BEYOND THE FINAL ITERATE: LEARNING FROM SOLVER TRAJECTORIES FOR INVERSE MEDIUM SCATTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Inverse medium scattering concerns the reconstruction of spatially varying media from limited and noisy wave measurements. We investigate whether the trajectory generated by a classical solver can itself serve as a useful representation for learned inverse reconstruction. We propose a Transformer-based neural operator that learns reconstruction corrections from function-space trajectories generated by Gauss–Newton (GN) iterations with total variation (TV) regularization. An attention mechanism with convolutional query, key, and value projections aggregates information across the trajectory while preserving spatial structure. For Helmholtz inverse medium reconstruction with nominal 3% observation noise, our method achieves a mean relative L^2 reconstruction error of 9.26%, compared with 13.96% for GN–TV and 12.70% for a comparably sized Fourier Neural Operator (FNO) baseline that corrects the final GN–TV iterate. Through extensive comparative experiments and ablation studies, we demonstrate the effectiveness of the proposed model. Without retraining, it achieves lower mean medium reconstruction errors than GN–TV across the tested noise levels and spatial resolutions, supporting its robustness to observation noise and discretization changes. These findings highlight solver trajectories as an effective interface between model-based inversion and operator learning.

1 INTRODUCTION

Inverse medium scattering reconstructs spatially varying media from measurements of scattered waves, with applications in acoustic imaging, geophysical exploration, and nondestructive testing (Colton & Kress, 2013). The dependence of the wave field on the unknown medium makes the inverse map nonlinear, while limited and noisy measurements make reconstruction unstable. Classical methods address these difficulties through physical modeling and regularization. Starting from a Born approximation, Gauss–Newton iterations with total variation regularization combine local linearization with a penalty that favors spatially coherent estimates and preserves discontinuities (Jin & Zhong, 2013). However, the gains of GN-TV iterations eventually diminish under fixed regularization and discretization, leaving a residual that motivates a learned correction. Neural operators learn mappings between function spaces, providing a natural framework for correcting spatial medium estimates (Kovachki et al., 2023). Synthesizing these ideas motivates the following questions:

*Can **solver trajectories** enhance inverse medium scattering beyond the final iterate?*

A short GN–TV trajectory records how medium estimates and adjoint residual fields evolve under the same measurements. The final estimate summarizes the outcome of these updates, but does not explicitly retain how spatial structures changed along the way. Intermediate estimates expose this evolution, while adjoint residual fields link each state to the remaining observation mismatch through local physical sensitivities. These quantities introduce no new measurements; their potential value lies in presenting the same inverse problem through several physically related states.

We therefore treat the short solver trajectory as a structured input to a learned reconstruction map. The central question is whether a network can identify and combine useful information across these states to improve the terminal estimate. This motivates our trajectory Transformer and the compar-

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

isons below, which examine reconstruction accuracy together with the roles of history and adjoint information.

Our contributions are fourfold:

- We formulate inverse reconstruction as learning a terminal correction from a precomputed GN–TV trajectory of medium estimates and adjoint residual fields, with supervision only on the final reconstruction.
- We design a Transformer-based neural operator that aggregates spatial fields across inverse iterations using convolutional attention and replaceable decoders. Our principal instantiation, Transformer–Local, produces a bounded correction projected onto the admissible medium range.
- At nominal 3% noise, Transformer–Local achieves 9.26% mean relative L^2 error over 144 test sequences, versus 13.96% for three-step GN–TV and 12.70% for single-frame FNO correction. These are complete-model comparisons with different inputs and architectures; both medium reconstruction accuracy and agreement with clean reference observations improve.
- We assess input interventions, direct trajectory concatenation, noise and grid changes, and complete reconstruction cost. Without retraining, Transformer–Local has lower mean medium errors than three-step GN–TV across the tested conditions; we report observation fit and limitations separately.

2 Related work

Neural operators and attention. FNO and DeepONet parameterize maps between function spaces (Li et al., 2021; Lu et al., 2021). Building on Transformer attention (Vaswani et al., 2017), Galerkin Transformer relates attention without softmax to Petrov–Galerkin projection (Cao, 2021). GNOT handles heterogeneous inputs and irregular meshes (Hao et al., 2023), while Transolver and its scalable extension Transolver++ apply attention to learned mesh slices (Wu et al., 2024; Luo et al., 2025). Continuum attention develops a function-space formulation and approximation theory (Calvello et al., 2025). Our attention aggregates over the inverse-iteration index, assigning scalar weights to spatial value fields.

Learning from PDE trajectories. DPOT pretrains operators through autoregressive denoising (Hao et al., 2024), while Poseidon combines a multiscale operator Transformer with semigroup-based training (Herde et al., 2024). Our trajectories are indexed by inverse-solver steps, with supervision on a single reconstructed medium rather than physical time evolution.

Learned iterations and solver corrections. LISTA learns sparse-coding updates (Gregor & LeCun, 2010), while recurrent learned optimizers generate gradient-based steps (Andrychowicz et al., 2016). Learned primal–dual reconstruction and MoDL combine learned updates or priors with forward-model operations (Adler & Öktem, 2018; Aggarwal et al., 2019). Solver-in-the-Loop trains corrections through a differentiable PDE solver (Um et al., 2020). Our GN–TV trajectory is fixed before training and aggregated into one terminal correction.

Learning inverse scattering. SwitchNet exploits scattering-map structure (Khoo & Ying, 2019), while WideBNet and MFISNet organize inversion around multiple frequencies (Li et al., 2022; Melia et al., 2025). Learned projectors alternate neural constraints with regularized GN updates (Li et al., 2024). Recent preprints combine DeepONet indicators with linear sampling for obstacle reconstruction (Chenu et al., 2026), or neural operators with inverse Born expansions (Schotland et al., 2026). Our single-frame FNO comparison and history study examine the value of solver trajectories; input and effective-capacity differences limit attribution to history or attention alone.

3 Problem setup and trajectory representation

Scattering and adjoint fields. We reconstruct a real medium contrast q, supported in a bounded domain $\Omega \subset \mathbb{R}^2$ and satisfying $0 \le q \le q_{\max}$, from noisy multi-frequency scattering data. For

plane-wave illumination at frequencies ν_f, indexed by $f = 1, \dots, F$, the total fields satisfy

$$\Delta u_f + k_f^2(1+q)u_f = 0, \qquad k_f = 2\pi\nu_f/c_0, \tag{1}$$

where c_0 is the background wave speed and the scattered fields are outgoing. Let $\mathcal{F}_f(q)$ denote the complex scattering data at wave number k_f. The measurements are $y_f = P_f\mathcal{F}_f(q^\dagger) + \eta_f$, with true contrast $q^\dagger$, observation restriction P_f, and noise η_f. For fixed scales $s_{d,f} > 0$, set $\bar{F}_f = P_f\mathcal{F}_f/s_{d,f}$ and $\bar{y}_f = y_f/s_{d,f}$. The data misfit and adjoint fields are

$$D(q;y) = \tfrac{1}{2}\sum_{f=1}^{F} \left\|\bar{F}_f(q) - \bar{y}_f\right\|_2^2, \tag{2}$$

$$g_f(q;y) = \bar{J}_f(q)^*\big(\bar{y}_f - \bar{F}_f(q)\big), \qquad \bar{J}_f(q) = \bar{F}_f'(q). \tag{3}$$

Here * denotes the real adjoint, and g_f is the negative gradient of the frequency-specific data misfit with respect to the discrete medium values. The forward model and adjoint calculation are detailed in Appendices A–B.

Classical reconstruction and trajectory generation. We use a classical reconstruction pipeline: Born initialization followed by TV-regularized Gauss–Newton iterations. We refer the reader to Colton & Kress (2013) for the Born approximation and to Salzo & Villa (2012) and Jin & Zhong (2013) for Gauss–Newton methods with convex penalties. On an $N \times N$ grid Ω_h with $n = N^2$ cells, Born initialization gives a clipped ridge estimate $q_0 \in \mathcal{C} = [0, q_{\max}]^n$ (Appendix A). Starting from q_0, successive GN–TV iterations generate the medium estimates $q_1, \dots, q_K$. At each iteration $t = 0, \dots, K-1$, we approximately solve

$$\min_{q \in \mathcal{C}_t} \ \tfrac{1}{2}\left\|A_t(q - q_t) - r_t\right\|_2^2 + \tfrac{\mu}{2}\left\|q - q_t\right\|_2^2 + \lambda\,\mathrm{TV}_h(q), \tag{4}$$

where A_t, r_t are the real-stacked normalized Jacobian and residual, $\mu > 0$ and $\lambda \geq 0$ are fixed regularization weights, and $\mathcal{C}_t$ is the feasible set. We solve this subproblem using the alternating direction method of multipliers (ADMM) (Boyd et al., 2011), followed by a line search to determine the next iterate q_{t+1}. Appendix C gives the constraints and solver details. We use a fixed iteration count K, chosen based on the numerical experiments reported in Section 5, to generate the trajectory $(q_0, \dots, q_K)$ for our model.

Function-valued trajectory representation. Each iterate is paired with its frequency-specific adjoint fields to form the input trajectory

$$\Gamma = (Z_t)_{t=0}^{K}, \qquad Z_t(x) = \big(q_t(x), g_1(q_t;y)(x), \dots, g_F(q_t;y)(x)\big). \tag{5}$$

We regard each token Z_t as an element of $L^2(\Omega;\mathbb{R}^{F+1})$ and represent it on the fixed computational grid Ω_h. The medium channel is normalized by a prescribed scale $s_q > 0$, and each adjoint channel by its root-mean-square value $s_{g,f}$ computed from the training set and held fixed. These feature scales are distinct from the observation scales $s_{d,f}$ in Equation (2). Supervised training pairs the precomputed trajectories Γ with the target contrasts $q^\dagger$, without differentiating through the physical solver. Clean scattering data and data at unobserved receiver directions are reserved for evaluation. Numerical settings are specified in Section 5.

4 Function-space trajectory Transformer

Given the solver trajectory Γ in Equation (5), we learn a reconstruction operator $\mathcal{T}_\theta : \Gamma \mapsto \widehat{q}$ that combines function-space attention with a replaceable terminal decoder. Figures 1 and 2 show the reconstruction pipeline and the Trajectory Transformer R_θ, which predicts the raw correction to q_K.

Trajectory embedding. With the fixed channel scales $S = \mathrm{diag}(s_q, s_{g,1}, \dots, s_{g,F})$, each function-valued token is embedded as $E_t = \phi_{\mathrm{emb}}(S^{-1}Z_t)$, where $E_t : \Omega_h \to \mathbb{R}^C$. The shared map ϕ_{emb} consists of a spatial convolution followed by GELU, lifting the input to C feature channels while preserving the computational grid.

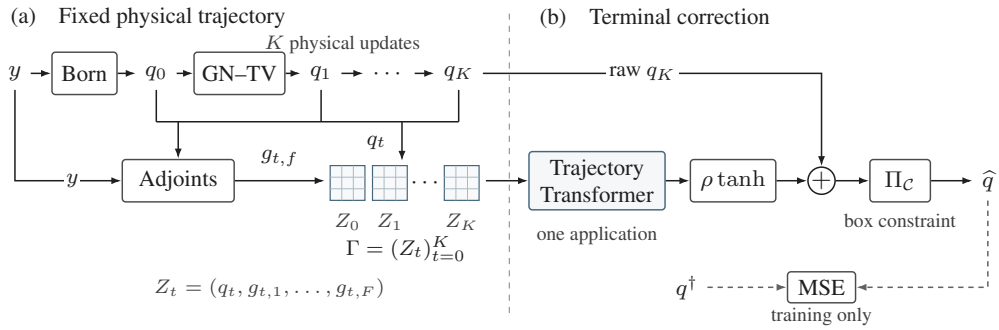


Figure 1: Reconstruction pipeline. The divider separates fixed preprocessing from learned correction; dashed arrows denote training-only supervision. Grid tiles are schematic tokens. Here $g_{t,f} = g_f(q_t; y)$ and $\Pi_{\mathcal{C}} = \mathrm{proj}_{\mathcal{C}}$.

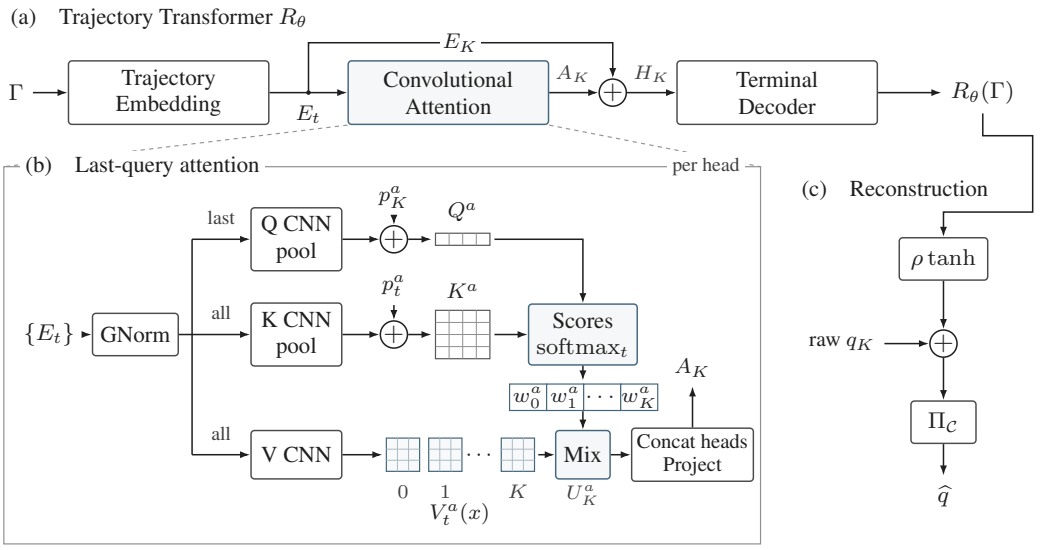


Figure 2: Trajectory Transformer architecture. Dashed guides expand the attention block. Here $A_K = W_O \,\mathrm{concat}_a\, U_K^a$; GNorm denotes group normalization and Mix the weighted sum in Equation (7). Descriptor cells and grids are schematic.

Convolutional attention. Group normalization (Wu & He, 2018), denoted by N_1, acts independently on each feature field E_t. In Figure 2(b), the final token supplies the query, while all $K+1$ tokens supply keys and values. Independent convolutional Q/K projections with spatial pooling produce n_h heads with d-dimensional descriptors. Writing $Q^a(E_K)$ and $K^a(E_t)$ for the head-a maps including N_1, and p_t^a for learned iteration embeddings, the attention weights are

$$\ell_t^a = \frac{\langle Q^a(E_K) + p_K^a, K^a(E_t) + p_t^a \rangle}{\sqrt{d}}, \qquad w_t^a = \frac{\exp(\ell_t^a)}{\sum_{j=0}^{K} \exp(\ell_j^a)}. \tag{6}$$

The embeddings encode the inverse-iteration index and are added only to Q/K descriptors.

The convolutional value projection maps $N_1(E_t)$ through C_V hidden channels to C channels without pooling or iteration embeddings. Assuming n_h divides C, splitting channels gives fields $V_t^a : \Omega_h \to \mathbb{R}^{C/n_h}$. Aggregation and residual fusion yield

$$U_K^a(x) = \sum_{t=0}^{K} w_t^a V_t^a(x), \qquad H_K = E_K + W_O \,\mathrm{concat}_{a=1}^{n_h} U_K^a, \tag{7}$$

where W_O is a pointwise channel projection and the residual E_K is taken before N_1. Each head assigns one scalar weight per token, shared over space. The convex combination in U_K^a concerns encoded values; it does not constrain $\widehat{q}$ to the convex hull of the medium iterates.

Terminal decoder. The terminal stage in Figure 2(a) maps H_K to $R_\theta(\Gamma)$ through a second group normalization N_2, a replaceable spatial decoder, and a pointwise scalar head. We use *Local* (pointwise channel maps), *FNO* (Fourier and pointwise branches), or *U-Net* (multiscale convolutions with skip connections), retaining the same embedding and function-space attention architecture. Local gives our principal instantiation, Transformer–Local. Reconstruction accuracy differs modestly among these choices in the primary comparison; detailed results are in Section 5.2.

With a prescribed pointwise correction bound $\rho > 0$, each decoder gives

$$\mathcal{T}_\theta(\Gamma) = \widehat{q} = \mathrm{proj}_{\mathcal{C}}\big[q_K + \rho \tanh R_\theta(\Gamma)\big]. \tag{8}$$

The raw q_K bypass is taken before input scaling, and projection onto $\mathcal{C}$ is pointwise clipping. Zero-initializing the scalar head gives the initial prediction q_K. The bound ρ is independent of the GN step radius δ_{GN} (Appendix C); decoder implementation details are in Appendix E.

Training and inference. We optimize θ on precomputed pairs $(\Gamma_i, q_i^\dagger)$, keeping the trajectories and input scales S fixed and excluding the physical solver from backpropagation. For a batch of size b and n grid points, the sole objective is normalized final-medium mean squared error:

$$\mathcal{L}(\theta) = \frac{1}{bn} \sum_{i=1}^{b} \sum_{j=1}^{n} \left(\frac{\mathcal{T}_\theta(\Gamma_i)_j - q_{i,j}^\dagger}{s_q} \right)^2. \tag{9}$$

At inference, visible observations and known physics generate Γ, and the model is applied once with the same fixed scaling. Complete ablation configurations and results are provided in Section 5.

5 Numerical Experiments

We evaluate whether a Transformer acting on function-valued solver trajectories improves inverse reconstruction, which parts of the representation contribute to this improvement, and how the learned map behaves under changes in noise and spatial discretization. Recovery is assessed against the true medium and clean reference scattering data. Local is our default, structurally simple decoder; FNO and U-Net provide alternative output maps within the same trajectory-Transformer framework.

5.1 Experimental setup and evaluation

Data and trajectory setup. We consider the domain $\Omega = [-1/2, 1/2]^2$, with a reference grid of 96×96 cells and admissible medium contrast $0 \le q \le 0.108$. All three datasets share four geometric families—multiscale disks, clustered disks, rings, and nonconvex domains—and three contrast profiles: piecewise constant, smoothly varying at coarse spatial scales, and smoothly varying at fine spatial scales. These define $4 \times 3 = 12$ geometry–profile groups. Each group contains 8, 1, and 6 medium realizations in the training, validation, and test sets, respectively. Every medium is paired with the two wave-number triples $(40, 70, 100)$ and $(40, 70, 120)$, producing two solver trajectories. This yields $4 \times 3 \times (8, 1, 6) \times 2 = (192, 24, 144)$ trajectory samples for the training, validation, and test sets, respectively. Geometric instances are sampled independently for each split and are not reused across splits.

At each frequency, 8 incident directions and 32 receivers provide 256 complex measurements, of which 96 are used for reconstruction. Across the three frequencies, these give $3 \times 96 = 288$ observed measurements with nominal 3% complex Gaussian noise. Born initialization followed by three GN–TV updates yields the input trajectory $\Gamma = (Z_0, \ldots, Z_3)$, where each Z_t combines the medium estimate with its adjoint fields as defined in Equation (5). The remaining 480 measurements are withheld from reconstruction, and scattering recovery is evaluated against clean reference data. Numerical and architectural details are given in Appendix E.

We perform at most 4000 ADMM iterations for each local GN–TV subproblem. If the stopping criterion is not met earlier, the finite feasible approximate solution obtained at the iteration limit is used in the subsequent line search.

Training and checkpoint selection. All models in the main comparison use random seed 123456, the same split and sample order, 100 epochs, and normalized final-medium MSE. AdamW (Loshchilov & Hutter, 2019) uses learning rate 2×10^{-4}, weight decay 10^{-4}, batch four, and gradient norm clipping with a threshold of 1.0. Every model produces a $0.02 \tanh$ correction to q_3, followed by projection onto the admissible range. Each checkpoint is selected independently by validation full-field relative L^2, averaging the two bands within each medium and then the 12 media; ties select the earlier epoch.

Evaluation metrics. We evaluate recovery using four relative L^2 errors. Let $\widehat{q}$ and $q^\dagger$ denote the reconstructed and reference media, I the interior mask constructed from $q^\dagger$, and M the visible measurement mask. With the independently evaluated full-wave response $F_h(\widehat{q})$ and clean reference data d, the metrics are

$$E_{\mathrm{field}}(\widehat{q}) = \frac{\left\|\widehat{q} - q^\dagger\right\|_2}{\left\|q^\dagger\right\|_2}, \quad E_{\mathrm{int}}(\widehat{q}) = \frac{\left\|I \odot (\widehat{q} - q^\dagger)\right\|_2}{\left\|I \odot q^\dagger\right\|_2}, \tag{10}$$

$$E_{\mathrm{vis}}(\widehat{q}) = \frac{\left\|M \odot \left(F_h(\widehat{q}) - d\right)\right\|_2}{\left\|M \odot d\right\|_2}, \quad E_{\mathrm{un}}(\widehat{q}) = \frac{\left\|(1-M) \odot \left(F_h(\widehat{q}) - d\right)\right\|_2}{\left\|(1-M) \odot d\right\|_2}. \tag{11}$$

Equation (10) gives the full-field and interior medium errors. In Equation (11), E_{vis} and E_{un} measure clean scattering recovery at visible and unobserved directions, respectively. Complex norms include real and imaginary parts. The interior-mask construction and error aggregation are detailed in Appendix F.

To quantify sampling uncertainty in mean error differences, we compute paired 95% bootstrap intervals from 10000 resamples of the 24 test geometries, retaining all profiles and both wave-number triples within each geometry. Mean recovery errors are reported as percentages.

5.2 Medium reconstruction and scattering recovery

Table 1 compares GN–TV, vanilla FNO and U-Net with single-frame or complete-trajectory inputs, and our trajectory Transformers with Local, FNO, and U-Net decoders.

Table 1: Recovery at 3% noise and grid 96^2: relative L^2 errors (%) on all 144 test sequences. Epoch denotes validation selection after the full 100-epoch budget. Local is the default implementation; all three Transformer rows instantiate the same trajectory framework.

Method	Parameters	Epoch	E_{field}	E_{int}	E_{vis}	E_{un}
Classical reconstruction						
GN–TV3	–	–	13.964	7.533	5.740	3.891
GN–TV8	–	–	13.957	7.516	5.734	3.885
Single-frame correction: input q_3						
vanilla FNO	131665	18	12.700	6.657	5.901	3.129
vanilla U-Net	125886	98	11.150	6.659	6.581	3.421
Direct concatenation: complete medium–adjoint trajectory						
vanilla FNO	131905	41	12.299	6.764	5.480	2.909
vanilla U-Net	126141	87	10.268	6.084	5.548	2.877
(Ours) *Trajectory Transformer: complete trajectory*						
Transformer–Local	152841	91	9.262	**5.350**	4.989	2.416
Transformer–FNO	447753	23	9.827	5.875	5.392	2.540
Transformer–U-Net	401601	57	**9.126**	5.637	**4.952**	**2.251**

Comparison with classical solver baselines. All three trajectory-Transformer variants achieve lower mean errors than GN–TV3 and GN–TV8 across all four reported metrics. Extending GN–TV from 3 to 8 iterations under the same regularization policy only reduces full-field error from 13.964% to 13.957%, whereas Local achieves 9.262%. This limited gain from further classical iterations is the key reason for choosing $K = 3$ for trajectory generation. Figure 3 complements these quantitative results with reconstructions and absolute-error maps for GN–TV3 and Transformer–Local on four test cases selected independently of reconstruction errors.

Comparison with neural operator baselines. The trajectory Transformers also offer substantial gains over vanilla FNO and U-Net whether the baselines use only q_3 or concatenate the complete medium–adjoint trajectory. Figures 4 and 5 in Appendix I extend the same four-case comparison to single-frame vanilla FNO and all three Transformer variants, showing their reconstructions and pointwise absolute errors, respectively, on shared color scales. We recommend Local as the default because it has the fewest parameters among the Transformer variants.

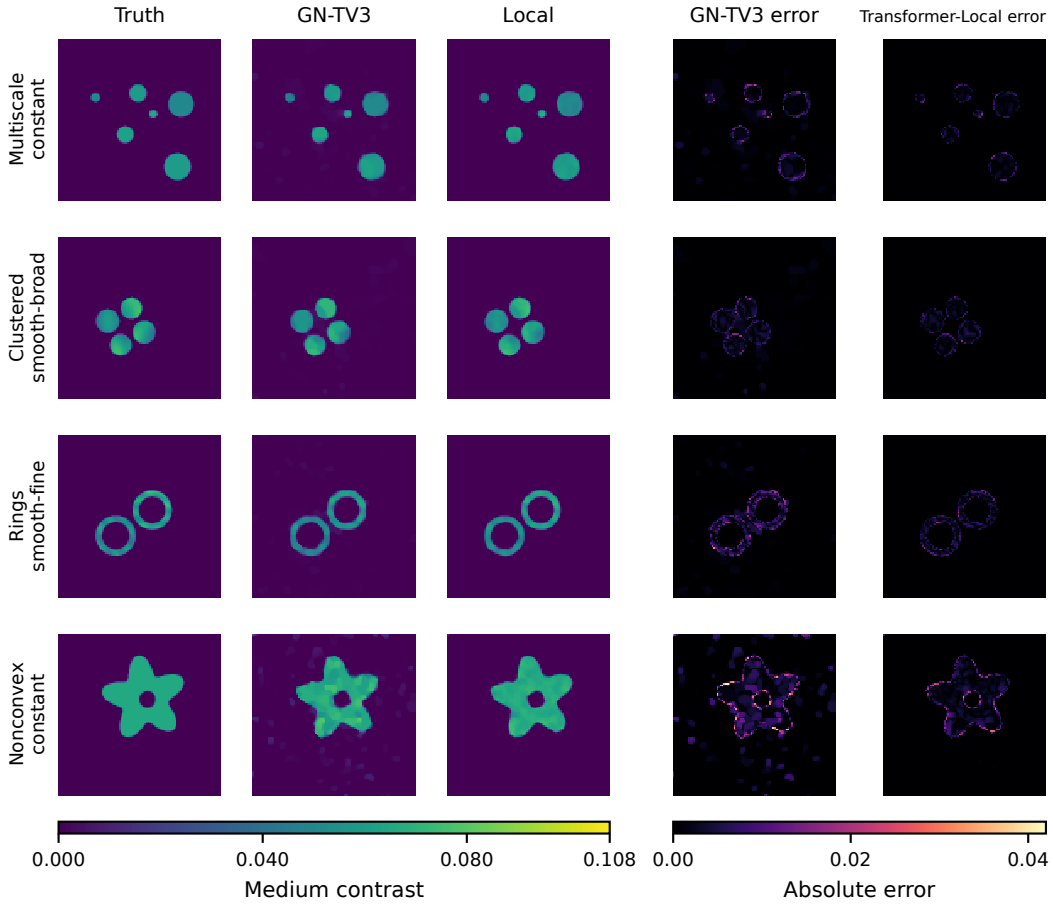


Figure 3: Four test examples selected by a fixed condition/profile rule, independently of errors: original-band shapes 000/constant, 006/smooth-broad, 012/smooth-fine, and 018/constant. Columns show truth, GN–TV3, Transformer–Local, and absolute errors with shared scales. The quantitative comparison of all methods is given in Table 1.

5.3 Ablations of trajectory information and adjoint fields

We distinguish the aggregation architecture from the information supplied to it. Table 2 reports the complete input, repetition of the final medium–adjoint token in all four slots, and removal of the three adjoint channels while retaining all medium states. Each intervention is retrained under the same budget and validation rule. For the vanilla models these inputs are concatenated; for the Transformer they remain separate function-valued tokens.

Does trajectory information matter? Compared with repeated-last inputs at unchanged channel counts, retaining intermediate states lowers full-field error from 9.386% to 9.262% for Transformer–Local and also improves both U-Net models, whereas both FNO models slightly favor repetition. The single-channel baselines in Table 1 also improve when given complete medium–adjoint trajectories: 12.700% to 12.299% for vanilla FNO and 11.150% to 10.268% for vanilla U-Net, reflecting

Table 2: Input interventions at 3% noise and grid 96^2: relative L^2 errors (%) on all 144 sequences. Repeated last preserves the final medium and its adjoint fields in every slot; Zero adjoints preserves the four medium states. Complete-input rows repeat Table 1 for paired interpretation.

Model	Input	Epoch	E_{field}	E_{int}	E_{vis}	E_{un}
Transformer–Local	Complete	91	9.262	5.350	4.989	2.416
	Repeated last	100	9.386	5.395	4.912	2.428
	Zero adjoints	100	10.155	5.785	5.553	2.773
Transformer–FNO	Complete	23	9.827	5.875	5.392	2.540
	Repeated last	21	9.751	5.901	5.309	2.578
	Zero adjoints	23	10.454	6.179	5.880	2.673
Transformer–U-Net	Complete	57	9.126	5.637	4.952	2.251
	Repeated last	70	9.349	5.738	5.217	2.350
	Zero adjoints	60	9.679	6.112	5.722	2.604
vanilla FNO	Complete	41	12.299	6.764	5.480	2.909
	Repeated last	56	12.131	6.449	5.677	3.154
	Zero adjoints	100	12.698	6.641	6.272	3.264
vanilla U-Net	Complete	87	10.268	6.084	5.548	2.877
	Repeated last	65	10.501	6.052	5.356	2.711
	Zero adjoints	93	10.500	6.457	5.993	2.711

the addition of both history and adjoint fields. These results suggest that trajectory information improves reconstruction only when the network architecture can exploit it effectively.

Remark. Repeating the final token and zeroing adjoint channels preserve the nominal architecture but change its algebraic dependence on the parameters. In Local, repeated identical values eliminate the influence of 114768 query/key and iteration-embedding parameters, leaving 38073 effective parameters; zero adjoints deactivate 648 input weights. These changes may influence optimization under the fixed training budget and contribute to the mixed ablation outcomes, although increased convergence difficulty has not been verified. Appendix G details the parameter dependencies. Nevertheless, all three Transformer variants retain lower mean full-field errors than both vanilla baselines given the same modified inputs under either intervention, demonstrating that their reconstruction advantage persists despite the restricted input information.

How can trajectory information be better utilized? With the same complete medium–adjoint trajectory, Transformer–Local lowers full-field error from 12.299% for vanilla FNO and 10.268% for vanilla U-Net to 9.262%, with comparable parameter counts (153k versus 132k and 126k; Table 1). Transformer–FNO and Transformer–U-Net also improve on their concatenation counterparts, from 12.299% to 9.827% and from 10.268% to 9.126%, respectively. These comparisons demonstrate that the proposed Transformer architecture makes more effective use of the same trajectory information than direct channel concatenation, improving reconstruction even with the simple Local decoder.

Do adjoint residual fields matter? With all medium states retained, including adjoint residual fields lowers mean full-field error in all five models compared with zero-adjoint inputs. For Transformer–Local, full-field error falls from 10.155% to 9.262%, and scattering errors at visible and unobserved directions fall from 5.553% to 4.989% and from 2.773% to 2.416%, respectively. The vanilla U-Net paired full-field difference interval includes zero, and some other metrics favor zero adjoints. The Local comparison therefore demonstrates that adjoint residual fields contribute useful information beyond the medium trajectory, benefiting both medium reconstruction and scattering recovery.

5.4 Noise robustness and application across grids

We apply the frozen models across observation noise levels of 0–5% and grids of 64^2, 96^2, and 128^2, varying one factor at a time. Appendix H gives the complete results in Table 6, together with native-grid errors, physical scaling, and numerical-status diagnostics.

Without retraining, Transformer–Local retains lower mean full-field errors than GN–TV3 across all tested conditions. A Transformer variant gives the lowest medium error at each noise level and at 128^2, while vanilla FNO leads at 64^2. All neural models have higher common-grid medium errors away from 96^2, and Local's scattering errors worsen relative to GN–TV3 at 64^2.

5.5 Computational cost and accuracy

We compare the complete reconstruction costs of Transformer–Local, GN–TV3, and GN–TV8 on all 144 primary-condition test sequences. The physical solver uses two threads on an AMD Ryzen 9 8945HX CPU, and network inference uses an RTX 5070 Laptop GPU in FP32.

Local timing includes per-sequence setup of the physical model, Born initialization, three GN–TV updates, computation of all required adjoint fields, input assembly, data transfers, and one network evaluation. The GN–TV baselines include the same setup and Born stages, with budgets of three or eight updates and the original stopping rule. Table 3 reports these costs alongside reconstruction and scattering errors.

Table 3: Deployment time and recovery at 3% noise on grid 96^2: relative L^2 errors (%) and mean times in seconds per sequence from a single exclusive timing run. Scattering errors are evaluated against clean reference data. Offline training, model loading and warmup, independent evaluation, and additional numerical checks are excluded from the deployment timings.

Method	Time (s)	E_{field}	E_{vis}	E_{un}
GN–TV3	8.593	13.964	5.740	3.891
GN–TV8	20.513	13.957	5.734	3.885
Transformer–Local	10.178	9.262	4.989	2.416

Compared with GN–TV8, Local reduces the mean deployment time per sequence from $20.513\,\mathrm{s}$ to $10.178\,\mathrm{s}$ and the mean full-field relative L^2 error from 13.957% to 9.262%, while also lowering both scattering errors. Compared with GN–TV3, the mean deployment time per sequence increases from $8.593\,\mathrm{s}$ to $10.178\,\mathrm{s}$, while the mean full-field relative L^2 error decreases from 13.964% to 9.262%.

These comparisons indicate that the additional runtime over GN–TV3 is reasonable in view of the reconstruction gains, supporting a favorable trade-off between accuracy and computational cost on this benchmark.

6 Discussion and conclusion

We have introduced an approach to operator learning that treats the trajectory of a classical solver as a representation for reconstruction. By aggregating function-valued iterates and adjoint residual fields, the proposed Transformer learns a correction that draws on the solver's intermediate computations. On Helmholtz inverse medium scattering at nominal 3% observation noise, Transformer–Local reduces the mean full-field relative L^2 error from 13.96% for three-step GN–TV to 9.26%, while improving interior and scattering recovery. Our systematic comparisons provide insight into how trajectory information and physical sensitivity can be combined, and how their value depends on the architecture and reconstruction setting. These findings support a broader role for classical solvers as generators of structured representations for learning, establishing solver trajectories as a practical interface between model-based inversion and operator learning.

The present approach integrates trajectory information through Transformer attention, while the dynamical structure of the iterates offers a further perspective. From a functional-analytic viewpoint, a classical solver can be viewed as generating discrete dynamics on an admissible function space. Successive states and residual fields then describe how the reconstruction evolves under the solver's update rule. Inspired by momentum-based optimization, future trajectory correction could explicitly use the directions and accumulated history of these updates to construct correction operators in function space. Developing this perspective calls for a formulation that accounts for function-space geometry, admissibility constraints, and discretization. We aim to advance its theoretical foundations and empirical evaluation, with the goal of understanding which features of solver dynamics enable more accurate and robust learned reconstruction.

AI USE STATEMENT

In this work, we used ChatGPT to refine the language, identify potentially relevant prior work for the Related Work and Introduction sections, and generate the code for the illustrations. The authors independently verified all references, reviewed and revised all suggestions, and take full responsibility for the final content.

ETHICS STATEMENT

This work studies a numerical inverse scattering problem using synthetically generated media and simulated wave measurements. No human participants or personal data are involved. The reported results concern controlled synthetic benchmarks and do not establish reliability for clinical, industrial, or other safety-critical applications.

REPRODUCIBILITY STATEMENT

We describe the experimental setup, data splits, training procedures, and checkpoint selection criteria in Section 5 and Appendix E. Appendices A–D detail the forward model and classical solver. Evaluation details and additional experimental protocols are documented in Appendices F–H. We will make the source code publicly available, including the data-generation, solver, training, and evaluation routines, to facilitate reproduction of the reported results.

REFERENCES

Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE Transactions on Medical Imaging*, 37(6):1322–1332, 2018.

Hemant K. Aggarwal, Merry P. Mani, and Mathews Jacob. MoDL: Model-based deep learning architecture for inverse problems. *IEEE Transactions on Medical Imaging*, 38(2):394–405, 2019.

Marcin Andrychowicz, Misha Denil, Sergio Gómez, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

Edoardo Calvello, Nikola B. Kovachki, Matthew E. Levine, and Andrew M. Stuart. Continuum attention for neural operators. *Journal of Machine Learning Research*, 26(300):1–52, 2025.

Shuhao Cao. Choose a transformer: Fourier or Galerkin. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Victor Chenu, Houssem Haddar, and Hadrien Montanelli. A neural operator framework for solving inverse scattering problems, 2026. URL https://arxiv.org/abs/2602.24147. arXiv preprint arXiv:2602.24147.

David Colton and Rainer Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*, volume 93 of *Applied Mathematical Sciences*. Springer, 3 edition, 2013.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 399–406, 2010.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. GNOT: A general neural operator transformer for operator learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12556–12569, 2023.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. DPOT: Auto-regressive denoising operator transformer for large-scale PDE pre-training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17616–17635, 2024.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Qinian Jin and Min Zhong. On the iteratively regularized Gauss–Newton method in Banach spaces with applications to parameter identification problems. *Numerische Mathematik*, 124(4):647–683, 2013.

Yuehaw Khoo and Lexing Ying. SwitchNet: A neural network model for forward and inverse scattering problems. *SIAM Journal on Scientific Computing*, 41(5):A3182–A3201, 2019.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Kai Li, Bo Zhang, and Haiwen Zhang. Reconstruction of inhomogeneous media by an iteration algorithm with a learned projector. *Inverse Problems*, 40:075008, 2024.

Matthew Li, Laurent Demanet, and Leonardo Zepeda-Núñez. Wide-band butterfly network: Stable and efficient inversion via multi-frequency neural networks. *Multiscale Modeling & Simulation*, 20(4):1191–1227, 2022.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3:218–229, 2021.

Huakun Luo, Haixu Wu, Hang Zhou, Lanxiang Xing, Yichen Di, Jianmin Wang, and Mingsheng Long. Transolver++: An accurate neural solver for PDEs on million-scale geometries. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 41432–41449, 2025.

Owen Melia, Olivia Tsang, Vasileios Charisopoulos, Yuehaw Khoo, Jeremy Hoskins, and Rebecca Willett. Multi-frequency progressive refinement for learned inverse scattering. *Journal of Computational Physics*, 527:113809, 2025.

Saverio Salzo and Silvia Villa. Convergence analysis of a proximal Gauss–Newton method. *Computational Optimization and Applications*, 53(2):557–589, 2012.

John C. Schotland, Aseel Titi, and Jenn-Nan Wang. Inverse Born series based neural operators, 2026. URL https://arxiv.org/abs/2608.18262. arXiv preprint arXiv:2608.18262.

Kiwon Um, Robert Brand, Yun (Raymond) Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative PDE-solvers. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53681–53705, 2024.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision*, pp. 3–19, 2018.

A FORWARD MODEL AND BORN INITIALIZATION

Scattering setup and outgoing convention. The contrast q satisfies $0 \le q \le q_{\max}$ for a prescribed $q_{\max} > 0$ and vanishes outside Ω. Fix a frequency index f, write $k = k_f$, and suppress the frequency index on the fields. For an incident direction $d \in \mathbb{S}^1$, the plane wave is $u_d^{\mathrm{inc}}(x) = e^{ikd\cdot x}$ and the total field is $u_d = u_d^{\mathrm{inc}} + u_d^{\mathrm{sca}}$. The scattered field satisfies the two-dimensional Sommerfeld radiation condition

$$\lim_{r\to\infty} r^{1/2}\left(\partial_r u_d^{\mathrm{sca}}(r\hat{x}) - iku_d^{\mathrm{sca}}(r\hat{x})\right) = 0, \qquad \hat{x} \in \mathbb{S}^1, \tag{12}$$

uniformly in the observation direction $\hat{x}$.

Following the volume-potential formulation of scattering theory (Colton & Kress, 2013), let $G_k(x) = \frac{i}{4}H_0^{(1)}(k|x|)$. The convention $(\Delta+k^2)G_k = -\delta$ gives the Lippmann–Schwinger equation

$$u_d(x) = e^{ikd\cdot x} + k^2 \int_\Omega G_k(x-z)q(z)u_d(z)\,dz. \tag{13}$$

Write $\mathcal{G}_k$ for this integral operator without the factor k^2, and M_q for multiplication by q. We assume $I - k^2\mathcal{G}_k M_q$ is invertible in a neighborhood of the states where derivatives are used.

The stored scattering convention is

$$\mathcal{F}_{k,o,d}(q) = k^2 \int_\Omega e^{-ik\hat{x}_o\cdot z} q(z)u_d(z)\,dz. \tag{14}$$

At $k = k_f$, collecting these values over the incidence and receiver directions gives $\mathcal{F}_f(q)$ in Section 3, and P_f selects the observed entries. Equation (14) omits the common frequency-dependent asymptotic prefactor of the far-field amplitude; the data, Born matrices, Jacobians, and evaluation all use this same normalization. We refer the reader to Colton & Kress (2013) for more details.

Observation normalization. Each frequency uses a fixed positive scale $s_{d,f}$ for both observations and predictions:

$$\bar{y}_f = \frac{y_f}{s_{d,f}}, \qquad \bar{F}_f(q) = \frac{P_f\mathcal{F}_f(q)}{s_{d,f}}, \qquad \bar{J}_f(q) = \frac{P_f\mathcal{F}_f'(q)}{s_{d,f}}.$$

The scales remain fixed across training and evaluation; their values are listed in Appendix E. The medium and adjoint input scales used by the network are separate from this observation normalization.

Fixed-grid realization. For a square domain of side length ℓ_Ω, the inversion grid has N cell centers per side and spacing $h = \ell_\Omega/N$; Section 5 sets $\ell_\Omega = 1$ and $N = 96$. The volume potential is represented by

$$(G_{k,h}v)_i = h^2 \sum_j G_k(x_i - x_j)v_j, \qquad (I - k^2 G_{k,h}\,\mathrm{diag}(q))u_d = u_d^{\mathrm{inc}}. \tag{15}$$

At the diagonal the implementation evaluates the Green function at radius $h/\sqrt{\pi}$; it does not use an exact analytical cell integral. All signed offsets are embedded in a $2N \times 2N$ zero-padded FFT convolution and the physical $N \times N$ block is retained. We use GMRES for reference evaluations.

Born linearization. At $q = 0$, $u_d = u_d^{\mathrm{inc}}$. Substituting the incident field into Equation (14) yields

$$(B_{f,\mathbb{C}})_{(o,d),j} = k_f^2 h^2 e^{ik_f(d-\hat{x}_o)\cdot x_j}. \tag{16}$$

After restricting each frequency to its visible mask and dividing by $s_{d,f}$, stack the real and imaginary parts to obtain $B \in \mathbb{R}^{2m_{\mathrm{obs}}\times n}$ and $\bar{y}_{\mathbb{R}} \in \mathbb{R}^{2m_{\mathrm{obs}}}$, where m_{obs} is the total number of visible complex observations across frequencies. For a ridge parameter $\alpha > 0$, the unconstrained problem

$$\min_q \ \tfrac{1}{2}\|Bq - \bar{y}_{\mathbb{R}}\|_2^2 + \tfrac{\alpha}{2}\|q\|_2^2 \tag{17}$$

has the unique minimizer $(B^\top B + \alpha I)^{-1}B^\top \bar{y}_{\mathbb{R}}$. The identity $(B^\top B + \alpha I)B^\top = B^\top(BB^\top + \alpha I)$ gives the row-space expression. After projection onto $\mathcal{C} = [0, q_{\max}]^n$, the initializer is

$$q_0 = \mathrm{proj}_{\mathcal{C}}\left[B^\top(BB^\top + \alpha I)^{-1}\bar{y}_{\mathbb{R}}\right]. \tag{18}$$

The clipping is a separate operation: clipping the ridge solution is generally *not* the minimizer of the box-constrained ridge problem.

Norm convention. We use Euclidean norms and real adjoints in all optimization and network implementations. For a piecewise-constant scalar field on the uniform grid, $\|v\|_{L^2(\Omega)} = h\,\|v\|_2$, so relative field errors agree in the two norms. The corresponding gradient conversion is derived in Appendix B.

B Full-wave derivative and adjoint residual fields

At a fixed wave number $k = k_f$, let q be a state where the volume operator is invertible, and let a be a real medium perturbation. Differentiating Equation (13) gives

$$(I - k^2 \mathcal{G}_k M_q)\dot{u}_d = k^2 \mathcal{G}_k(a u_d). \tag{19}$$

Differentiating the scattering functional first, without invoking reciprocity, yields

$$\mathcal{F}'_{k,o,d}(q)a = k^2 \int_\Omega e^{-ik\hat{x}_o\cdot x}\big(a(x)u_d(x) + q(x)\dot{u}_d(x)\big)\,dx. \tag{20}$$

This expression includes the change of the total field. Omitting its second term would give a frozen-field approximation rather than the full-wave Jacobian.

Reciprocity representation. Let v_o solve Equation (13) with incident wave $e^{-ik\hat{x}_o\cdot x}$. Symmetry of the Green kernel, $G_k(x-z) = G_k(z-x)$, gives the bilinear transpose identity

$$\mathcal{F}'_{k,o,d}(q)a = k^2 \int_\Omega v_o(x)u_d(x)a(x)\,dx. \tag{21}$$

For completeness, set $A = I - k^2 G M_q$, with symmetric $G^\top = G$, and $e_o(x) = e^{-ik\hat{x}_o\cdot x}$. In discretized bilinear notation, Equation (20) is

$$k^2 e_o^\top \big[I + M_q A^{-1} k^2 G\big] M_{u_d} a = k^2 e_o^\top (I - k^2 M_q G)^{-1} M_{u_d} a = k^2 v_o^\top M_{u_d} a. \tag{22}$$

The first equality uses $(I-XY)^{-1} = I + X(I-YX)^{-1}Y$ and the second uses transpose symmetry. Uniform quadrature supplies the common cell-area factor. There is no complex conjugation in this reciprocity product. The discrete entries are therefore $J_{(o,d),j} = k^2 h^2 v_{o,j} u_{d,j}$. In the aligned direction grids, each incident direction is present among the reversed receiver directions, allowing incident total fields to be reused from receiver solves.

Adjoint residual fields. Equip $\mathbb{C}^m$ with the real inner product $\langle z, w\rangle_{\mathbb{R}} = \Re(z^H w)$. For a real perturbation a and complex matrix J,

$$\langle Ja, r\rangle_{\mathbb{R}} = a^\top \Re(J^H r), \qquad J^* r = \Re(J^H r). \tag{23}$$

Define $r_f = \bar{y}_f - \bar{F}_f(q)$ and $D_f(q;y) = \frac{1}{2}\,\|r_f\|_2^2$, so that $D(q;y) = \sum_{f=1}^F D_f(q;y)$. The directional derivative with respect to the discrete medium values is

$$D'_f(q;y)[a] = -\left\langle \bar{J}_f(q)a, r_f\right\rangle_{\mathbb{R}} = -a^\top g_f(q;y), \qquad g_f(q;y) = -\nabla_q D_f(q;y). \tag{24}$$

Thus Equation (3) maps the observed residual to a real negative-gradient field on the medium grid. The two factors $s_{d,f}^{-1}$, one in $\bar{J}_f$ and one in r_f, give an overall factor $s_{d,f}^{-2}$. The objective D_f sums over observed entries. Network input normalization is applied to the resulting field separately.

Adjoint field scaling across grids. The cached field $g_{f,h}$ is the negative Euclidean gradient with respect to cell values; the Jacobian includes the cell-area factor h^2 from quadrature. With the discrete L^2 inner product $\langle a, b\rangle_h = h^2 a^\top b$,

$$D'_f(q;y)[a] = -a^\top g_{f,h} = -\left\langle a, h^{-2} g_{f,h}\right\rangle_h.$$

Hence the corresponding negative L^2 gradient is $h^{-2} g_{f,h}$. Training uses the cached fields on the reference grid with fixed input normalization.

C GN–TV SUBPROBLEM AND ADMM SOLUTION

Subproblem formulation. At outer iteration t, the current medium $q_t \in \mathcal{C} = [0, q_{\max}]^n$ is fixed and q is the optimization variable, as in Equation (4). Stack the real and imaginary parts of the normalized visible data to define $\bar{F}_{\mathbb{R}}$ and $\bar{y}_{\mathbb{R}}$. Then $A_t = \bar{F}'_{\mathbb{R}}(q_t)$ and $r_t = \bar{y}_{\mathbb{R}} - \bar{F}_{\mathbb{R}}(q_t)$. The feasible set combines the medium box with a bound on the change from q_t:

$$\mathcal{C}_t = \{q \in \mathcal{C} : \|q - q_t\|_\infty \le \delta_{\mathrm{GN}}\}, \qquad \delta_{\mathrm{GN}} > 0. \tag{25}$$

Thus the step radius enters Equation (4) through $\mathcal{C}_t$. Projection onto this set clips cell j to $[\max(0, (q_t)_j - \delta_{\mathrm{GN}}), \min(q_{\max}, (q_t)_j + \delta_{\mathrm{GN}})]$. With $I_{\mathcal{C}_t}$ denoting the convex indicator, the local objective is

$$\Psi_t(q) = \tfrac{1}{2}\|A_t(q - q_t) - r_t\|^2 + \tfrac{\mu}{2}\|q - q_t\|^2 + \lambda\,\mathrm{TV}_h(q) + I_{\mathcal{C}_t}(q), \tag{26}$$

where $\mu > 0$ and $\lambda \ge 0$ are fixed regularization weights. The set $\mathcal{C}_t$ is nonempty because it contains q_t.

Proximal formulation. Let $H_t = A_t^\top A_t + \mu I$ and $\zeta_t = q_t + H_t^{-1} A_t^\top r_t$. Completing the square gives

$$\tfrac{1}{2}\|A_t(q - q_t) - r_t\|^2 + \tfrac{\mu}{2}\|q - q_t\|^2 = \tfrac{1}{2}\|q - \zeta_t\|^2_{H_t} + \tfrac{1}{2}\|r_t\|^2 - \tfrac{1}{2}\left\|H_t^{-1} A_t^\top r_t\right\|^2_{H_t}. \tag{27}$$

The unique exact minimizer $q_t^\star$ therefore satisfies

$$\begin{aligned} q_t^\star &= \mathrm{prox}^{H_t}_{\lambda\,\mathrm{TV}_h + I_{\mathcal{C}_t}}(\zeta_t), \\ 0 &\in A_t^\top(A_t(q_t^\star - q_t) - r_t) + \mu(q_t^\star - q_t) + \lambda\partial\,\mathrm{TV}_h(q_t^\star) + N_{\mathcal{C}_t}(q_t^\star). \end{aligned} \tag{28}$$

Strong convexity and the nonempty compact feasible set ensure existence and uniqueness. This local variable-metric proximal interpretation belongs to the established family of Gauss–Newton methods with convex penalties (Salzo & Villa, 2012; Jin & Zhong, 2013).

Discrete total variation. Let $D_h q = (D_{h,1}q, D_{h,2}q)$, with $(D_{h,1}q)_{ij} = q_{i,j+1} - q_{ij}$ and $(D_{h,2}q)_{ij} = q_{i+1,j} - q_{ij}$ when the neighbor exists, and zero at the terminal boundary. The frozen penalty is

$$\mathrm{TV}_h(q) = \sum_{i,j} \sqrt{(D_{h,1}q)^2_{ij} + (D_{h,2}q)^2_{ij}} = \max_{\|p_{ij}\|_2 \le 1} \langle p, D_h q\rangle. \tag{29}$$

The linear difference operator D_h is distinct from the scalar data misfit $D(q; y)$ in Equation (2).

ADMM updates. Set $b_t = A_t q_t + r_t$ and $f_t(q) = \frac{1}{2}\|A_t q - b_t\|^2 + \frac{\mu}{2}\|q - q_t\|^2$. Introducing $z = D_h q$ and $v = q$ separates the TV term and the constraints:

$$\min_{q,z,v}\; f_t(q) + \lambda \sum_{i,j} \|z_{ij}\|_2 + I_{\mathcal{C}_t}(v) \quad \text{subject to} \quad z = D_h q, \quad v = q. \tag{30}$$

We apply ADMM with blocks q and (z, v) (Boyd et al., 2011). During this inner solve, t is fixed and ℓ indexes ADMM iterations. Initialize $q^0 = v^0 = q_t$, $z^0 = D_h q_t$, and the scaled duals $u^0 = w^0 = 0$. For penalty $\beta_t > 0$, the updates are

$$\begin{aligned} (A_t^\top A_t + \mu I + \beta_t D_h^\top D_h + \beta_t I) q^{\ell+1} &= A_t^\top b_t + \mu q_t + \beta_t D_h^\top (z^\ell - u^\ell) + \beta_t (v^\ell - w^\ell), \\ z_{ij}^{\ell+1} &= \mathrm{shrink}_{\lambda/\beta_t}((D_h q^{\ell+1} + u^\ell)_{ij}), \\ v^{\ell+1} &= \mathrm{proj}_{\mathcal{C}_t}(q^{\ell+1} + w^\ell), \\ u^{\ell+1} &= u^\ell + D_h q^{\ell+1} - z^{\ell+1}, \qquad w^{\ell+1} = w^\ell + q^{\ell+1} - v^{\ell+1}. \end{aligned} \tag{31}$$

Here $\mathrm{shrink}_\tau(a) = (1 - \tau/\|a\|)_+ a$. The penalty $\beta_t = \sqrt{\mu(\sigma_{\max}(A_t)^2 + \mu)}$ is fixed within each local solve. The quadratic update uses a DCT inverse for $(\mu + \beta_t)I + \beta_t D_h^\top D_h$ and a row-space Woodbury correction.

Stopping criteria and line search. The inner solve permits at most 4000 ADMM iterations and checks the local gap certificate from Appendix D every 10 iterations and at termination. Early stopping requires a gap of at most $9 \times 10^{-6} \max(1, \Psi_t(q_t))$ and a local objective no larger than $\Psi_t(q_t) + 10^{-8}$.

The projected variable v^ℓ supplies the candidate $q_t^{\mathrm{cand}} \in \mathcal{C}_t$. A line search tests $q_t + a(q_t^{\mathrm{cand}} - q_t)$ for $a \in \mathcal{A} = \{1, 1/2, 1/4, 1/8, 1/16, 0\}$ using the nonlinear objective $\Phi(q) = D(q; y) + \lambda \, \mathrm{TV}_h(q)$, with D defined in Equation (2). The lowest-value trial gives q_{t+1} only if it strictly improves $\Phi(q_t)$; otherwise the path stops and retains q_t.

D Local gap certificate

Proposition 1 (A computable local gap bound)**.** *Fix the outer index t, let f_t be the quadratic part defined in Appendix C, and let $q \in \mathcal{C}_t$. For any dual field with $\|p_{ij}\| \le \lambda$, define*

$$a = \nabla f_t(q) + D_h^\top p, \qquad s = \mathrm{proj}_{\mathcal{C}_t}(q - a/\mu) - q, \tag{32}$$

$$G_t(q, p) = \lambda \, \mathrm{TV}_h(q) - \langle p, D_h q \rangle - \langle a, s \rangle - \tfrac{\mu}{2} \|s\|^2 . \tag{33}$$

For the unique minimizer $q_t^\star$ of Equation (26),

$$0 \le \Psi_t(q) - \Psi_t(q_t^\star) \le G_t(q, p), \qquad \|q - q_t^\star\| \le \sqrt{2 G_t(q, p)/\mu}. \tag{34}$$

Proof. Strong convexity of f_t and dual feasibility imply, for every feasible v,

$$\Psi_t(v) \ge f_t(q) + \langle \nabla f_t(q), v - q \rangle + \tfrac{\mu}{2} \|v - q\|^2 + \langle p, D_h v \rangle \tag{35}$$

$$= f_t(q) + \langle p, D_h q \rangle + \langle a, v - q \rangle + \tfrac{\mu}{2} \|v - q\|^2 . \tag{36}$$

Completing the square shows that the last two terms have their minimum over $\mathcal{C}_t$ at $v = q + s$. Hence $L = f_t(q) + \langle p, D_h q \rangle + \langle a, s \rangle + \mu \|s\|^2 / 2$ is a lower bound on $\Psi_t(q_t^\star)$, and $\Psi_t(q) - L = G_t(q, p)$. To obtain the distance estimate, apply strong convexity on the convex box: for $0 < \theta < 1$,

$$\Psi_t(q_t^\star) \le \Psi_t((1-\theta) q_t^\star + \theta q) \le (1-\theta) \Psi_t(q_t^\star) + \theta \Psi_t(q) - \tfrac{\mu}{2} \theta (1-\theta) \|q - q_t^\star\|^2 .$$

Divide by θ and let $\theta \to 0$, giving $\Psi_t(q) - \Psi_t(q_t^\star) \ge \mu \|q - q_t^\star\|^2 / 2$. Combining the bounds proves the claim. □

For the ADMM candidate $q = v^\ell$, the code obtains a feasible dual by projecting $\beta_t u^\ell$ onto $\{p : \|p_{ij}\|_2 \le \lambda\}$, using $p = 0$ if $\lambda = 0$. This gives a bound for the fixed local subproblem without requiring an exact dual optimizer.

E Implementation details

Numerical constants. For $N = 96$, $n = N^2$, $q_{\max} = 0.108$, and $F = K = 3$, the solver uses $\alpha = 0.3 \|B\|_F^2 / n$, $\mu = 0.3\alpha$, $\lambda = 0.6 \alpha s_q$, $\delta_{\mathrm{GN}} = 0.02$, and line-search fractions $\{1, 1/2, 1/4, 1/8, 1/16, 0\}$. The Transformer uses $C = 24$, $n_h = 4$, $d = 24$, $C_V = 80$, and correction radius $\rho = 0.02$.

Inputs have shape $[b, 4, 4, 96, 96]$ (batch, iteration, channel, and spatial axes); outputs have shape $[b, 96, 96]$. The 288 visible complex observations give $B \in \mathbb{R}^{576 \times 9216}$. Convolutions use biases, unit stride, and zero padding for 3×3 kernels. Table 4 specifies the Transformer–Local model.

Alternative decoders. The Transformer variants differ only in the decoder after the second group normalization. The FNO decoder has two width-24 Fourier layers and a final 1×1 projection. Each layer uses orthonormal `rfft2`/`irfft2`, retaining indices 0:8 and -8: on the first axis and 0:8 on the second, with two independent complex $24 \times 24 \times 8 \times 8$ multipliers. Other coefficients are zero. A 1×1 convolutional branch is added before GELU. The decoder feeds the scalar head without an internal input skip connection. The U-Net decoder uses the block below with $w = 24$. The shared components have 151041 parameters; the Local, FNO, and U-Net decoders have 1800, 296712, and 250560.

Table 4: Exact Transformer–Local architecture. Parameters include biases and affine normalization.

Component	Specification	Real parameters
Stem	3×3, $4 \to 24$, GELU	888
Query / key	Each: 3×3 conv $24 \to 24 \to 48 \to 96$; GELU after each; average pool 2 after first two; global average pool after third	114384
Values	3×3 conv $24 \to 80 \to 24$, GELU after each	34664
Attention output	1×1 conv $24 \to 24$	600
Two normalizations	GroupNorm(1,24), affine, $\varepsilon = 10^{-5}$	96
Iteration embedding	4×96; added to Q/K descriptors	384
Local decoder	Two 1×1 convolutions $24 \to 24$, each followed by GELU; final 1×1 projection $24 \to 24$	1800
Output head	1×1 conv $24 \to 1$, zero initialization	25
Total	Four heads; 24 descriptor and six value channels per head	152841

U-Net block. The encoder and bottleneck have widths $w, 2w, 4w$. Each block has two 3×3 convolutions with GELU; each encoder block is followed by 2×2 average pooling with stride two. Each decoder stage upsamples bilinearly to the skip-feature size (`align_corners=False`), halves the channels with a 1×1 convolution, and concatenates the encoder feature. Two-convolution blocks then map $4w \to 2w \to 2w$ and $2w \to w \to w$. A final 1×1 projection maps $w \to w$. This block has no normalization or dropout.

Vanilla baselines and input assembly. Single-frame models receive q_3/s_q. Complete inputs concatenate $(S^{-1}Z_0, \ldots, S^{-1}Z_3)$ into 16 channels, ordered by iteration, then medium and three adjoint fields. A 1×1 lift with GELU gives width 16 for FNO or 17 for U-Net. FNO uses two Fourier layers as above, with $16 \times 16 \times 8 \times 8$ complex multipliers, followed directly by a 1×1 scalar head. U-Net uses the width-17 block followed by the same type of head. All heads are zero-initialized, with the bounded output defined in Equation (8).

Table 5: Architectures in Table 1. Counts include biases and affine normalization; complex weights count as two real parameters.

Model	Input	Spatial network	Parameters
Transformer–Local	Four tokens	Width 24, two pointwise layers + projection	152841
Transformer–FNO	Four tokens	Width 24, two Fourier layers	447753
Transformer–U-Net	Four tokens	Two levels, widths $24, 48, 96$	401601
vanilla FNO	q_3	Width 16, two Fourier layers	131665
vanilla FNO	Concatenated	Width 16, two Fourier layers	131905
vanilla U-Net	q_3	Two levels, widths $17, 34, 68$	125886
vanilla U-Net	Concatenated	Two levels, widths $17, 34, 68$	126141

Normalization scales and receiver selection. The fixed input scales are $s_q = 0.08$ and adjoint RMS values $(0.0984411375, 0.0604584918, 0.0448125535)$, computed from distinct states within each training trajectory. Repeated states are excluded only from this RMS calculation. Observation scales are $(2.97023368, 7.76225424, 15.01252556)$, shared by frequency position across both bands, including their different third wave numbers. The fixed visible receiver indices (zero-based) are

$$(0, 2, 3, 9, 11, 13, 14, 17, 21, 23, 28, 31),$$
$$(1, 2, 5, 6, 9, 10, 11, 14, 15, 19, 23, 28),$$
$$(1, 2, 3, 6, 9, 10, 11, 13, 14, 15, 21, 23).$$

All eight incidences are retained at each selected receiver and angles are uniform over $[0, 2\pi)$.

Reference data and noise distribution. Truth at 96^2 uses cell averages over four subcells per axis. Reference scattering is computed at 192^2 and checked against 288^2. Noise is $d_{\text{noisy}} = d +$

$0.03 \|d\|_2 (a + ib)/\sqrt{2m}$, where $m = 3 \cdot 256$ and $a, b \in \mathbb{R}^m$ have independent standard Gaussian entries.

F INTERIOR MASK AND ERROR AGGREGATION

Interior definition. The support is $q^\dagger \geq 0.02$ on the 96^2 grid. It is eroded once with the default nearest-neighbor binary-erosion structure. The original support is split into connected components; only eroded components with at least four remaining pixels contribute to I. This truth-derived mask is used only for evaluation.

Error aggregation. Each relative error in Equations (10)–(11) is computed per sequence, then averaged over the three profiles and two bands within each shape and over the 24 shapes. The bootstrap intervals in Section 5.1 use the 2.5th and 97.5th percentiles of the resampled mean paired differences. They describe variation across test shapes for fixed trained models.

G INPUT INTERVENTIONS AND PARAMETER DEPENDENCE

Table 2 reports the input interventions under the training protocol in Section 5.1. Within each architecture, the runs share initialization, sample order, optimizer settings, and fixed normalization, with checkpoints selected independently by validation error.

Repeated last token. Identical tokens produce identical values, so $\sum_t w_t^a V^a = V^a$. The output therefore becomes independent of 114768 query/key and iteration-embedding parameters in each Transformer. The Transformer–Local model retains 38073 effective parameters out of 152841. The nominal architecture is unchanged, but its effective capacity is reduced.

Zero adjoint channels. Zeroing the three adjoint channels preserves all medium states and makes 648 input weights ineffective in each Transformer, leaving 152193 effective parameters in Transformer–Local.

Concatenated and single q_3 inputs. Vanilla FNO and U-Net with concatenated inputs have 131905 and 126141 parameters; either intervention removes dependence on 192 and 204 parameters, respectively. Their single q_3 counterparts have 131665 and 125886 parameters.

H NOISE ROBUSTNESS AND GRID TRANSFER

Table 6 varies noise at grid 96^2 and grid resolution at 3% noise, using all 144 test sequences per condition. Neural models retain their complete-input checkpoints trained at 3% noise on 96^2, without retraining. The table reports common-grid medium errors and native-grid scattering errors against clean references; Table 7 additionally reports native-grid medium errors.

Table 6: Noise and grid transfer: relative L^2 errors (%) on all 144 sequences. Neural models use complete trajectories and are trained at 3% noise on 96^2. †In 93 of 144 test sequences at 0% noise and 1 of 144 at 1%, at least one inner GN–TV solve within the first three steps reached the 4000-iteration ADMM limit without meeting the stopping criterion. The resulting feasible approximations were used in the subsequent line search.

	Noise at grid 96^2				Grid at 3% noise	
Method	$0\%^\dagger$	$1\%^\dagger$	3%	5%	64^2	128^2
Medium reconstruction: E_{field}						
GN–TV3	9.023	9.646	13.964	20.304	20.277	14.739
GN–TV8	8.979	9.610	13.957	20.344	20.271	14.714
vanilla FNO	8.915	9.284	12.299	17.333	18.966	12.452
vanilla U-Net	8.082	8.271	10.268	14.135	21.248	11.457
Transformer–Local	6.136	6.531	9.262	14.099	19.186	10.780
Transformer–FNO	6.968	7.320	9.827	14.255	19.026	11.063
Transformer–U-Net	6.673	6.892	9.126	13.261	20.323	10.504
Visible scattering: E_{vis}						
GN–TV3	3.378	3.682	5.740	8.698	6.495	5.615
GN–TV8	3.366	3.672	5.734	8.696	6.491	5.608
vanilla FNO	3.990	4.059	5.480	8.294	6.721	5.357
vanilla U-Net	4.492	4.553	5.548	7.582	12.090	6.154
Transformer–Local	3.225	3.434	4.989	7.942	9.101	4.770
Transformer–FNO	3.702	3.914	5.392	7.778	8.828	5.208
Transformer–U-Net	3.635	3.728	4.952	7.315	10.864	4.812
Unobserved scattering: E_{un}						
GN–TV3	2.107	2.271	3.891	6.460	4.547	3.824
GN–TV8	2.098	2.262	3.885	6.459	4.543	3.816
vanilla FNO	2.475	2.373	2.909	4.575	3.871	2.718
vanilla U-Net	2.227	2.253	2.877	4.038	6.253	3.526
Transformer–Local	1.647	1.702	2.416	4.005	4.783	2.429
Transformer–FNO	2.013	1.959	2.540	3.616	5.064	2.526
Transformer–U-Net	1.869	1.839	2.251	3.286	5.565	2.317

Table 7: Native-grid full-field recovery at 3% noise: relative L^2 errors (%) on all 144 sequences. These results supplement the common-grid results in Table 6. All neural models use complete-input checkpoints trained at 96^2.

Method	64^2 native	96^2 native	128^2 native
GN–TV3	13.048	13.964	15.027
GN–TV8	13.048	13.957	15.016
vanilla FNO	12.031	12.299	13.187
vanilla U-Net	15.380	10.268	11.793
Transformer–FNO	11.852	9.827	11.111
Transformer–U-Net	13.954	9.126	10.461
Transformer–Local	12.213	9.262	10.661

I QUALITATIVE RECONSTRUCTION AND ERROR COMPARISONS

Figures 4 and 5 extend the fixed-case visualization in Figure 3 with the single-frame vanilla FNO and the Transformer variants with FNO and U-Net decoders. The paired views show the reconstructed structures and the spatial distribution of their errors for the same four test media.

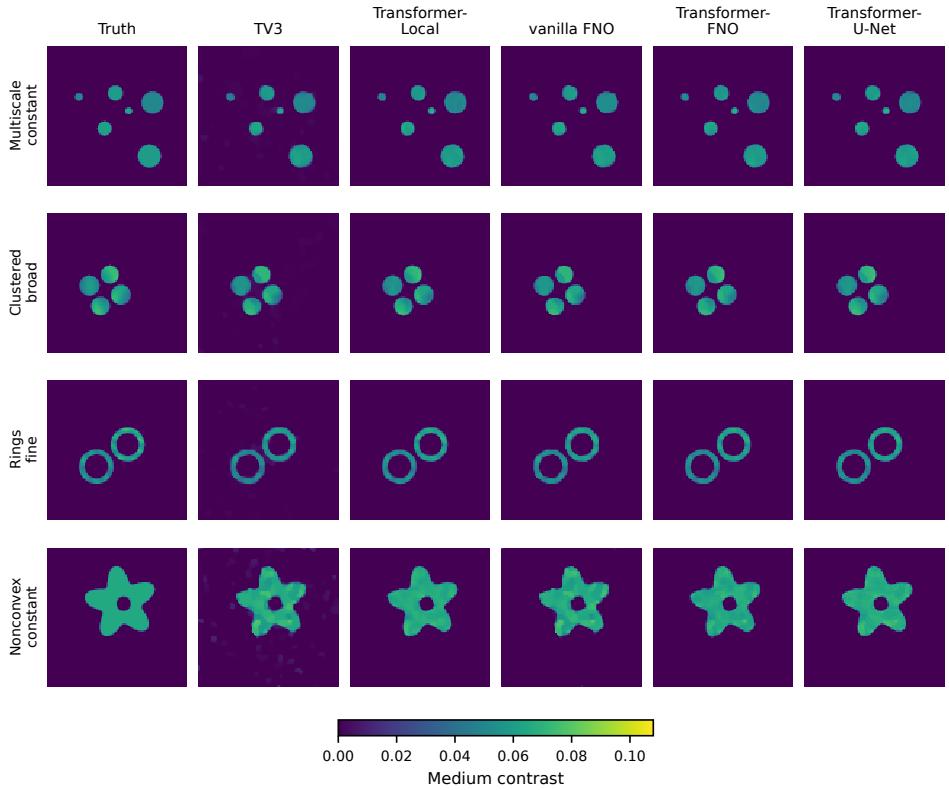


Figure 4: Medium reconstructions at nominal 3% observation noise on the 96×96 grid. Rows show four fixed original-band test cases: multiscale disks (shape 000, constant profile), clustered disks (006, smooth-broad), rings (012, smooth-fine), and a nonconvex domain (018, constant). Cases were selected by a predefined geometry–profile rule, independently of reconstruction errors. From left to right: ground truth, three-step GN–TV (labeled TV3), Transformer–Local, vanilla FNO, Transformer–FNO, and Transformer–U-Net. All panels use the same contrast scale $[0, 0.108]$. The neural models use the validation-selected checkpoints reported in Table 1; vanilla FNO corrects the final iterate q_3, while the three Transformer variants use the complete medium–adjoint trajectory.

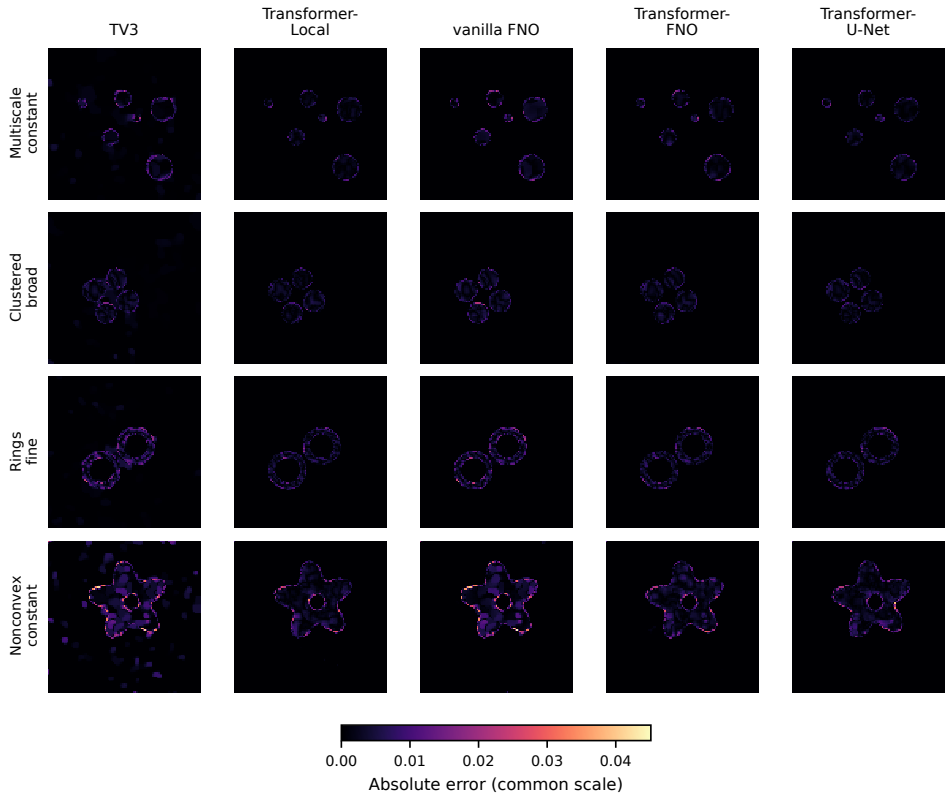


Figure 5: Pointwise absolute errors $|\widehat{q} - q^{\dagger}|$ corresponding to Figure 4. The four cases retain the same row order; columns show TV3, Transformer–Local, vanilla FNO, Transformer–FNO, and Transformer–U-Net, respectively. All panels share a color scale from zero to the largest displayed absolute error, with darker colors indicating smaller errors. This common scale reveals the spatial distribution of boundary errors, interior variations, and background artifacts. Errors are displayed without interpolation, and no panel is normalized separately.