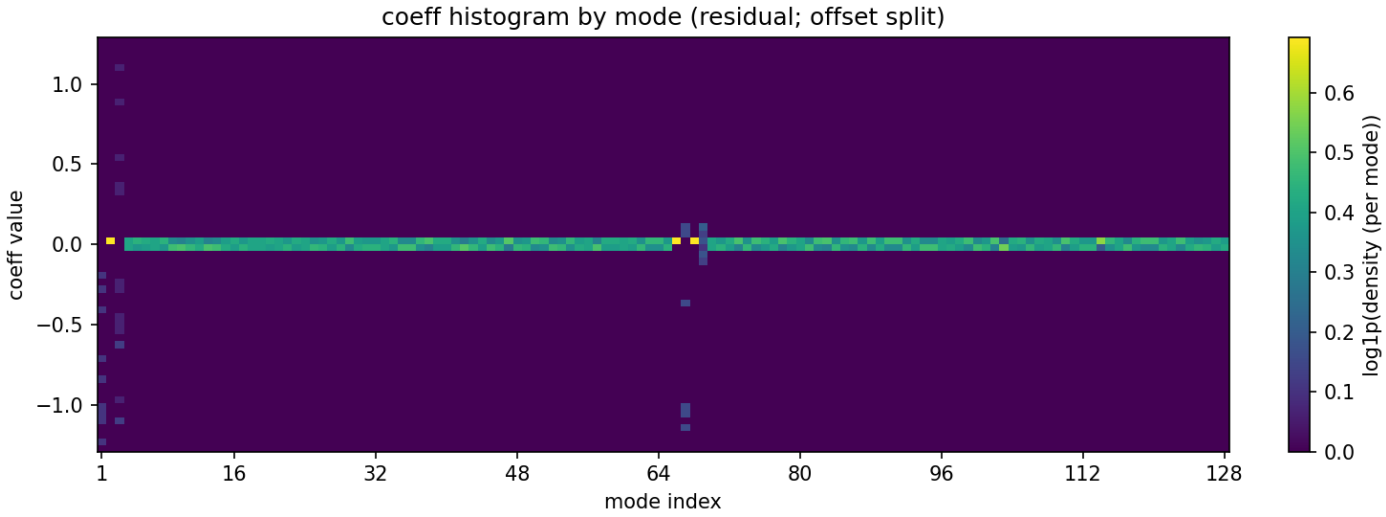
pod (pod)



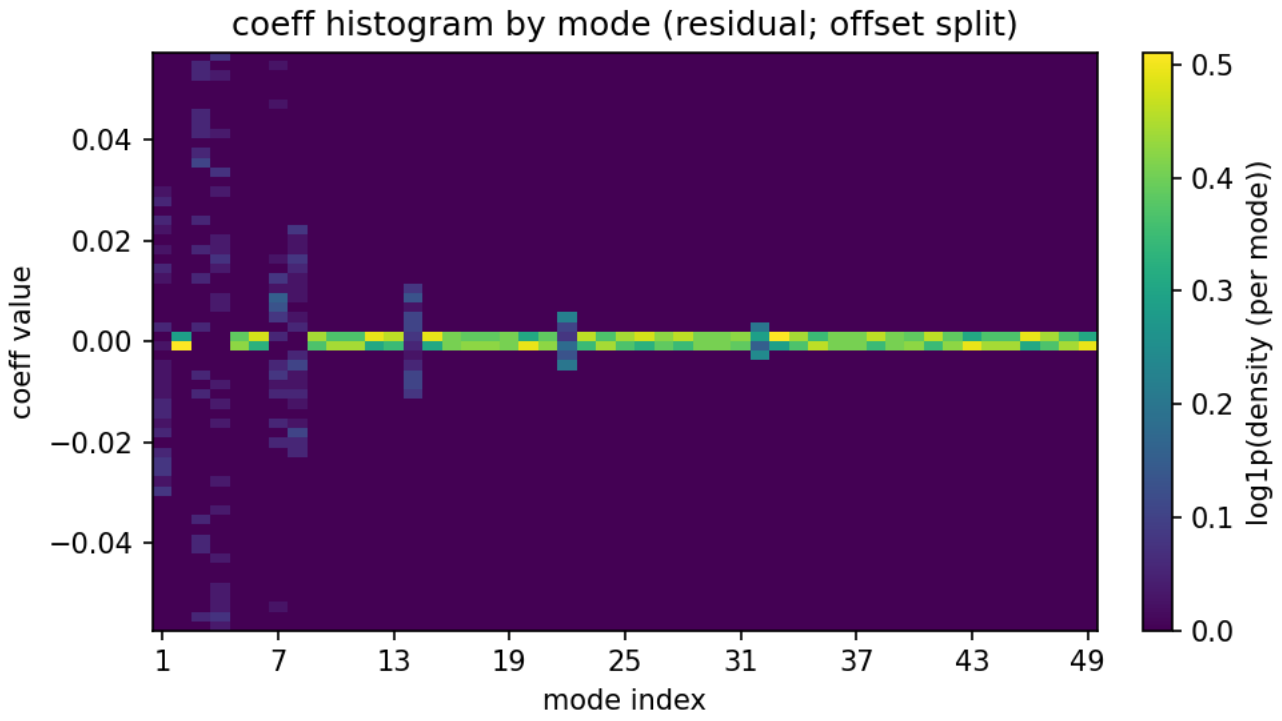
pod_em (pod_em)



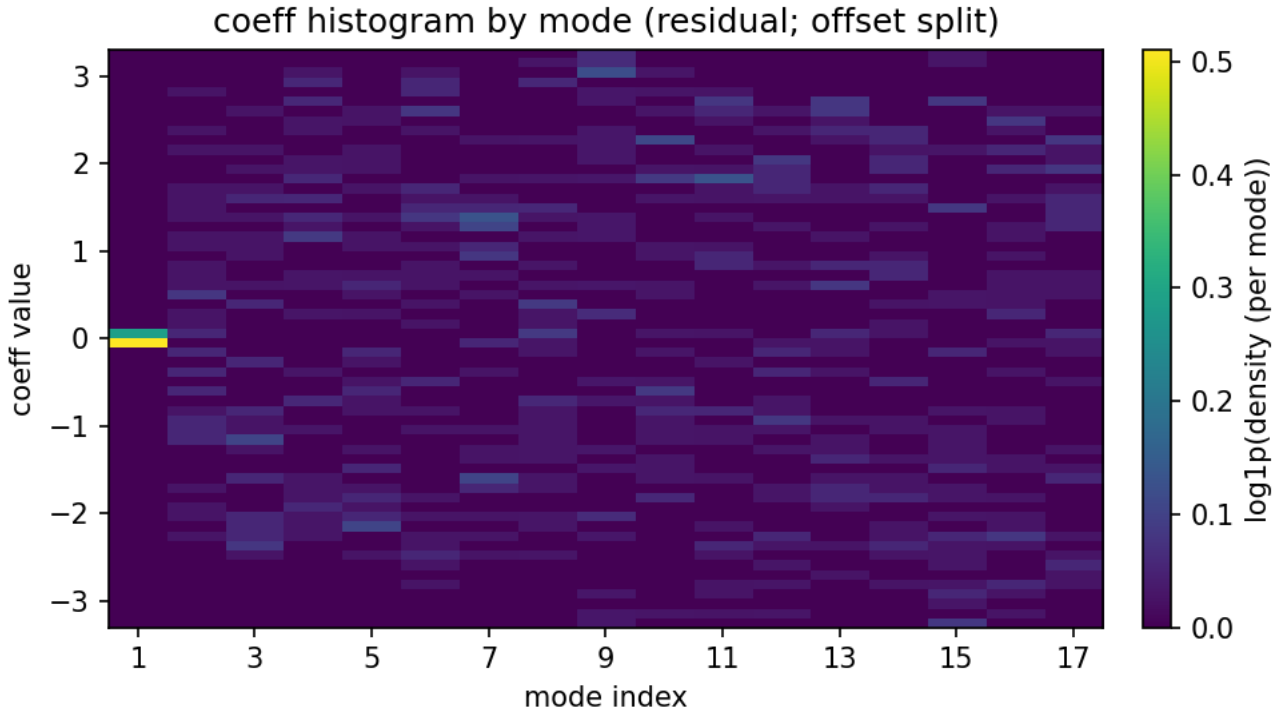
polar_fft (polar_fft)



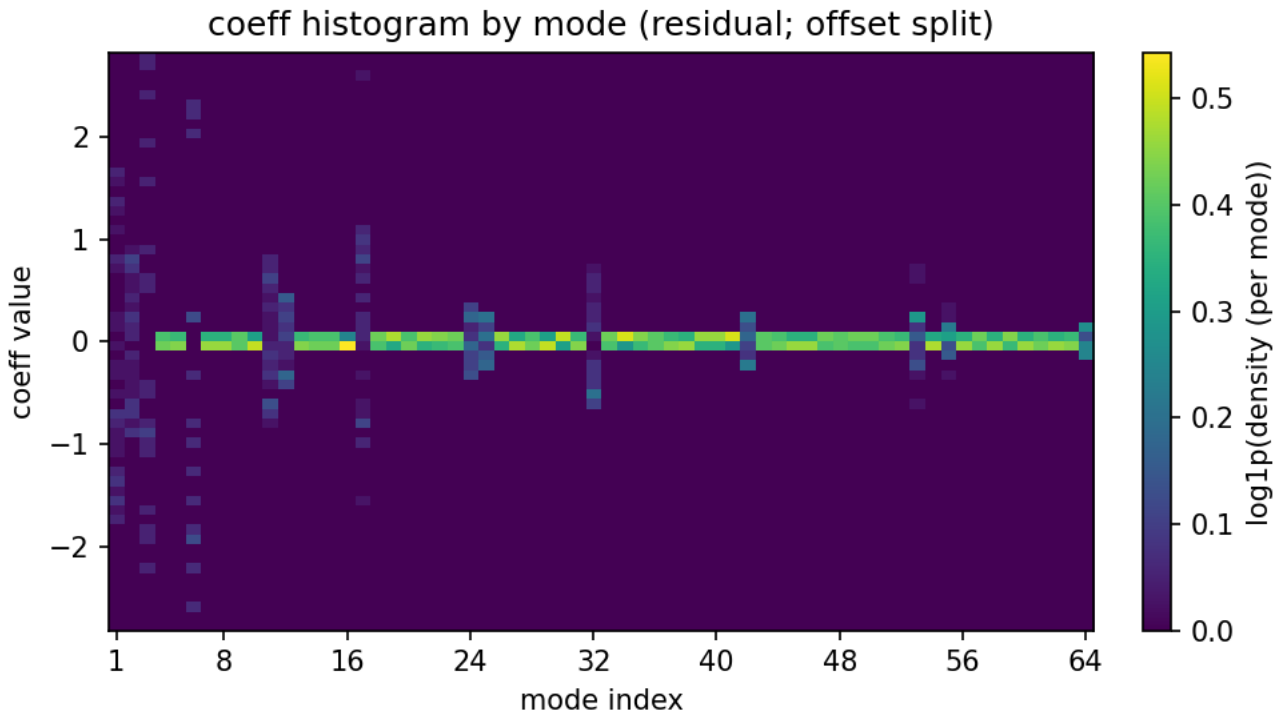
pseudo_zernike (pseudo_zernike)



autoencoder_bench (autoencoder)

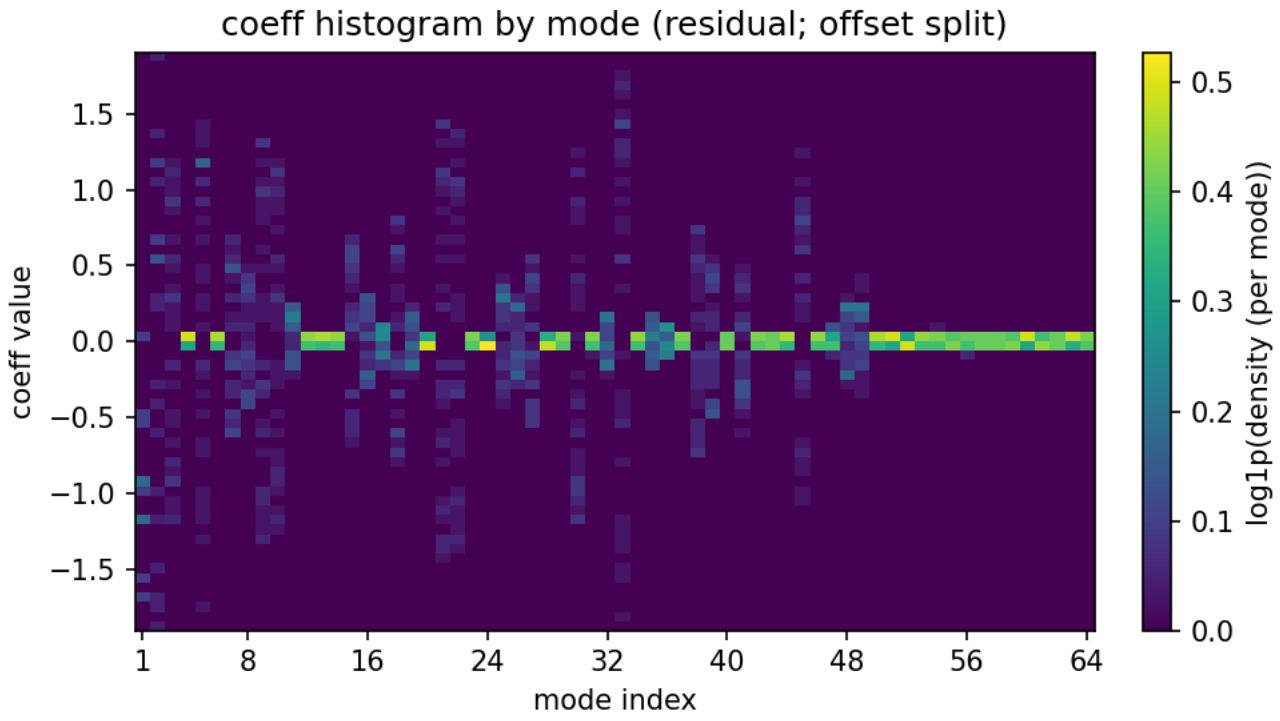


graph_fourier_bench (graph_fourier)

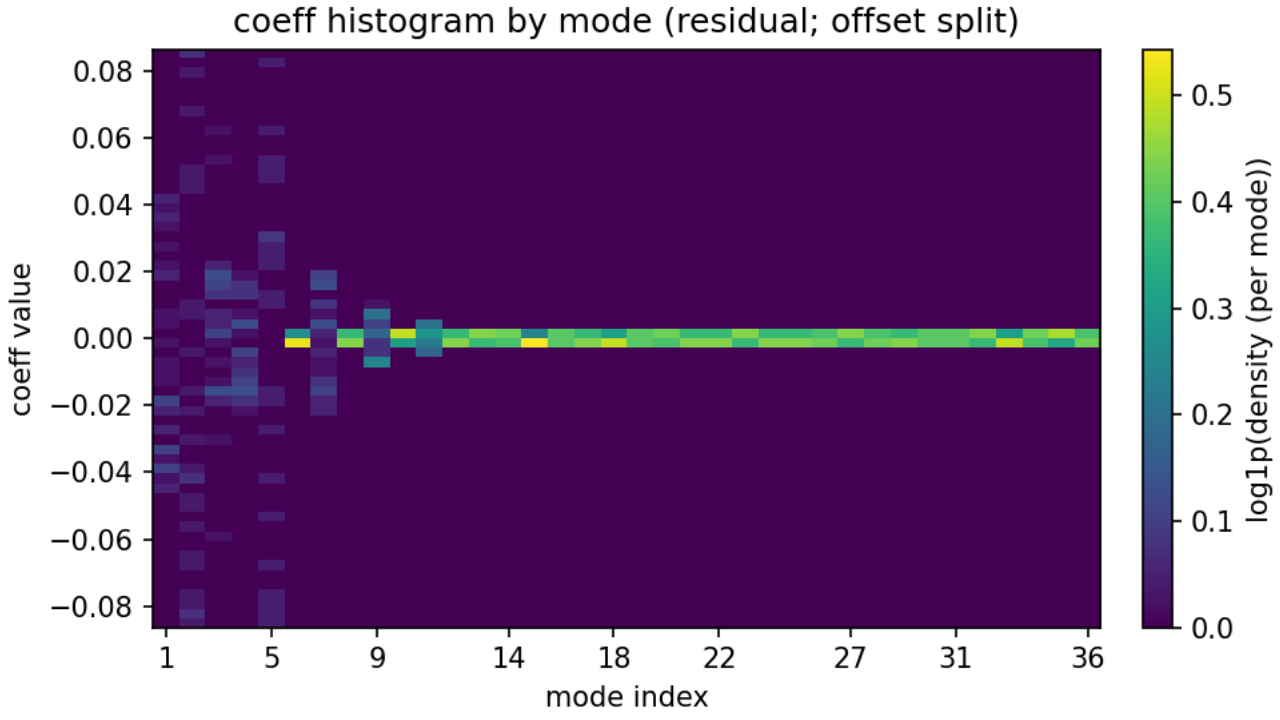


fourier_jacobi (fourier_jacobi)

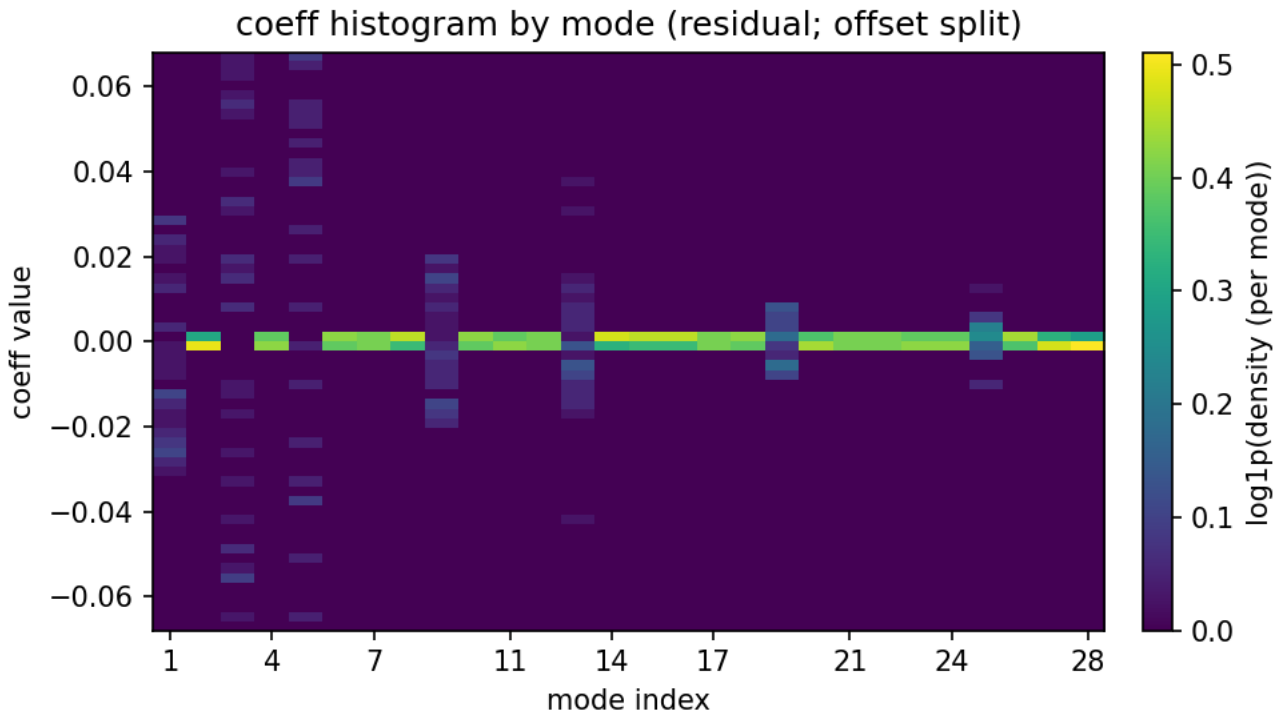
disk_slepian_bench (disk_slepian)



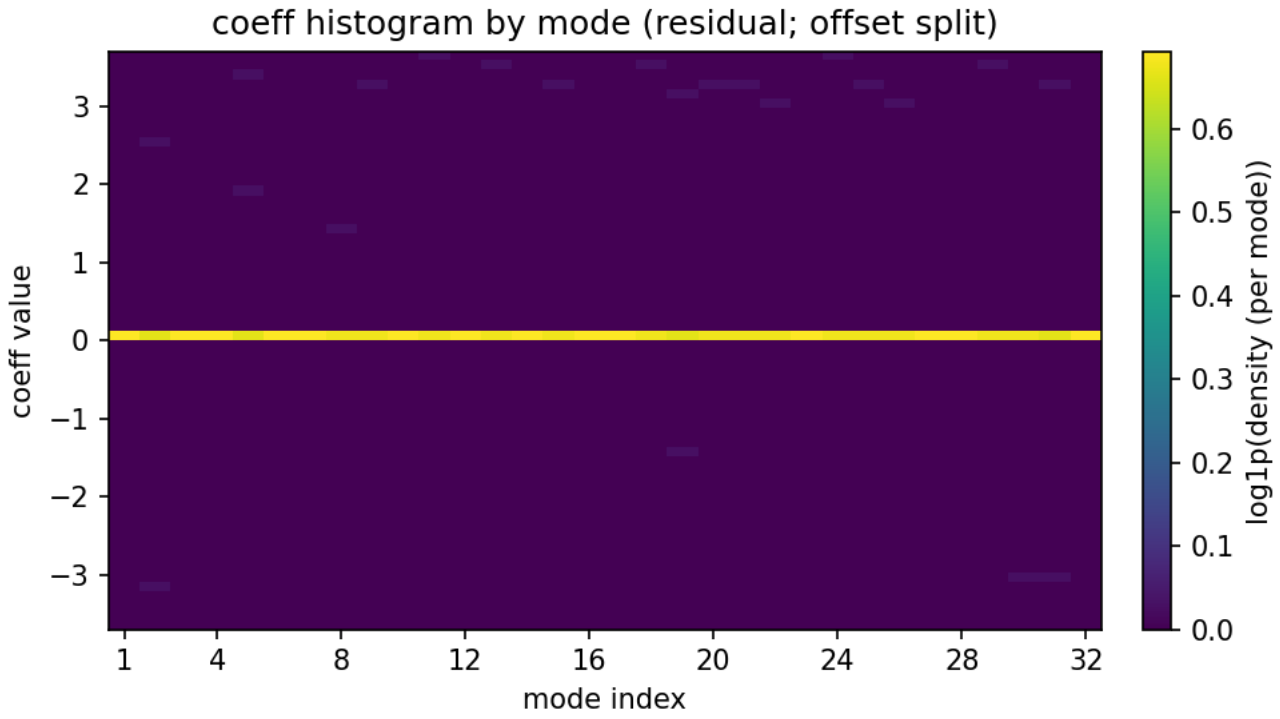
fourier_bessel_neumann (fourier_bessel)



zernike (zernike)



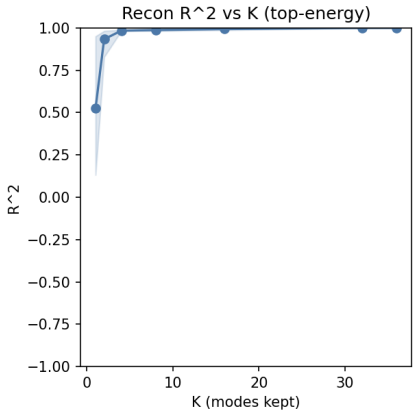
dict_learning_bench (dict_learning)

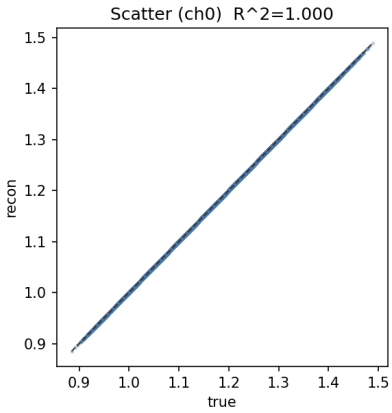


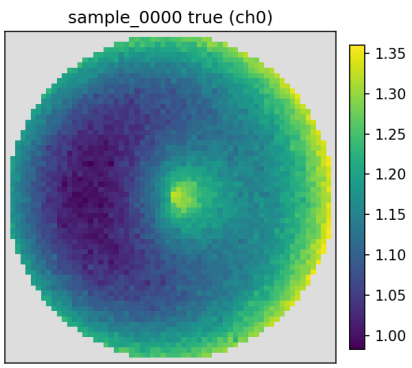
Compression leaderboard (Top-energy @K)

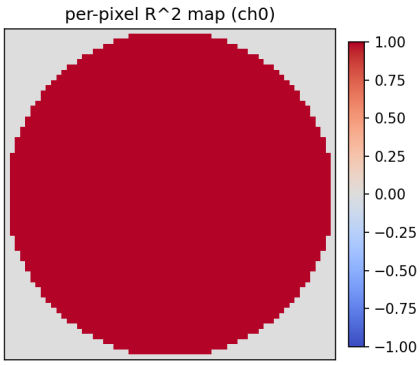
decompose(cfg)	method	r2_topk_k64	r2_topk_k16	k_req_r2_0.95	n_req
pod	pod	1.000000	0.989785	4	2
pod_em	pod_em	0.989785	0.989785	4	2
pseudo_zernike	pseudo_zernike	0.979626	0.979437	8	2
autoencoder_bench	autoencoder	0.976731	0.911240	17	16
graph_fourier_bench	graph_fourier	0.975489	0.975068	8	6

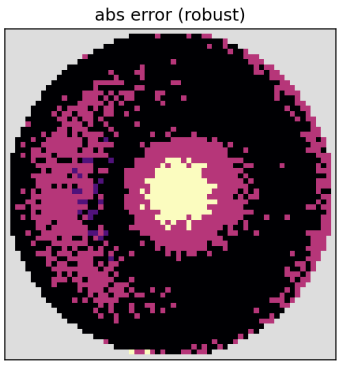
Key decomposition plots (best_rmse=pod)

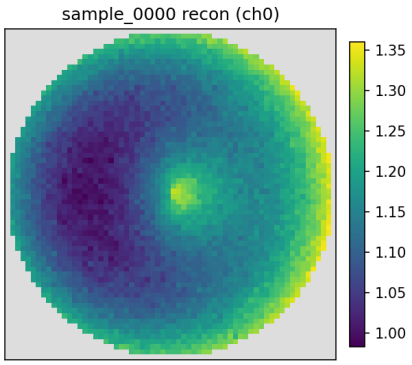

Recon R^2 vs K (top-energy)
R^2
K (modes kept)

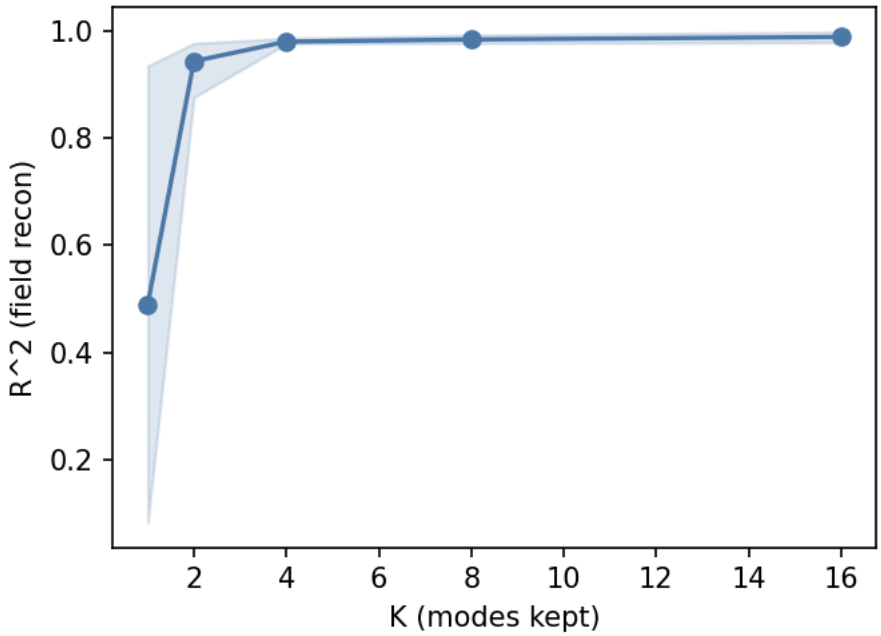

Scatter (ch0) R^2=1.000
recon
true

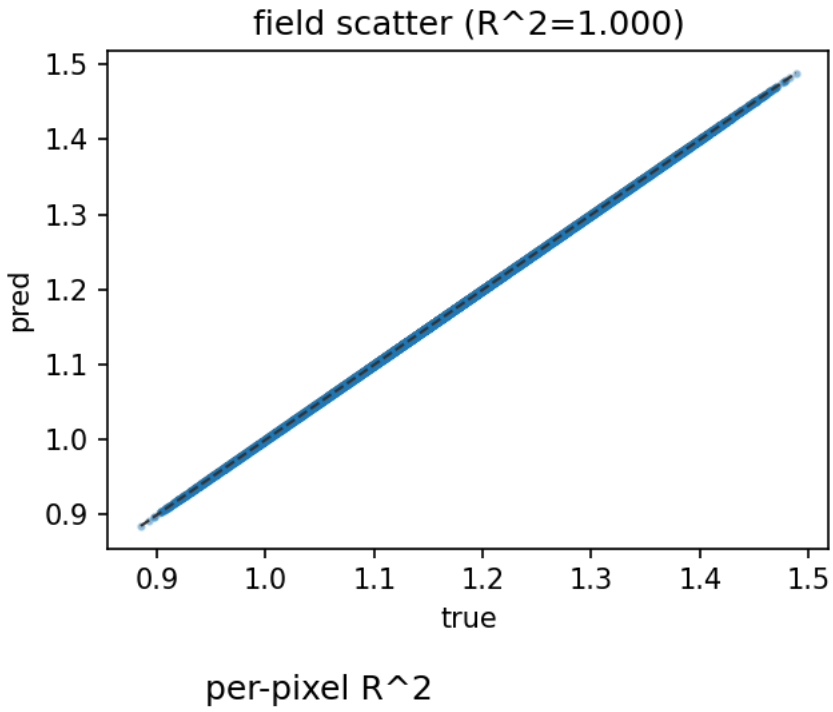

sample_0000 true (ch0)

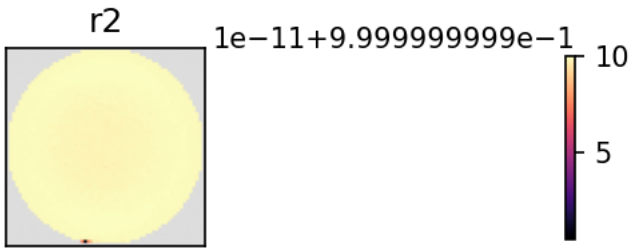

per-pixel R^2 map (ch0)


abs error (robust)
1e−7


sample_0000 recon (ch0)


R^2 (field recon)
K (modes kept)





Train (cond -> coeff prediction)

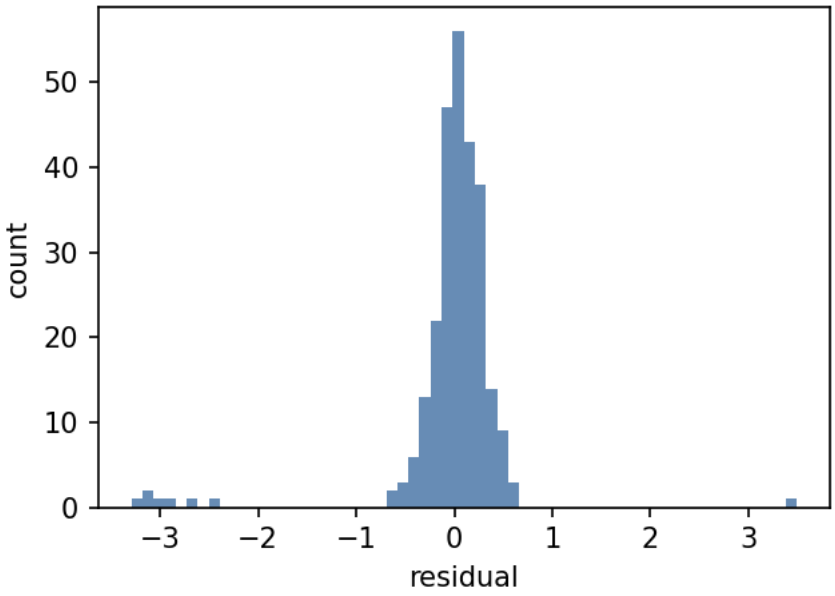
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2	fit	run
pseudo_zernike	pseudo_zernike	0.979626	ridge	ok	4.240e-03	0.910056	2.967e-02	0.858169	1.6ms	run
zernike	zernike	0.967842	ridge	ok	5.414e-03	0.913078	2.881e-02	0.856603	1.7ms	run
fourier_jacobi	fourier_jacobi	0.975130	ridge	ok	6.611e-03	0.875332	2.927e-02	0.857242	1.8ms	run
fourier_bessel_neumann	fourier_bessel	0.968277	ridge	ok	7.113e-03	0.872845	2.929e-02	0.857099	1.7ms	run
polar_fft	polar_fft	0.986563	ridge	ok	2.716e-02	0.914507	2.996e-02	0.854888	7.4ms	run
disk_slepian_bench	disk_slepian	0.968803	ridge	ok	2.026e-01	0.846915	2.930e-02	0.857758	2.9ms	run
graph_fourier_bench	graph_fourier	0.975489	ridge	ok	2.049e-01	0.845193	2.963e-02	0.858125	1.8ms	run
pod	pod	1.000000	ridge	ok	2.880e-01	0.829865	3.144e-02	0.844380	1.6ms	run
pod_em	pod_em	0.989785	ridge	ok	4.109e-01	0.839204	3.040e-02	0.853350	1.7ms	run

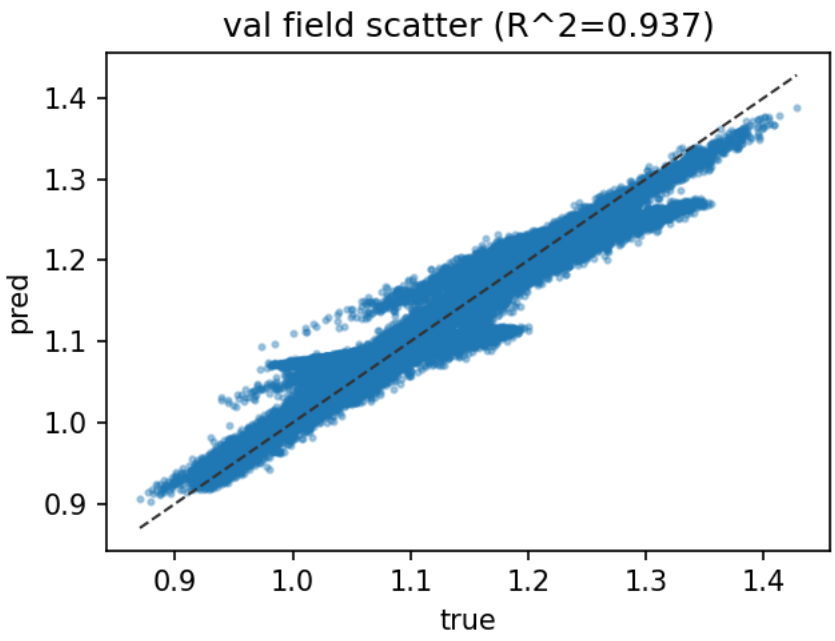
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2	fit	run
dict_learning_bench	dict_learning	0.946666	ridge	ok	5.703e-01	0.147275	2.431e-02	0.845986	1.8ms	run
autoencoder_bench	autoencoder	0.976731	ridge	ok	6.626e-01	0.829666	3.035e-02	0.860531	2.1ms	run

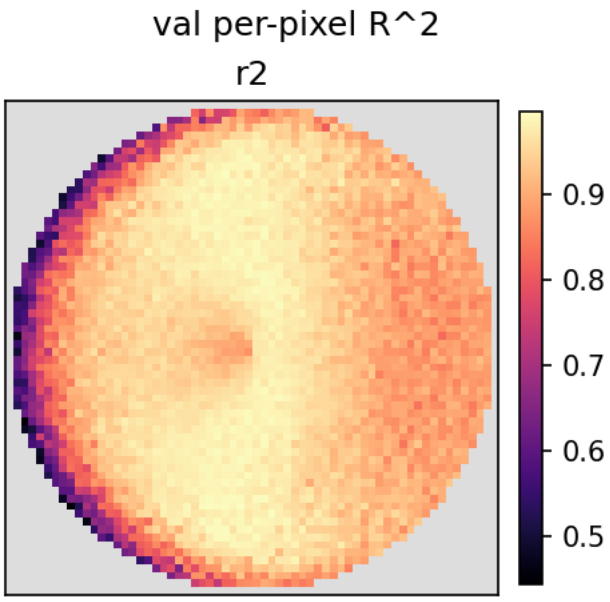
Train leaderboard (field-space)

decompose(cfg)	method	model	val_field_rmse	val_field_r2	run
dict_learning_bench	dict_learning	ridge	2.431e-02	0.845986	run
zernike	zernike	ridge	2.881e-02	0.856603	run
fourier_jacobi	fourier_jacobi	ridge	2.927e-02	0.857242	run
fourier_bessel_neumann	fourier_bessel	ridge	2.929e-02	0.857099	run
disk_slepian_bench	disk_slepian	ridge	2.930e-02	0.857758	run

Key train plots (best_field_eval=dict_learning_bench)







annulus_scalar

Problem setting

- domain: `annulus` (center=[0.0, 0.0], r_inner=0.35, r_outer=1.0)
- field: `scalar`
- grid: `64x64`, x=[-1.0, 1.0], y=[-1.0, 1.0]
- cond: `(N,4)`
- mask: geometric domain mask (outside is 0-filled; evaluation uses inside only)

Highlights (auto)

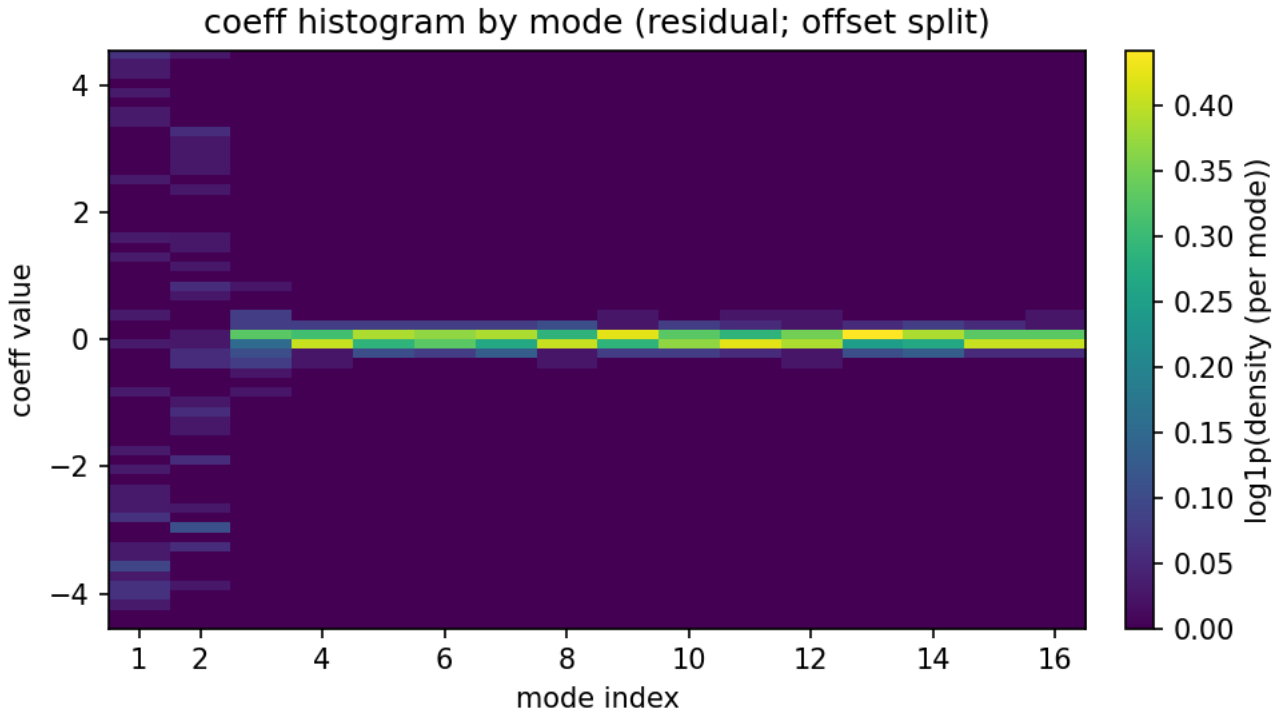
- decomposition: best full recon = `pod_em` (`pod_em`) (field_rmse=7.847e-03, field_r2=0.989865)
- decomposition: best compression proxy = `pod_em` (`pod_em`) (k_req_r2_0.95=2, r2_topk_k64=0.989865)
- decomposition: best top-energy@64 = `pod_em` (`pod_em`) (r2_topk_k64=0.989865, k_req_r2_0.95=2)
- train: best coeff-space = `annular_zernike` (`annular_zernike`) (`ridge`) (val_rmse=5.552e-03, val_r2=0.945274)
- train: best field-space = `annular_zernike` (`annular_zernike`) (`ridge`) (val_field_rmse=3.134e-02, val_field_r2=0.836965)

Decomposition (field reconstruction)

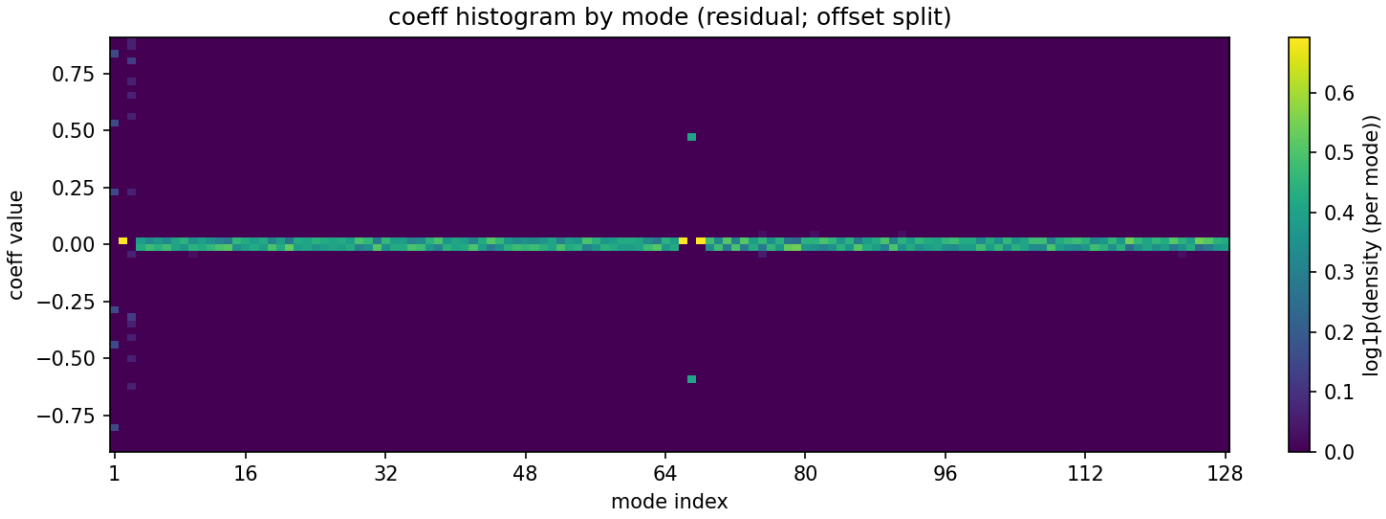
decompose(cfg)	method	status	rmse	r2	r2_k1	r2_k64	fit	n_req	k_req_r2_0.95	run
`pod_em`	`pod_em`	ok	7.847e-03	0.989865	0.656117	0.989865	272.8ms	2	2	run
`polar_fft`	`polar_fft`	ok	9.357e-03	0.986988			2.4ms	34		run
`rbf_expansion_k64`	`rbf_expansion`	ok	1.181e-02	0.979281	-3.447748	0.979281	2.1ms	60	64	run
`graph_fourier_bench`	`graph_fourier`	ok	1.222e-02	0.977856	-0.000000	0.977856	123.0ms	8	4	run
`annular_zernike`	`annular_zernike`	ok	1.683e-02	0.957231	-0.064201	0.957231	23.3ms	3	8	run

Coefficient histograms by mode (per decomposer)

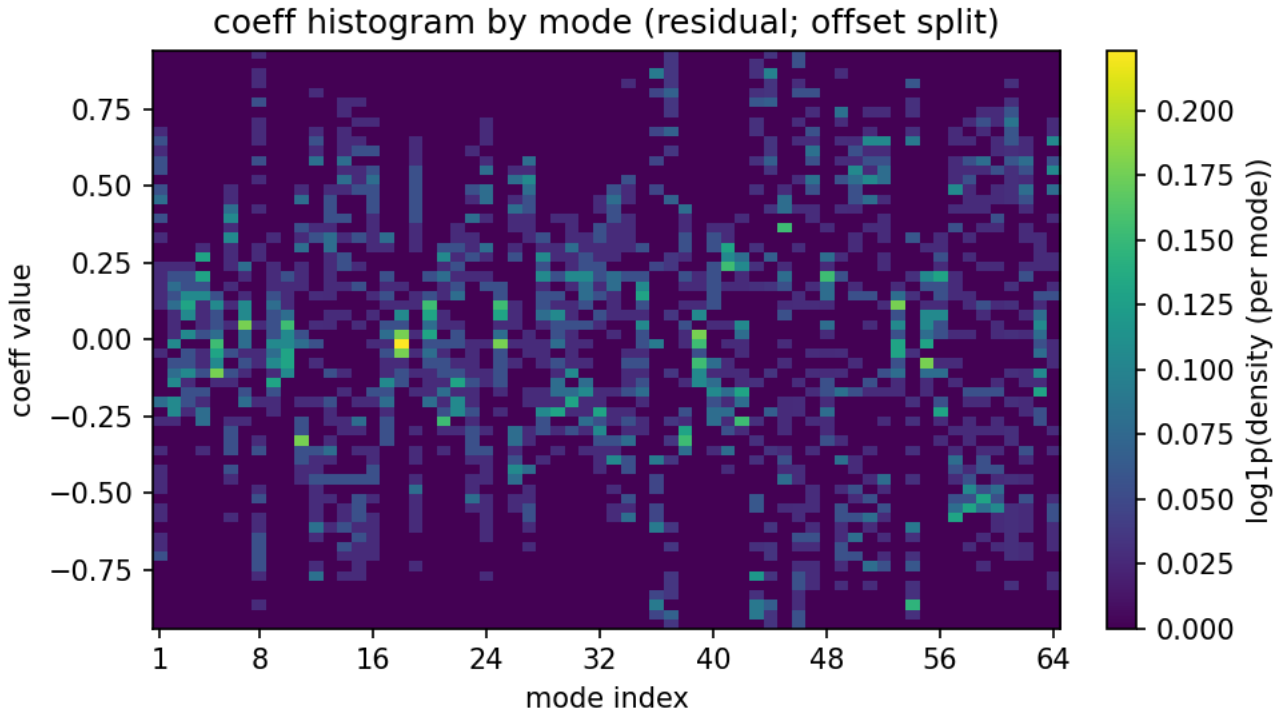
pod_em (pod_em)



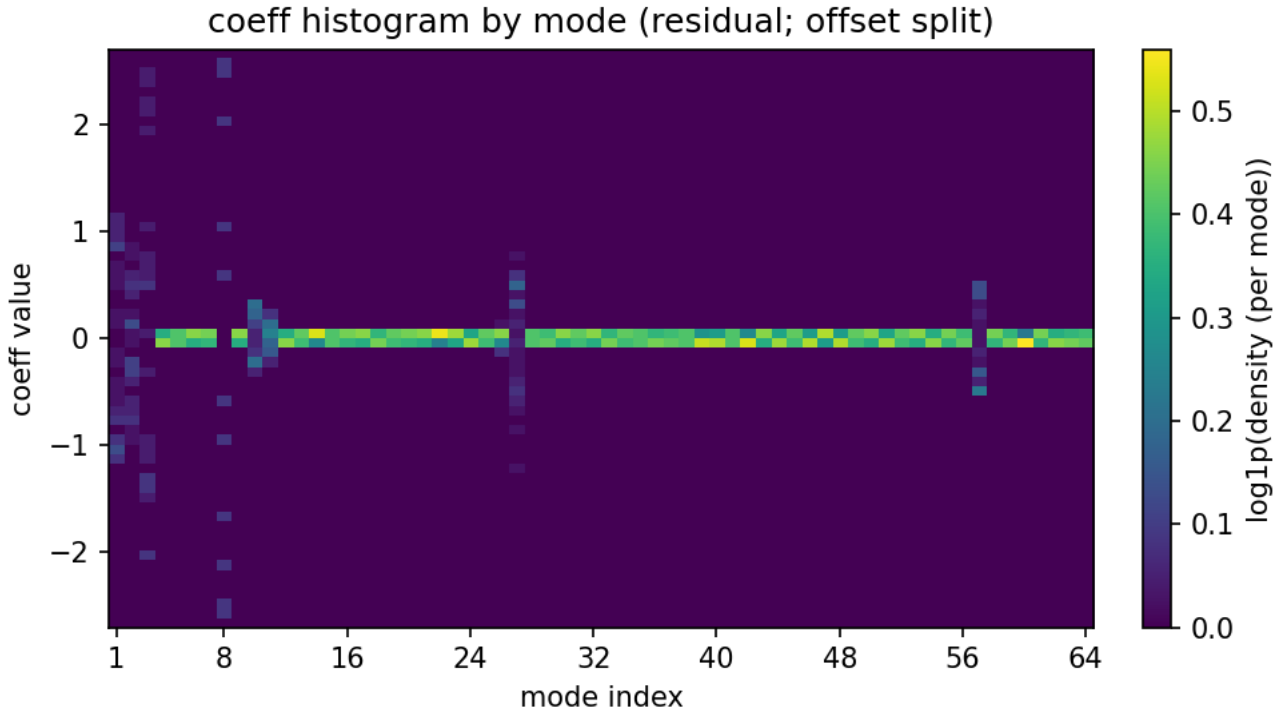
polar_fft (polar_fft)



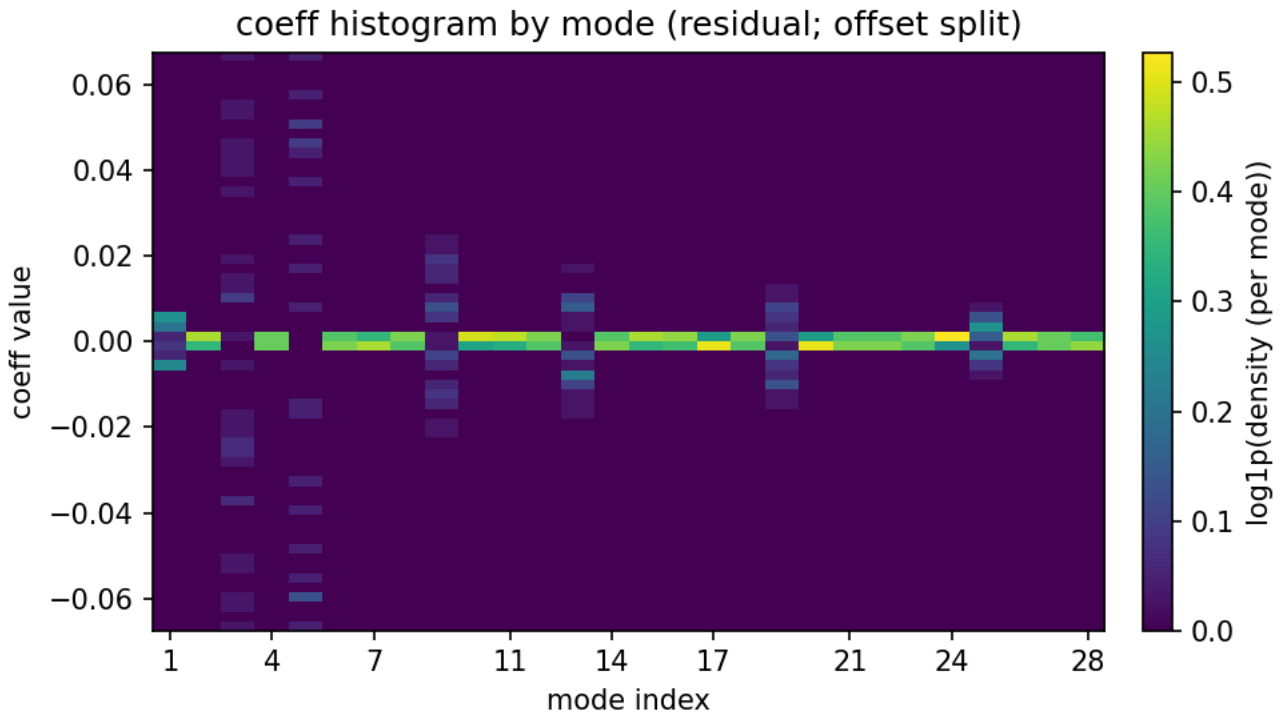
rbf_expansion_k64 (rbf_expansion)



graph_fourier_bench (graph_fourier)



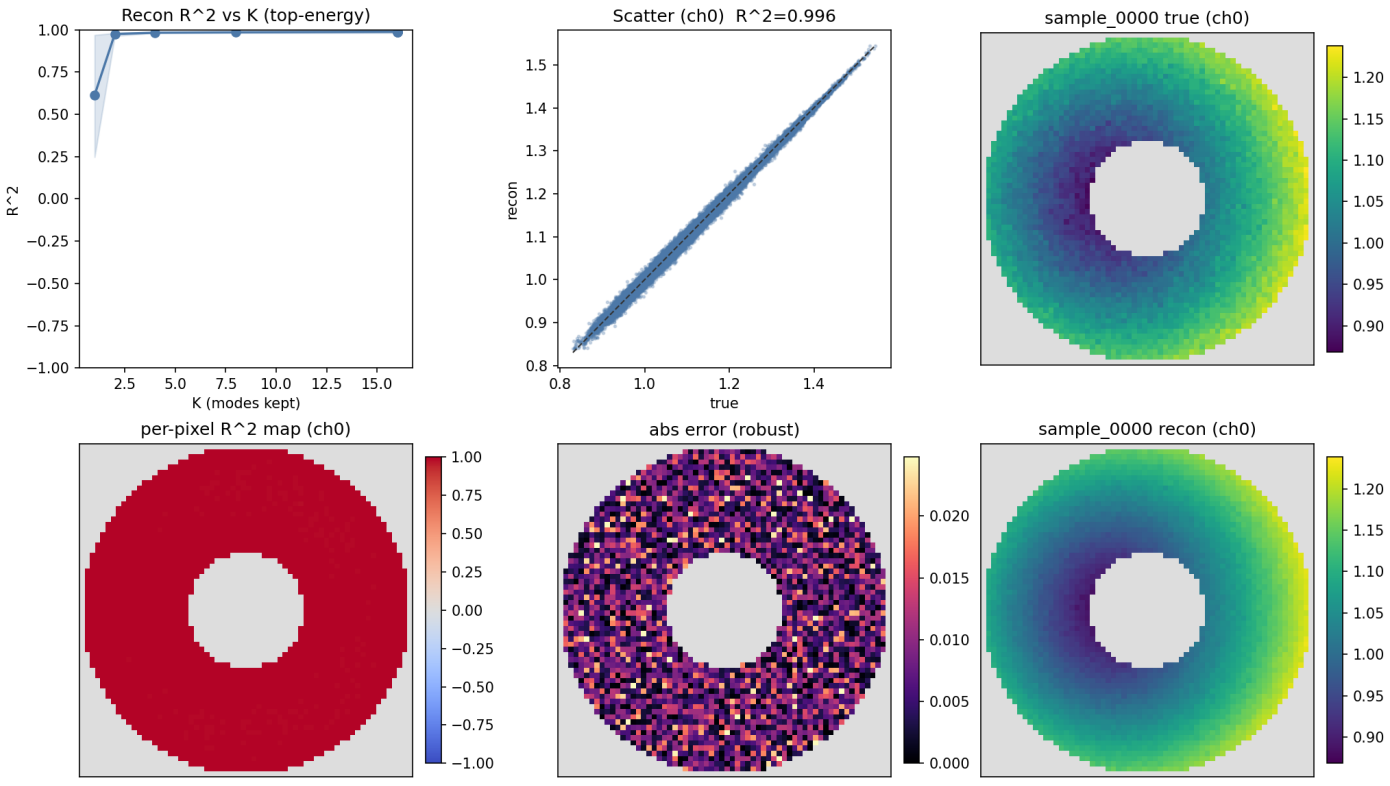
annular_zernike (annular_zernike)

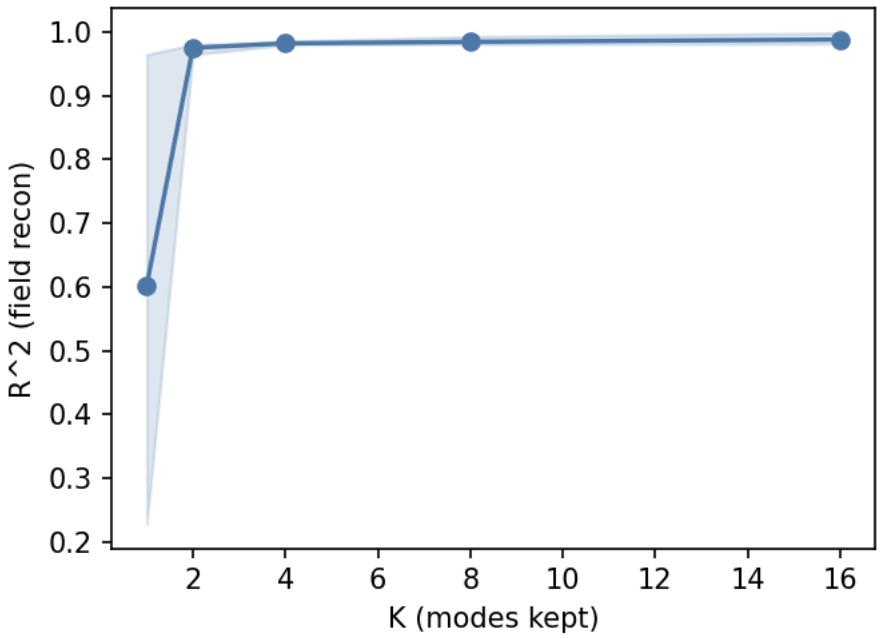


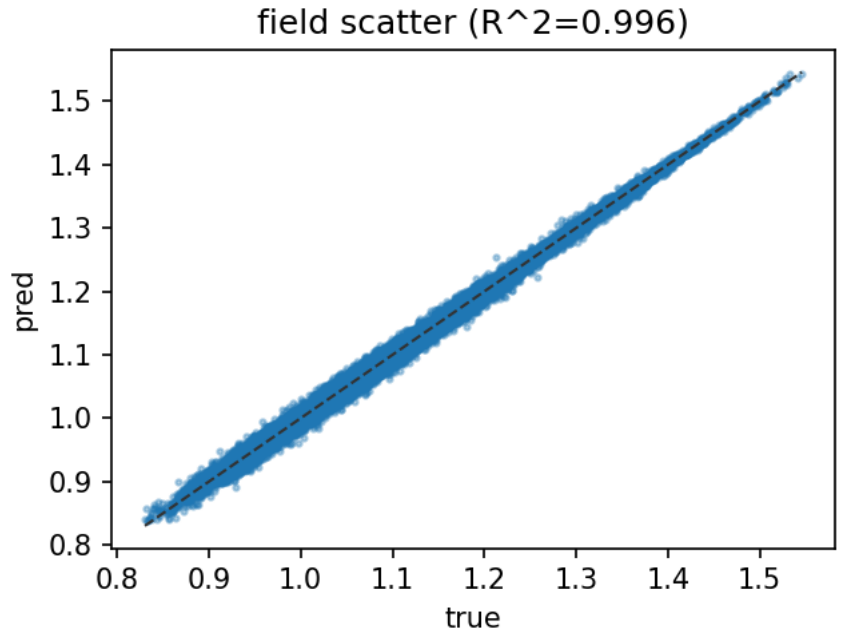
Compression leaderboard (Top-energy @K)

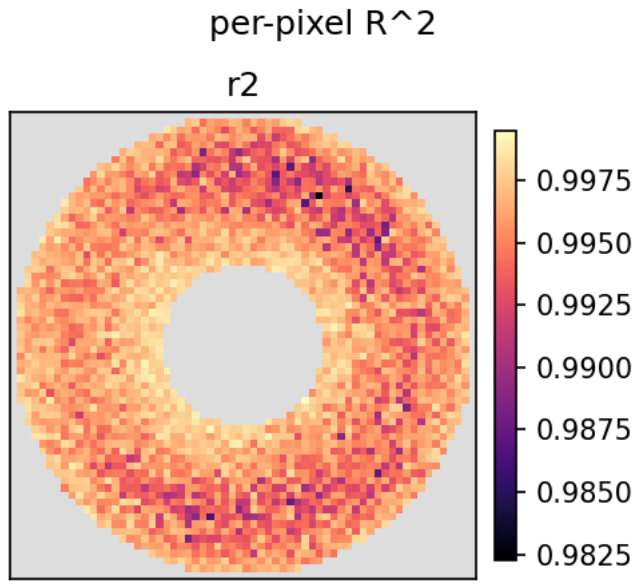
decompose(cfg)	method	r2_topk_k64	r2_topk_k16	k_req_r2_0.95	n_req
pod_em	pod_em	0.989865	0.989865	2	2
rbf_expansion_k64	rbf_expansion	0.979281	-36.156671	64	60
graph_fourier_bench	graph_fourier	0.977856	0.977513	4	8
annular_zernike	annular_zernike	0.957231	0.957151	8	3

Key decomposition plots (best_rmse=pod_em)









Train (cond -> coeff prediction)

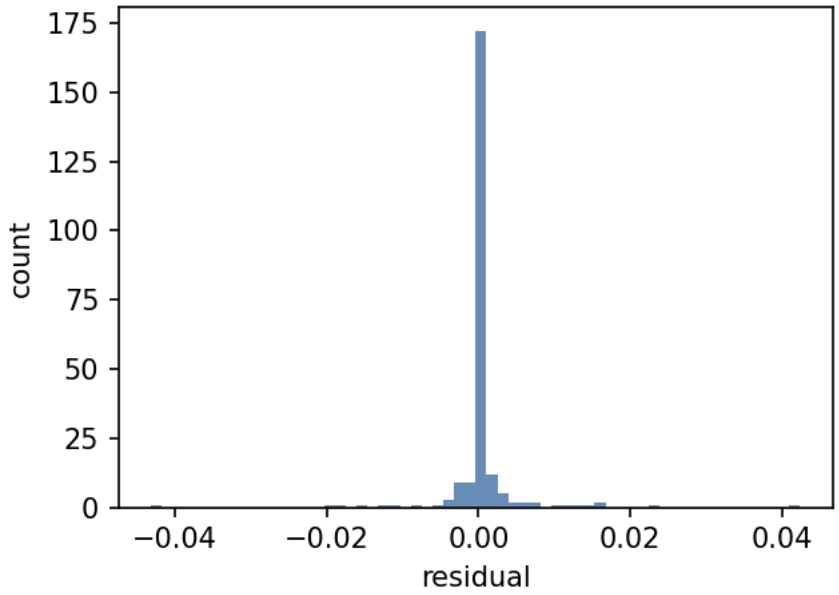
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2	fit	run

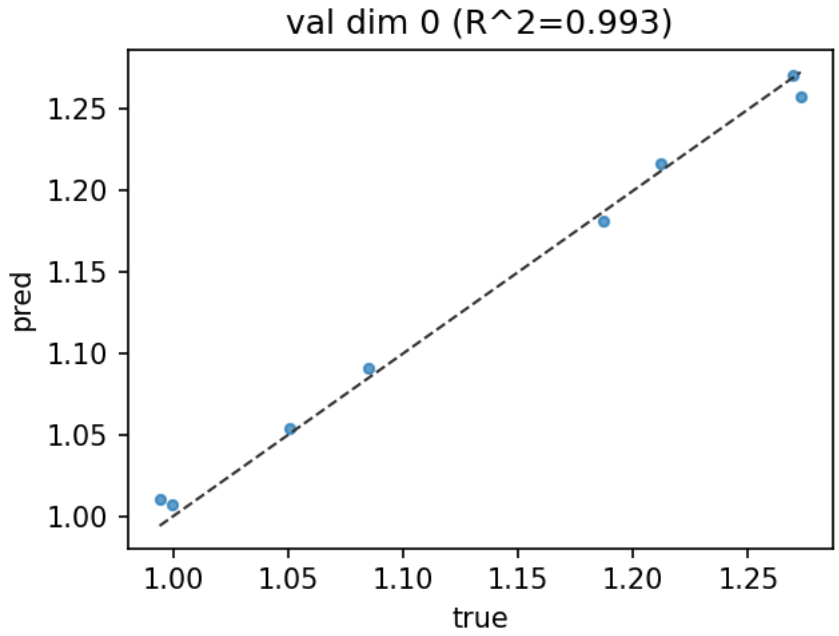
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2	fit	run
annular_zernike	annular_zernike	0.957231	ridge	ok	5.552e-03	0.945274	3.134e-02	0.836965	2.3ms	run
polar_fft	polar_fft	0.986988	ridge	ok	3.533e-02	0.817919	3.209e-02	0.833762	13.7ms	run
rbf_expansion_k64	rbf_expansion	0.979281	ridge	ok	1.746e-01	0.804485	3.168e-02	0.837207	2.0ms	run
graph_fourier_bench	graph_fourier	0.977856	ridge	ok	2.092e-01	0.823287	3.174e-02	0.837035	3.5ms	run
pod_em	pod_em	0.989865	ridge	ok	4.179e-01	0.817504	3.246e-02	0.832306	1.8ms	run

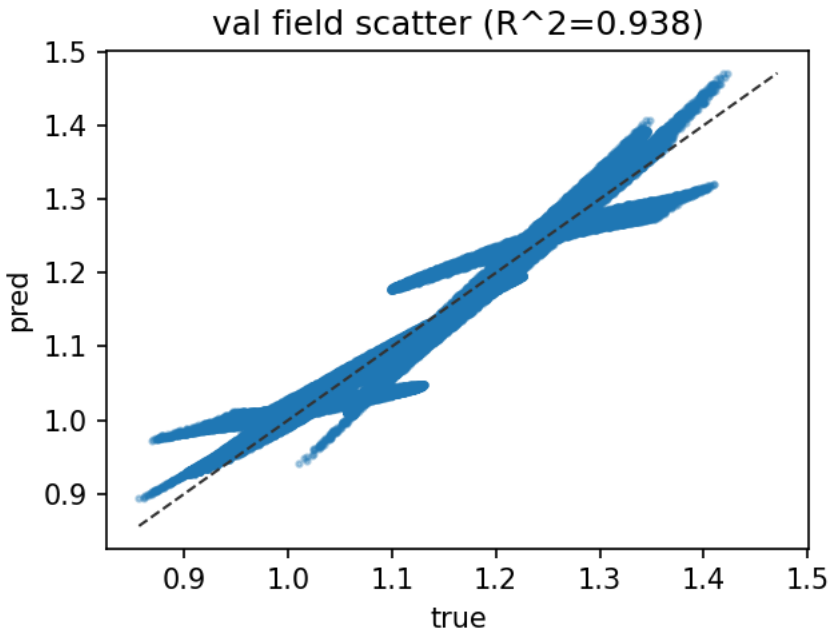
Train leaderboard (field-space)

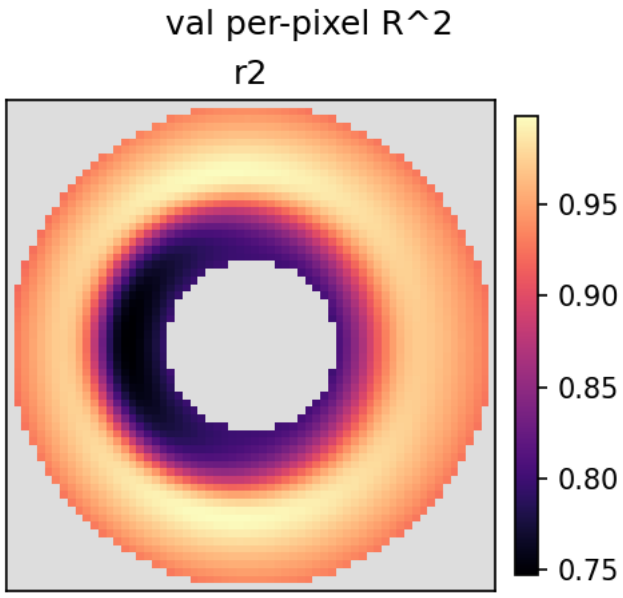
decompose(cfg)	method	model	val_field_rmse	val_field_r2	run
annular_zernike	annular_zernike	ridge	3.134e-02	0.836965	run
rbf_expansion_k64	rbf_expansion	ridge	3.168e-02	0.837207	run
graph_fourier_bench	graph_fourier	ridge	3.174e-02	0.837035	run
polar_fft	polar_fft	ridge	3.209e-02	0.833762	run
pod_em	pod_em	ridge	3.246e-02	0.832306	run

Key train plots (best_field_eval=annular_zernike)









arbitrary_mask_scalar

Problem setting

- domain: `arbitrary_mask` (mask=domain_mask.npy)
- field: `scalar`
- grid: `64x64`, x=[-1.0, 1.0], y=[-1.0, 1.0]
- cond: `(N,4)`
- mask: fixed irregular mask (`domain_mask.npy`; evaluation uses mask==true only)

Highlights (auto)

- decomposition: best full recon = `pod` (`pod`) (field_rmse=9.150e-08, field_r2=1.000000)
- decomposition: best compression proxy = `pod` (`pod`) (k_req_r2_0.95=4, r2_topk_k64=1.000000)
- decomposition: best top-energy@64 = `pod` (`pod`) (r2_topk_k64=1.000000, k_req_r2_0.95=4)
- train: best coeff-space = `wavelet2d_k64` (`wavelet2d`) (`ridge`) (val_rmse=1.032e-01, val_r2=0.944795)
- train: best field-space = `dict_learning_bench` (`dict_learning`) (`ridge`) (val_field_rmse=1.548e-02, val_field_r2=0.932046)
- train: mismatch detected (best coeff-space != best field-space)

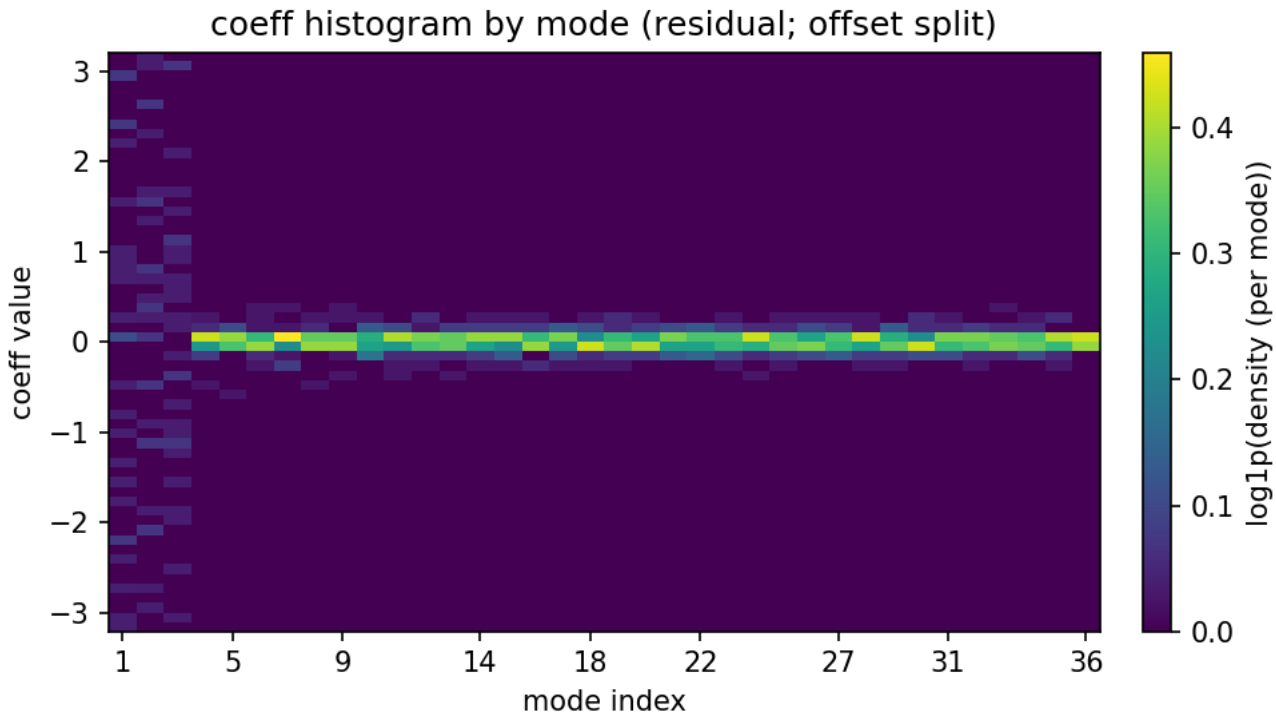
Decomposition (field reconstruction)

decompose(cfg)	method	status	rmse	r2	r2_k1	r2_k64	fit	n_req	k_req_r2_0.95
pod	pod	ok	9.150e-08	1.000000	0.390700	1.000000	10.6ms	3	4
pod_em	pod_em	ok	7.686e-03	0.989886	0.390700	0.989886	238.7ms	3	4

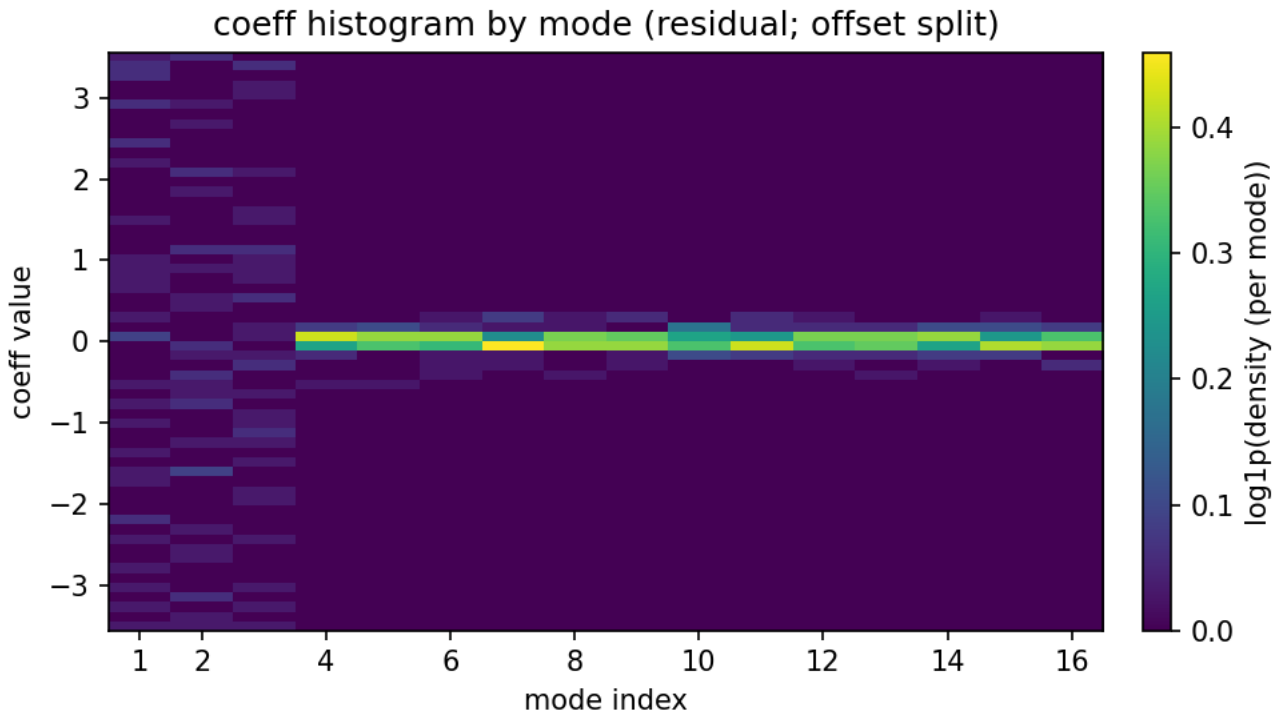
decompose(cfg)	method	status	rmse	r2	r2_k1	r2_k64	fit	n_req	k_req_r2_0.95
autoencoder_bench	autoencoder	ok	1.216e-02	0.976808	0.002144	0.976808	7.476s	16	16
graph_fourier_bench	graph_fourier	ok	1.229e-02	0.976361	-0.000000	0.976361	105.8ms	24	32
gappy_graph_fourier_bench	gappy_graph_fourier	ok	1.229e-02	0.976360	-0.000000	0.976360	105.9ms	24	32
rbf_expansion_k64	rbf_expansion	ok	1.434e-02	0.967712	-132.270793	0.967712	2.4ms	59	64
dict_learning_bench	dict_learning	ok	1.988e-02	0.936680	0.026165	0.936680	234.9ms	29	
wavelet2d_k64	wavelet2d	ok	3.085e-02	0.850876	0.000033	0.850876	2.4ms	47	

Coefficient histograms by mode (per decomposer)

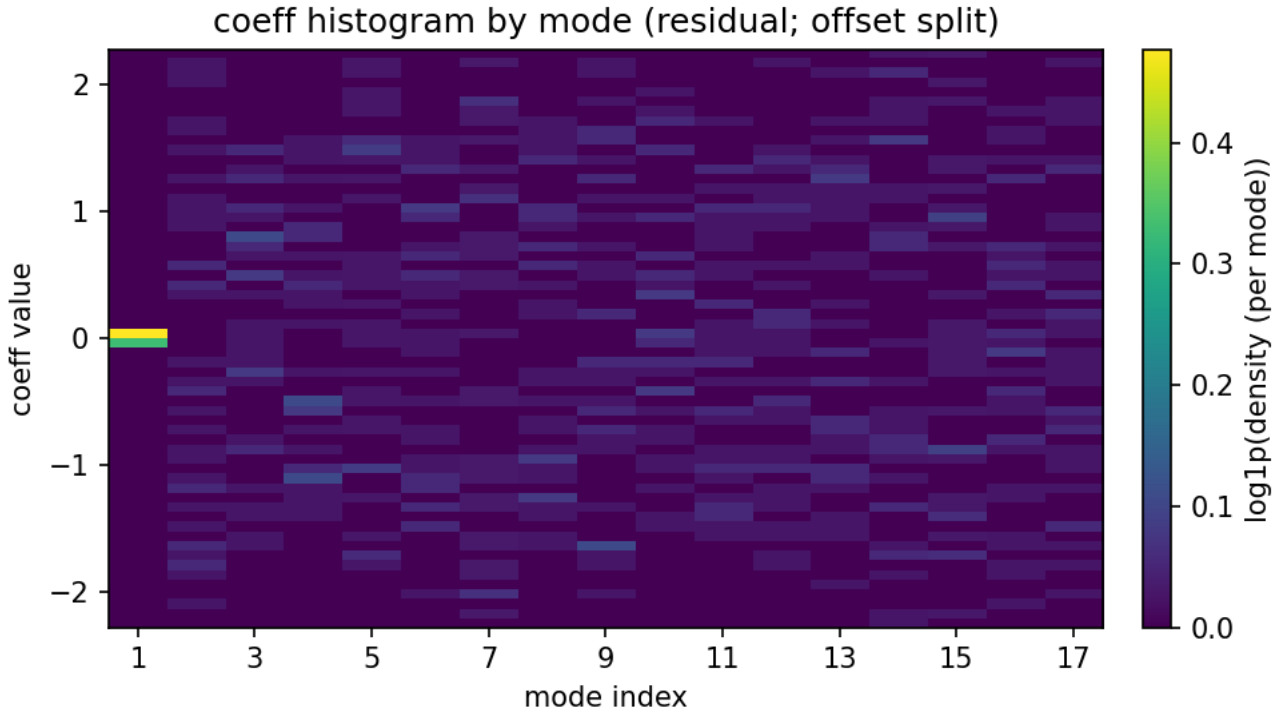
pod (pod)



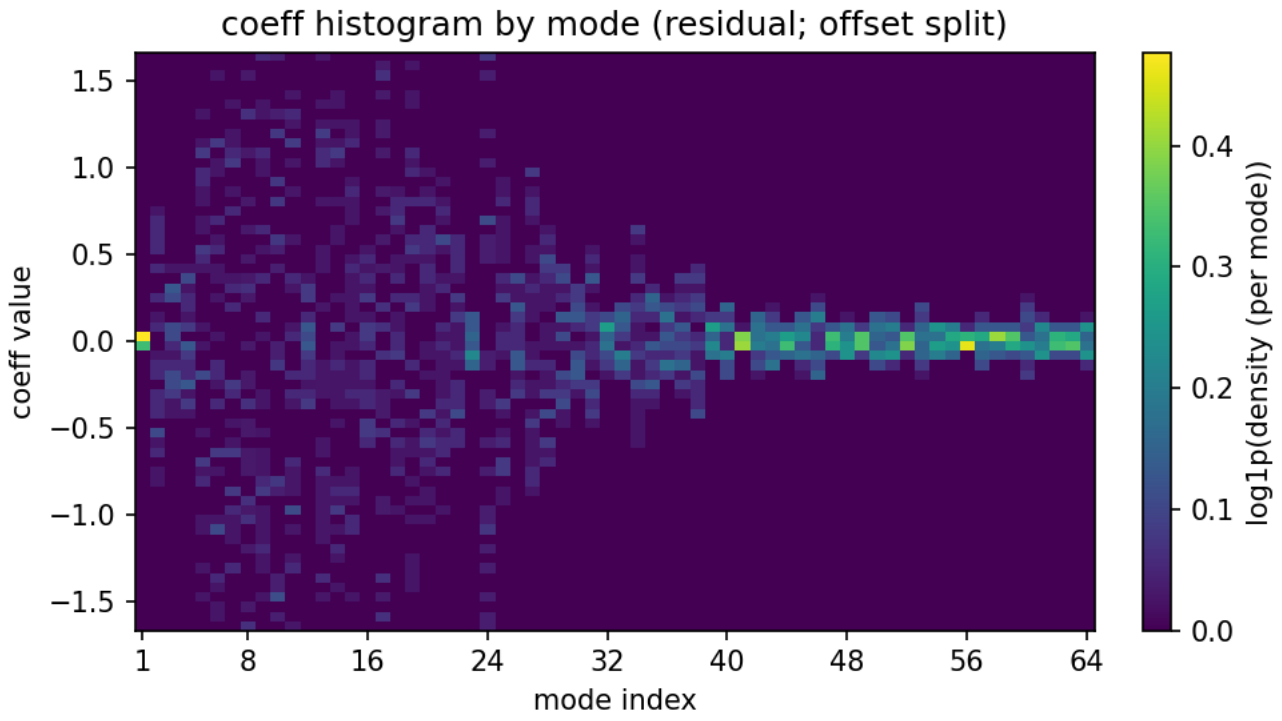
pod_em (pod_em)



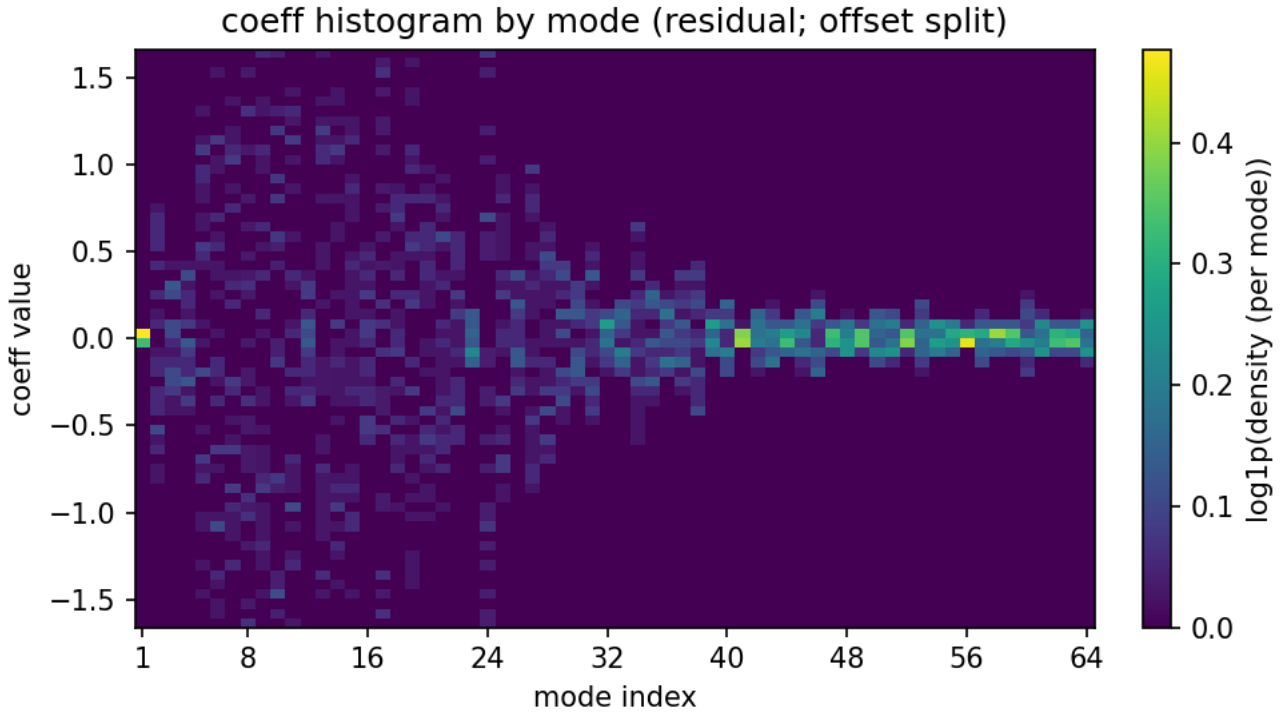
autoencoder_bench (autoencoder)



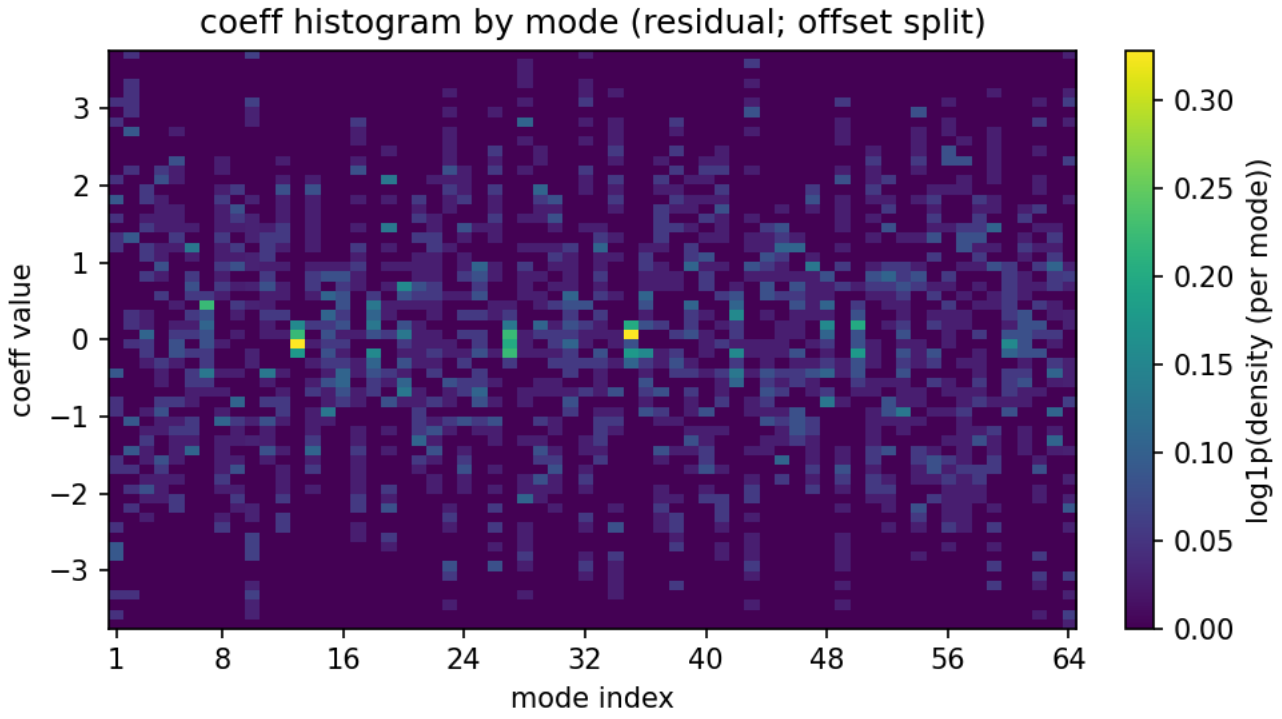
graph_fourier_bench (graph_fourier)



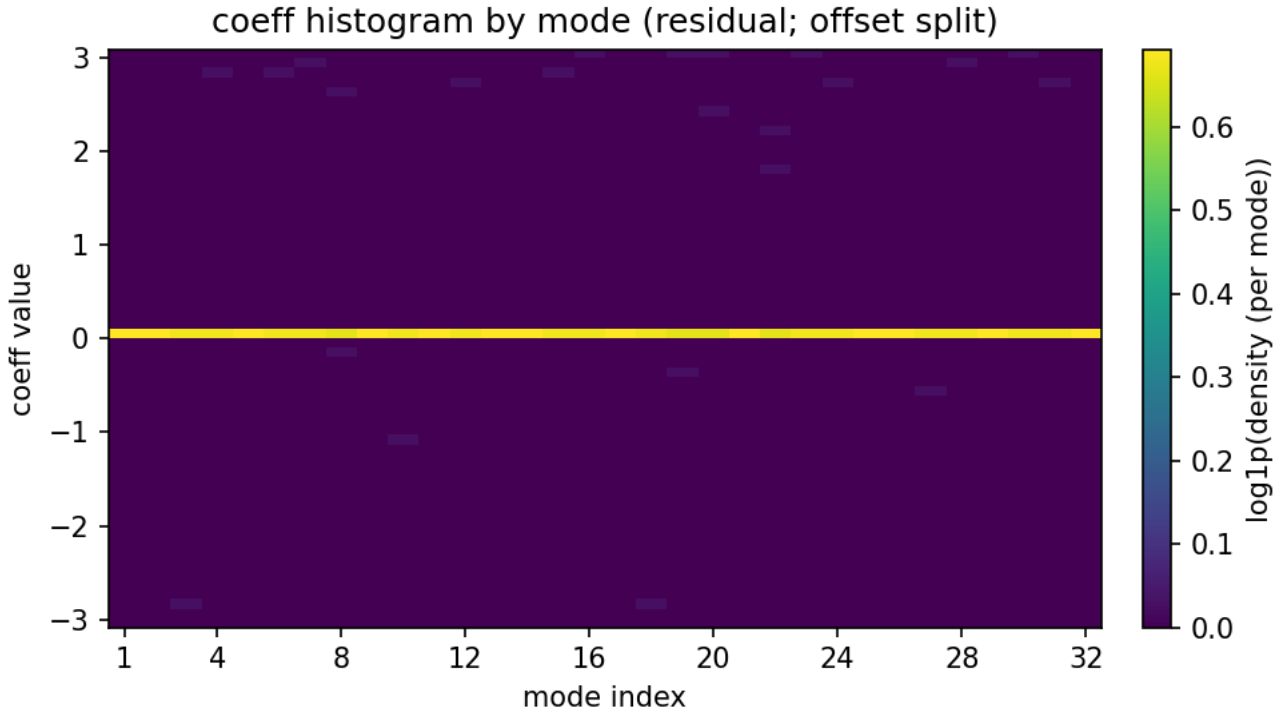
gappy_graph_fourier_bench (gappy_graph_fourier)



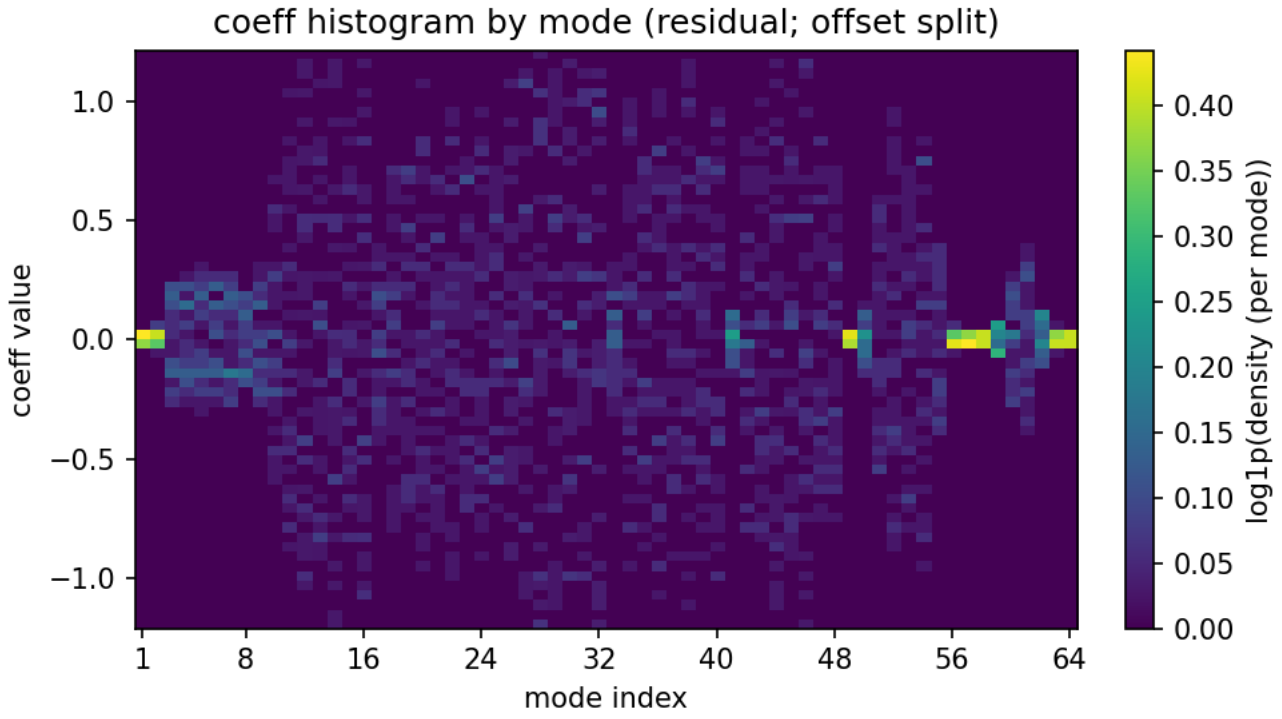
rbf_expansion_k64 (rbf_expansion)



dict_learning_bench (dict_learning)



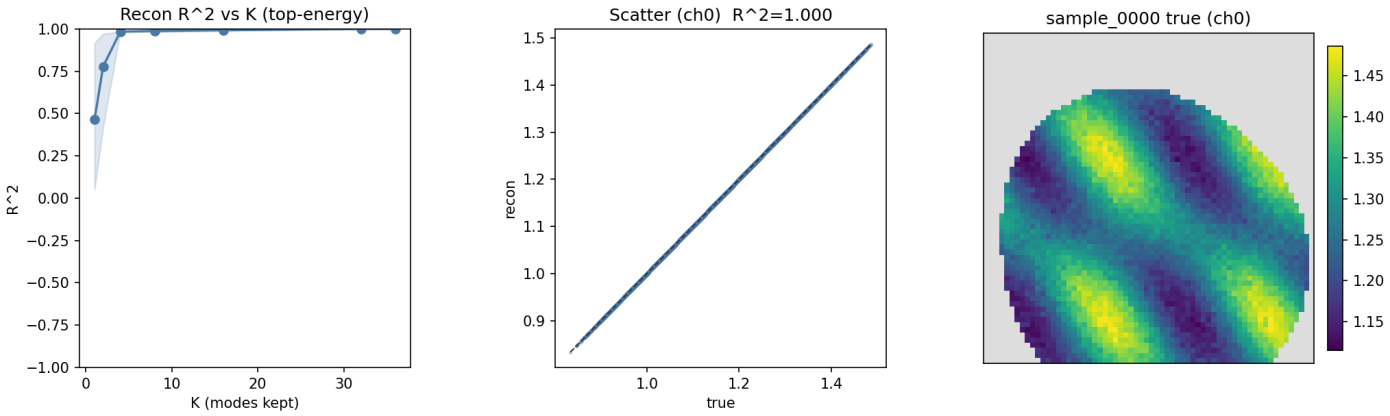
wavelet2d_k64 (wavelet2d)

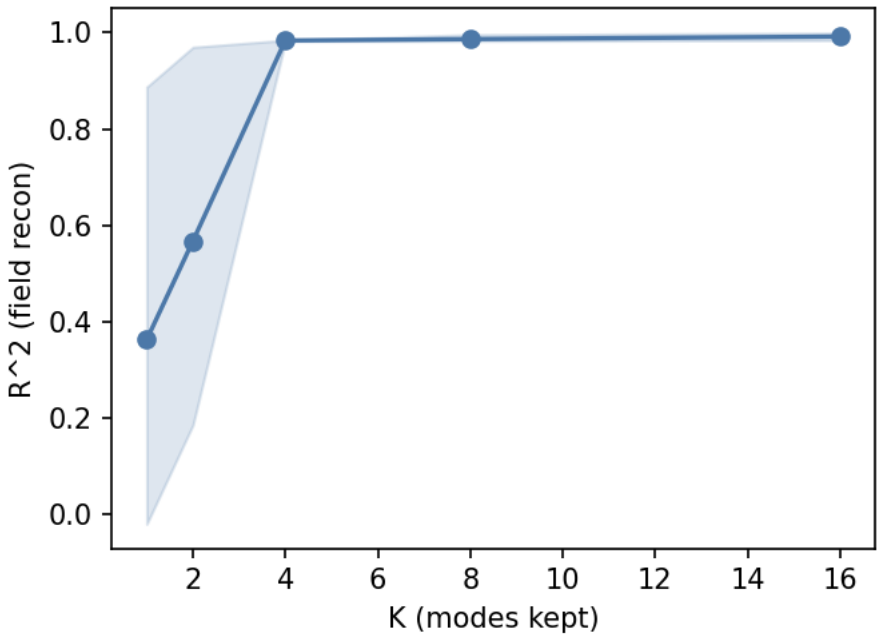


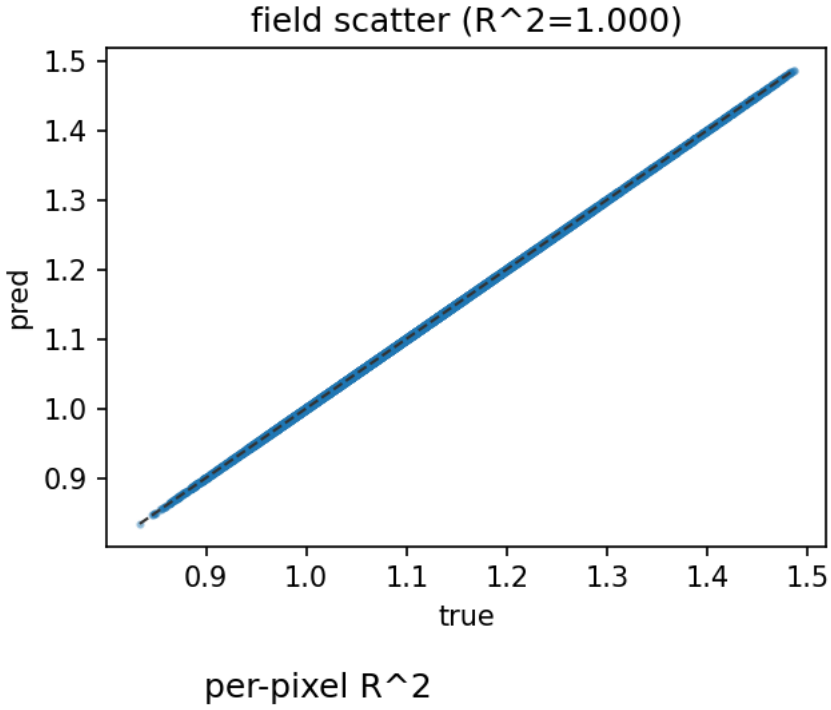
Compression leaderboard (Top-energy @K)

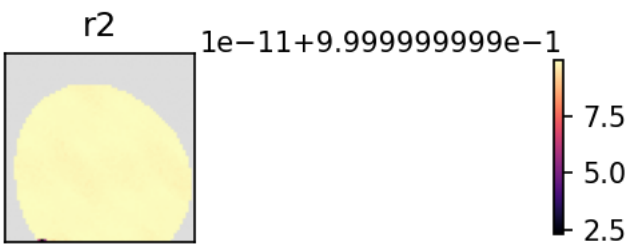
decompose(cfg)	method	r2_topk_k64	r2_topk_k16	k_req_r2_0.95	n_req
pod	pod	1.000000	0.989886	4	3
pod_em	pod_em	0.989886	0.989886	4	3
autoencoder_bench	autoencoder	0.976808	0.976808	16	16
graph_fourier_bench	graph_fourier	0.976361	0.851829	32	24
gappy_graph_fourier_bench	gappy_graph_fourier	0.976360	0.851828	32	24

Key decomposition plots (best_rmse=pod)









Train (cond -> coeff prediction)

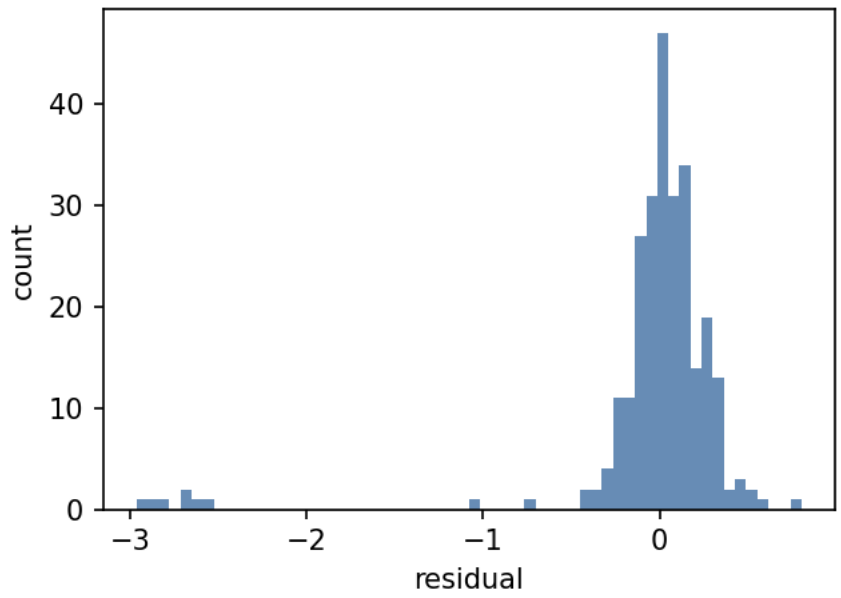
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2	fit

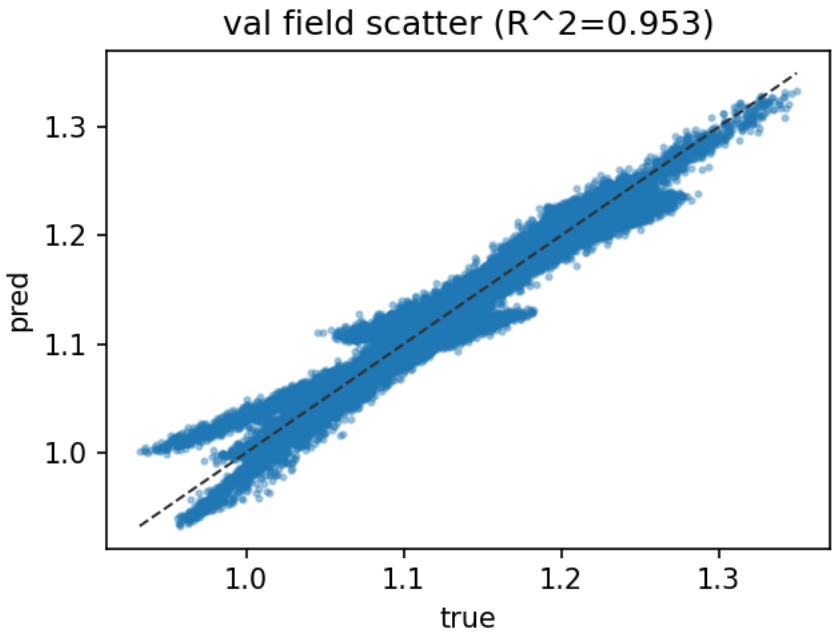
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2	fit
wavelet2d_k64	wavelet2d	0.850876	ridge	ok	1.032e-01	0.944795	1.584e-02	0.947848	2.9m
gappy_graph_fourier_bench	gappy_graph_fourier	0.976360	ridge	ok	1.132e-01	0.944137	1.774e-02	0.947691	3.8m
graph_fourier_bench	graph_fourier	0.976361	ridge	ok	1.133e-01	0.944137	1.776e-02	0.947692	1.7m
pod	pod	1.000000	ridge	ok	1.745e-01	0.926291	2.062e-02	0.931667	1.7m
pod_em	pod_em	0.989886	ridge	ok	2.392e-01	0.935770	1.917e-02	0.940455	1.9m
autoencoder_bench	autoencoder	0.976808	ridge	ok	2.468e-01	0.943179	1.861e-02	0.946424	2.3m
rbf_expansion_k64	rbf_expansion	0.967712	ridge	ok	3.593e-01	0.924369	1.743e-02	0.948115	1.9m
dict_learning_bench	dict_learning	0.936680	ridge	ok	4.913e-01	0.103301	1.548e-02	0.932046	1.7m

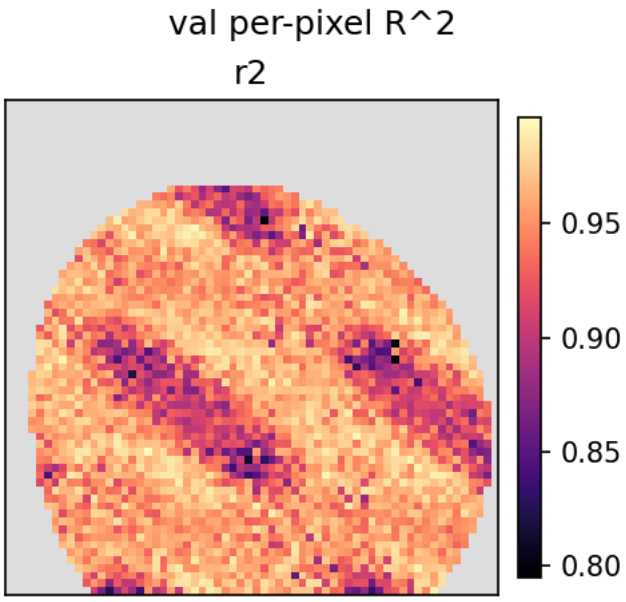
Train leaderboard (field-space)

decompose(cfg)	method	model	val_field_rmse	val_field_r2	run
dict_learning_bench	dict_learning	ridge	1.548e-02	0.932046	run
wavelet2d_k64	wavelet2d	ridge	1.584e-02	0.947848	run
rbf_expansion_k64	rbf_expansion	ridge	1.743e-02	0.948115	run
gappy_graph_fourier_bench	gappy_graph_fourier	ridge	1.774e-02	0.947691	run
graph_fourier_bench	graph_fourier	ridge	1.776e-02	0.947692	run

Key train plots (best_field_eval=dict_learning_bench)







sphere_grid_scalar

Problem setting

- domain: `sphere_grid` (n_lat=18, n_lon=36, lon_range=[-180.0, 170.0])
- field: `scalar`
- grid: `18x36`, x=[-180.0, 170.0], y=[-90.0, 90.0]
- cond: `(N,4)`
- mask: all-valid (no mask)

Highlights (auto)

- decomposition: best full recon = `dct2` (`dct2`) (field_rmse=4.517e-09, field_r2=1.000000)
- decomposition: best compression proxy = `graph_fourier_bench` (`graph_fourier`) (k_req_r2_0.95=8, r2_topk_k64=0.980022)
- decomposition: best top-energy@64 = `graph_fourier_bench` (`graph_fourier`) (r2_topk_k64=0.980022, k_req_r2_0.95=8)
- train: best coeff-space = `spherical_harmonics_scipy_bench` (`spherical_harmonics`) (`ridge`) (val_rmse=9.012e-03, val_r2=0.901779)
- train: best field-space = `spherical_slepian_scipy` (`spherical_slepian`) (`ridge`) (val_field_rmse=1.129e-02, val_field_r2=0.831449)
- train: mismatch detected (best coeff-space != best field-space)

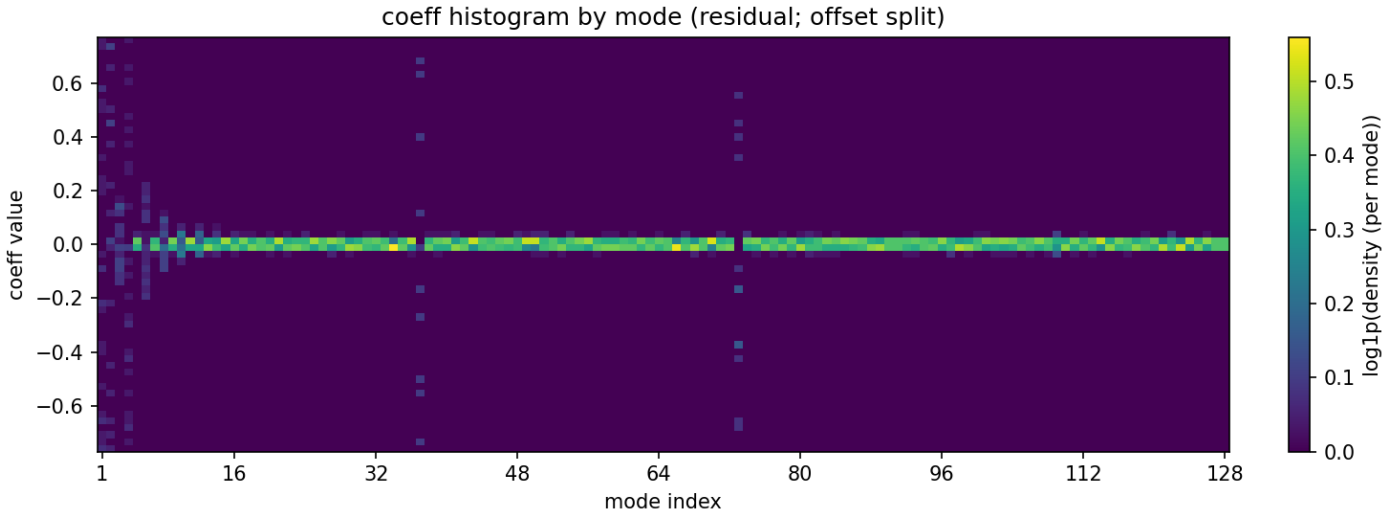
Decomposition (field reconstruction)

decompose(cfg)	method	status	rmse	r2	r2_k1	r2_k64	fit	n_req	k_req_r2_
`dct2`	`dct2`	ok	4.517e-09	1.000000	0.000000	0.980436	970.6us	73	
`graph_fourier_bench`	`graph_fourier`	ok	1.133e-02	0.980022	-0.000000	0.980022	31.6ms	9	8

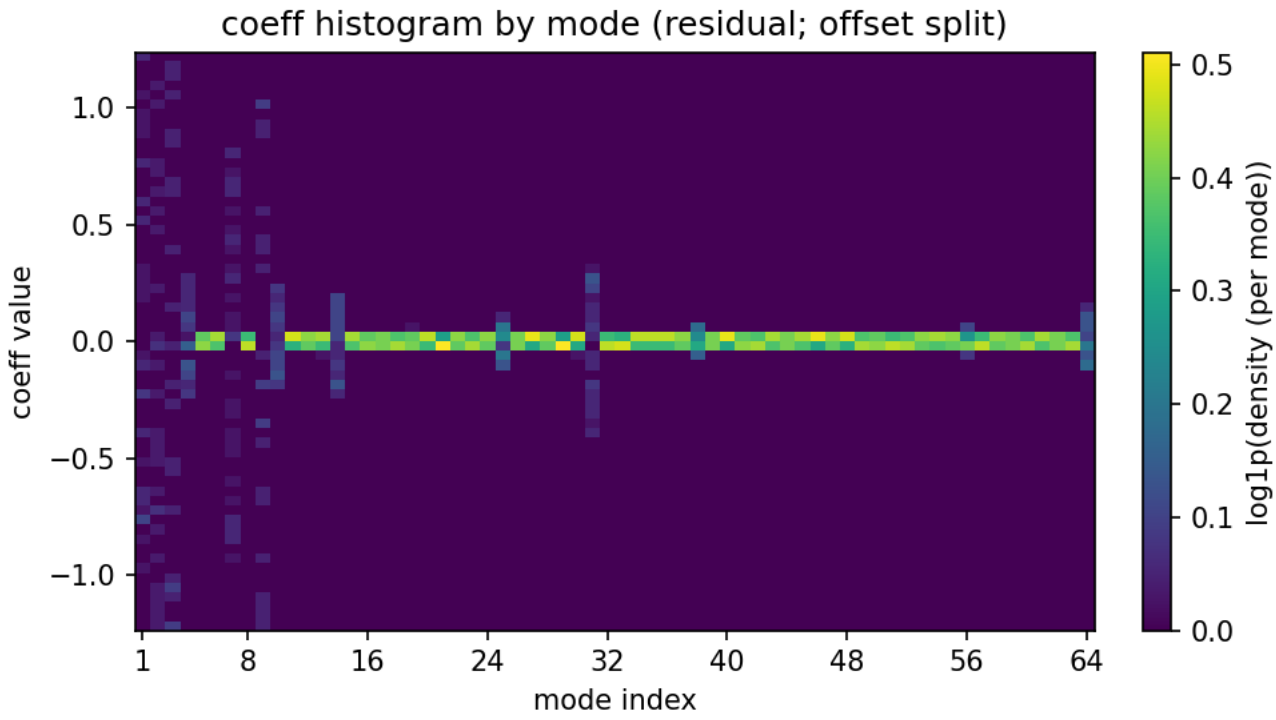
decompose(cfg)	method	status	rmse	r2	r2_k1	r2_k64	fit	n_req	k_req_r2_
spherical_harmonics_scipy_bench	spherical_harmonics	ok	1.996e-02	0.937276	-0.104138	0.937276	1.0ms	5	
spherical_slepian_scipy	spherical_slepian	ok	7.313e-02	0.168643	-0.094486	0.168643	4.6ms	7	

Coefficient histograms by mode (per decomposer)

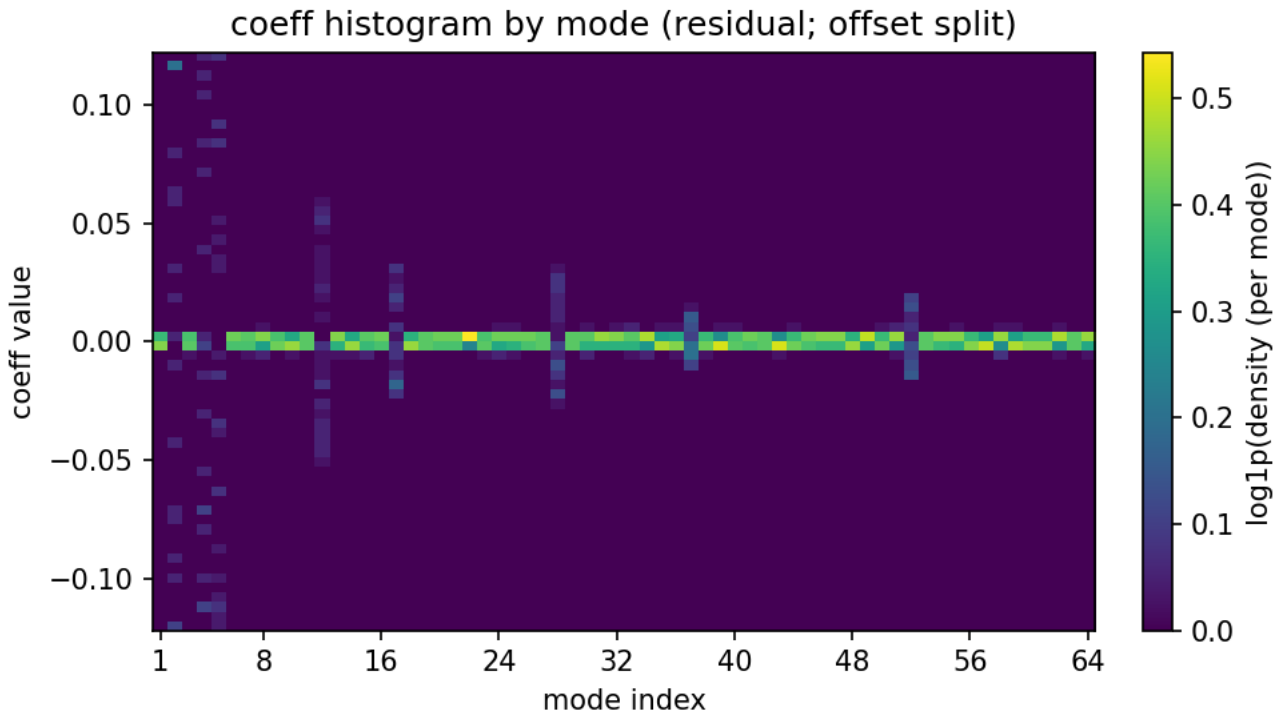
dct2 (dct2)



graph_fourier_bench (graph_fourier)



spherical_harmonics_scipy_bench (spherical_harmonics)

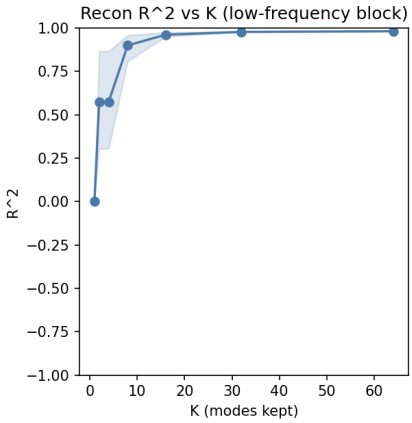


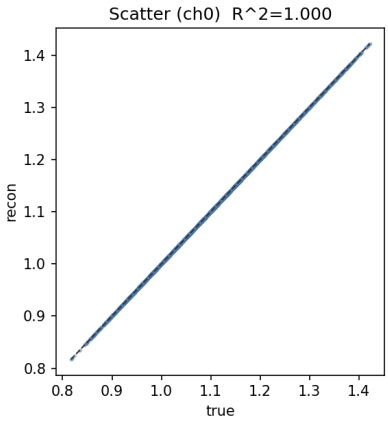
spherical_slepian_scipy (spherical_slepian)

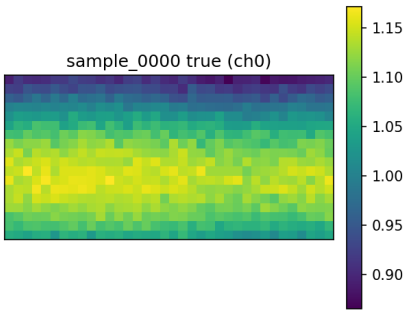
Compression leaderboard (Top-energy @K)

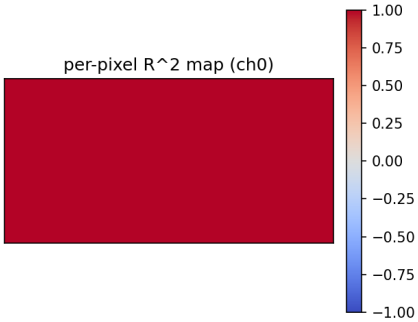
decompose(cfg)	method	r2_topk_k64	r2_topk_k16	k_req_r2_0.95	n_req
graph_fourier_bench	graph_fourier	0.980022	0.978535	8	9
spherical_harmonics_scipy_bench	spherical_harmonics	0.937276	0.935933		5
spherical_slepian_scipy	spherical_slepian	0.168643	0.168643		7

Key decomposition plots (best_rmse=dct2)

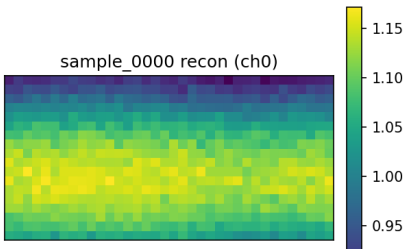

Recon R^2 vs K (low-frequency block)
R^2
K (modes kept)

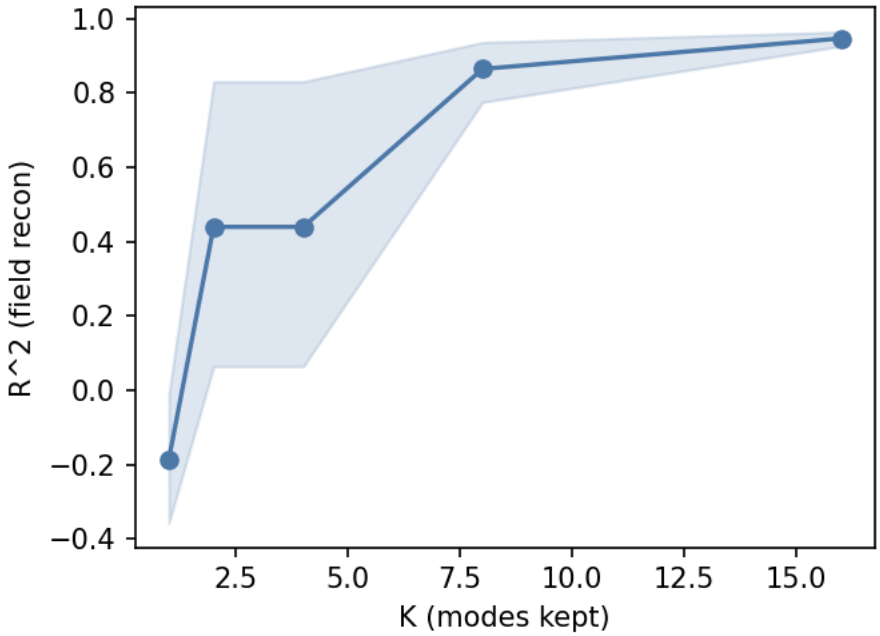

Scatter (ch0) R^2=1.000
recon
true

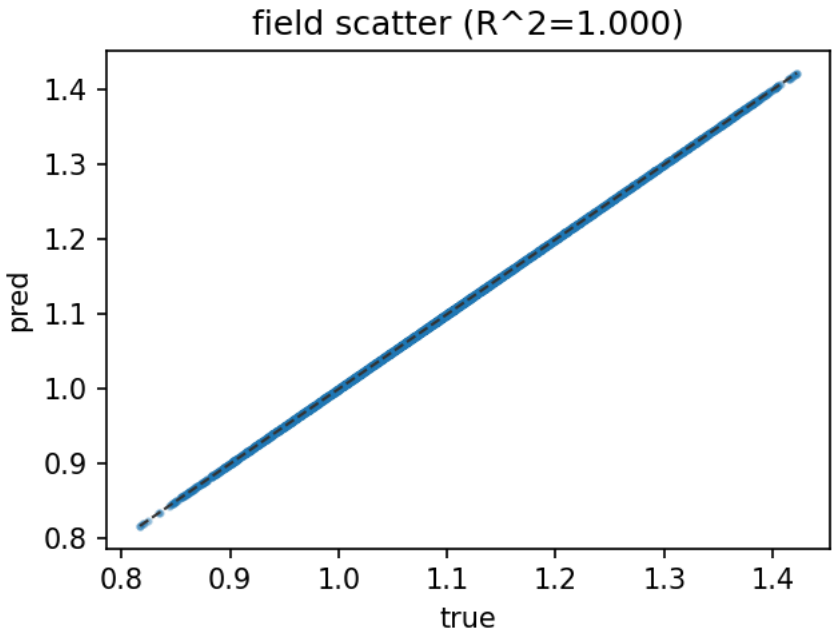

sample_0000 true (ch0)

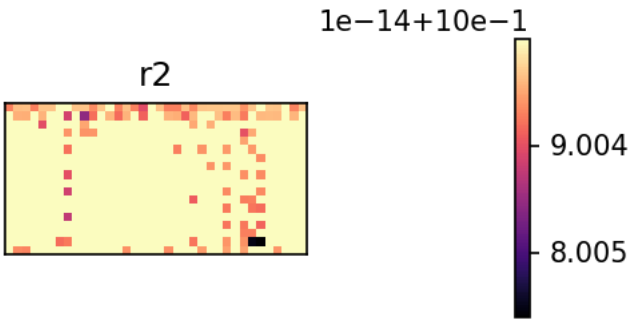

per-pixel R^2 map (ch0)


abs error (robust)


sample_0000 recon (ch0)


R^2 (field recon)
K (modes kept)





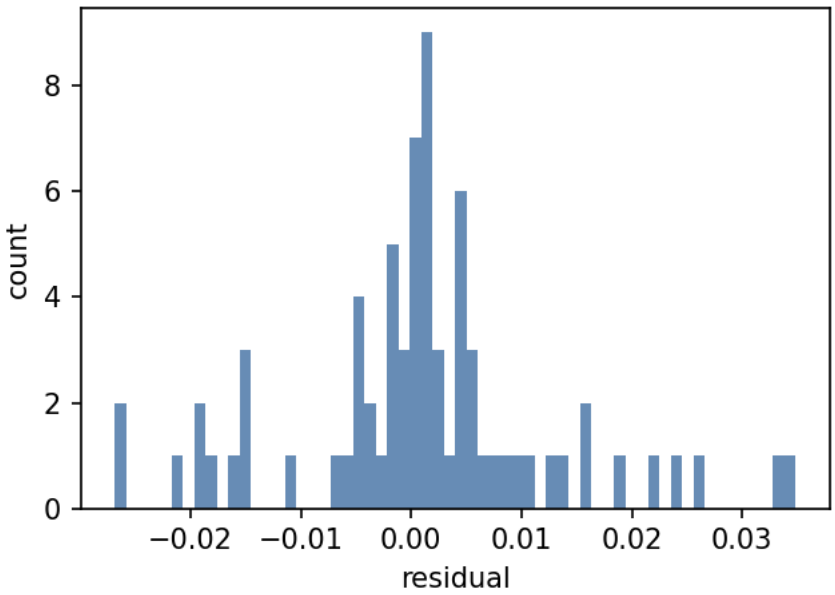
Train (cond -> coeff prediction)

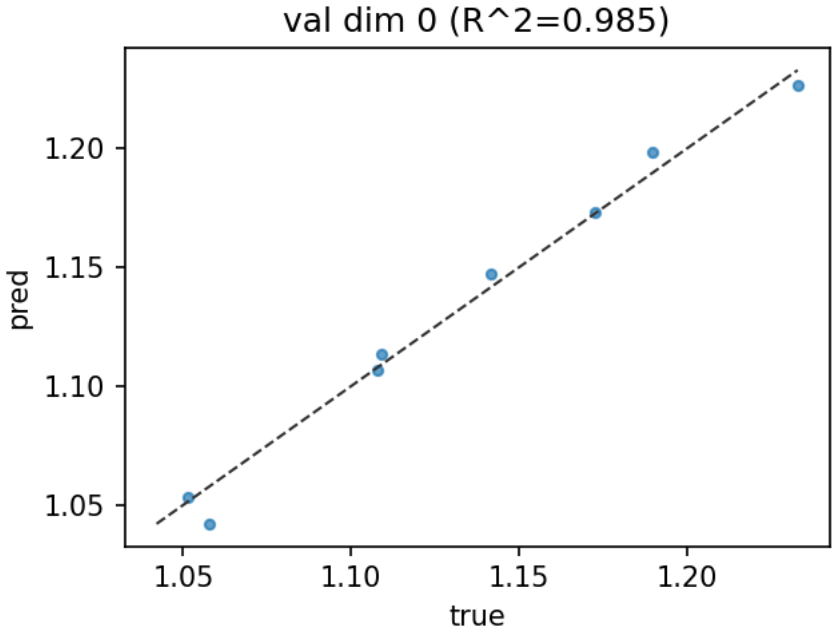
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2
spherical_harmonics_scipy_bench	spherical_harmonics	0.937276	ridge	ok	9.012e-03	0.901779	2.440e-02	0.887717
spherical_slepian_scipy	spherical_slepian	0.168643	ridge	ok	1.192e-02	0.919739	1.129e-02	0.831449
dct2	dct2	1.000000	ridge	ok	2.877e-02	0.880635	2.759e-02	0.873499
graph_fourier_bench	graph_fourier	0.980022	ridge	ok	8.497e-02	0.893887	2.563e-02	0.887421

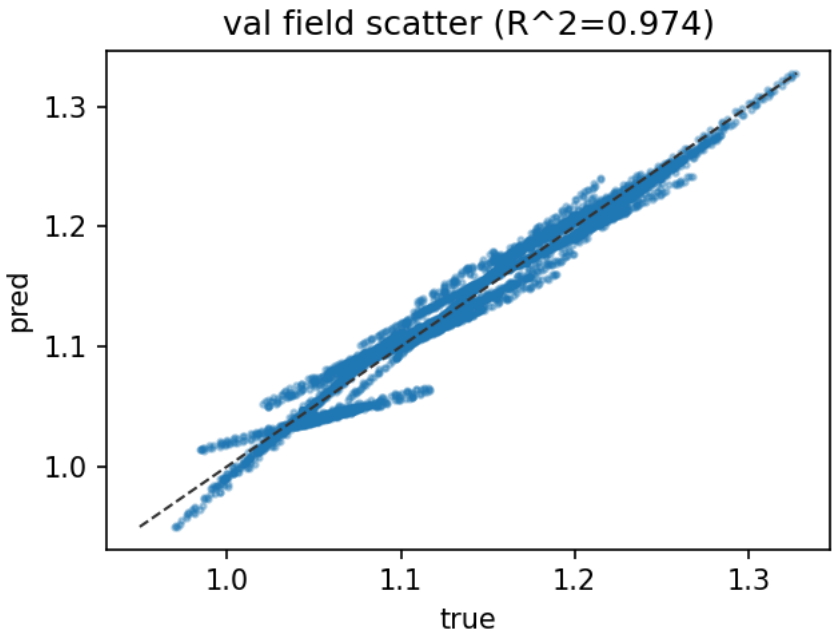
Train leaderboard (field-space)

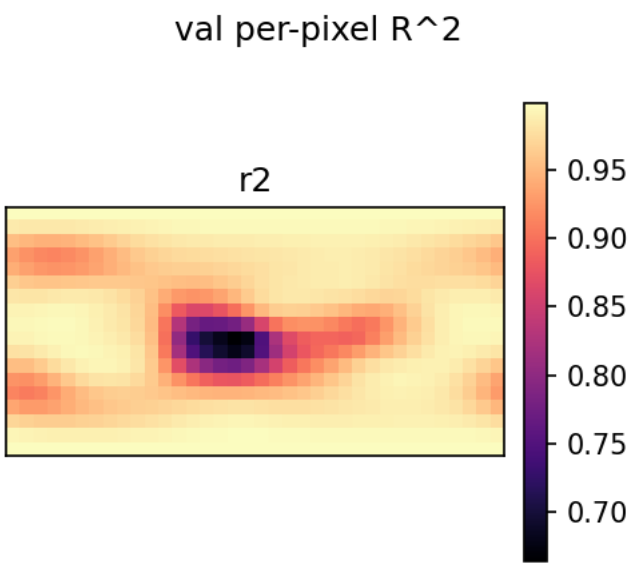
decompose(cfg)	method	model	val_field_rmse	val_field_r2	run
spherical_slepian_scipy	spherical_slepian	ridge	1.129e-02	0.831449	run
spherical_harmonics_scipy_bench	spherical_harmonics	ridge	2.440e-02	0.887717	run
graph_fourier_bench	graph_fourier	ridge	2.563e-02	0.887421	run
dct2	dct2	ridge	2.759e-02	0.873499	run

Key train plots (best_field_eval=spherical_slepian_scipy)


count
residual
0
2
4
6
8
−0.02
−0.01
0.00
0.01
0.02
0.03


val dim 0 (R^2=0.985)
pred
true
1.05
1.10
1.15
1.20


val field scatter (R^2=0.974)
pred
true
1.0
1.1
1.2
1.3



rectangle_vector

Problem setting

- domain: `rectangle` ()
- field: `vector`
- grid: `64x64`, x=[-1.0, 1.0], y=[-1.0, 1.0]
- cond: `(N,8)`
- mask: all-valid (no mask)

Highlights (auto)

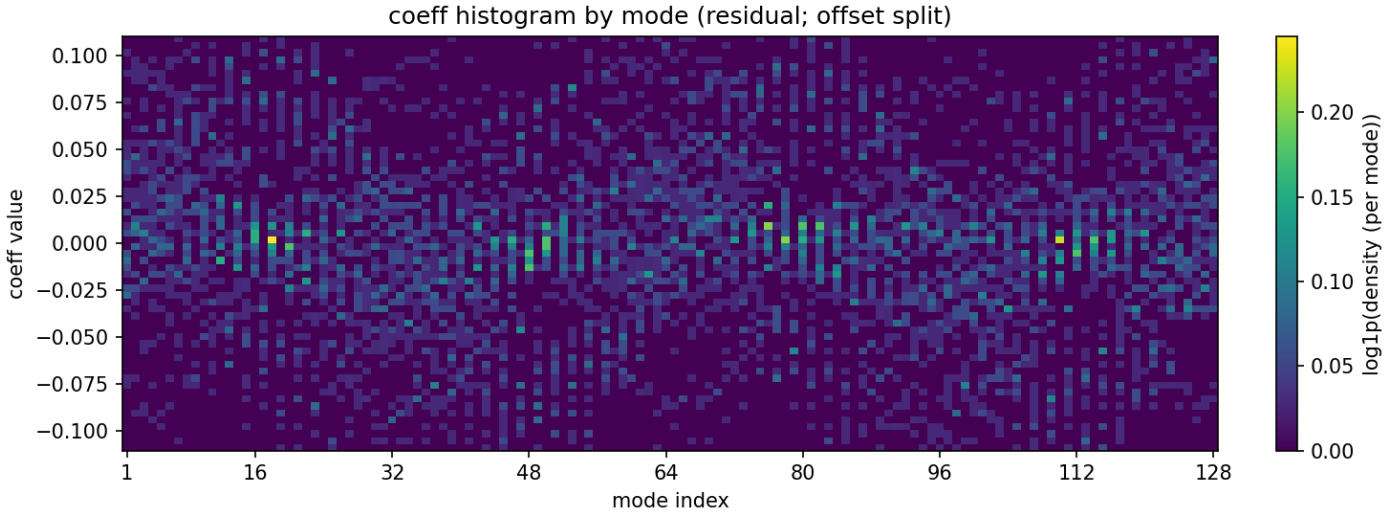
- decomposition: best full recon = `helmholtz` (`helmholtz`) (field_rmse=0.000e+00, field_r2=1.000000)
- decomposition: best compression proxy = `pod_joint_em` (`pod_joint_em`) (k_req_r2_0.95=8, r2_topk_k64=0.992829)
- decomposition: best top-energy@64 = `pod_joint_em` (`pod_joint_em`) (r2_topk_k64=0.992829, k_req_r2_0.95=8)
- train: best coeff-space = `helmholtz` (`helmholtz`) (`ridge`) (val_rmse=1.816e-02, val_r2=0.883531)
- train: best field-space = `graph_fourier_bench` (`graph_fourier`) (`ridge`) (val_field_rmse=2.367e-02, val_field_r2=0.954785)
- train: mismatch detected (best coeff-space != best field-space)

Decomposition (field reconstruction)

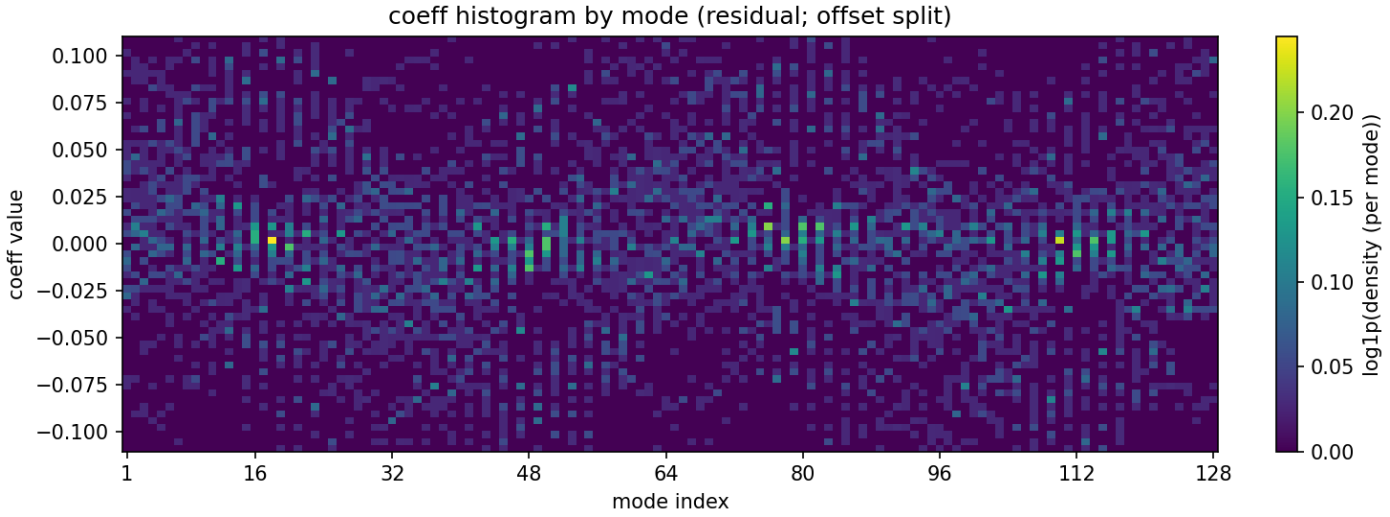
decompose(cfg)	method	status	rmse	r2	r2_k1	r2_k64	fit	n_req	k_req_r2_0.95	run
helmholtz	helmholtz	ok	0.000e+00	1.000000			4.6ms	7322		run
helmholtz_poisson	helmholtz_poisson	ok	0.000e+00	1.000000			19.2ms	7322		run
dct2	dct2	ok	5.485e-09	1.000000	0.303503	0.978527	5.6ms	321		run
pod	pod	ok	1.094e-07	1.000000			26.4ms	3		run
pod_joint_em	pod_joint_em	ok	8.122e-03	0.992829	0.535751	0.992829	1.061s	5	8	run
pod_joint	pod_joint	ok	8.334e-03	0.992514	0.508488	0.992514	25.5ms	5	8	run
autoencoder_bench	autoencoder	ok	1.436e-02	0.978326	0.310887	0.978326	7.250s	17	16	run
graph_fourier_bench	graph_fourier	ok	1.506e-02	0.976130	0.303503	0.976130	182.4ms	32	16	run

Coefficient histograms by mode (per decomposer)

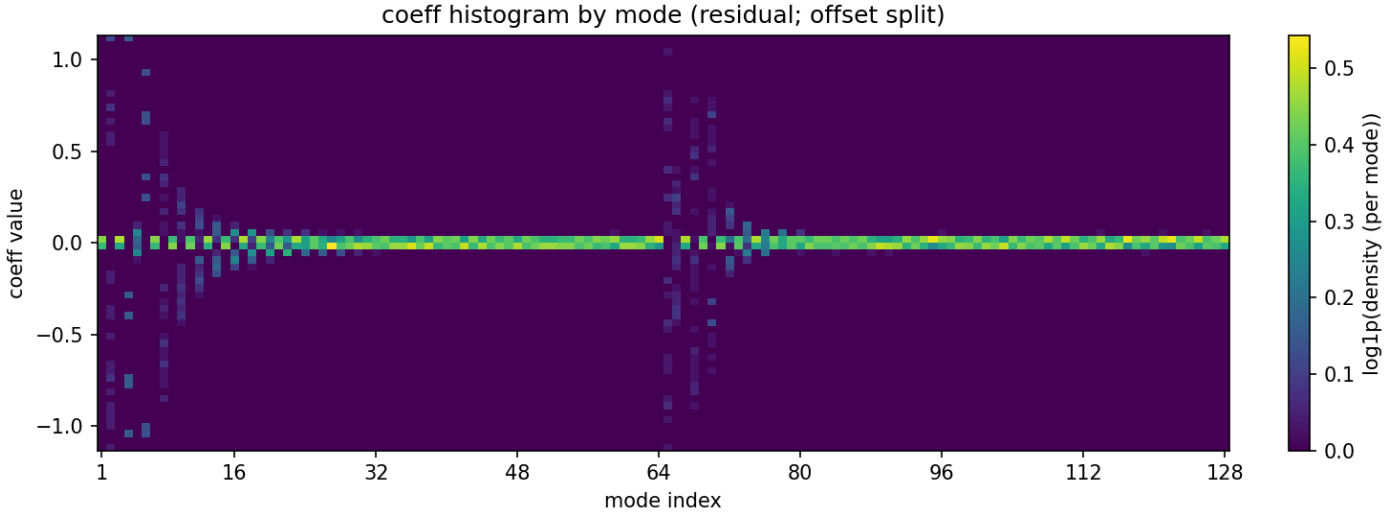
helmholtz (helmholtz)



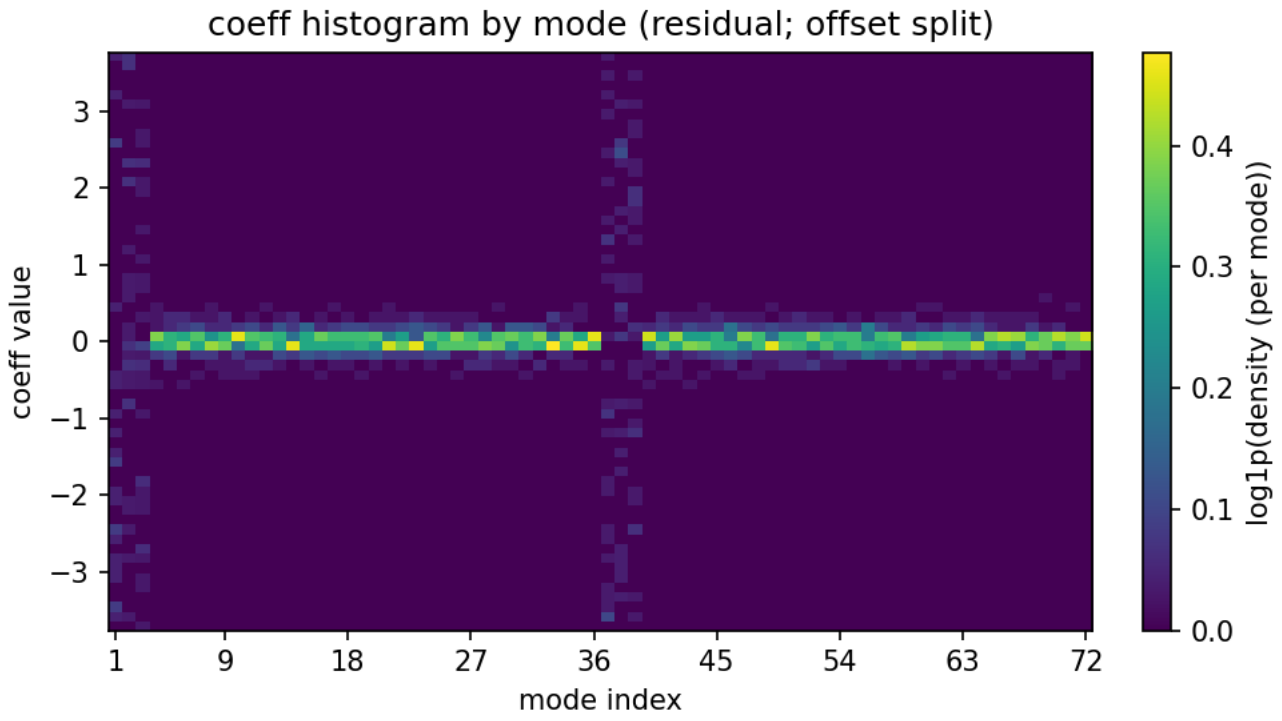
helmholtz_poisson (helmholtz_poisson)



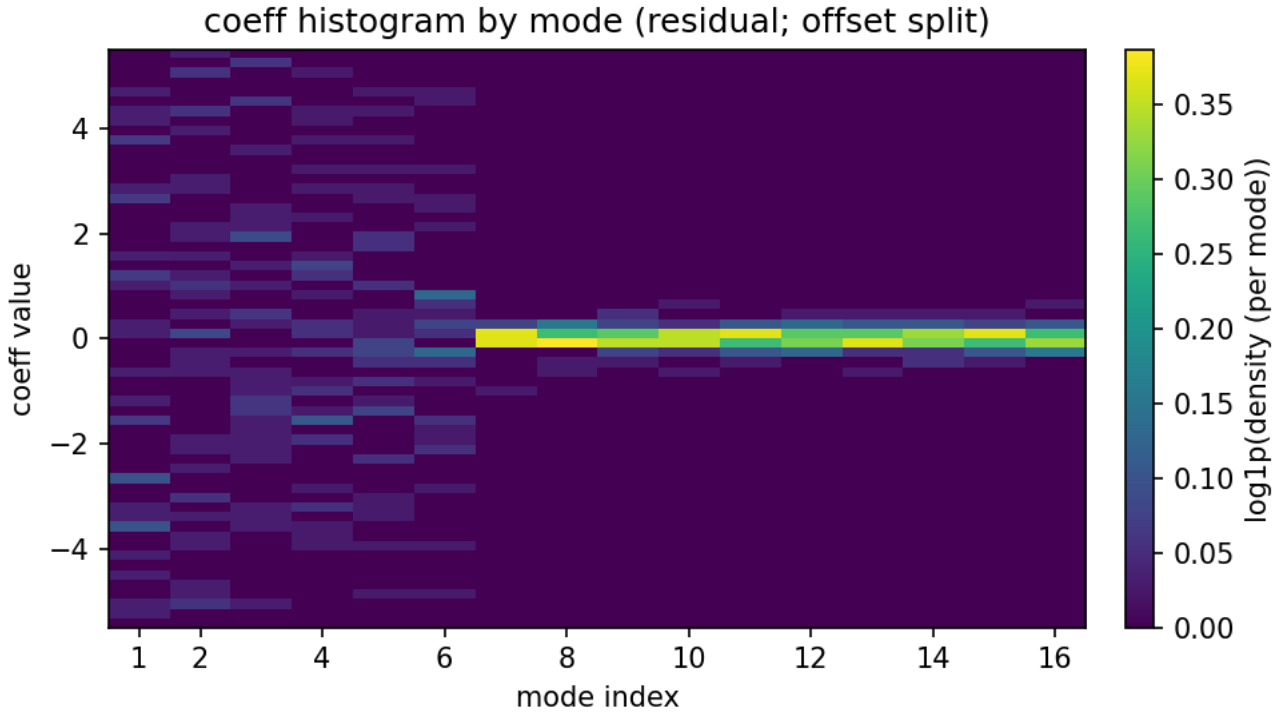
dct2 (dct2)



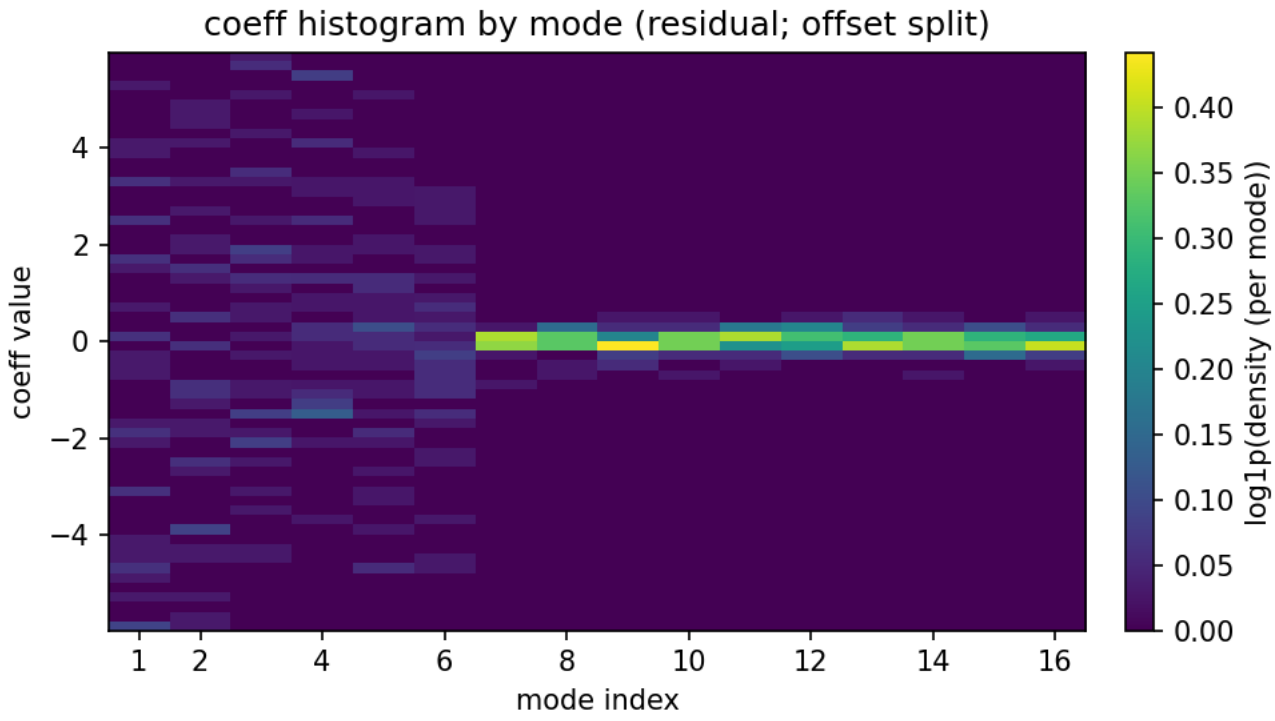
pod (pod)



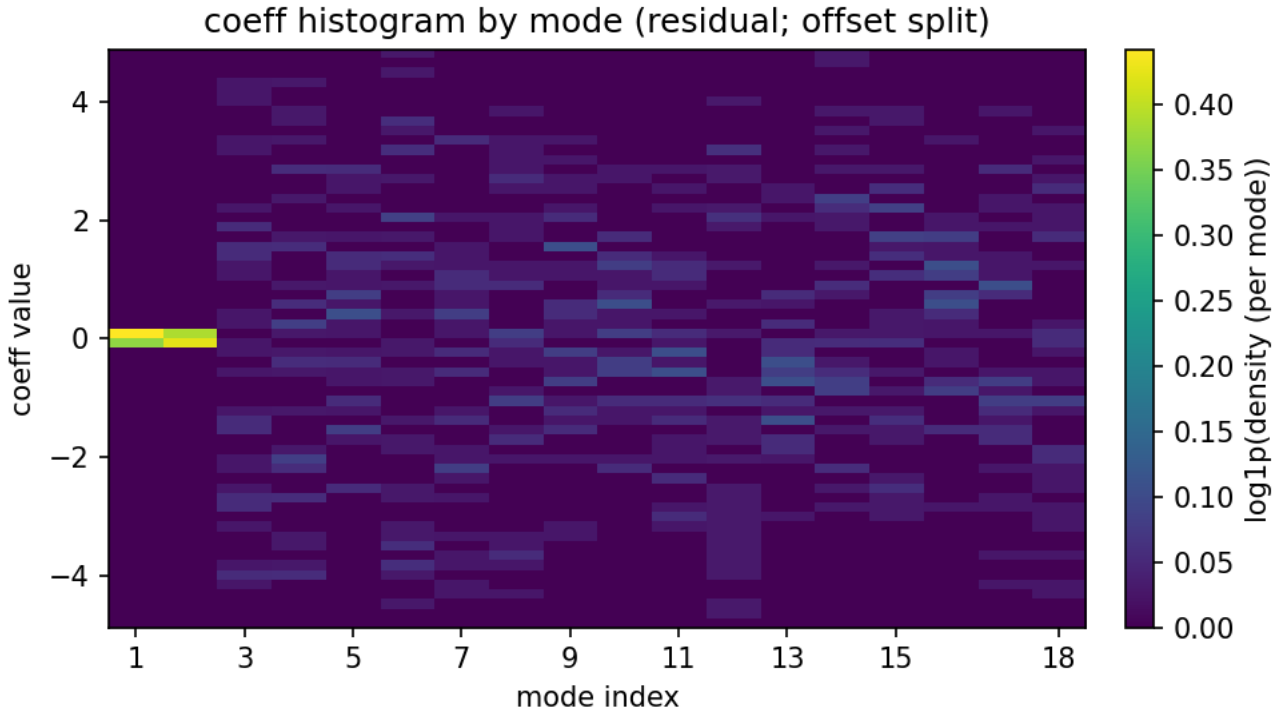
pod_joint_em (pod_joint_em)



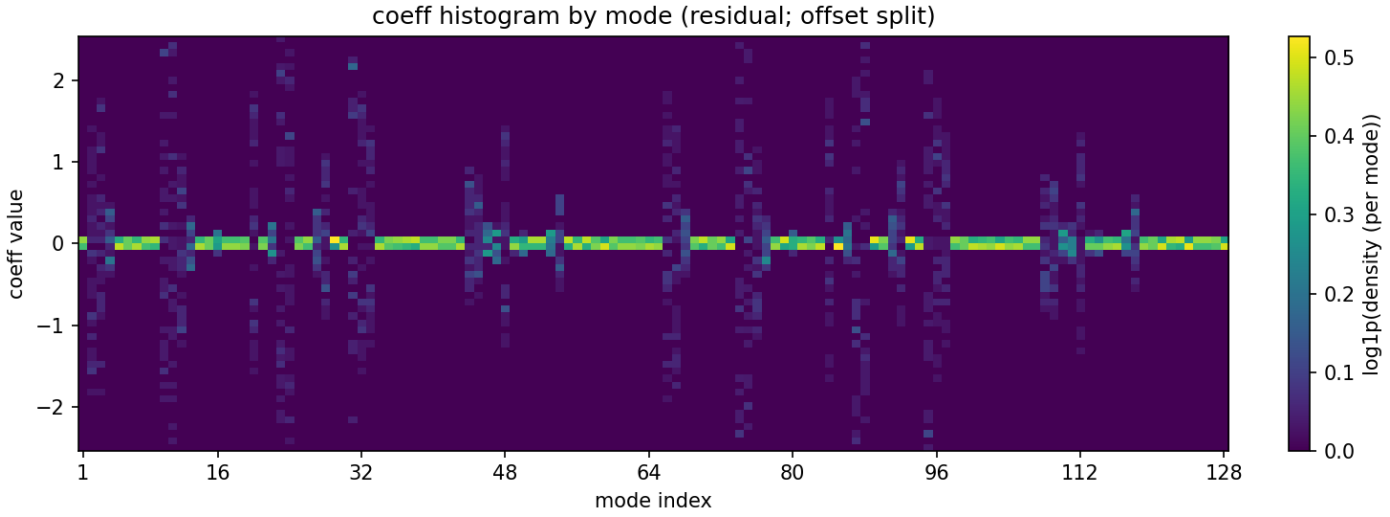
pod_joint (pod_joint)



autoencoder_bench (autoencoder)



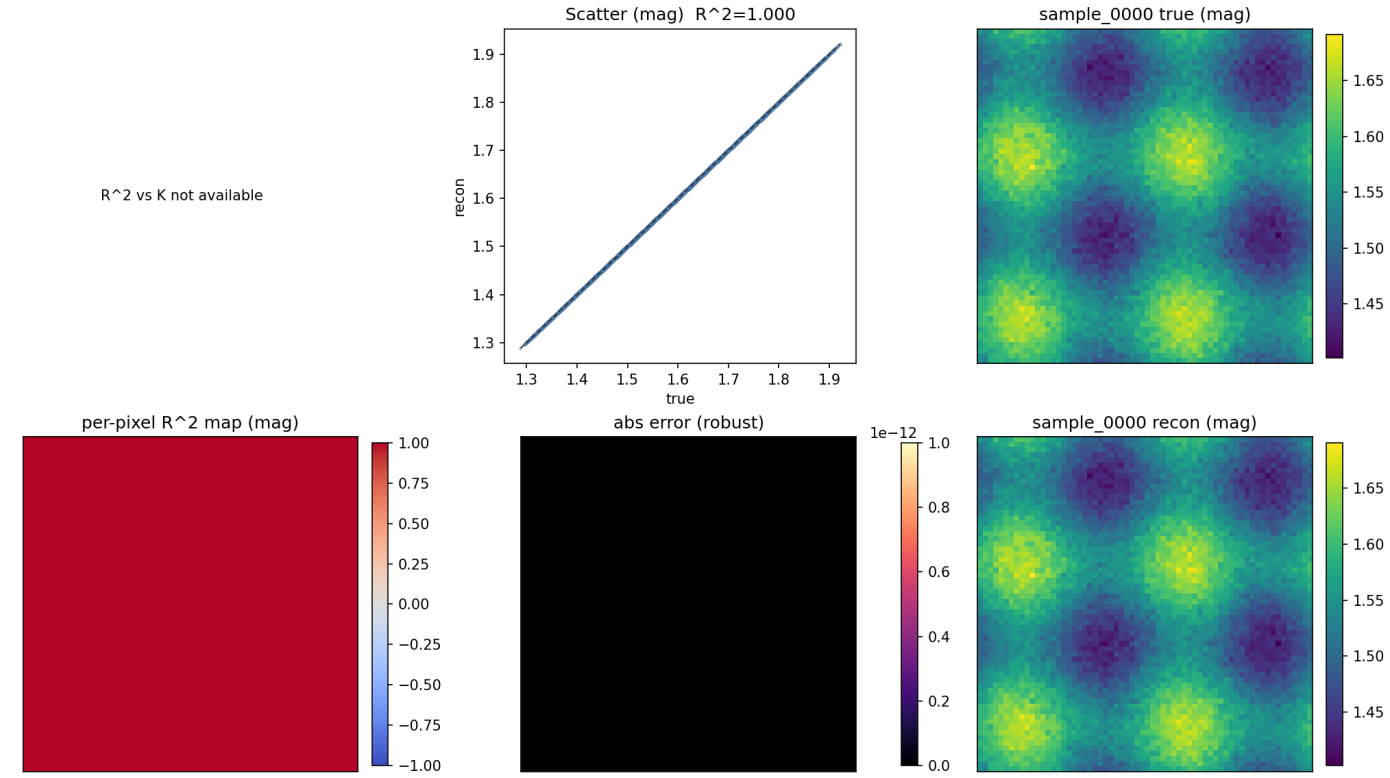
graph_fourier_bench (graph_fourier)

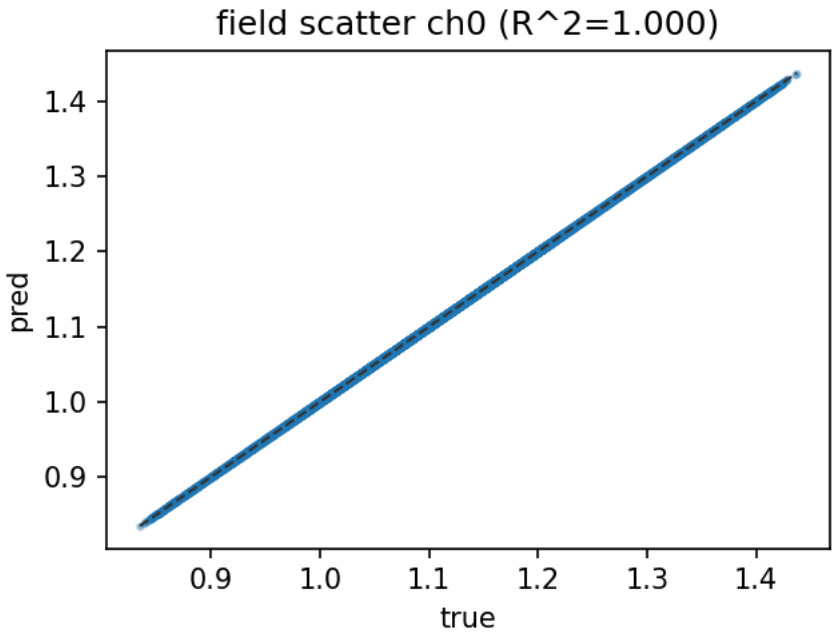


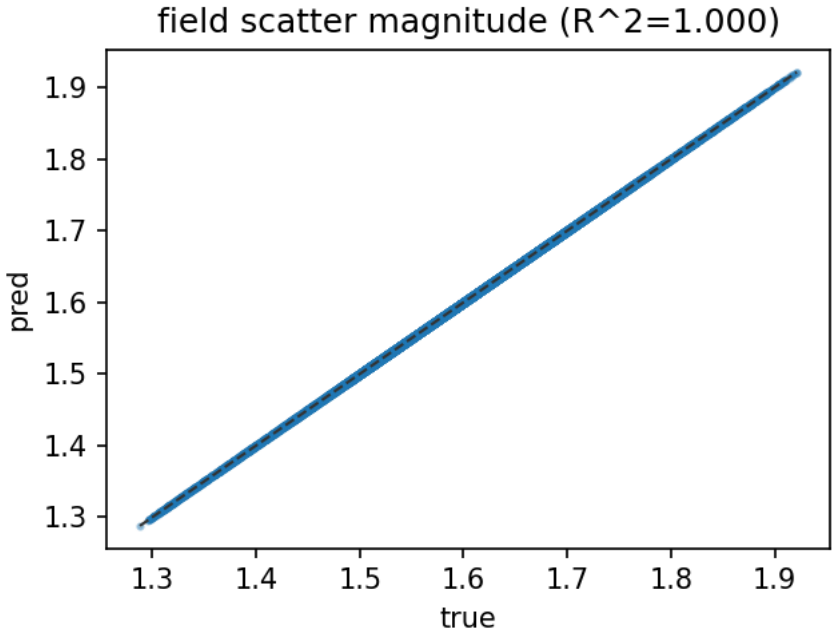
Compression leaderboard (Top-energy @K)

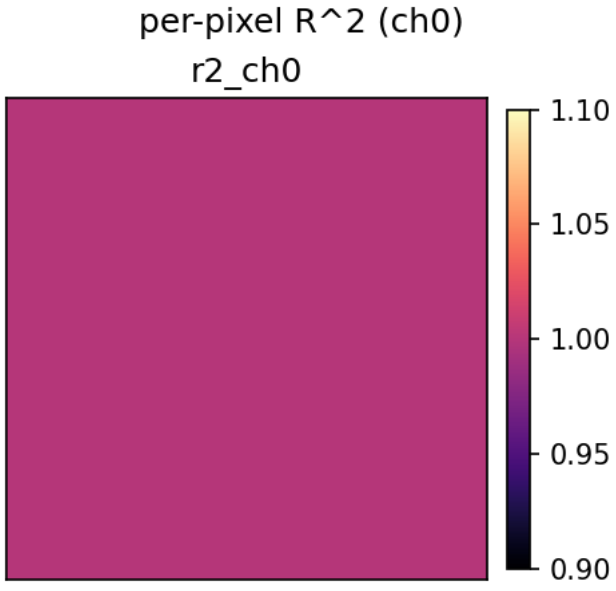
decompose(cfg)	method	r2_topk_k64	r2_topk_k16	k_req_r2_0.95	n_req
pod_joint_em	pod_joint_em	0.992829	0.992829	8	5
pod_joint	pod_joint	0.992514	0.992514	8	5
autoencoder_bench	autoencoder	0.978326	0.978326	16	17
graph_fourier_bench	graph_fourier	0.976130	0.970453	16	32

Key decomposition plots (best_rmse=helmholtz)









Train (cond -> coeff prediction)

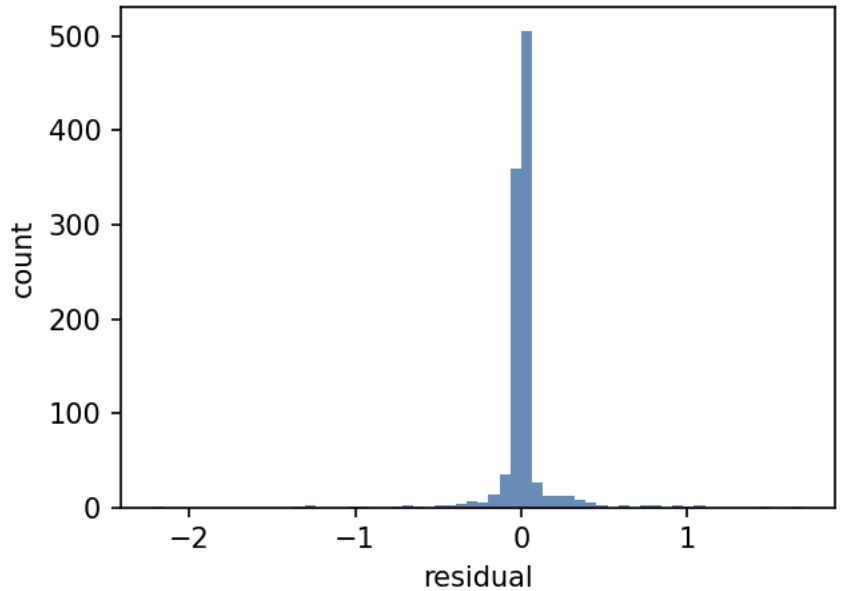
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2	fit	run

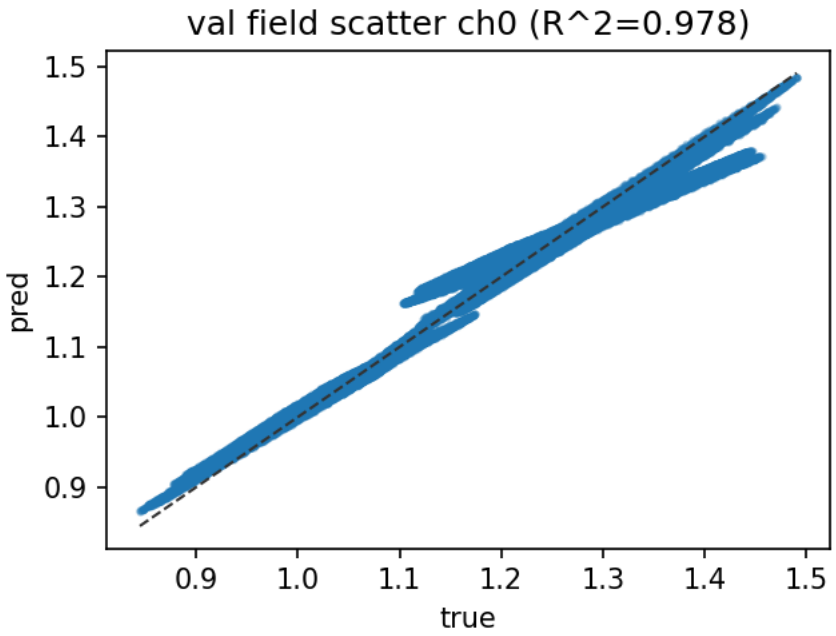
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2	fit	run
helmholtz	helmholtz	1.000000	ridge	ok	1.816e-02	0.883531	2.591e-02	0.946499	13.0ms	run
helmholtz_poisson	helmholtz_poisson	1.000000	ridge	ok	1.816e-02	0.883531	2.591e-02	0.946499	11.2ms	run
dct2	dct2	1.000000	ridge	ok	2.570e-02	0.883388	2.591e-02	0.946499	7.0ms	run
graph_fourier_bench	graph_fourier	0.976130	ridge	ok	1.861e-01	0.899452	2.367e-02	0.954785	2.2ms	run
pod	pod	1.000000	ridge	ok	2.705e-01	0.883387	2.591e-02	0.946499	1.7ms	run
pod_joint	pod_joint	0.992514	ridge	ok	5.294e-01	0.890511	2.503e-02	0.950003	1.6ms	run
pod_joint_em	pod_joint_em	0.992829	ridge	ok	5.303e-01	0.890198	2.507e-02	0.949914	1.7ms	run
autoencoder_bench	autoencoder	0.978326	ridge	ok	5.677e-01	0.899857	2.406e-02	0.953719	1.7ms	run

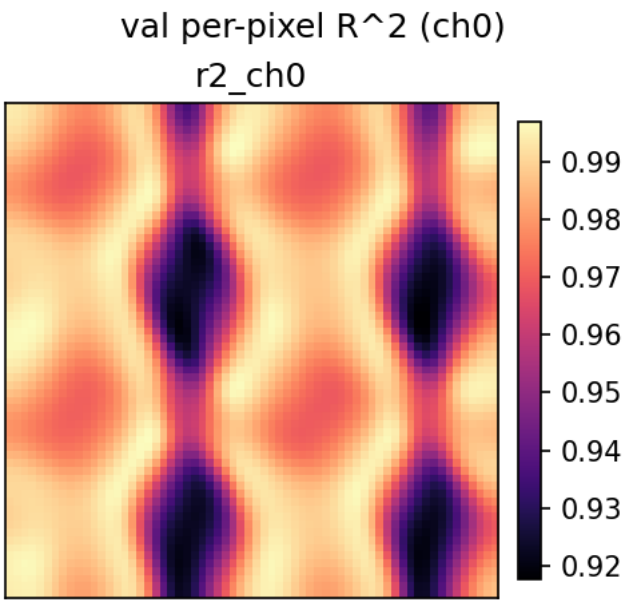
Train leaderboard (field-space)

decompose(cfg)	method	model	val_field_rmse	val_field_r2	run
graph_fourier_bench	graph_fourier	ridge	2.367e-02	0.954785	run
autoencoder_bench	autoencoder	ridge	2.406e-02	0.953719	run
pod_joint	pod_joint	ridge	2.503e-02	0.950003	run
pod_joint_em	pod_joint_em	ridge	2.507e-02	0.949914	run
helmholtz	helmholtz	ridge	2.591e-02	0.946499	run

Key train plots (best_field_eval=graph_fourier_bench)







disk_vector

Problem setting

- domain: `disk` (center=[0.0, 0.0], radius=1.0)
- field: `vector`
- grid: `64x64`, x=[-1.0, 1.0], y=[-1.0, 1.0]
- cond: `(N,8)`
- mask: geometric domain mask (outside is 0-filled; evaluation uses inside only)

Highlights (auto)

- decomposition: best full recon = `pod_joint_em` (`pod_joint_em`) (field_rmse=7.975e-03, field_r2=0.991907)
- decomposition: best compression proxy = `pod_joint_em` (`pod_joint_em`) (k_req_r2_0.95=4, r2_topk_k64=0.991907)
- decomposition: best top-energy@64 = `pod_joint_em` (`pod_joint_em`) (r2_topk_k64=0.991907, k_req_r2_0.95=4)
- train: best coeff-space = `pseudo_zernike` (`pseudo_zernike`) (`ridge`) (val_rmse=4.822e-03, val_r2=0.891683)
- train: best field-space = `gappy_graph_fourier_bench` (`gappy_graph_fourier`) (`ridge`) (val_field_rmse=3.095e-02, val_field_r2=0.893211)
- train: mismatch detected (best coeff-space != best field-space)

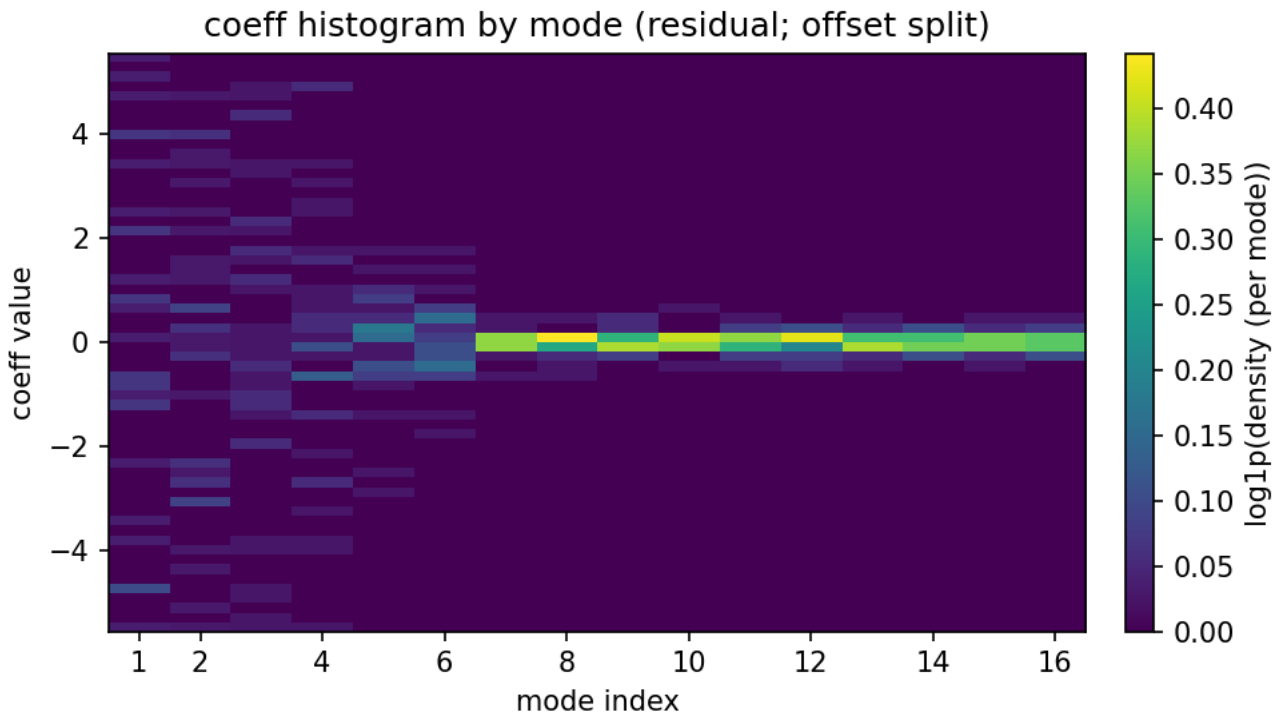
Decomposition (field reconstruction)

decompose(cfg)	method	status	rmse	r2	r2_k1	r2_k64	fit	n_req	k_req_r2_0.95
`pod_joint_em`	`pod_joint_em`	ok	7.975e-03	0.991907	0.512433	0.991907	571.8ms	4	4
`polar_fft`	`polar_fft`	ok	9.371e-03	0.989683			11.4ms	34	

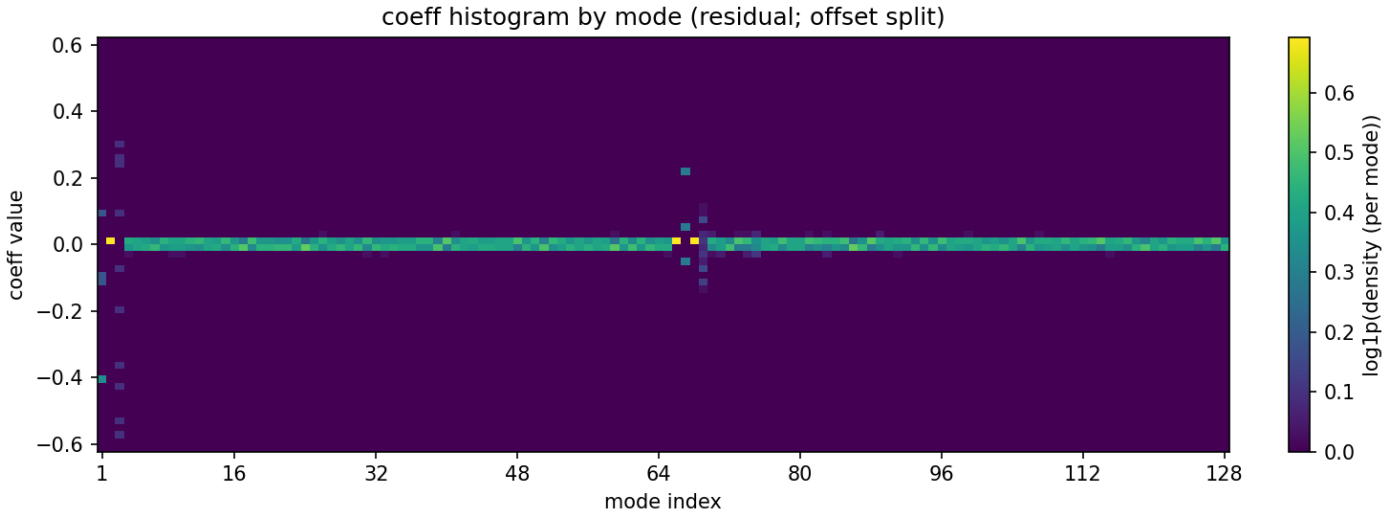
decompose(cfg)	method	status	rmse	r2	r2_k1	r2_k64	fit	n_req	k_req_r2_0.95
pseudo_zernike	pseudo_zernike	ok	1.148e-02	0.984511	0.235689	0.984511	8.2ms	2	4
graph_fourier_bench	graph_fourier	ok	1.259e-02	0.981307	0.235748	0.981307	143.6ms	6	8
rbf_expansion_k64	rbf_expansion	ok	1.284e-02	0.980485	-1.323432	0.980485	7.7ms	58	64
gappy_graph_fourier_bench	gappy_graph_fourier	ok	1.295e-02	0.980237	0.235735	0.980237	143.9ms	6	4
fourier_bessel_neumann	fourier_bessel	ok	1.419e-02	0.976025	0.235730	0.976025	5.7ms	5	4

Coefficient histograms by mode (per decomposer)

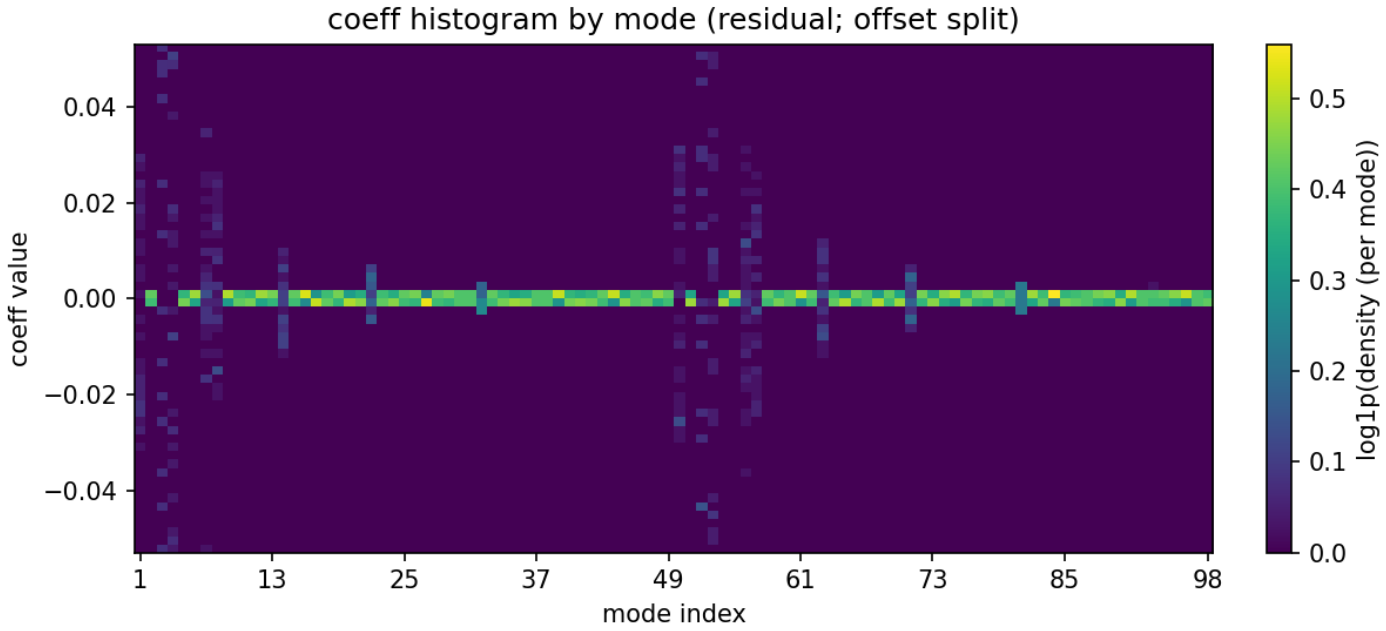
pod_joint_em (pod_joint_em)



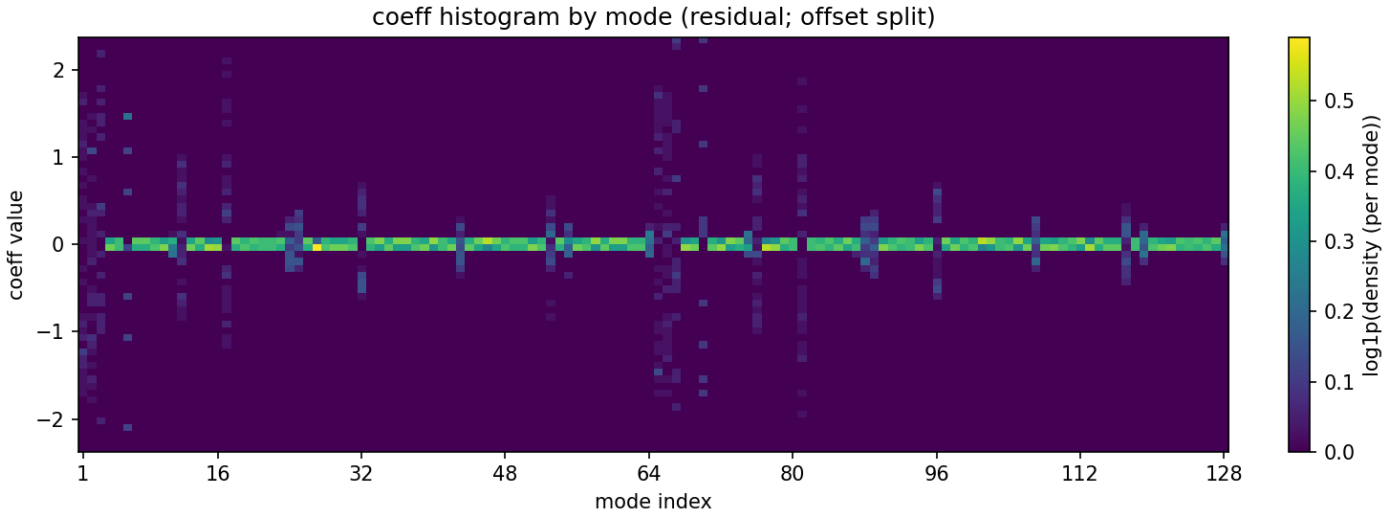
polar_fft (polar_fft)



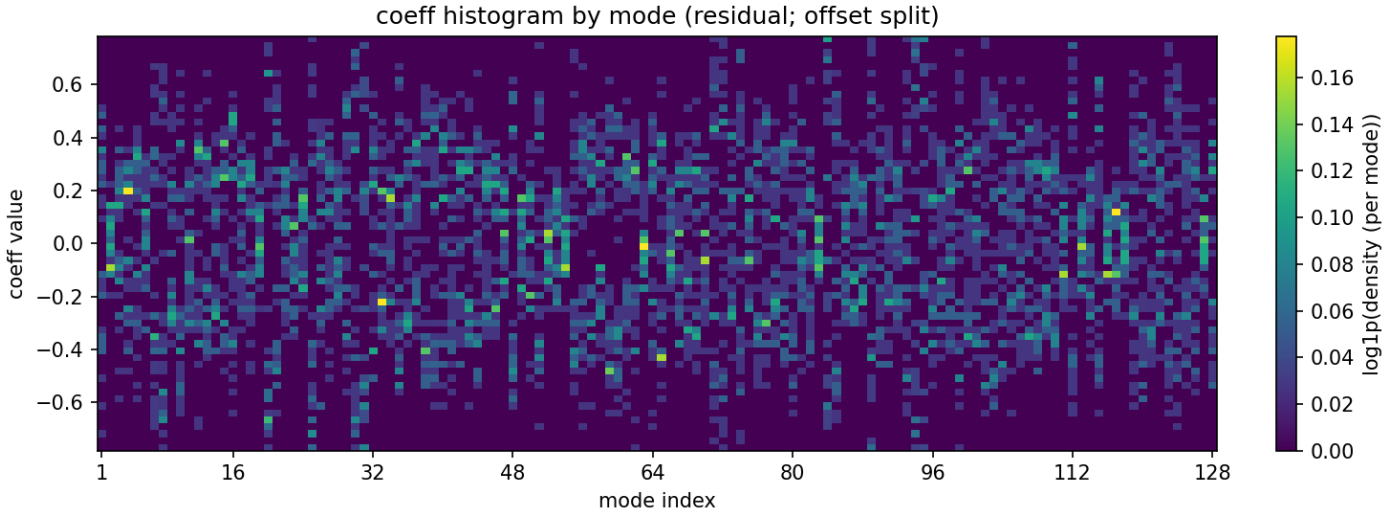
pseudo_zernike (pseudo_zernike)



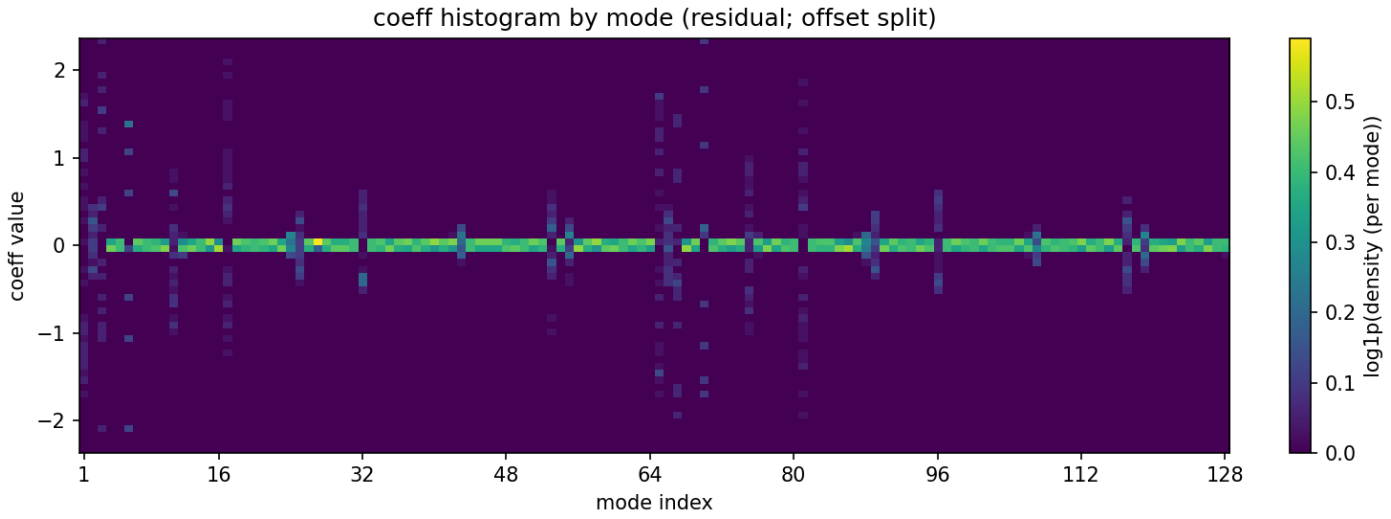
graph_fourier_bench (graph_fourier)



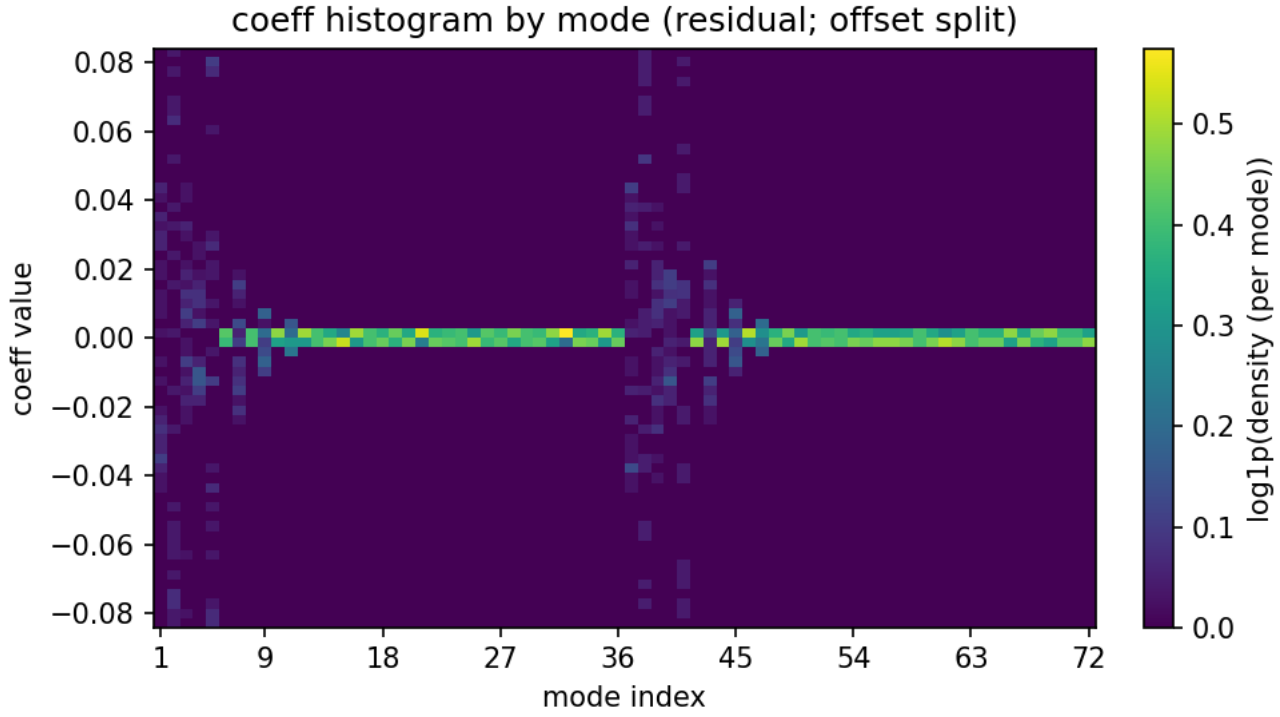
rbf_expansion_k64 (rbf_expansion)



gappy_graph_fourier_bench (gappy_graph_fourier)



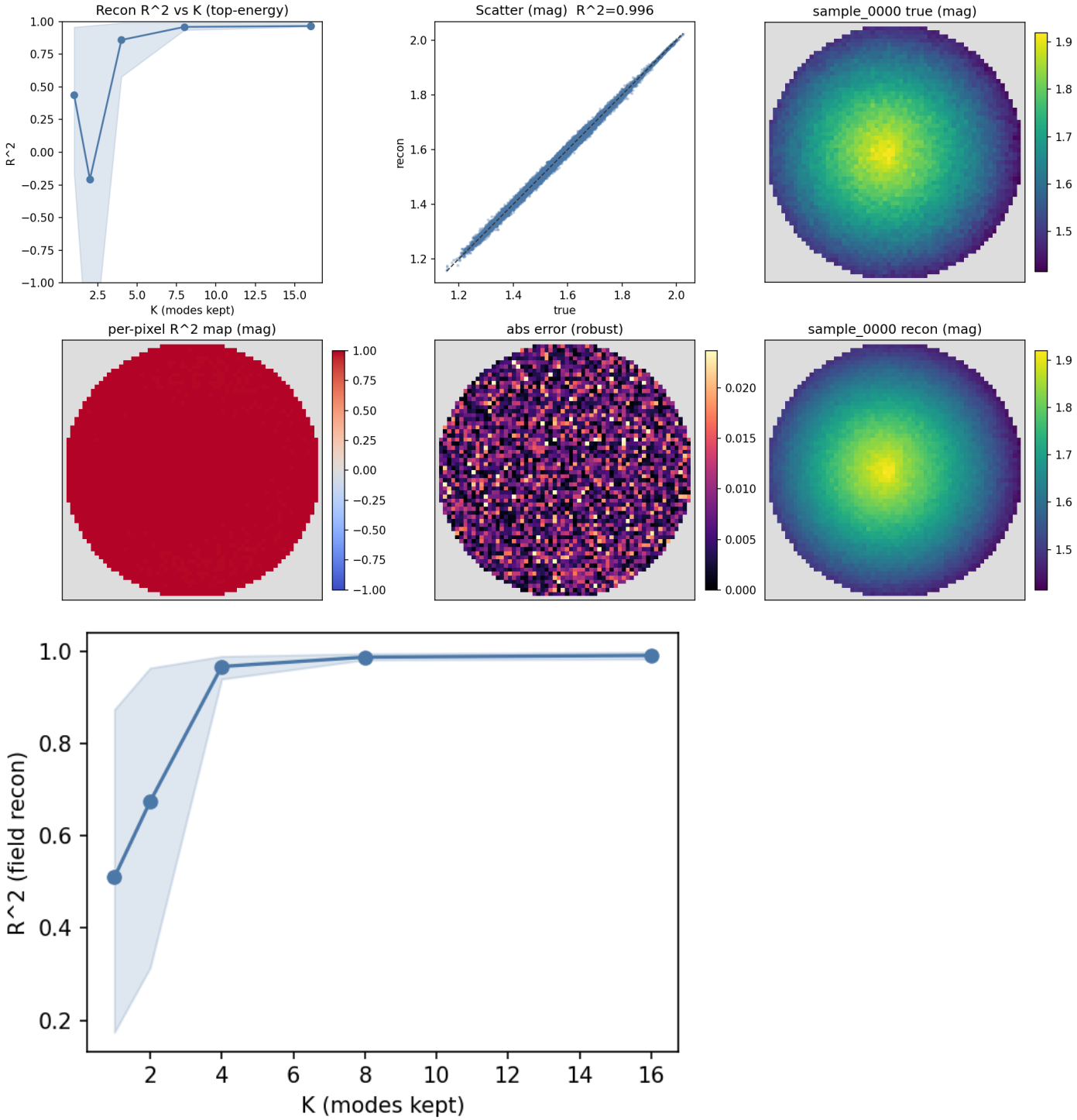
fourier_bessel_neumann (fourier_bessel)

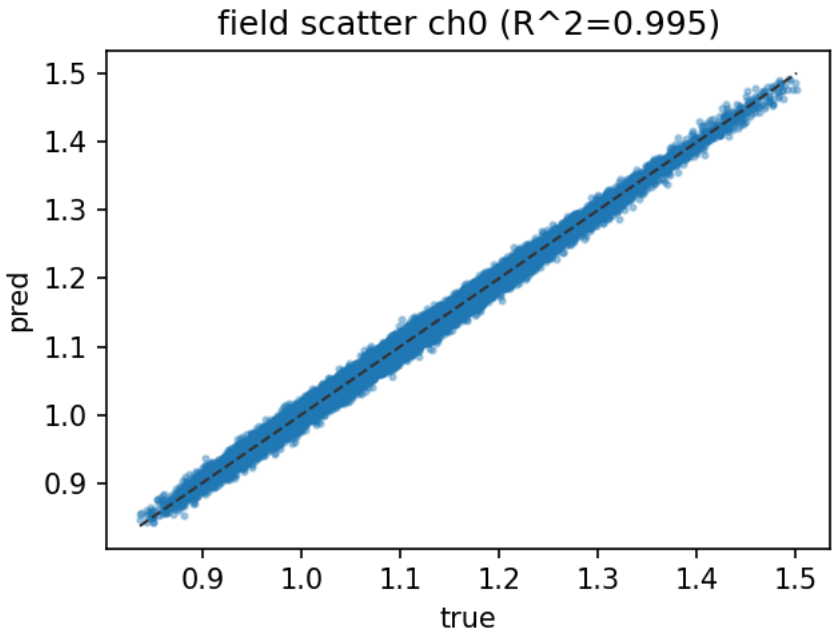


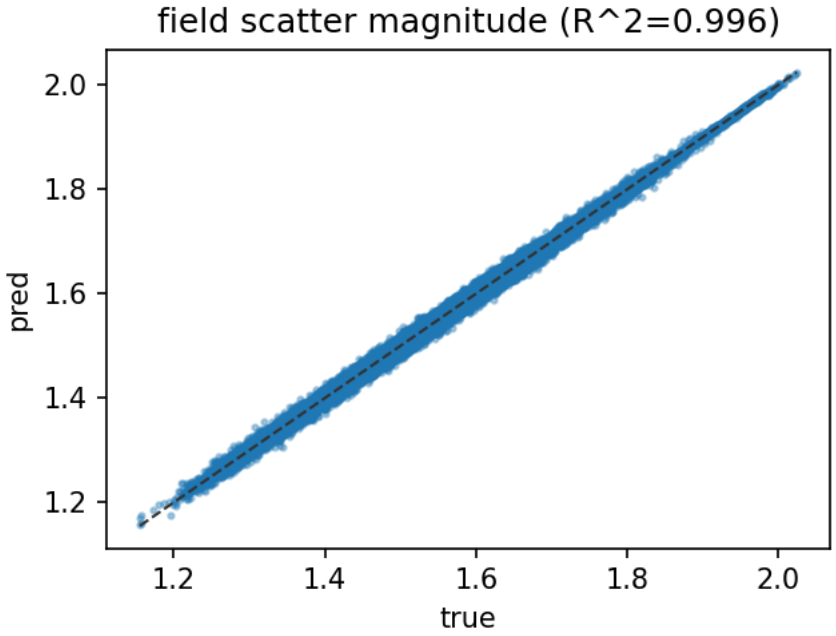
Compression leaderboard (Top-energy @K)

decompose(cfg)	method	r2_topk_k64	r2_topk_k16	k_req_r2_0.95	n_req
pod_joint_em	pod_joint_em	0.991907	0.991907	4	4
pseudo_zernike	pseudo_zernike	0.984511	0.984354	4	2
graph_fourier_bench	graph_fourier	0.981307	0.981009	8	6
rbf_expansion_k64	rbf_expansion	0.980485	-233.889839	64	58
gappy_graph_fourier_bench	gappy_graph_fourier	0.980237	0.979946	4	6

Key decomposition plots (best_rmse=pod_joint_em)


Recon R^2 vs K (top-energy)
R^2
K (modes kept)
Scatter (mag) R^2=0.996
recon
true
sample_0000 true (mag)
per-pixel R^2 map (mag)
abs error (robust)
sample_0000 recon (mag)
R^2 (field recon)
K (modes kept)





Train (cond -> coeff prediction)

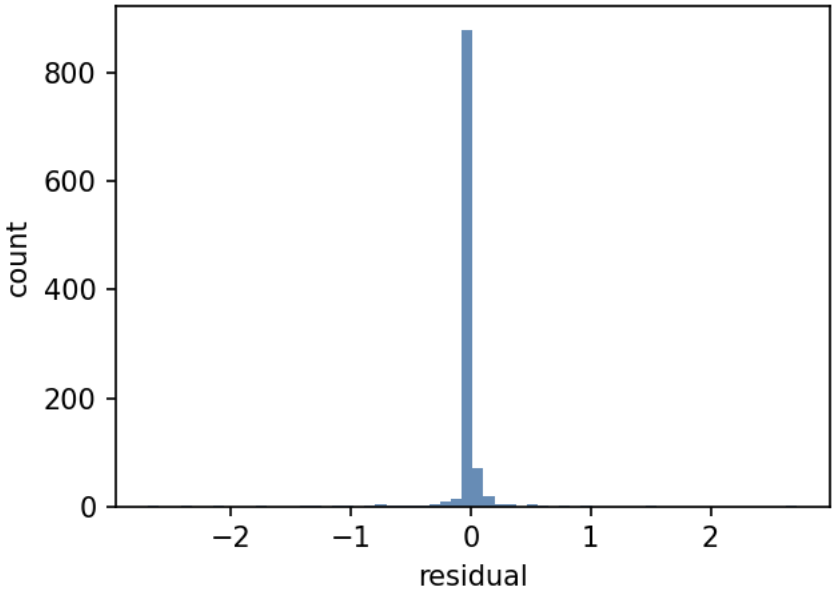
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2	fit
pseudo_zernike	pseudo_zernike	0.984511	ridge	ok	4.822e-03	0.891683	3.120e-02	0.892964	1.7m
fourier_bessel_neumann	fourier_bessel	0.976025	ridge	ok	7.175e-03	0.883284	3.133e-02	0.892700	1.7m
polar_fft	polar_fft	0.989683	ridge	ok	4.193e-02	0.774807	3.138e-02	0.891811	10.0
rbf_expansion_k64	rbf_expansion	0.980485	ridge	ok	1.437e-01	0.779620	3.106e-02	0.893047	2.0m
gappy_graph_fourier_bench	gappy_graph_fourier	0.980237	ridge	ok	2.224e-01	0.830318	3.095e-02	0.893211	2.0m
graph_fourier_bench	graph_fourier	0.981307	ridge	ok	2.240e-01	0.830207	3.113e-02	0.893147	1.9m
pod_joint_em	pod_joint_em	0.991907	ridge	ok	6.123e-01	0.826294	3.171e-02	0.890163	1.7m

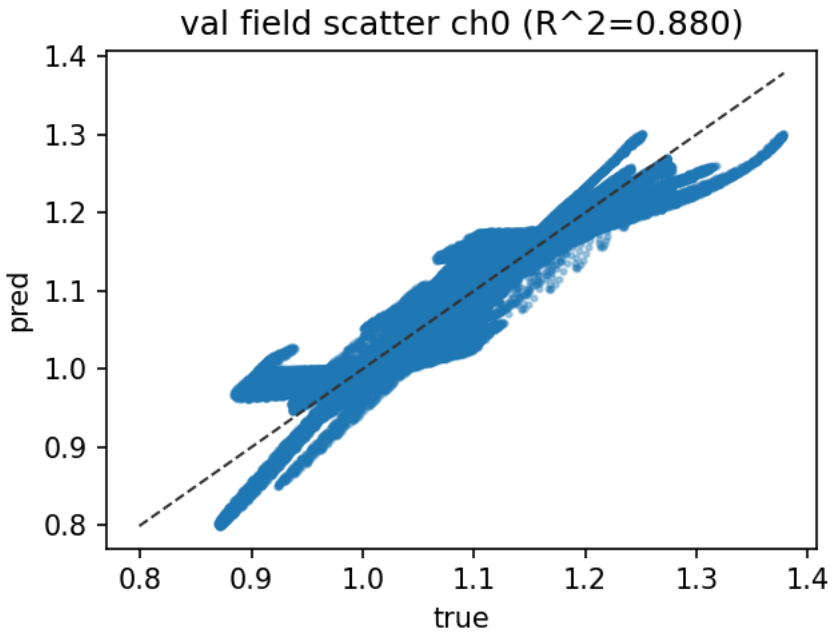
Train leaderboard (field-space)

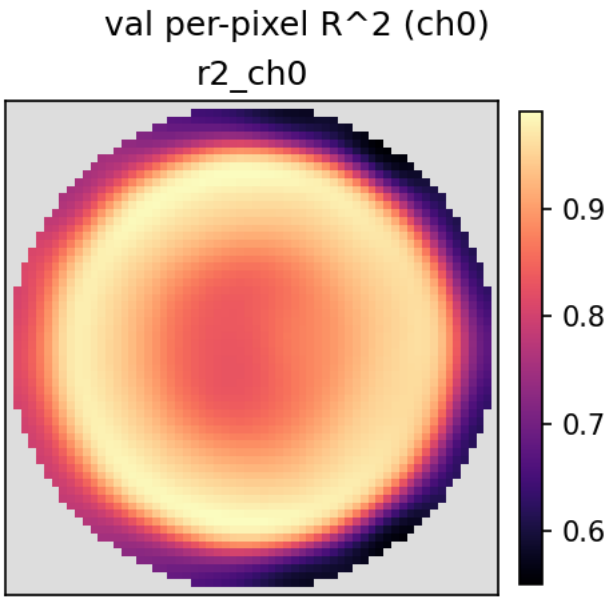
decompose(cfg)	method	model	val_field_rmse	val_field_r2	run
gappy_graph_fourier_bench	gappy_graph_fourier	ridge	3.095e-02	0.893211	run

decompose(cfg)	method	model	val_field_rmse	val_field_r2	run
rbf_expansion_k64	rbf_expansion	ridge	3.106e-02	0.893047	run
graph_fourier_bench	graph_fourier	ridge	3.113e-02	0.893147	run
pseudo_zernike	pseudo_zernike	ridge	3.120e-02	0.892964	run
fourier_bessel_neumann	fourier_bessel	ridge	3.133e-02	0.892700	run

Key train plots (best_field_eval=gappy_graph_fourier_bench)







annulus_vector

Problem setting

- domain: `annulus` (center=[0.0, 0.0], r_inner=0.35, r_outer=1.0)
- field: `vector`
- grid: `64x64`, x=[-1.0, 1.0], y=[-1.0, 1.0]
- cond: `(N,8)`
- mask: geometric domain mask (outside is 0-filled; evaluation uses inside only)

Highlights (auto)

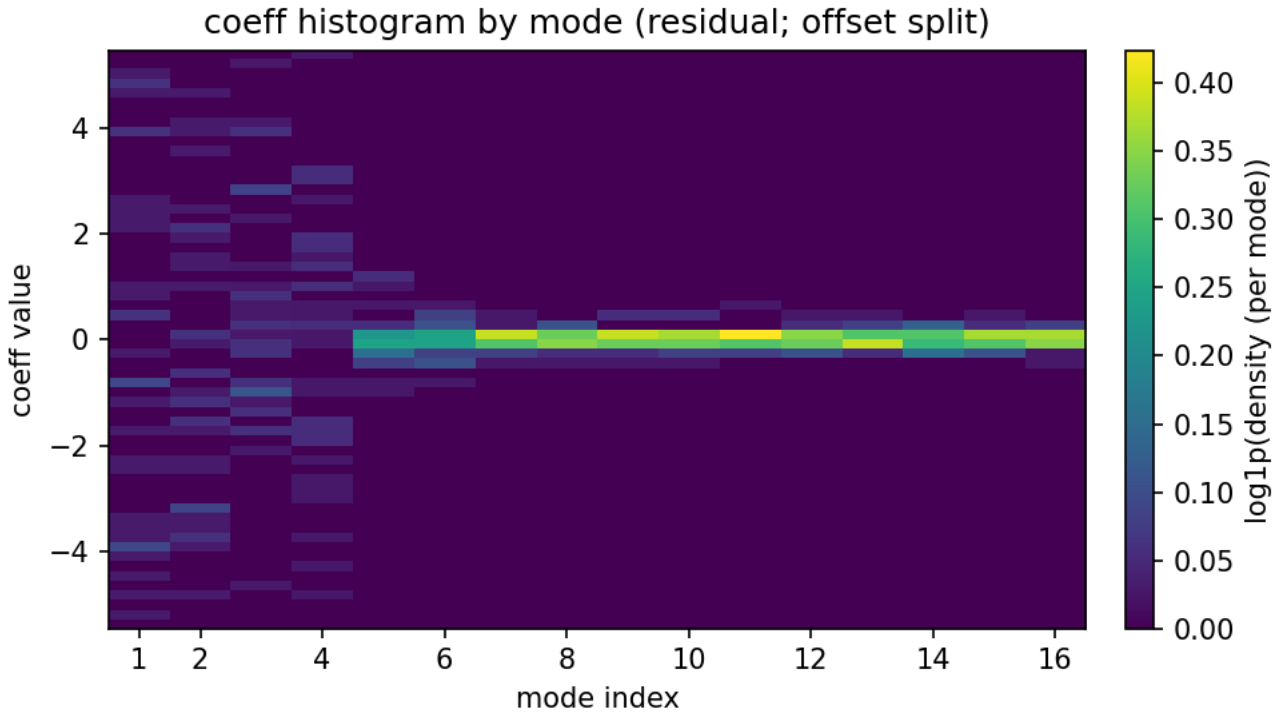
- decomposition: best full recon = `pod_joint_em` (`pod_joint_em`) (field_rmse=8.012e-03, field_r2=0.992121)
- decomposition: best compression proxy = `pod_joint_em` (`pod_joint_em`) (k_req_r2_0.95=4, r2_topk_k64=0.992121)
- decomposition: best top-energy@64 = `pod_joint_em` (`pod_joint_em`) (r2_topk_k64=0.992121, k_req_r2_0.95=4)
- train: best coeff-space = `polar_fft` (`polar_fft`) (`ridge`) (val_rmse=2.342e-02, val_r2=0.897195)
- train: best field-space = `rbf_expansion_k64` (`rbf_expansion`) (`ridge`) (val_field_rmse=2.516e-02, val_field_r2=0.926714)
- train: mismatch detected (best coeff-space != best field-space)

Decomposition (field reconstruction)

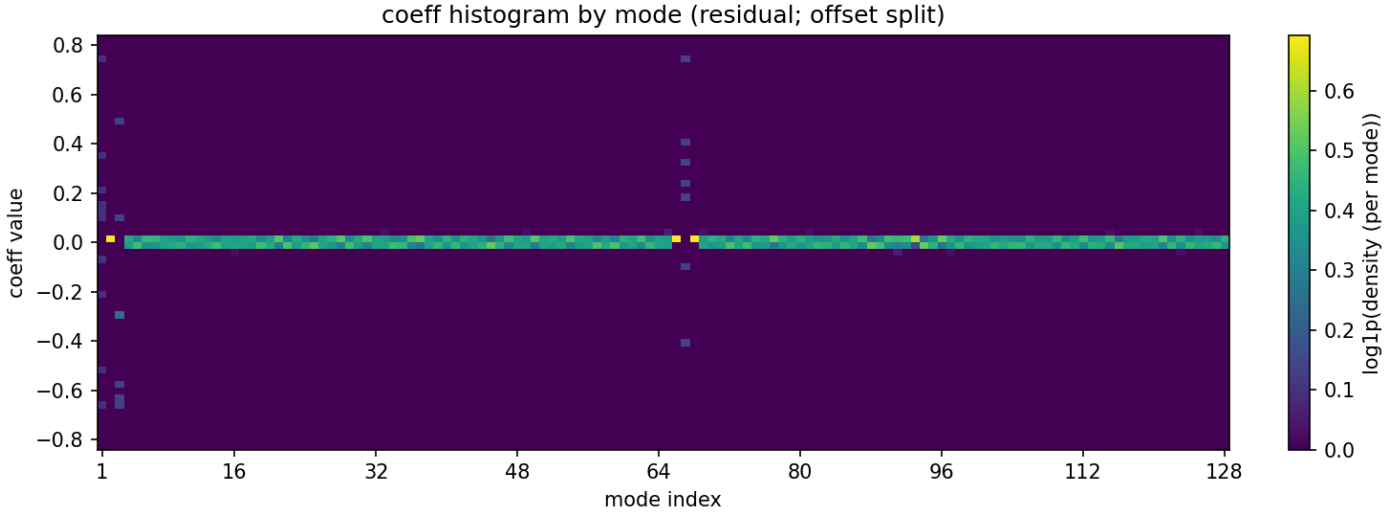
decompose(cfg)	method	status	rmse	r2	r2_k1	r2_k64	fit	n_req	k_req_r2_0.95
`pod_joint_em`	`pod_joint_em`	ok	8.012e-03	0.992121	0.553036	0.992121	350.6ms	4	4
`polar_fft`	`polar_fft`	ok	9.267e-03	0.990358			11.5ms	34	
`rbf_expansion_k64`	`rbf_expansion`	ok	1.165e-02	0.984744	-1.851114	0.984744	7.8ms	60	64
`graph_fourier_bench`	`graph_fourier`	ok	1.200e-02	0.983787	0.256411	0.983787	122.6ms	8	4
`gappy_graph_fourier_bench`	`gappy_graph_fourier`	ok	1.201e-02	0.983751	0.256411	0.983751	114.7ms	8	4

Coefficient histograms by mode (per decomposer)

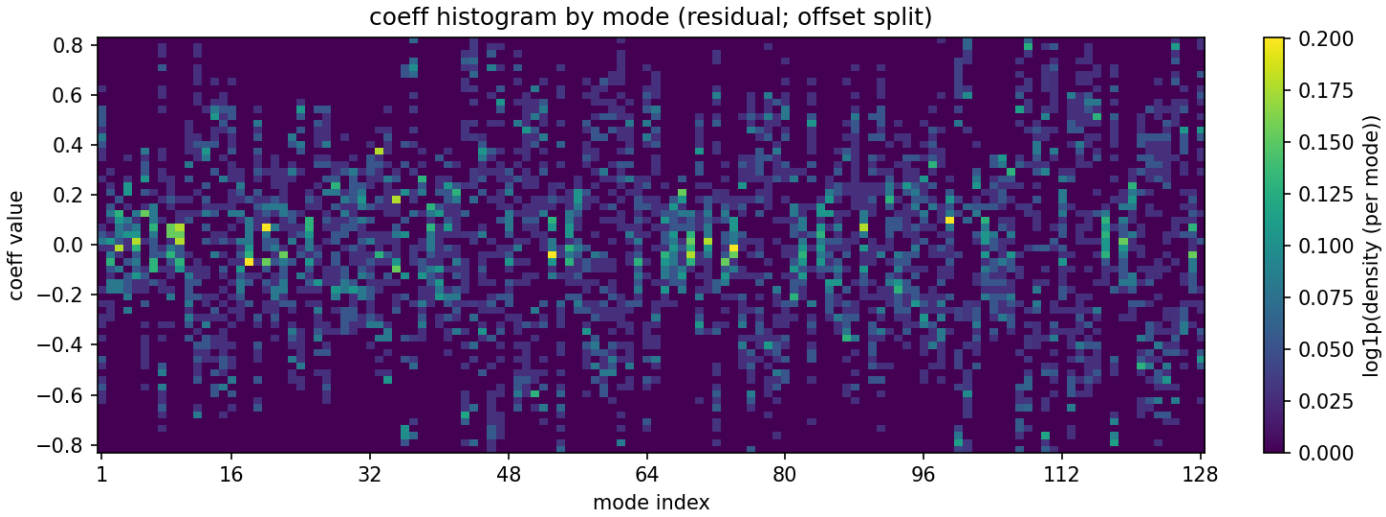
pod_joint_em (pod_joint_em)



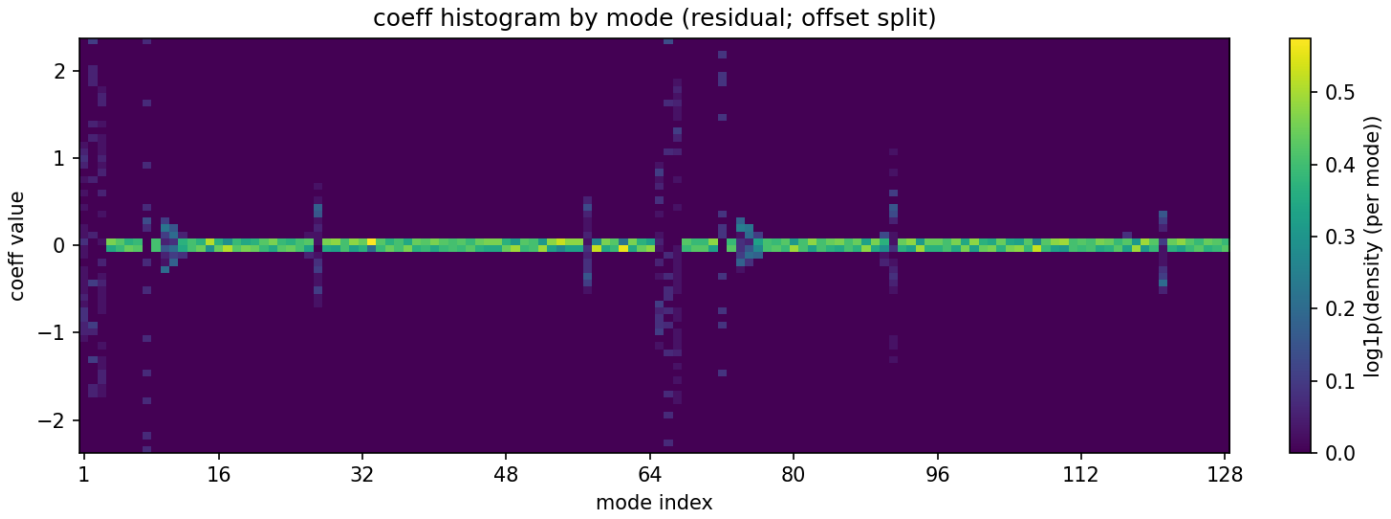
polar_fft (polar_fft)



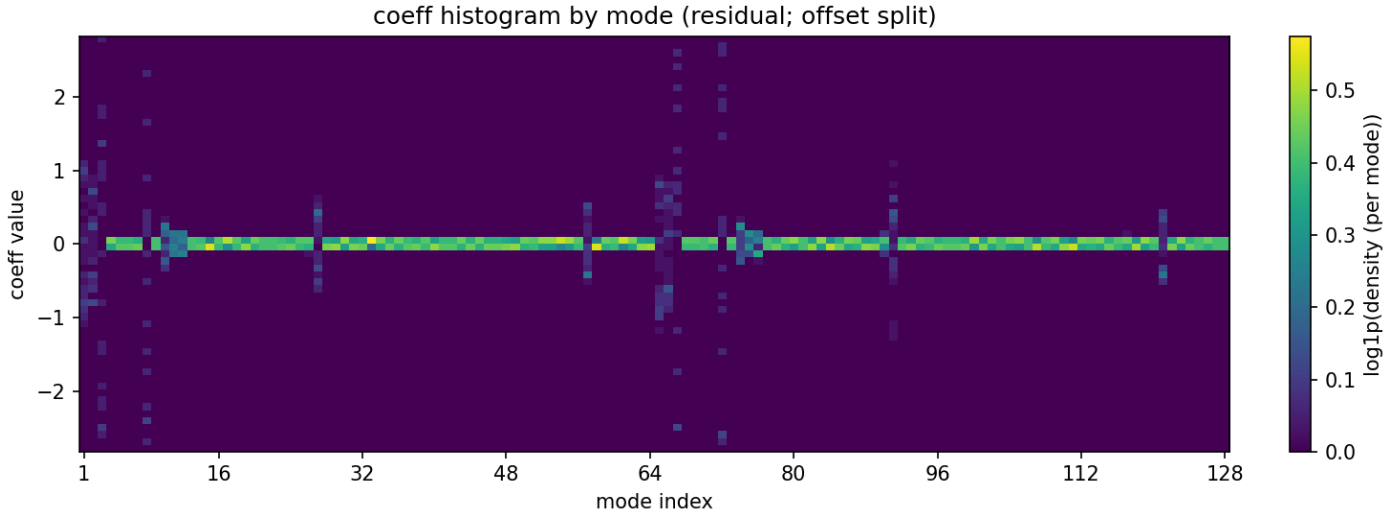
rbf_expansion_k64 (rbf_expansion)



graph_fourier_bench (graph_fourier)



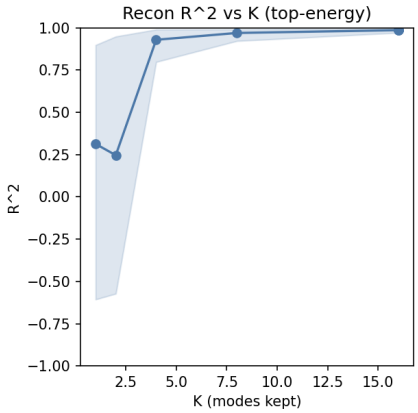
gappy_graph_fourier_bench (gappy_graph_fourier)

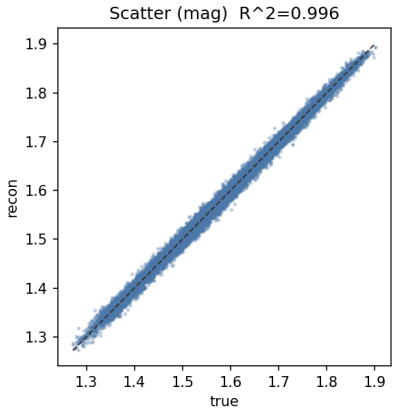


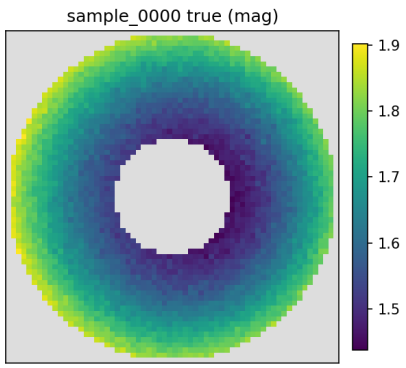
Compression leaderboard (Top-energy @K)

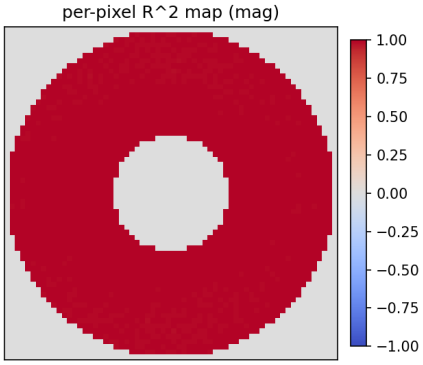
decompose(cfg)	method	r2_topk_k64	r2_topk_k16	k_req_r2_0.95	n_req
pod_joint_em	pod_joint_em	0.992121	0.992121	4	4
rbf_expansion_k64	rbf_expansion	0.984744	-9.931224	64	60
graph_fourier_bench	graph_fourier	0.983787	0.983533	4	8
gappy_graph_fourier_bench	gappy_graph_fourier	0.983751	0.983501	4	8

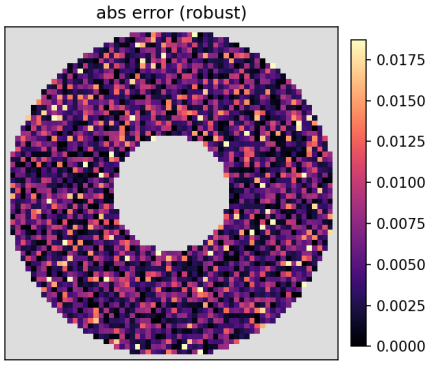
Key decomposition plots (best_rmse=pod_joint_em)

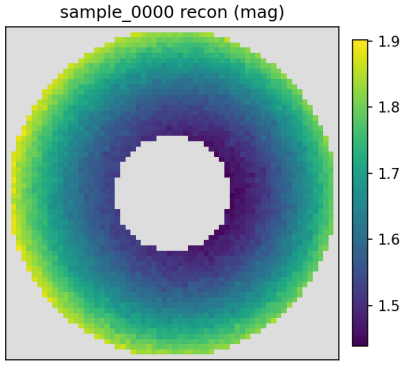

Recon R^2 vs K (top-energy)
R^2
K (modes kept)

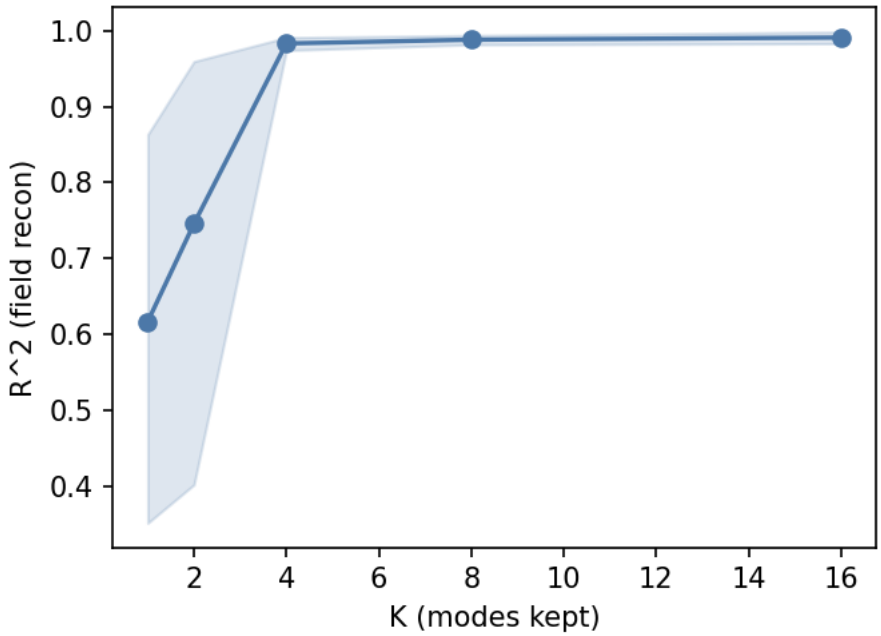

Scatter (mag) R^2=0.996
recon
true

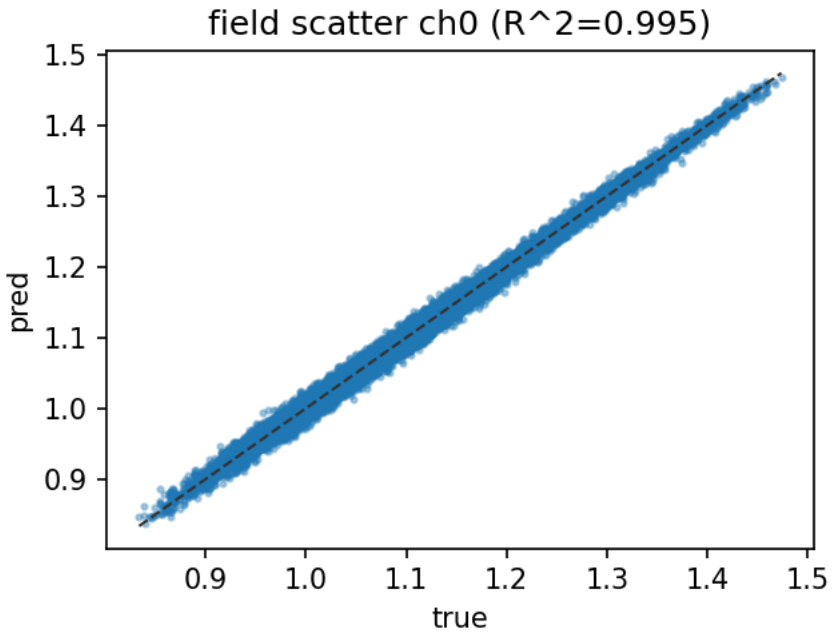

sample_0000 true (mag)

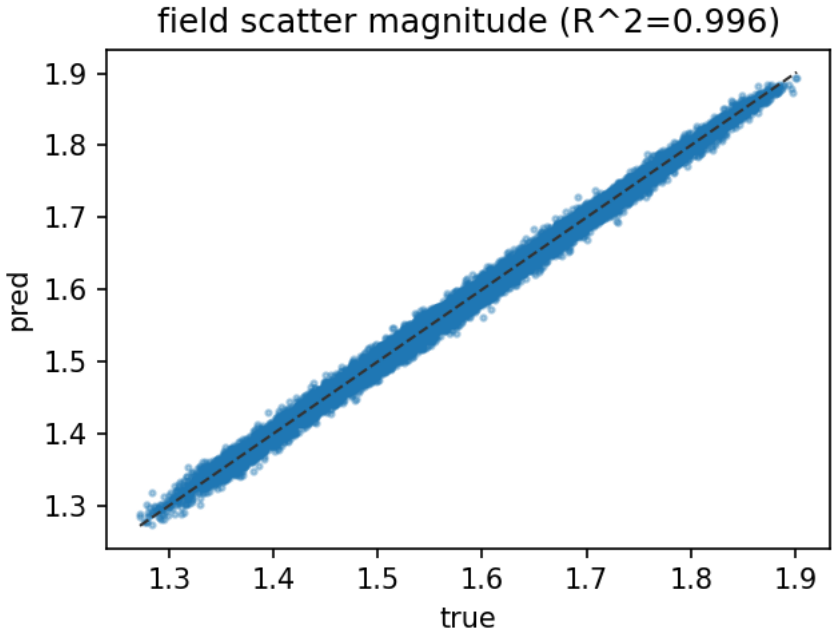

per-pixel R^2 map (mag)


abs error (robust)


sample_0000 recon (mag)


R^2 (field recon)
K (modes kept)





Train (cond -> coeff prediction)

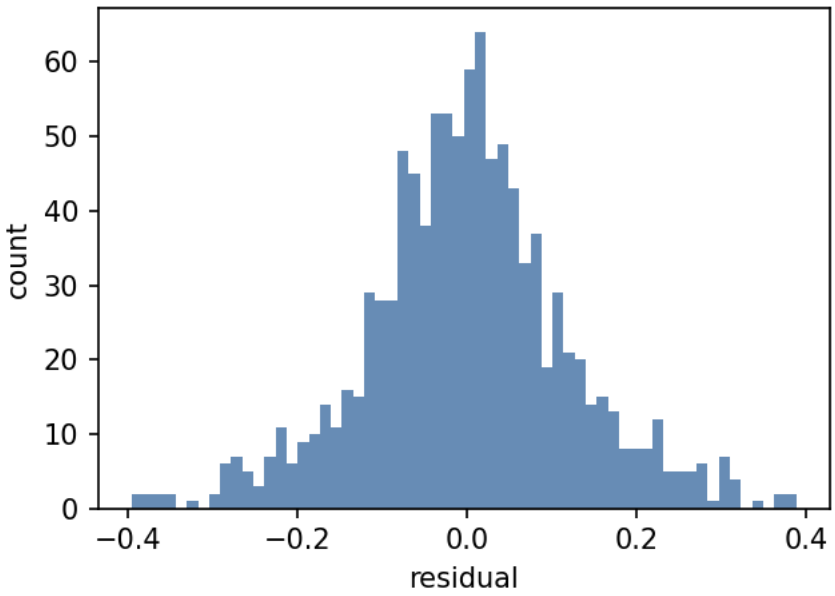
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2	fit
polar_fft	polar_fft	0.990358	ridge	ok	2.342e-02	0.897195	2.565e-02	0.923987	30.5
rbf_expansion_k64	rbf_expansion	0.984744	ridge	ok	1.209e-01	0.869495	2.516e-02	0.926714	2.0m
gappy_graph_fourier_bench	gappy_graph_fourier	0.983751	ridge	ok	1.638e-01	0.898751	2.516e-02	0.926611	1.8m
graph_fourier_bench	graph_fourier	0.983787	ridge	ok	1.639e-01	0.898757	2.518e-02	0.926576	1.9m
pod_joint_em	pod_joint_em	0.992121	ridge	ok	4.548e-01	0.893233	2.601e-02	0.922609	2.8m

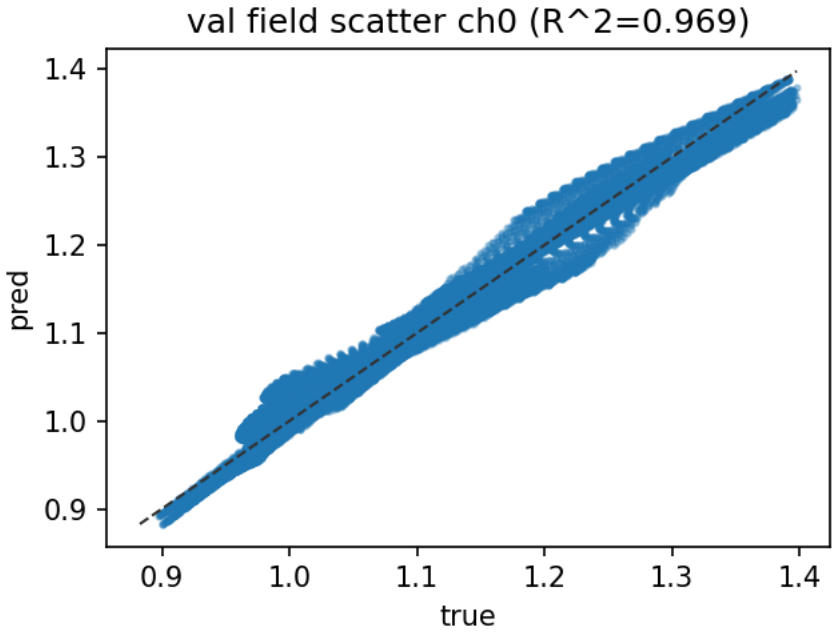
Train leaderboard (field-space)

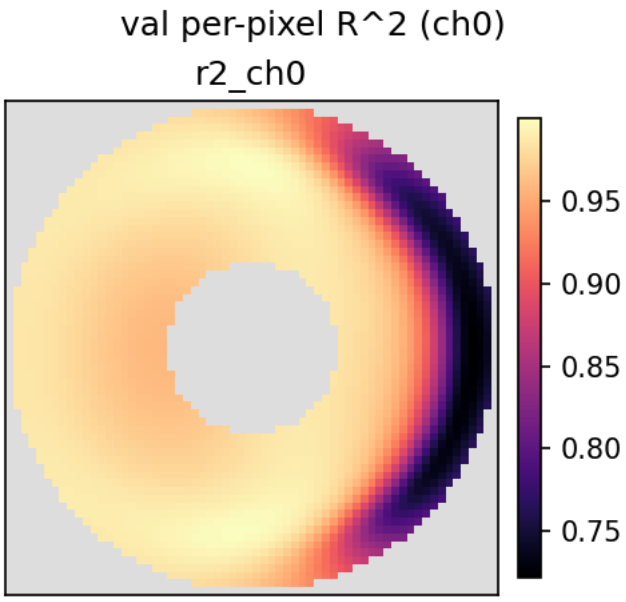
decompose(cfg)	method	model	val_field_rmse	val_field_r2	run
rbf_expansion_k64	rbf_expansion	ridge	2.516e-02	0.926714	run
gappy_graph_fourier_bench	gappy_graph_fourier	ridge	2.516e-02	0.926611	run
graph_fourier_bench	graph_fourier	ridge	2.518e-02	0.926576	run

decompose(cfg)	method	model	val_field_rmse	val_field_r2	run
polar_fft	polar_fft	ridge	2.565e-02	0.923987	run
pod_joint_em	pod_joint_em	ridge	2.601e-02	0.922609	run

Key train plots (best_field_eval=rbf_expansion_k64)







arbitrary_mask_vector

Problem setting

- domain: `arbitrary_mask` (mask=domain_mask.npy)
- field: `vector`
- grid: `64x64`, x=[-1.0, 1.0], y=[-1.0, 1.0]
- cond: `(N,8)`
- mask: fixed irregular mask (`domain_mask.npy`; evaluation uses mask==true only)

Highlights (auto)

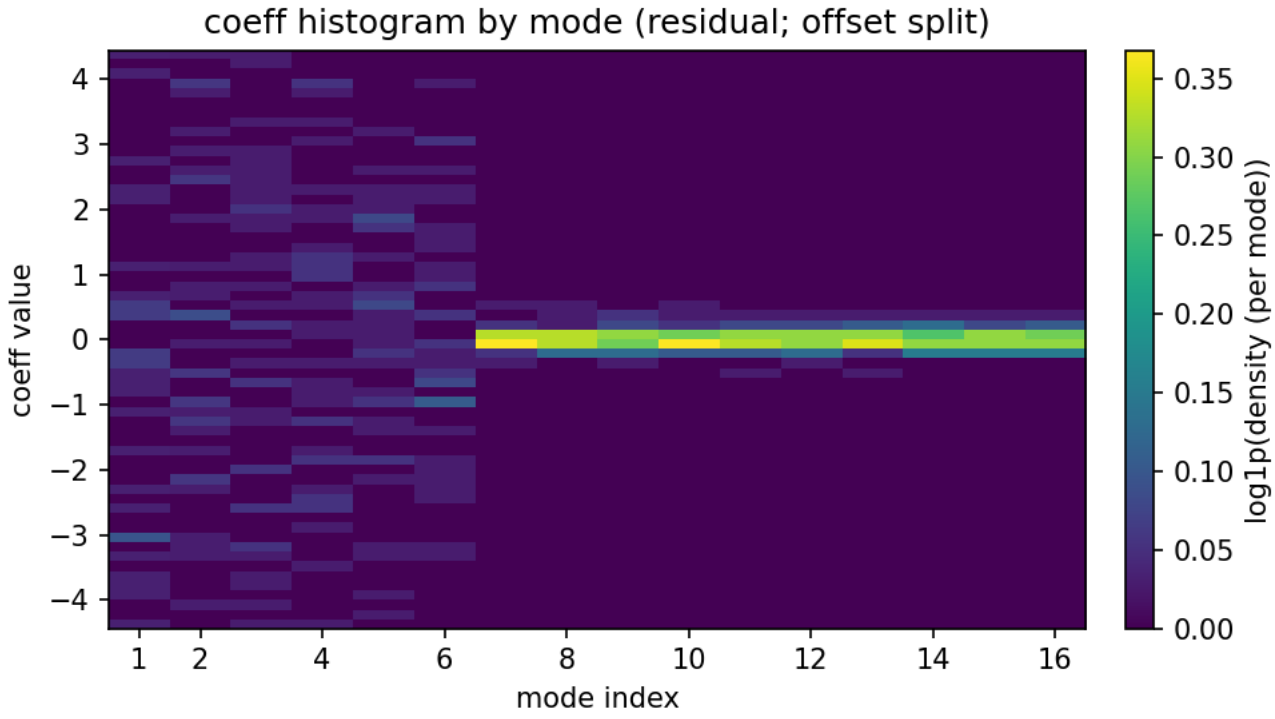
- decomposition: best full recon = `pod_joint_em` (`pod_joint_em`) (field_rmse=8.016e-03, field_r2=0.993009)
- decomposition: best compression proxy = `pod_joint_em` (`pod_joint_em`) (k_req_r2_0.95=8, r2_topk_k64=0.993009)
- decomposition: best top-energy@64 = `pod_joint_em` (`pod_joint_em`) (r2_topk_k64=0.993009, k_req_r2_0.95=8)
- train: best coeff-space = `gappy_graph_fourier_bench` (`gappy_graph_fourier`) (`ridge`) (val_rmse=1.331e-01, val_r2=0.921325)
- train: best field-space = `rbf_expansion_k64` (`rbf_expansion`) (`ridge`) (val_field_rmse=2.061e-02, val_field_r2=0.953578)
- train: mismatch detected (best coeff-space != best field-space)

Decomposition (field reconstruction)

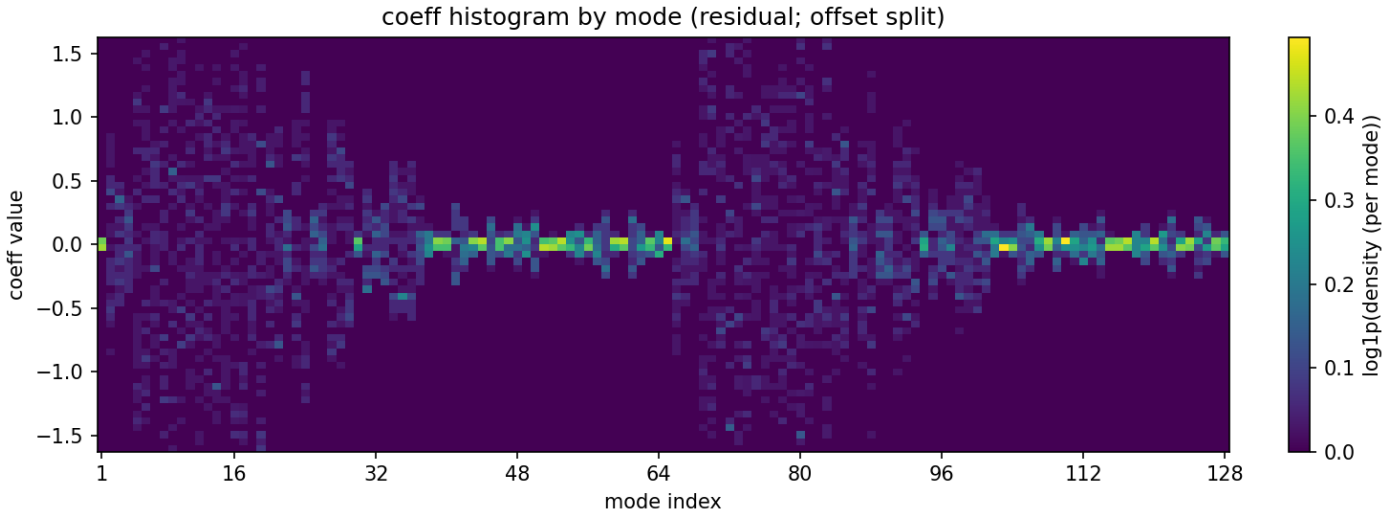
decompose(cfg)	method	status	rmse	r2	r2_k1	r2_k64	fit	n_req	k_req_r2_0.95
`pod_joint_em`	`pod_joint_em`	ok	8.016e-03	0.993009	0.508376	0.993009	345.1ms	5	8
`gappy_graph_fourier_bench`	`gappy_graph_fourier`	ok	1.249e-02	0.983967	0.327570	0.983967	110.8ms	24	32
`rbf_expansion_k64`	`rbf_expansion`	ok	1.386e-02	0.980067	-78.853494	0.980067	5.7ms	60	64

Coefficient histograms by mode (per decomposer)

pod_joint_em (pod_joint_em)



gappy_graph_fourier_bench (gappy_graph_fourier)



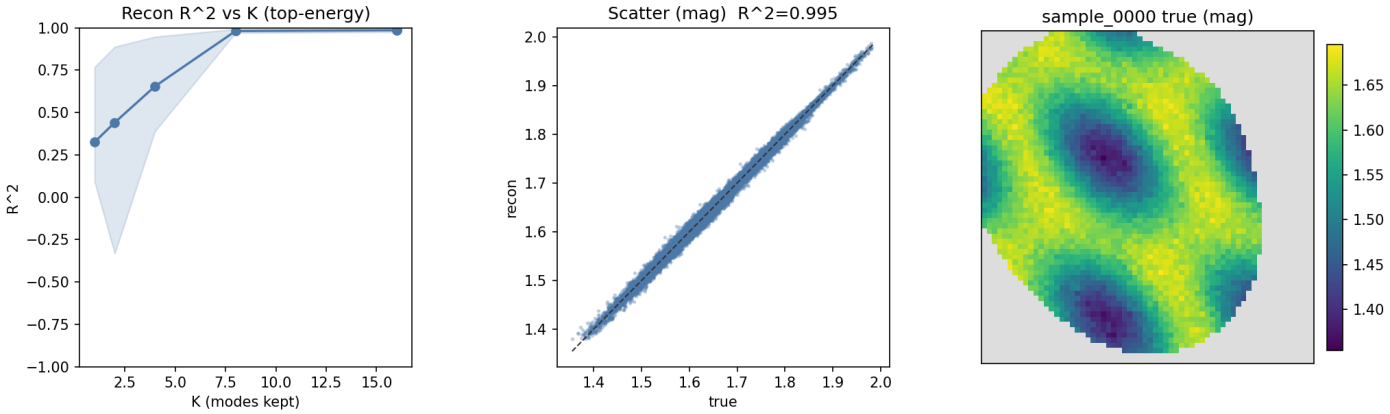
rbf_expansion_k64 (rbf_expansion)

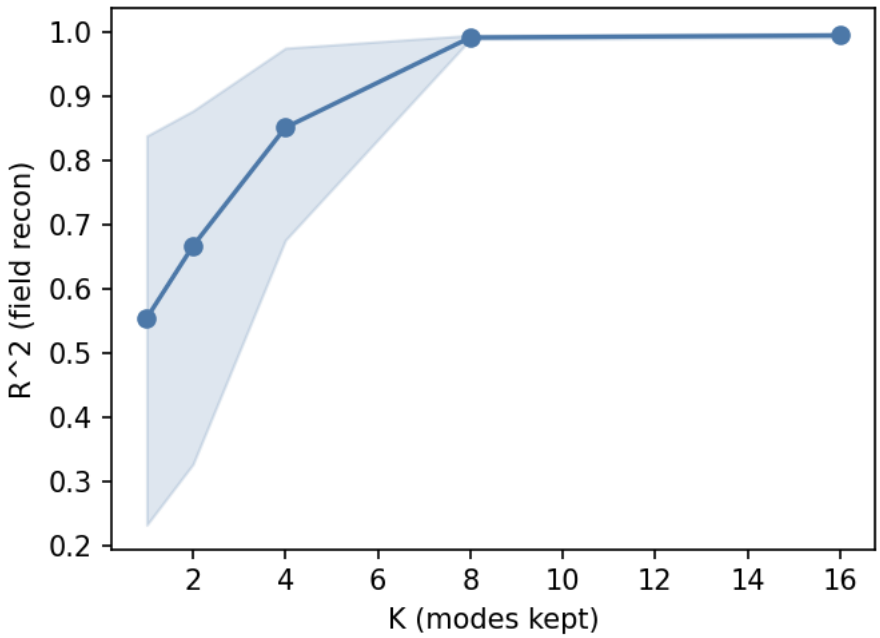
Compression leaderboard (Top-energy @K)

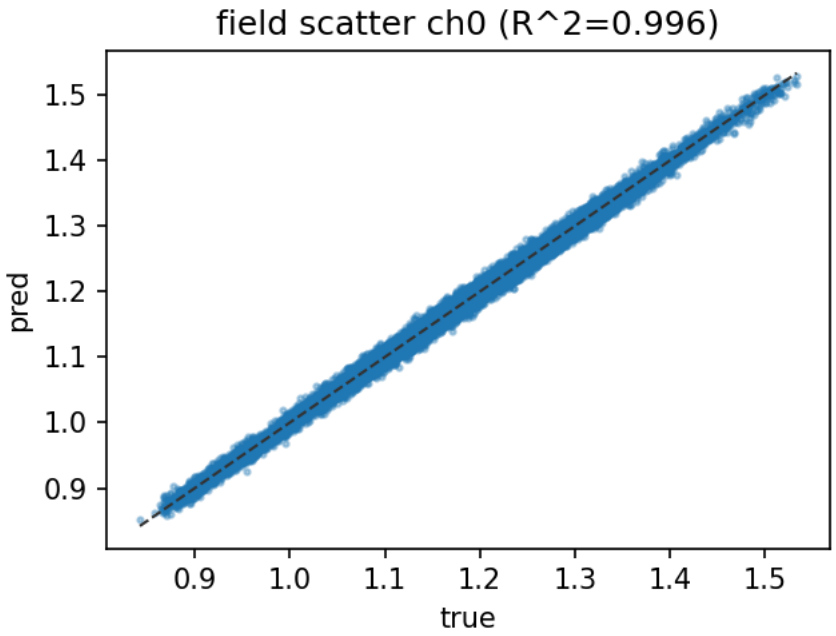
decompose(cfg)	method	r2_topk_k64	r2_topk_k16	k_req_r2_0.95	n_req

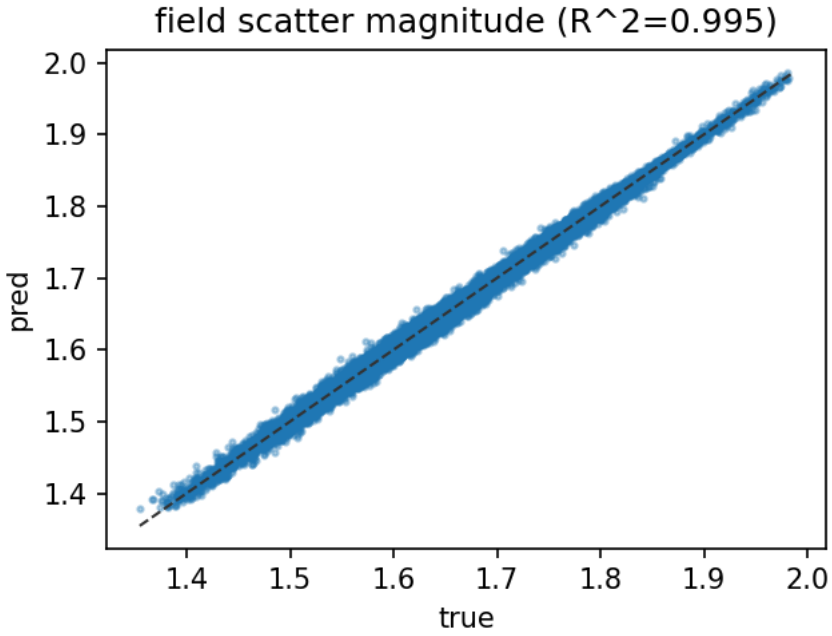
decompose(cfg)	method	r2_topk_k64	r2_topk_k16	k_req_r2_0.95	n_req
pod_joint_em	pod_joint_em	0.993009	0.993009	8	5
gappy_graph_fourier_bench	gappy_graph_fourier	0.983967	0.884176	32	24
rbf_expansion_k64	rbf_expansion	0.980067	-864.877811	64	60

Key decomposition plots (best_rmse=pod_joint_em)









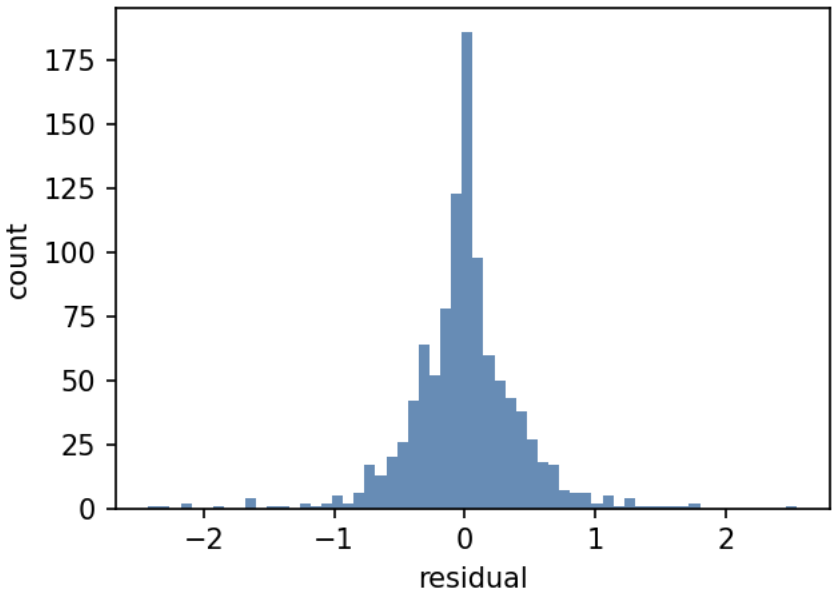
Train (cond -> coeff prediction)

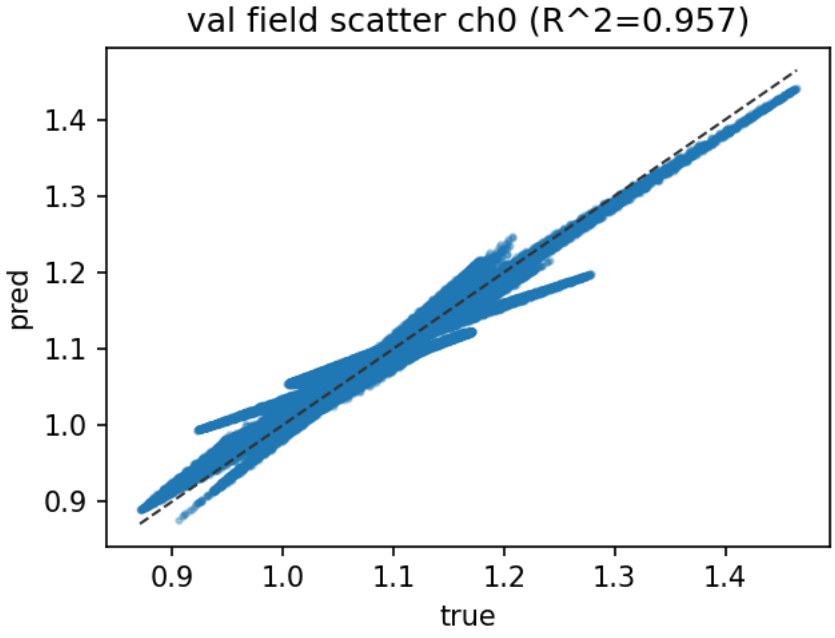
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2	fit
gappy_graph_fourier_bench	gappy_graph_fourier	0.983967	ridge	ok	1.331e-01	0.921325	2.091e-02	0.953054	2.1m
pod_joint_em	pod_joint_em	0.993009	ridge	ok	3.714e-01	0.916491	2.170e-02	0.950205	1.7m
rbf_expansion_k64	rbf_expansion	0.980067	ridge	ok	4.312e-01	0.904053	2.061e-02	0.953578	2.2m

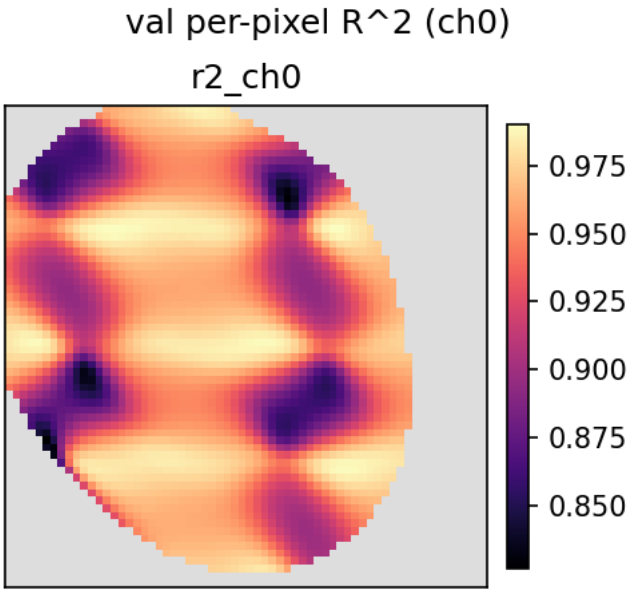
Train leaderboard (field-space)

decompose(cfg)	method	model	val_field_rmse	val_field_r2	run
rbf_expansion_k64	rbf_expansion	ridge	2.061e-02	0.953578	run
gappy_graph_fourier_bench	gappy_graph_fourier	ridge	2.091e-02	0.953054	run
pod_joint_em	pod_joint_em	ridge	2.170e-02	0.950205	run

Key train plots (best_field_eval=rbf_expansion_k64)







sphere_grid_vector

Problem setting

- domain: `sphere_grid` (n_lat=18, n_lon=36, lon_range=[-180.0, 170.0])

- field: `vector`
- grid: `18x36`, x=[-180.0, 170.0], y=[-90.0, 90.0]
- cond: `(N,8)`
- mask: all-valid (no mask)

Highlights (auto)

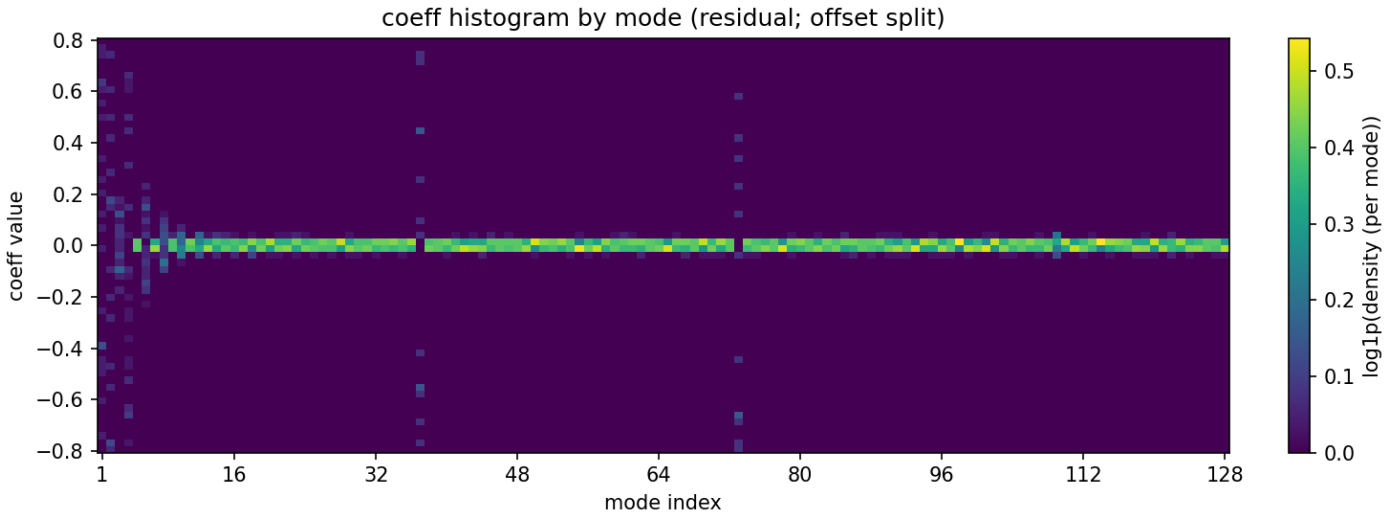
- decomposition: best full recon = `dct2` (`dct2`) (field_rmse=3.300e-09, field_r2=1.000000)
- decomposition: best compression proxy = `graph_fourier_bench` (`graph_fourier`) (k_req_r2_0.95=8, r2_topk_k64=0.983974)
- decomposition: best top-energy@64 = `graph_fourier_bench` (`graph_fourier`) (r2_topk_k64=0.983974, k_req_r2_0.95=8)
- train: best coeff-space = `spherical_harmonics_scipy_bench` (`spherical_harmonics`) (`ridge`) (val_rmse=9.733e-03, val_r2=0.897570)
- train: best field-space = `spherical_slepian_scipy` (`spherical_slepian`) (`ridge`) (val_field_rmse=1.158e-02, val_field_r2=0.964466)
- train: mismatch detected (best coeff-space != best field-space)

Decomposition (field reconstruction)

decompose(cfg)	method	status	rmse	r2	r2_k1	r2_k64	fit	n_req	k_req_r2_0
`dct2`	`dct2`	ok	3.300e-09	1.000000	0.261890	0.985590	1.4ms	73	
`graph_fourier_bench`	`graph_fourier`	ok	1.200e-02	0.983974	0.261890	0.983974	39.6ms	9	8
`spherical_harmonics_scipy_bench`	`spherical_harmonics`	ok	2.005e-02	0.955897	0.166197	0.955897	1.5ms	5	8
`spherical_slepian_scipy`	`spherical_slepian`	ok	7.484e-02	0.377093	0.174895	0.377093	6.2ms	7	

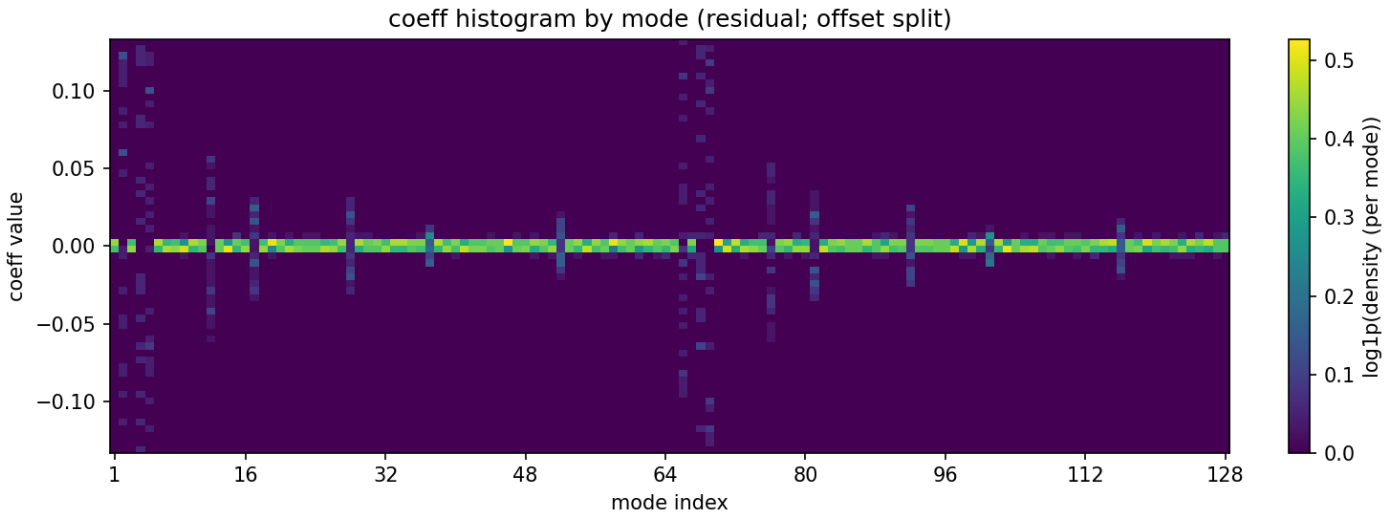
Coefficient histograms by mode (per decomposer)

dct2 (dct2)



graph_fourier_bench (graph_fourier)

spherical_harmonics_scipy_bench (spherical_harmonics)

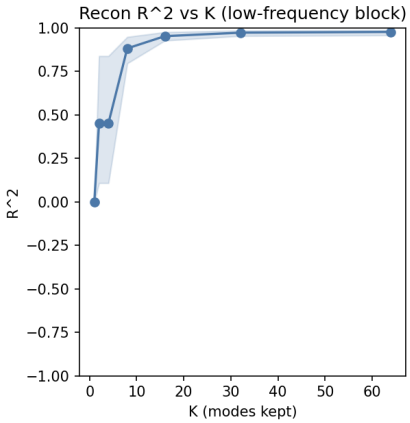


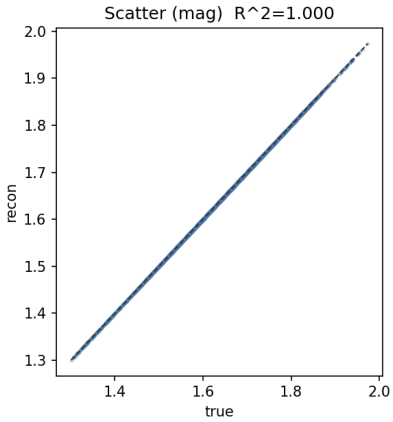
spherical_slepian_scipy (spherical_slepian)

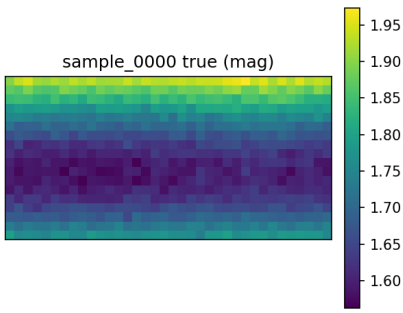
Compression leaderboard (Top-energy @K)

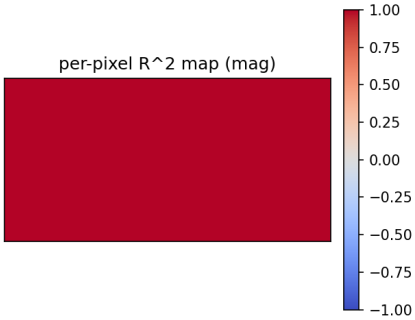
decompose(cfg)	method	r2_topk_k64	r2_topk_k16	k_req_r2_0.95	n_req
graph_fourier_bench	graph_fourier	0.983974	0.982908	8	9
spherical_harmonics_scipy_bench	spherical_harmonics	0.955897	0.954760	8	5
spherical_slepian_scipy	spherical_slepian	0.377093	0.377093		7

Key decomposition plots (best_rmse=dct2)

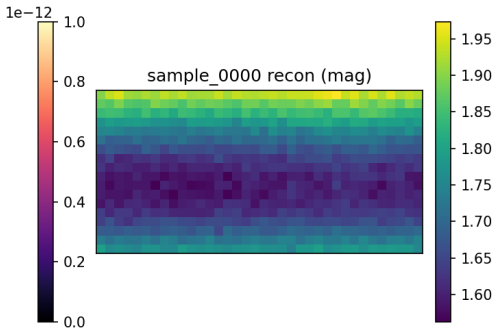

Recon R^2 vs K (low-frequency block)
R^2
K (modes kept)

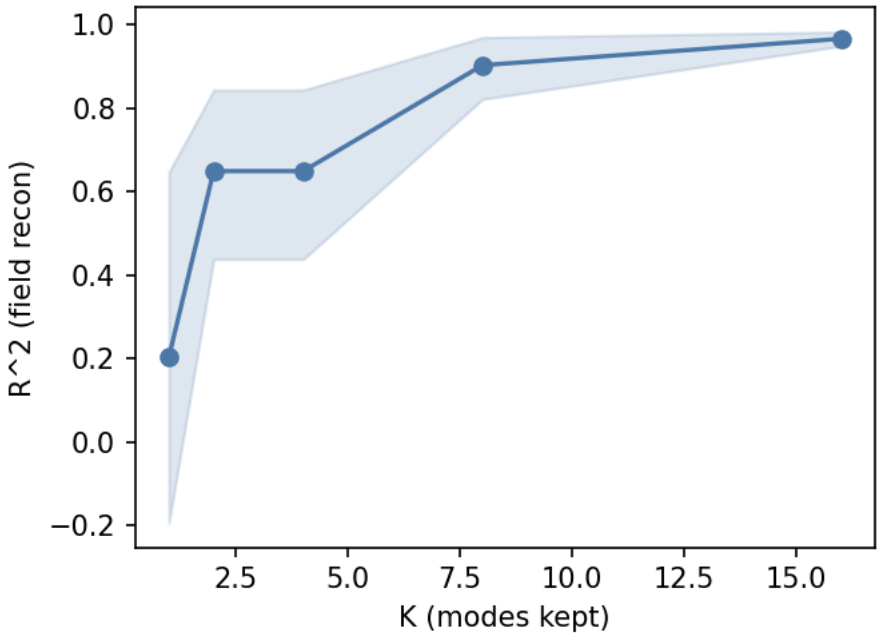

Scatter (mag) R^2=1.000
recon
true

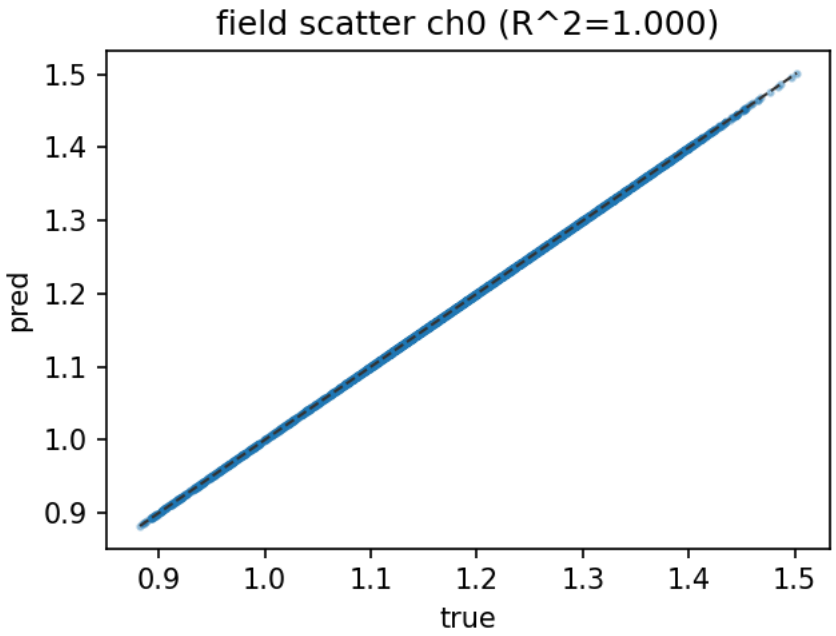

sample_0000 true (mag)

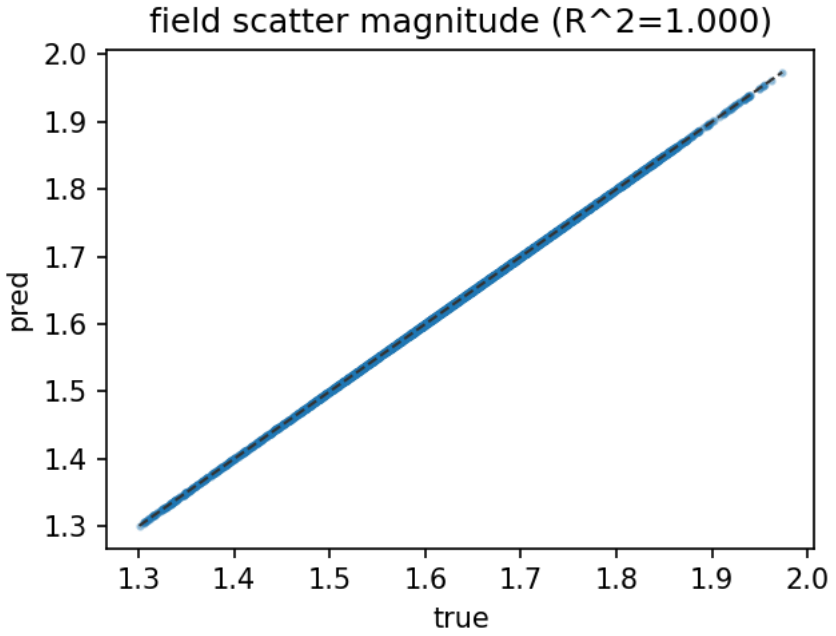

per-pixel R^2 map (mag)


abs error (robust)


1e−12
sample_0000 recon (mag)


R^2 (field recon)
K (modes kept)





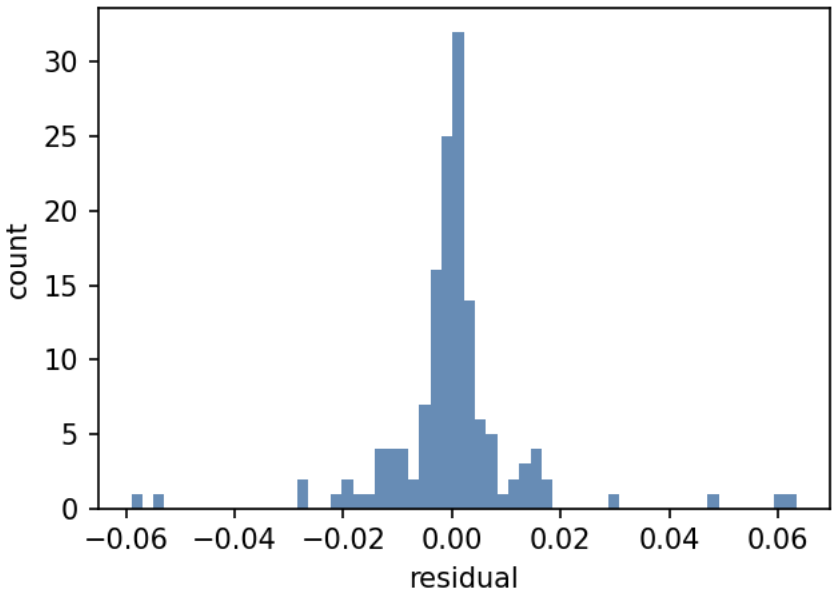
Train (cond -> coeff prediction)

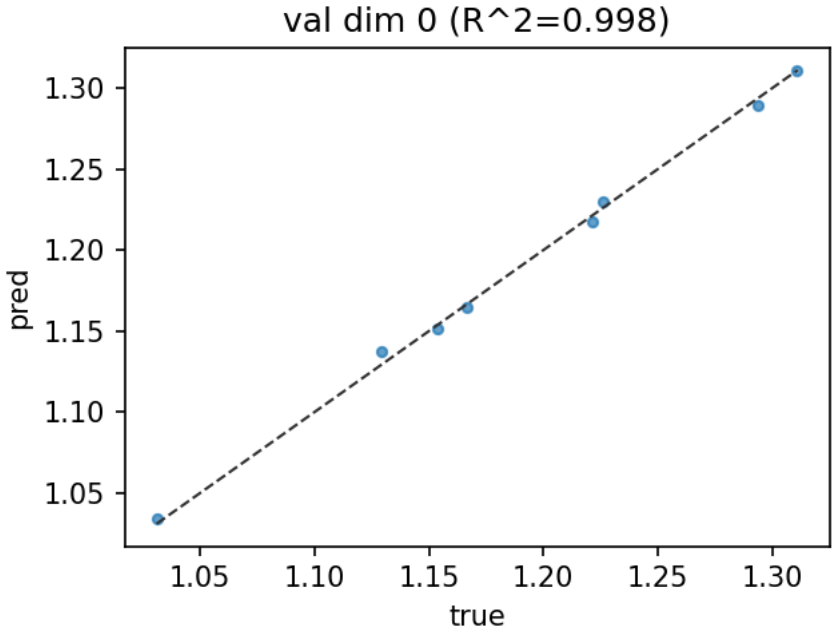
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2
spherical_harmonics_scipy_bench	spherical_harmonics	0.955897	ridge	ok	9.733e-03	0.897570	2.354e-02	0.929476
spherical_slepian_scipy	spherical_slepian	0.377093	ridge	ok	1.324e-02	0.924401	1.158e-02	0.964466
dct2	dct2	1.000000	ridge	ok	2.691e-02	0.895908	2.653e-02	0.919973
graph_fourier_bench	graph_fourier	0.983974	ridge	ok	7.946e-02	0.907406	2.473e-02	0.928265

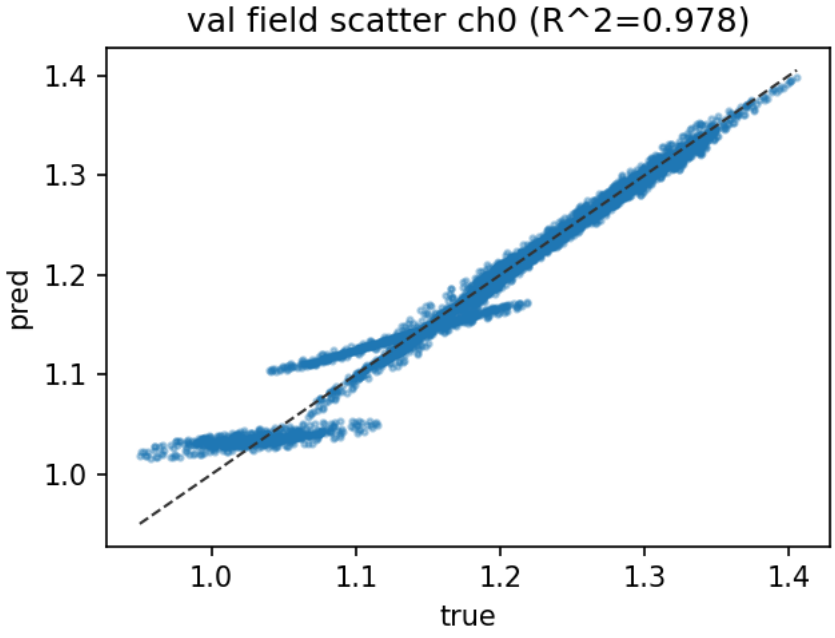
Train leaderboard (field-space)

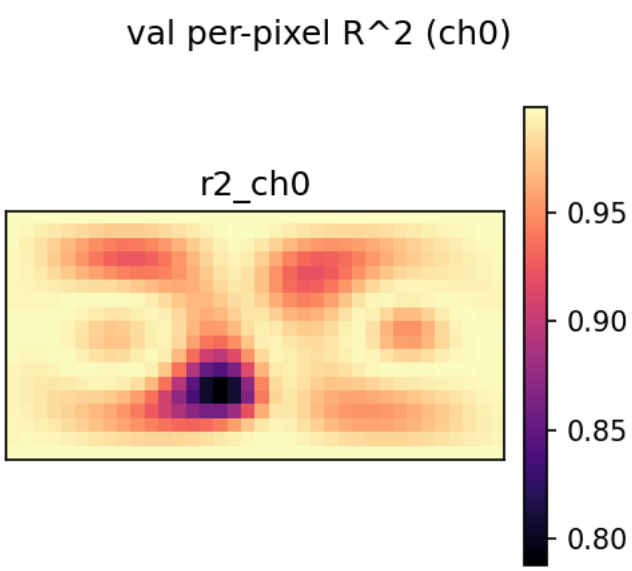
decompose(cfg)	method	model	val_field_rmse	val_field_r2	run
spherical_slepian_scipy	spherical_slepian	ridge	1.158e-02	0.964466	run
spherical_harmonics_scipy_bench	spherical_harmonics	ridge	2.354e-02	0.929476	run
graph_fourier_bench	graph_fourier	ridge	2.473e-02	0.928265	run
dct2	dct2	ridge	2.653e-02	0.919973	run

Key train plots (best_field_eval=spherical_slepian_scipy)


count
residual
30
25
20
15
10
5
0
−0.06
−0.04
−0.02
0.00
0.02
0.04
0.06


val dim 0 (R^2=0.998)
pred
true
1.30
1.25
1.20
1.15
1.10
1.05


val field scatter ch0 (R^2=0.978)
pred
true
1.4
1.3
1.2
1.1
1.0



mesh_scalar

Problem setting

- domain: `mesh` (planar triangulated grid mesh (289 verts))
- field: `scalar`
- grid: `289x1`, x=[-1.0, 1.0], y=[-1.0, 1.0]
- cond: `(N,4)`
- mask: n/a (values are on vertices)

Highlights (auto)

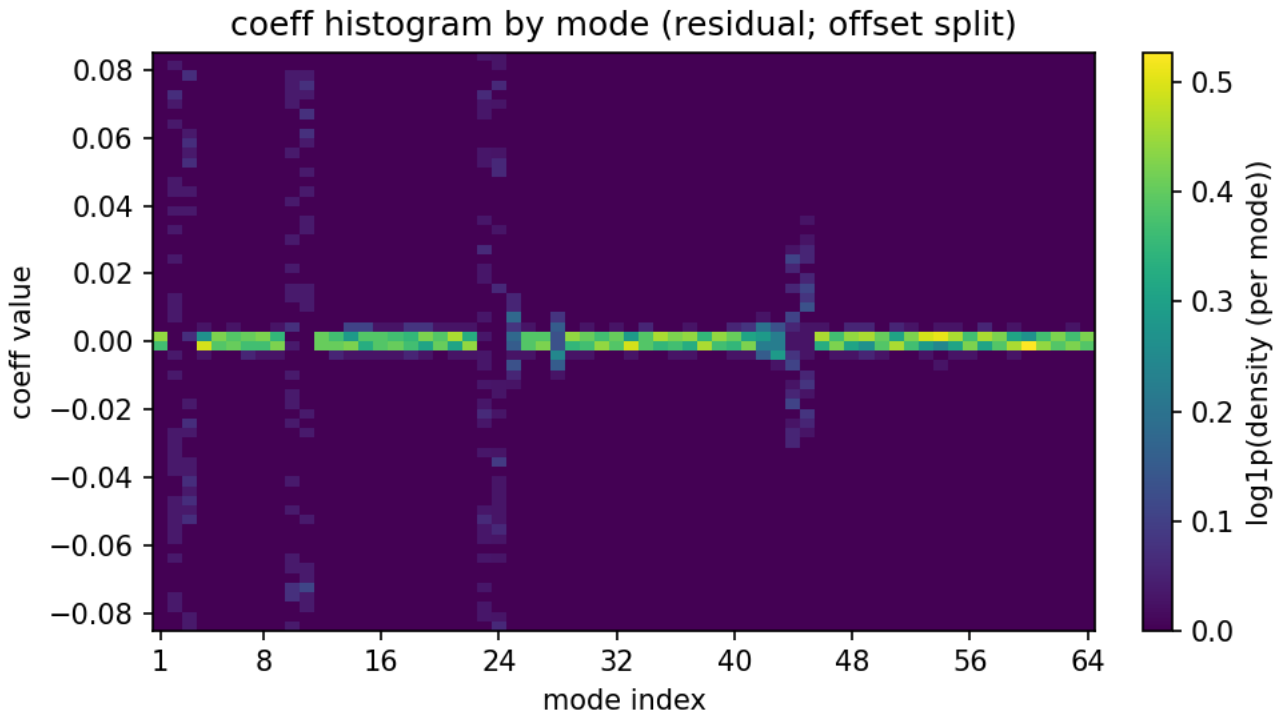
- decomposition: best full recon = `laplace_beltrami` (`laplace_beltrami`) (field_rmse=1.592e-02, field_r2=0.961517)
- decomposition: best compression proxy = `laplace_beltrami` (`laplace_beltrami`) (k_req_r2_0.95=8, r2_topk_k64=0.961517)
- decomposition: best top-energy@64 = `laplace_beltrami` (`laplace_beltrami`) (r2_topk_k64=0.961517, k_req_r2_0.95=8)
- train: best coeff-space = `laplace_beltrami` (`laplace_beltrami`) (`ridge`) (val_rmse=7.961e-03, val_r2=0.826122)
- train: best field-space = `laplace_beltrami` (`laplace_beltrami`) (`ridge`) (val_field_rmse=3.168e-02, val_field_r2=0.852310)

Decomposition (field reconstruction)

decompose(cfg)	method	status	rmse	r2	r2_k1	r2_k64	fit	n_req	k_req_r2_0.95	run
`laplace_beltrami`	`laplace_beltrami`	ok	1.592e-02	0.961517	-0.000004	0.961517	30.1ms	24	8	run

Coefficient histograms by mode (per decomposer)

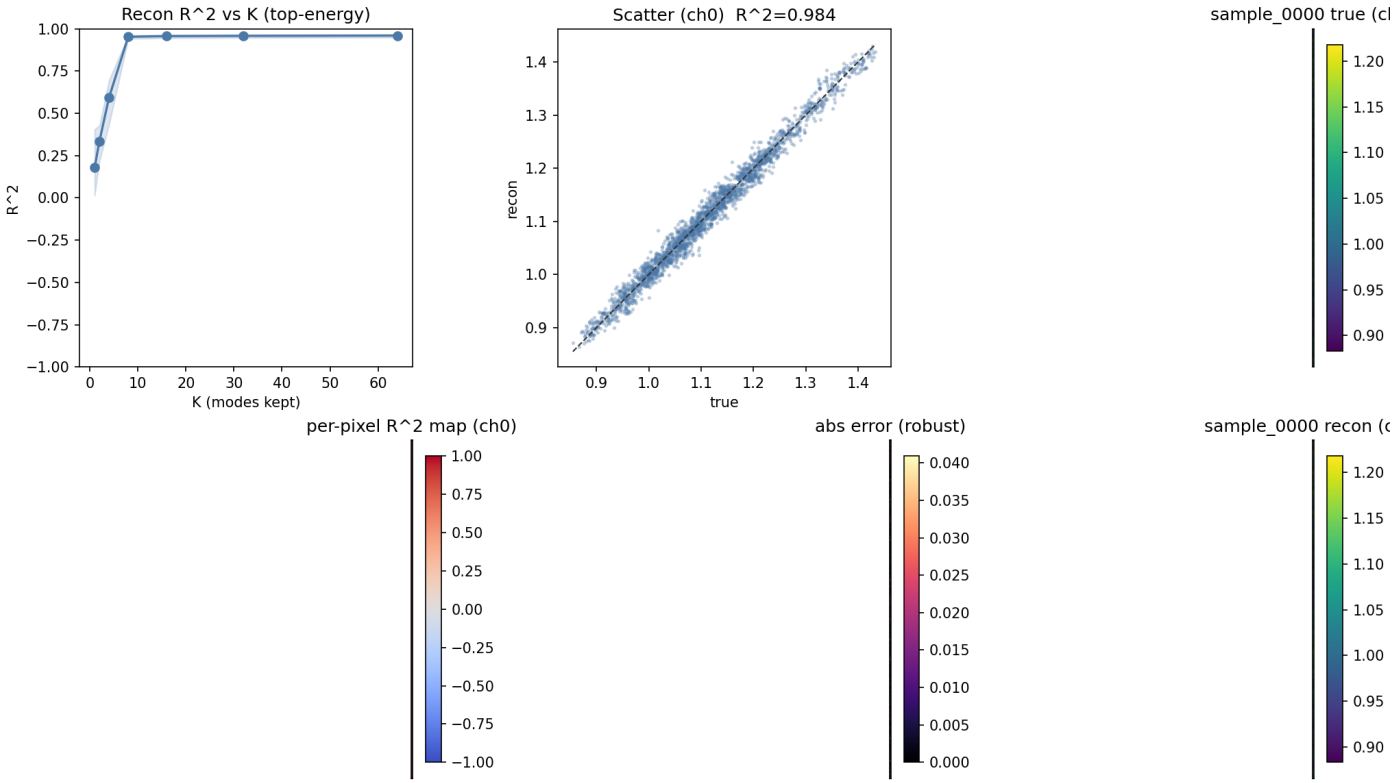
laplace_beltrami (laplace_beltrami)

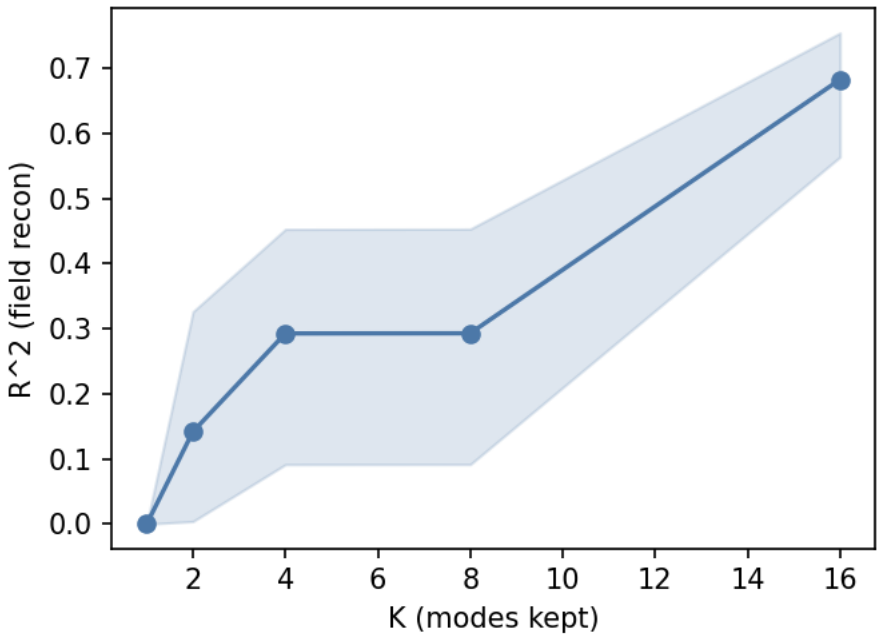


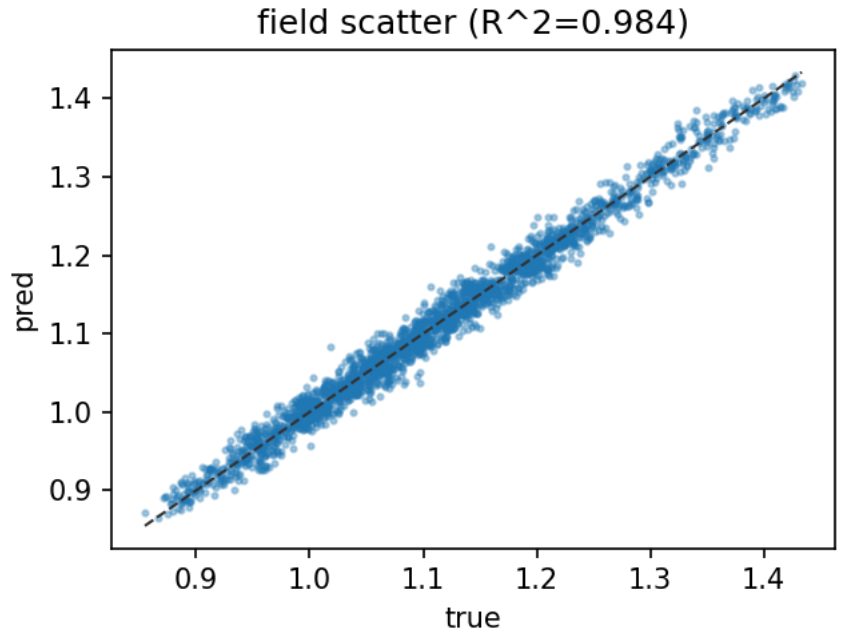
Compression leaderboard (Top-energy @K)

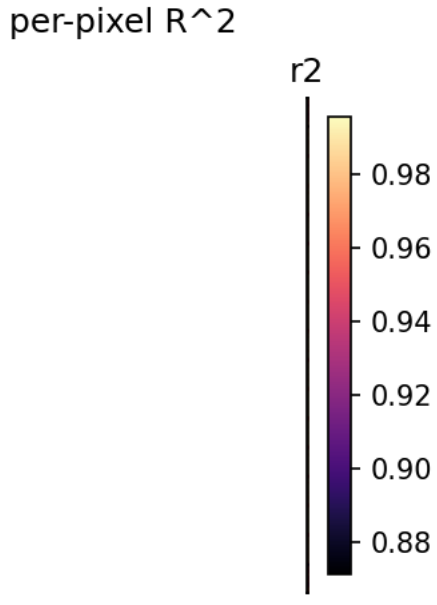
decompose(cfg)	method	r2_topk_k64	r2_topk_k16	k_req_r2_0.95	n_req
laplace_beltrami	laplace_beltrami	0.961517	0.958266	8	24

Key decomposition plots (best_rmse=laplace_beltrami)









Train (cond -> coeff prediction)

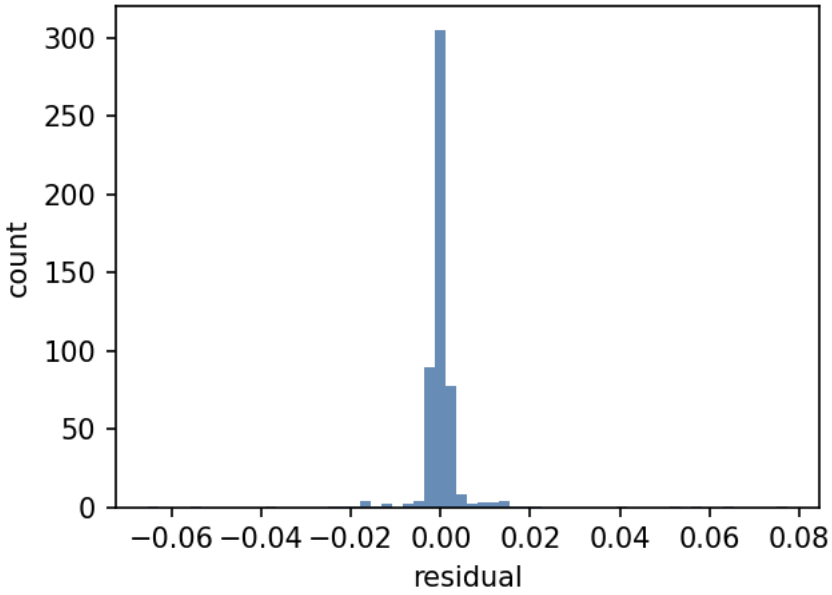
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2	fit	run

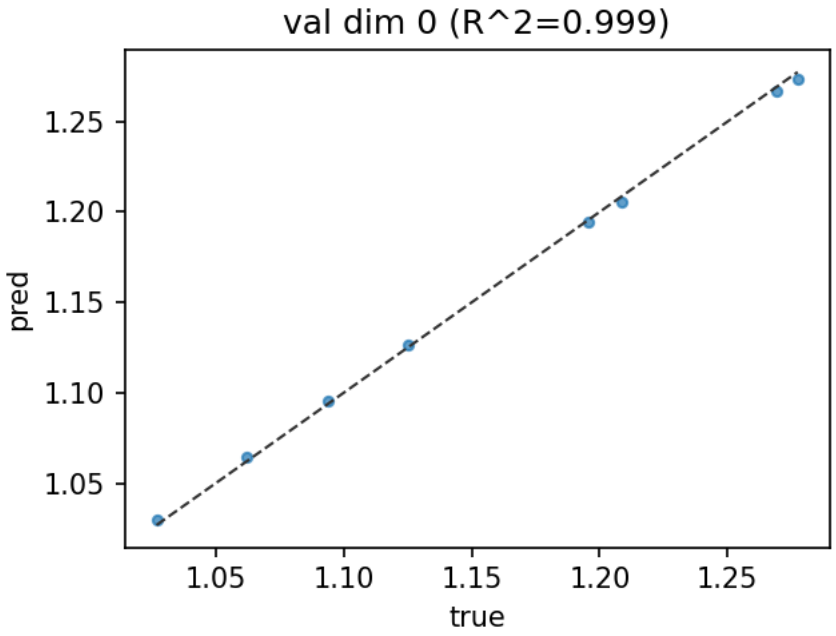
decompose(cfg)	method	decomp_r2	model	status	val_rmse	val_r2	val_field_rmse	val_field_r2	fit	run
laplace_beltrami	laplace_beltrami	0.961517	ridge	ok	7.961e-03	0.826122	3.168e-02	0.852310	1.8ms	run

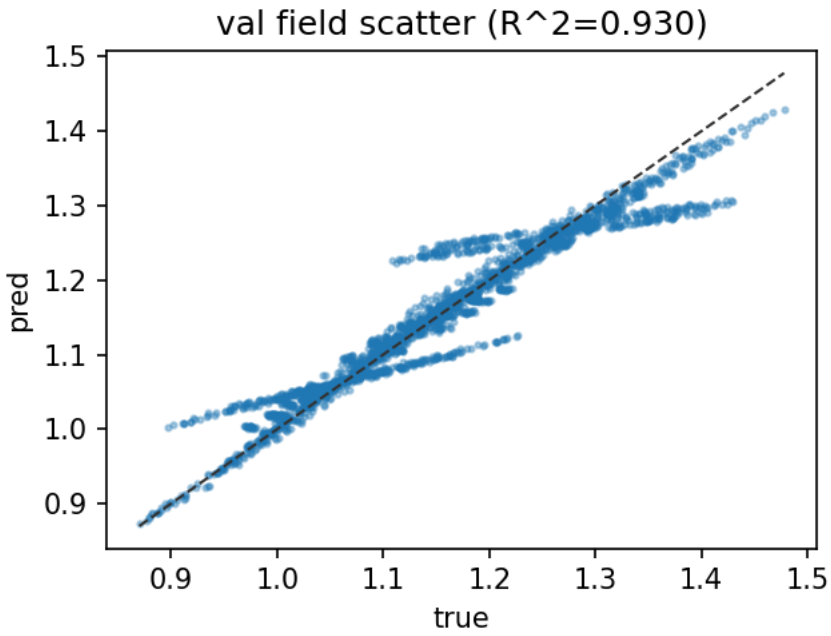
Train leaderboard (field-space)

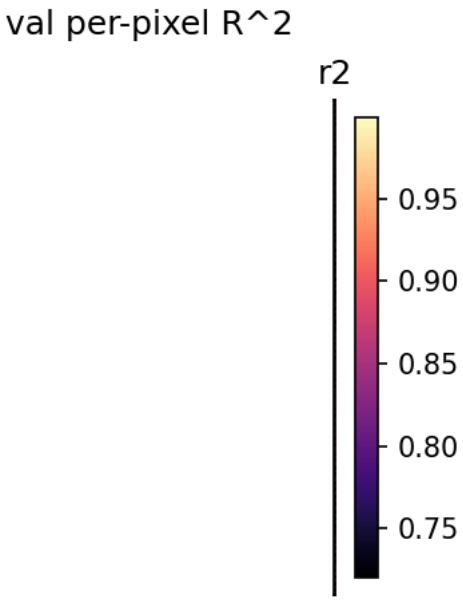
decompose(cfg)	method	model	val_field_rmse	val_field_r2	run
laplace_beltrami	laplace_beltrami	ridge	3.168e-02	0.852310	run

Key train plots (best_field_eval=laplace_beltrami)







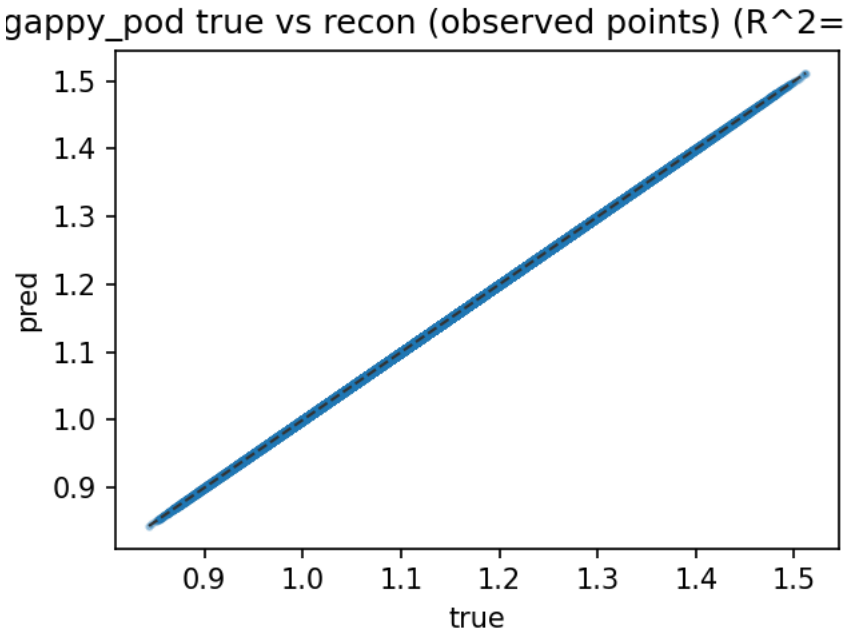


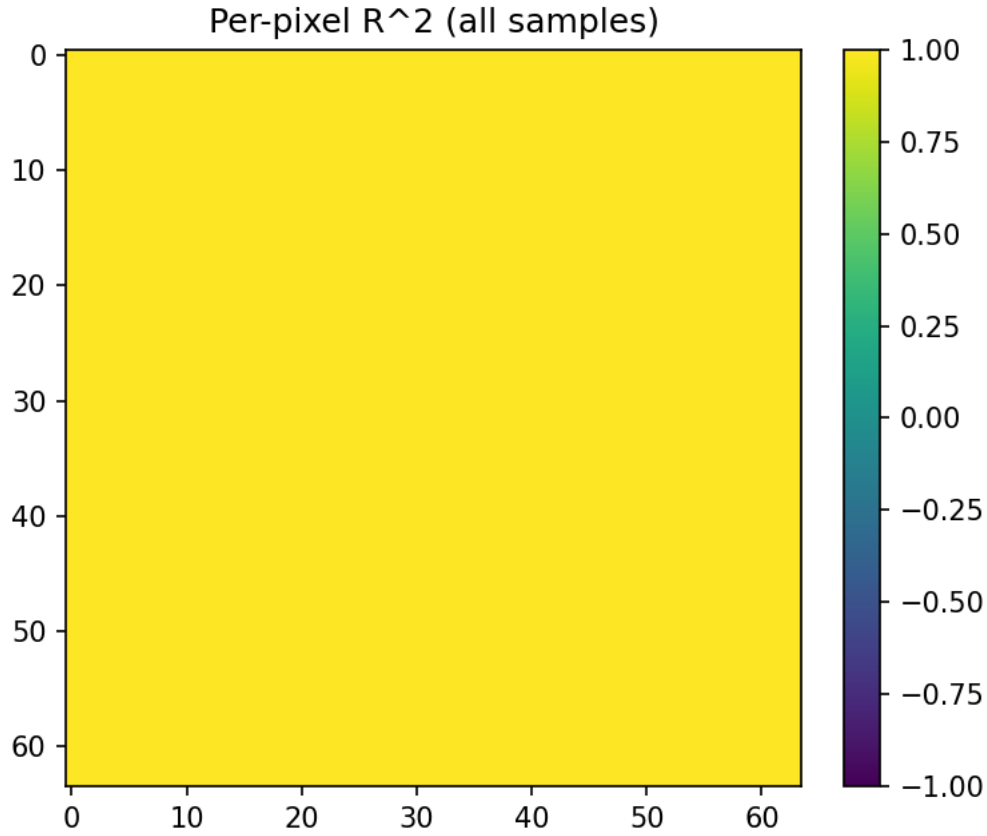
Special Evaluation: gappy_pod (rectangle_scalar, observed mask)

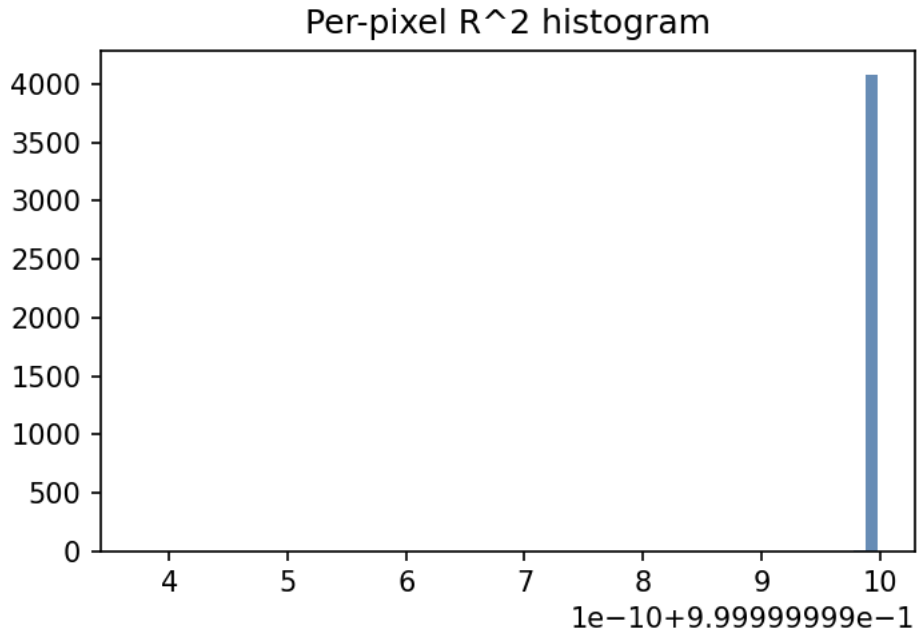
- metrics: `runs/benchmarks/v1_missing_methods/gappy_pod_rectangle_scalar/metrics.json`

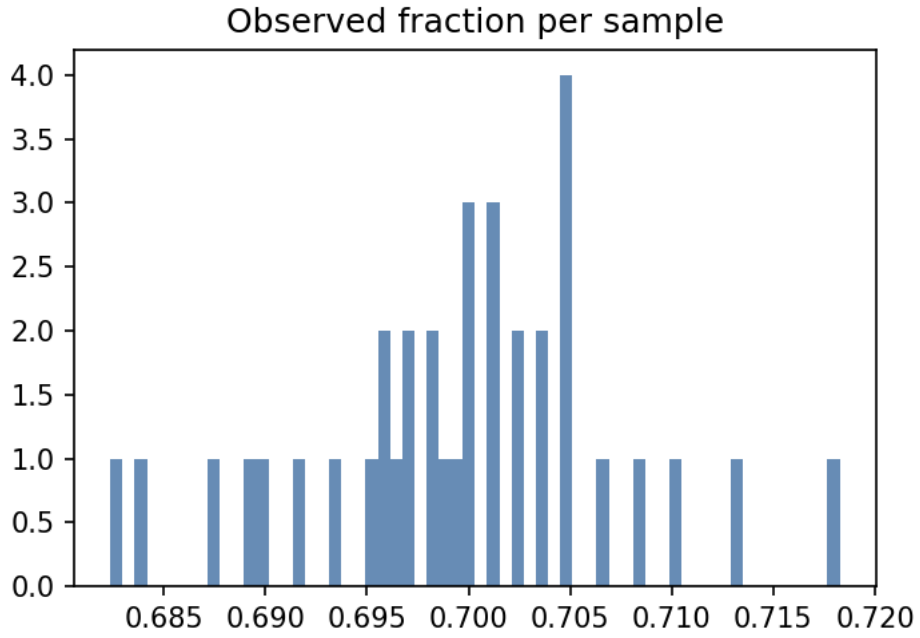
metric	value
n_samples	36
grid	[64, 64]
obs_frac	0.7
reg_lambda	1e-06
field_rmse	2.497201592177589e-07
field_r2	0.9999999999903206
field_rmse_obs	2.470346203153895e-07
field_r2_obs	0.9999999999905478

Key gappy_pod plots


gappy_pod true vs recon (observed points) (R^2=
pred
true
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.9
1.0
1.1
1.2
1.3
1.4
1.5


Per-pixel R^2 (all samples)
0
10
20
30
40
50
60
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00





PDF conversion

- 画像は Markdown の `![](...)` で埋め込み済み（相対パス）。
- 例: `pandoc summary_benchmark.md -o summary_benchmark.pdf`