WHAT DO INRS FIND DIFFICULT? UNDERSTANDING AND EXPLOITING COORDINATE-LEVEL TRAINING DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Implicit neural representations (INRs) are typically evaluated through aggregate reconstruction quality, obscuring how individual coordinates are learned over the course of optimization. We study this process through *coordinate difficulty*, defined from the reconstruction-error trajectory of each coordinate during training. Across 50 image and 50 audio signals and four INR representations, we find that coordinate difficulty exhibits substantial structure: it can be highly reproducible across random seeds, is partly predictable from local properties of the target signal, and differs systematically across representations even when they learn the same signal. We then ask whether these training dynamics are useful for optimization. Sampling coordinates according to difficulty accumulated over the complete training history improves the hardest regions of a signal but can degrade overall reconstruction, particularly for ReLU-based representations. We find that how difficulty is aggregated over time is critical: estimators that emphasize recent errors yield substantially more consistent behavior than complete-history difficulty. A simple recent-difficulty sampler improves reconstruction over uniform sampling across all eight architecture–modality settings and remains competitive with existing adaptive sampling methods under a common fixed-step evaluation protocol. Together, these results show that coordinate-level training trajectories reveal both signal-associated and representation-dependent structure hidden by aggregate reconstruction metrics, and provide an actionable signal for understanding and guiding INR training.

1 INTRODUCTION

Implicit neural representations (INRs) store signals in a neural network by learning a function that maps coordinates to signal values, and are used for images, audio, and neural fields alike (Xie et al., 2022). A wide range of architectures have been introduced for INRs: SIREN uses periodic activations to represent the signal Sitzmann et al. (2020); Fourier-feature and positional-encoding schemes transform coordinates into a higher-frequency space before passing them to a ReLU network (Tancik et al., 2020; Mildenhall et al., 2020); and multiresolution grids instead interpolate learned, spatially local features (Müller et al., 2022). These architectures are usually compared by how well the resulting network reconstructs the signal. But two networks that reach the same final reconstruction quality can arrive there by very different routes: one may fit most of a signal almost immediately and spend the rest of training on a small stubborn region, while another may improve more gradually across the signal. This difference is invisible to a single end-of-training number.

We call this coordinate-level view *coordinate difficulty*: for each coordinate in a signal, we track its reconstruction error over training and summarize the trajectory as one score. Figure 1 illustrates why this matters: three coordinates can reach similar final errors while following substantially different trajectories — one converges early, one degrades and recovers, and one remains difficult until late in training. Aggregate PSNR or MSE cannot distinguish these cases, but coordinate difficulty can.

We use this view to ask four questions. *Is it reproducible?* If independent training runs on the same signal repeatedly flag the same coordinates as hard, difficulty reflects stable structure in the signal–representation pair rather than only stochastic variation across runs. *How much of it is explained*

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

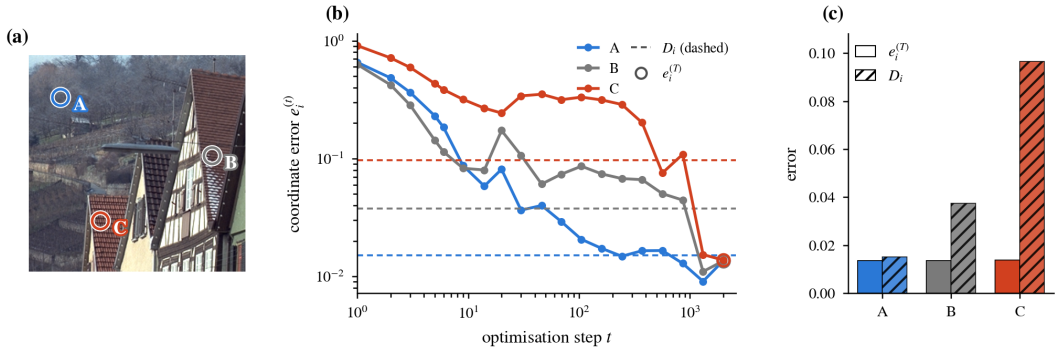


Figure 1: **Final error can hide substantial differences in how coordinates are learned.** (a) Three coordinates from the same image. (b) Their reconstruction errors throughout training; dashed lines indicate mean trajectory error D_i, while circles indicate final error $e_i^{(T)}$. (c) Although the three coordinates reach similar final errors, their coordinate difficulties differ substantially because their errors persist for different amounts of training.

by the target signal itself? If coordinate difficulty can largely be predicted from properties already present in the target, then it may mostly reflect signal structure rather than the learning process. We therefore ask how much difficulty is explained by the target alone, and what structure remains after accounting for it. *Does it transfer across representations?* If different architectures identify the same coordinates as difficult, then much of that difficulty may be inherent to the signal. If they disagree, the differences reveal how the choice of representation changes what the model finds difficult to learn. *Can it be used to improve training?* If coordinate-level error trajectories contain actionable information about learning, then that information may also be useful for allocating training effort. We therefore test whether emphasizing difficult coordinates improves reconstruction.

Across 50 image and 50 audio signals and four INR representations (SIREN, ReLU with Fourier features, ReLU with positional encodings, and a dense multiresolution grid), we find that coordinate difficulty exhibits strongly architecture-dependent reproducibility across seeds, is partly but not fully predictable from properties of the target signal, and depends substantially on the representation used to learn it — different representations of the same signal can disagree sharply on which parts of it were hard to learn. Turning to whether this is actionable, we find that sampling coordinates by difficulty accumulated over the complete training history can improve the hardest coordinates, but can substantially degrade overall reconstruction and behaves inconsistently across representations. How coordinate errors are aggregated over time turns out to matter: estimators that emphasize recent errors provide a more consistent training signal than complete-history difficulty, although the preferred temporal scale varies across representations and modalities. The resulting recent-difficulty sampler improves reconstruction over uniform sampling in every one of our eight architecture–modality settings and remains competitive with existing adaptive samplers under a common fixed-step evaluation protocol.

We summarize our main contributions as:

- We define and systematically study coordinate difficulty, a framework for summarizing per-coordinate training-error trajectories in INRs, and use it to characterize reproducibility, target-signal dependence, and representation dependence (Section 4).
- We study coordinate difficulty as an online training signal and show that its temporal aggregation matters: emphasizing recent errors provides more consistent gains than retaining the complete training history. A fixed 128-step recent-difficulty sampler improves reconstruction over uniform sampling in all eight architecture–modality settings and remains competitive with existing adaptive samplers (Section 5).
- We provide a cross-architecture, cross-modality evaluation of coordinate-level INR training dynamics, spanning four representation families and two modalities under a common experimental protocol.

2 Related Work

2.1 Training Dynamics and Example Difficulty

Training dynamics have been used to characterize differences between individual training examples. Example forgetting (Toneva et al., 2019), dataset cartography (Swayamdipta et al., 2020), and error-based measures such as EL2N and GraNd (Paul et al., 2021) show that examples can exhibit persistent differences in how they are learned, with some of this structure reproducible across seeds or useful for selecting training data (Baldock et al., 2021; Agarwal et al., 2022).

We adapt this perspective from individual examples to coordinates within a continuously represented signal. Unlike example-level analyses, coordinates within a signal are explicitly spatially or temporally related and share the same underlying signal, while their errors depend on the target, representation, and optimization state.

2.2 Adaptive Coordinate Selection for INRs and Neural Fields

Several methods use reconstruction error or related signals to allocate INR training toward selected coordinates. Soft Mining (Kheradmand et al., 2024) moves sampling particles toward high-loss regions using a Langevin-style update, while INT (Zhang et al., 2024) selects coordinates using reconstruction discrepancy. EVOS (Zhang et al., 2025) maintains an error ranking between periodic full-domain evaluations and updates selections using evolutionary operations, and NINT (Zhang et al., 2026) combines reconstruction-based selection with an NTK-derived score (Jacot et al., 2018).

Related work also uses previously observed errors for sampling. Otonari et al. (2022) use last-observed per-pixel losses for NeRF sampling, while ERF uses running error statistics (Aroudj et al., 2022). More broadly, adaptive sample selection has used current losses, historical loss statistics, and changes in loss over training (Loshchilov & Hutter, 2016; Schaul et al., 2016; Chang et al., 2017).

These works show that reconstruction error can guide coordinate selection, but differ in how that error is represented and used. We study the structure of coordinate-level training dynamics directly, and examine how their temporal aggregation affects adaptive sampling across representations and modalities.

3 Coordinate Difficulty in Implicit Neural Representations

We examine INR optimization at the level of individual coordinates by following their reconstruction errors over time. Each trajectory is summarized by a scalar *coordinate difficulty*, yielding a difficulty map over the signal (Figure 2).

3.1 Coordinate Difficulty

For a signal $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, an INR f_θ maps coordinate $\mathbf{x}_i$ to signal value $\mathbf{y}_i$. At optimization step t, we define the reconstruction error at coordinate i as

$$e_i^{(t)} = \| f_{\theta_t}(\mathbf{x}_i) - \mathbf{y}_i \|_2 \,. \tag{1}$$

For images, we use RGB Euclidean error, $e_i^{(t)} = \|\hat{\mathbf{y}}_i^{(t)} - \mathbf{y}_i\|_2$, while for audio we use absolute amplitude error, $e_i^{(t)} = |\hat{y}_i^{(t)} - y_i|$.

Rather than considering only the final error $e_i^{(T)}$, we follow its evolution over optimization:

$$\mathcal{E}_i = \left(e_i^{(0)}, e_i^{(1)}, \ldots, e_i^{(T)} \right). \tag{2}$$

We summarize this trajectory by its mean,

$$D_i = \frac{1}{T+1} \sum_{t=0}^{T} e_i^{(t)}, \tag{3}$$

which we refer to as the *coordinate difficulty* of coordinate i. Thus, D_i captures how much reconstruction error persists over optimization, rather than only the error remaining at the final step.

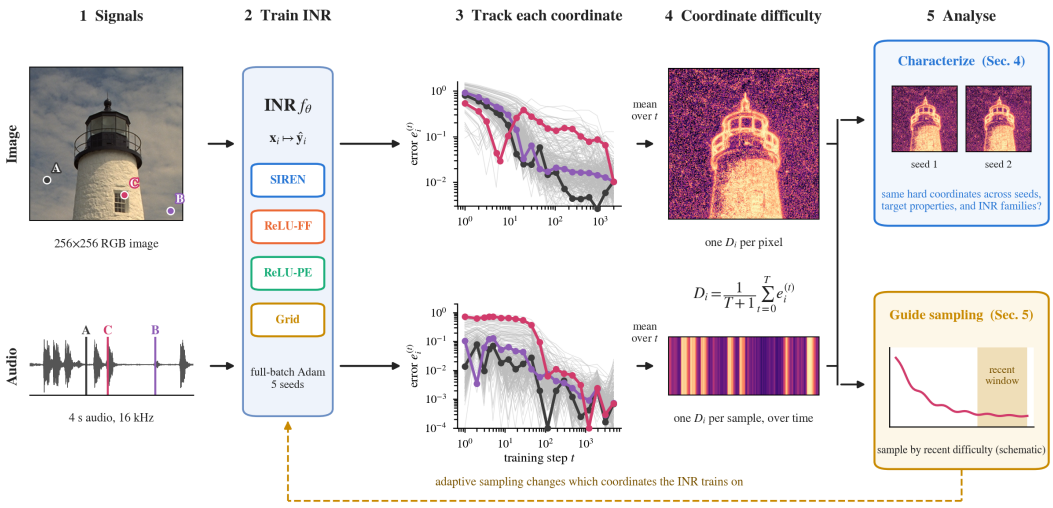


Figure 2: **Overview of coordinate difficulty.** Reconstruction error is recorded for each coordinate throughout optimization and averaged over time to obtain D_i. The resulting difficulty map is used for subsequent analysis and adaptive sampling.

3.2 Dataset

We evaluate 50 image and 50 audio signals spanning a range of signal structures. The image collection contains natural photographs, textures, blurred images, and synthetic signals; the audio collection contains speech, environmental sounds, musical and vocal recordings, and synthetic signals. Images are represented at 256×256 RGB resolution, while audio clips are four-second, 16 kHz mono recordings. Dataset composition and preprocessing are detailed in Appendix A.1.

3.3 INR Architectures

We consider four INR representations: SIREN, ReLU with Fourier features (ReLU-FF), ReLU with positional encodings (ReLU-PE), and a dense multiresolution feature grid. The dense grid is an unhashed implementation of the multiresolution grid used in Instant-NGP. These models span substantially different ways of encoding coordinate-dependent structure: periodic activations, fixed input encodings, and learned spatially localized features.

We use architecture-specific learning rates chosen for stable convergence and reasonable reconstruction quality, while approximately matching the number of trainable parameters across models. This limits differences in model size as a potential confound.

3.4 Experimental Protocol

For the analyses in Section 4, each signal–architecture pair is trained with five independent random seeds using full-batch Adam (Kingma & Ba, 2015) and the corresponding architecture-specific learning rate. Image models are optimized for 2,000 steps and audio models for 5,000 steps.

Full-batch optimization lets us observe every coordinate at every step. These errors are used to construct $\mathcal{E}_i$ and D_i, giving a difficulty value for every coordinate, signal, architecture, and seed.

The adaptive sampling experiments in Section 5 instead operate online, where difficulty must be estimated from coordinates encountered during optimization. Complete dataset, architecture, encoding, and optimization details are provided in the Appendix.

4 UNDERSTANDING COORDINATE DIFFICULTY

4.1 IS COORDINATE DIFFICULTY REPRODUCIBLE?

We first ask whether independent runs of the same INR identify the same coordinates as difficult. For each signal and architecture, we compute pairwise Spearman correlations between the coordinate-difficulty rankings produced by the five training seeds, giving ten seed pairs per setting.

Table 1: **Coordinate-level reproducibility across independent training seeds.** Values are median pairwise Spearman correlations across signals.

Modality	SIREN	ReLU-FF	ReLU-PE	Grid
Audio	0.902	0.234	0.316	0.913
Image	0.933	0.624	0.668	0.920

Table 1 shows that SIREN and the dense grid produce highly consistent difficulty rankings across seeds, with median correlations above 0.90 in both modalities. ReLU-FF and ReLU-PE are less reproducible, particularly for audio.

These results concern the ranking of coordinates within each signal. Reproducibility is even higher at the level of entire signals. When D_i is averaged over all coordinates of each signal, seed-to-seed rankings within the same architecture have Spearman correlations of 0.989–0.999. This agreement remains high after excluding the 12 easiest and 12 hardest signals within each architecture ($\rho = 0.933$–0.998), rather than being driven only by the extremes of the difficulty distribution. For audio, mean signal difficulty is strongly associated with mean squared signal amplitude ($\rho = 0.82$–0.98 across architectures), so this result partly reflects differences in signal scale. The corresponding association with mean squared pixel intensity is much weaker for images ($\rho = 0.10$–0.30).

4.2 CAN THE TARGET SIGNAL EXPLAIN COORDINATE DIFFICULTY?

The reproducibility of difficulty suggests that it contains stable structure, but that structure could simply reflect properties of the target signal. We therefore ask how well coordinate difficulty can be predicted without using any information from the trained INR.

We compute 12 local descriptors from the target signal and use them to predict median log difficulty across seeds using histogram gradient boosting (Pedregosa et al., 2011) under held-out evaluation. No model predictions, gradients, parameters, or optimization state are used as features.

Table 2: **Target predictability and reproducibility of the remaining difficulty.** Target R^2 measures held-out prediction from target-only descriptors. Residual ρ measures seed reproducibility after subtracting the out-of-fold prediction.

Modality	Measure	SIREN	ReLU-FF	ReLU-PE	Grid
Audio	Target R^2	0.844	0.443	0.590	0.759
	Residual ρ	0.624	0.045	0.056	0.638
Image	Target R^2	0.843	0.601	0.648	0.690
	Residual ρ	0.623	0.307	0.349	0.759

Target-only descriptors predict a substantial fraction of coordinate difficulty, although the amount varies across representations. Difficulty generally increases with local audio energy and with image gradient magnitude and contrast, but the strength of these relationships differs substantially across architectures. In particular, ReLU-FF shows a flatter relationship than SIREN across both modalities.

Target-predictable structure is not the whole story. After subtracting the out-of-fold target-only prediction, the remaining difficulty is still reproducible across seeds for SIREN and the dense grid (Table 2). These residuals may contain target structure not captured by our descriptors, so they should not be interpreted as purely representation-specific. They nevertheless show that the measured target properties do not explain all repeatable coordinate difficulty.

4.3 Does Difficulty Transfer Across INR Families?

We next ask whether different INR families identify the same coordinates as difficult. Figure 3 illustrates this on a synthetic radial signal. All four architectures concentrate difficulty around the signal's concentric structure, but they select visibly different coordinates within it. For this example, the SIREN–grid hard sets overlap substantially more than the other SIREN pairs.

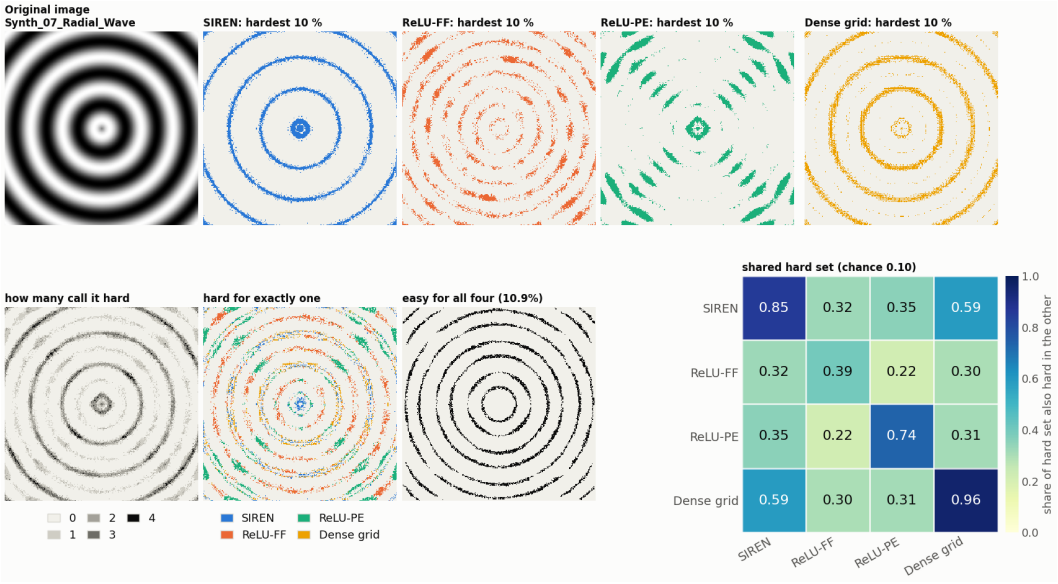


Figure 3: **Coordinate difficulty across INR families on a synthetic radial image.** The top row shows the target and the hardest 10% of coordinates for each architecture. The bottom row shows how often architectures agree on hard coordinates, coordinates identified as hard by only one architecture, coordinates considered easy by all four, and pairwise hard-set overlap.

Across the full dataset, we compare coordinate-difficulty rankings between each pair of architectures. Because some agreement may arise simply because the architectures respond to the same target structure, we also compare what remains after accounting for the 12 measured target descriptors.

For this comparison, we use a linear rank residualization. Difficulty ranks for each architecture are regressed on the ranks of the target descriptors, and the resulting residuals are aggregated across seeds into a consensus map. This differs from the nonlinear out-of-fold predictor in Table 2, which is used to measure how well target properties predict difficulty.

Table 3: Cross-architecture Spearman correlations of coordinate difficulty before and after accounting for measured target properties.

		S–FF	S–PE	S–Grid	FF–PE	FF–Grid	PE–Grid
Audio	Raw	.425	.518	.823	.462	.397	.491
	Residual	.060	.093	.292	.121	.035	.059
Image	Raw	.372	.733	.742	.473	.515	.645
	Residual	.209	.381	.364	.349	.376	.295

Raw difficulty shows substantial agreement for several architecture pairs, but this agreement falls after accounting for measured target properties (Table 3). For audio, cross-architecture correlations fall from 0.397–0.823 to 0.035–0.292 after residualization. For images, they fall from 0.372–0.742 to 0.209–0.381.

Low cross-architecture agreement is meaningful only when the residual difficulty of each representation is itself reproducible. Using the same residualization procedure on disjoint groups of seeds, residual difficulty remains strongly reproducible for audio SIREN and the dense grid ($\rho = 0.755$

and 0.792) and for all four image representations ($\rho = 0.606$–0.898). The substantially lower cross-architecture correlations in these settings therefore cannot be explained by seed variation alone. In contrast, residual reproducibility is low for audio ReLU-FF and ReLU-PE ($\rho = 0.115$ and 0.132), so the corresponding cross-architecture comparisons are inconclusive.

Together, these results show that coordinate difficulty contains both signal-associated and architecture-dependent structure. The target influences where difficulty arises, while the representation influences how it is distributed. We next ask whether this information can be used to guide training.

5 Application: Difficulty-Guided Coordinate Sampling

5.1 Can Coordinate Difficulty Guide Training?

We first ask whether coordinate difficulty can be used to guide optimization. A natural approach is to sample difficult coordinates more frequently, giving them more opportunities to train.

We begin by accumulating each coordinate's error over the complete training history. This successfully improves the region it targets: the hardest 10% of coordinates improve across all eight architecture–modality settings, with gains of up to $+3.98$ dB. However, this does not consistently improve the signal as a whole. Full-history sampling benefits SIREN and some dense-grid settings, but substantially degrades both ReLU representations. In these cases, the difficult coordinates improve while reconstruction of the remaining coordinates deteriorates enough to outweigh those gains.

Table 4: **Median reconstruction gain (dB) of full-history difficulty-guided sampling over uniform sampling.** Full-history sampling improves the hardest 10% of coordinates in every setting, but can degrade the remainder of the signal enough to reduce overall reconstruction quality.

	Image				Audio			
Region	**SIREN**	**FF**	**PE**	**Grid**	**SIREN**	**FF**	**PE**	**Grid**
Whole	+1.27	-2.08	-0.61	+0.05	+0.45	-2.80	-2.69	+0.61
Hard 10%	+3.11	+2.74	+2.51	+1.62	+1.95	+3.56	+3.98	+1.56
Other 90%	-0.15	-2.69	-1.42	-1.27	-0.22	-3.41	-3.37	-0.09

Coordinate difficulty therefore contains useful information about where training effort can be directed, but errors accumulated over the complete training history are not always a useful guide to where that effort should be allocated next.

5.2 How Much History Should the Sampler Remember?

The mixed performance of full-history sampling suggests that the usefulness of difficulty may depend on when the errors were observed. We therefore vary only how past coordinate errors are aggregated. We compare the most recently observed error (*Instant*), exponentially decayed history (*EMA*), complete training history (*History*), and a bounded window over the previous 128 training steps (*Window128*).

Temporal aggregation has a substantial effect (Figure 4). Complete-history difficulty performs poorly for the ReLU representations, whereas estimators that emphasize recent errors are more consistent. Window128 is the only estimator whose 95% bootstrap confidence interval lies above uniform in all eight settings, and it improves over both Instant and History in paired comparisons across all eight.

We sweep temporal horizons on the development set and select a 128-step window as a fixed operating point for the remaining experiments. Among the bounded windows, 128 steps provides the most consistent performance across representations and modalities, ranking highest in seven of eight settings. The preferred horizon nevertheless varies across settings, so 128 steps should not be interpreted as universally optimal (Appendix B).

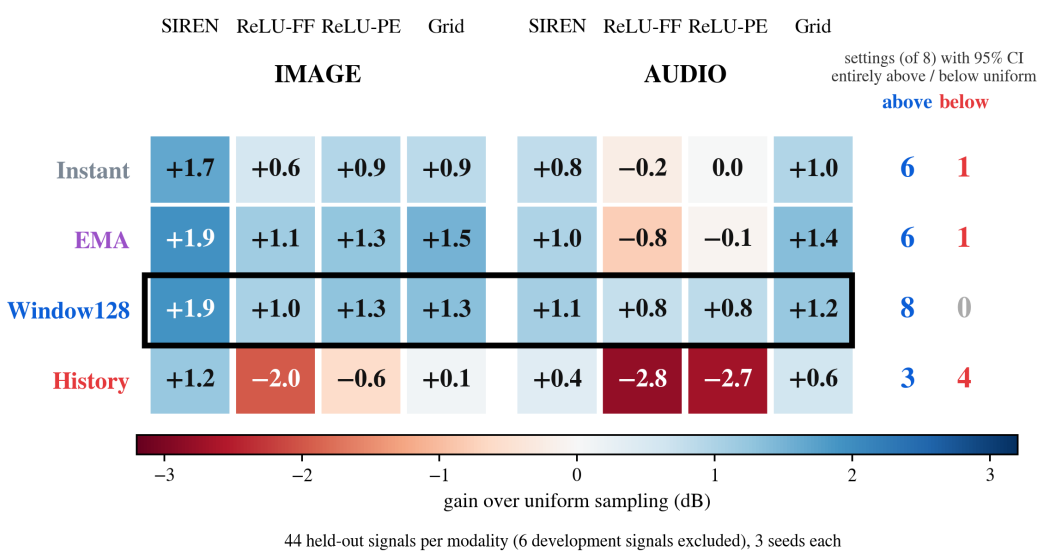


Figure 4: **Temporal aggregation determines whether difficulty is a useful sampling signal.** Cells show median whole-signal reconstruction gain (dB) over uniform sampling across the 44 held-out evaluation signals. The columns on the right count settings whose 95% bootstrap confidence interval lies entirely above or below zero. Complete-history difficulty performs poorly for the ReLU representations, whereas estimators emphasizing recent errors are more consistent. Window128 is the only estimator shown whose confidence interval lies above uniform in all eight architecture–modality settings.

5.3 Can Recent Difficulty Guide Training Across Representations?

We use this finding to define Recent-Difficulty Sampling (RDS), a simple sampler that prioritizes coordinates according to difficulty accumulated over a bounded recent history. RDS maintains observed coordinate errors, periodically ranks coordinates by their recent mean error, and samples higher-ranked coordinates more frequently.

For a coordinate i, let $r_i \in \{1, \dots, N\}$ denote its ascending rank by recent mean error. Its sampling probability is

$$p_i = \frac{(r_i/N)^\alpha}{\sum_{j=1}^{N} (r_j/N)^\alpha}, \qquad \alpha = 2. \tag{4}$$

We use the fixed 128-step operating point above, denoted *RDS-128*. RDS uses errors already observed during training and therefore requires no additional model evaluations for coordinate scoring. Full implementation details are provided in Appendix C.

On the 44 held-out evaluation signals per modality, RDS-128 improves whole-signal reconstruction across all eight architecture–modality settings, with gains of $+0.76$–$+1.92$ dB and improvements in 339 of 352 signal–architecture comparisons (96.3%). The gains are not confined to the hardest coordinates: both the hardest 10% and remaining 90% improve across all eight settings (Appendix C). RDS-128 also remains positive against uniform sampling without replacement, with gains of $+0.39$–$+1.87$ dB.

5.4 How Does Recent-Difficulty Sampling Compare with Existing Methods?

We finally compare RDS-128 with Soft Mining, INT, NINT, and EVOS under a common fixed-step protocol. Existing methods can achieve larger gains in particular settings, but their behavior varies substantially across representations. RDS-128 improves over uniform sampling across all eight architecture–modality settings under this protocol.

Table 5: **Median whole-signal reconstruction gain (dB) over uniform sampling under a common fixed-step protocol.** "–" indicates a modality not evaluated in the original paper.

	Image				Audio			
Method	**SIREN**	**FF**	**PE**	**Grid**	**SIREN**	**FF**	**PE**	**Grid**
RDS-128	+1.92	+1.04	+1.28	+1.34	+1.09	+0.76	+0.84	+1.20
Soft Mining	+1.64	+0.95	+0.77	+0.01	–	–	–	–
INT	+1.78	-2.45	-0.78	+1.82	+2.94	-4.62	-3.90	+2.84
NINT	+2.24	-0.81	-0.37	+2.34	+2.57	-3.89	-3.21	+2.24
EVOS	+0.85	-1.94	-0.71	-0.29	–	–	–	–

NINT produces larger median gains than RDS-128 on SIREN and Grid, while INT does so on audio SIREN and on Grid in both modalities. Both, however, substantially degrade the ReLU representations under this protocol. RDS-128 instead provides consistent positive gains without additional full-domain evaluations for scoring, at approximately 1.04–$1.28\times$ the wall-clock time of uniform sampling. Detailed timing results and exploratory equal-wall-clock comparisons are reported in Appendix D.1.

Taken together, these experiments distinguish retrospective difficulty from actionable difficulty. Errors accumulated over the complete training history identify coordinates that have been persistently difficult, but are not necessarily the best guide to where optimization effort should be allocated next. Across the representations considered here, emphasizing recent difficulty provides a more consistent training signal.

6 Limitations

Coordinate difficulty reflects a particular training process rather than an intrinsic property of a coordinate, and may therefore depend on optimization choices not systematically varied here. Our study is also limited to four INR representations and two modalities. The target-only descriptors capture only selected signal properties, so residual difficulty cannot be interpreted as purely representation-specific.

RDS is a proof-of-concept use of coordinate difficulty rather than an optimized sampler. Its performance may depend on the temporal estimator and other optimization choices, and our results do not establish 128 steps as a universally optimal window. Finally, the peer comparison uses a common fixed-step protocol rather than exhaustive method-specific tuning, and fixed-step gains do not imply a general wall-clock advantage.

7 Conclusion

We study INR learning through the reconstruction trajectories of individual coordinates. Coordinate difficulty reveals structure hidden by aggregate reconstruction metrics: difficult coordinates can be reproducible across training runs, are partly explained by properties of the target signal, and depend on the representation used to learn it.

This structure is also actionable. Sampling by complete-history difficulty can improve persistently difficult coordinates while harming overall reconstruction, whereas emphasizing recent difficulty provides a more consistent training signal across the representations studied. These results suggest that understanding INRs requires considering not only what they ultimately represent, but how different parts of a signal are learned over optimization.

AI use statement

In this work, we used generative AI tools to assist with code development, automating batches of experiments, literature search, drafting, and improving the flow and presentation of portions of the paper. AI-generated code was inspected and tested against the underlying experimental outputs. AI-assisted literature searches were checked against the cited papers rather than relying on generated summaries. We reviewed and edited all AI-assisted text and take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

Reproducibility statement

We provide code and configurations for all experiments, together with dataset construction procedures, random seeds, and analysis scripts used to produce the reported results. The main paper specifies the coordinate-difficulty definition, INR representations, training protocols, and adaptive-sampling procedure. The appendix provides the corresponding architecture and optimization hyperparameters, dataset composition and preprocessing, development–evaluation split, temporal-horizon selection procedure, RDS implementation details, and adaptive-sampling baseline configurations. Development signals used for configuration selection are explicitly separated from the 44 held-out evaluation signals. We also document implementation choices and computational costs for the adaptive-sampling baselines to support reproduction of the common-protocol comparisons.

References

Chirag Agarwal, Daniel D'souza, and Sara Hooker. Estimating example difficulty using variance of gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10358–10368, 2022. doi: 10.1109/CVPR52688.2022.01012.

Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1122–1131, 2017. doi: 10.1109/CVPRW.2017.150.

Samir Aroudj, Steven Lovegrove, Eddy Ilg, Tanner Schmidt, Michael Goesele, and Richard Newcombe. ERF: Explicit radiance field reconstruction from scratch. arXiv preprint arXiv:2203.00051, 2022. URL https://arxiv.org/abs/2203.00051.

Robert Baldock, Hartmut Maennel, and Behnam Neyshabur. Deep learning through the lens of example difficulty. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10876–10889, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/5a4b25aaed25c2ee1b74de72dc03c14e-Abstract.html.

G. Chaladze and L. Kalatozishvili. Linnaeus 5 dataset for machine learning, 2017. URL https://chaladze.com/l5/.

Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/2f37d10131f2a483a8dd005b3d14b0d9-Abstract.html.

Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. Neural audio synthesis of musical notes with WaveNet autoencoders. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1068–1077, 2017. URL https://proceedings.mlr.press/v70/engel17a.html.

Rich Franzen. Kodak lossless true color image suite, 1999. URL https://r0k.us/graphics/kodak/. Images released by Eastman Kodak Company for unrestricted usage.

Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5197–5206, 2015. doi: 10.1109/CVPR.2015.7299156.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html.

Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. Accelerating neural field training via soft mining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20071–20080, 2024. doi: 10.1109/CVPR52733.2024.01897.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. In *International Conference on Learning Representations (Workshop Track)*, 2016. arXiv:1511.06343.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Computer Vision – ECCV 2020*, pp. 405–421, 2020. doi: 10.1007/978-3-030-58452-8_24.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):1–15, 2022. doi: 10.1145/3528223.3530127.

Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 257–265, 2017. doi: 10.1109/CVPR.2017.35.

Takashi Otonari, Satoshi Ikehata, and Kiyoharu Aizawa. Non-uniform sampling strategies for NeRF on 360° images. In *33rd British Machine Vision Conference (BMVC)*, 2022. URL https://bmvc2022.mpi-inf.mpg.de/0344.pdf.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015. doi: 10.1109/ICASSP.2015.7178964.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20596–20607, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/ac56f8fe9eea3e4a365f29f0f1957c55-Abstract.html.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12 (85):2825–2830, 2011. URL https://www.jmlr.org/papers/v12/pedregosa11a.html.

Karol J. Piczak. ESC: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 1015–1018, 2015. doi: 10.1145/2733373.2806390.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7462–7473, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/53c04118df112c13a8c34b38343b9c10-Abstract.html.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9275–9293, 2020. doi: 10.18653/v1/2020.emnlp-main.746.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7537–7547, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/55053683268957697aa39fba6f231c68-Abstract.html.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

Allan G. Weber. The USC-SIPI image database: Version 5. USC-SIPI Report 315, University of Southern California, Signal and Image Processing Institute, 1997. URL https://sipi.usc.edu/database/.

Julia Wilkins, Prem Seetharaman, Alison Wahl, and Bryan Pardo. VocalSet: A singing voice dataset. In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 468–474, 2018.

Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 41(2):641–676, 2022. doi: 10.1111/cgf.14505.

Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, pp. 711–730, 2012. doi: 10.1007/978-3-642-27413-8_47.

Chen Zhang, Steven Tin Sui Luo, Jason Chun Lok Li, Yik Chung Wu, and Ngai Wong. Nonparametric teaching of implicit neural representations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 59435–59458, 2024. URL https://proceedings.mlr.press/v235/zhang24ap.html.

Chen Zhang, Wei Zuo, Bingyang Cheng, Yikun Wang, Wei-Bin Kou, Yik-Chung Wu, and Ngai Wong. NTK-guided implicit neural teaching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17248–17258, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhang_NTK-Guided_Implicit_Neural_Teaching_CVPR_2026_paper.html.

Weixiang Zhang, Shuzhao Xie, Chengwei Ren, Siyi Xie, Chen Tang, Shijia Ge, Mingzi Wang, and Zhi Wang. EVOS: Efficient implicit neural training via EVOlutionary selector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30472–30482, 2025. doi: 10.1109/CVPR52734.2025.02837.

APPENDIX

A EXPERIMENTAL DETAILS

A.1 DATASETS AND PREPROCESSING

The collection of audio and images is curated rather than randomly sampled from a single set. Table 6 summarizes the source composition.

Table 6: Composition of the image and audio datasets.

Images

Source	n
Blurred [16]	5
DIV2K [2]	8
Kodak [8]	6
Linnaeus5 [5]	7
SIPI [27]	7
Set14 [30]	4
Synthetic	7
Urban100 [9]	6
Total	50

Audio

Source	n
ESC-50 [21]	12
LibriSpeech [18]	10
NSynth [7]	9
Synthetic	12
VocalSet [28]	7
Total	50

Images are represented as 256×256 RGB signals with values in $[0, 1]$. Source images are cropped to this resolution without resizing. Crop locations are selected per source image. The image collection includes natural photographs, textures, blurred images, and synthetic signals.

Audio signals are four-second, 16 kHz mono clips. Signals are resampled where necessary using polyphase resampling and peak-normalized before training. The collection contains environmental sounds, speech, musical and vocal recordings, and synthetic signals.

The seven synthetic images include low, high, and multi frequency sinusoids, a checkerboard, Perlin-like noise, $1/f$ fractal noise, and a radial wave. The twelve synthetic audio signals include pure tones, a chirp, harmonic and inharmonic stacks, noise, amplitude and frequency modulated signals, and Gaussian bursts.

Development and evaluation signals. Six signals from each modality are designated as development signals and are used for exploratory experiments, temporal-horizon selection, and adaptive sampling baseline tuning. For comparisons affected by these choices, the remaining 44 signals per modality are used for evaluation and are not used to select the corresponding hyperparameters or configurations. Table 7 lists the development signals.

Table 7: Development signals used for exploratory analysis and configuration selection. The remaining 44 signals per modality are excluded from the corresponding selection decisions and used for final evaluation.

Modality	Development signals
Image	`kodim01`, `DIV2K_0240`, `Blurred_1013`, `SIPI_1.1.09`, `Synth_01_Low_Freq_Sine`, `Urban_img_005`
Audio	`ESC-01`, `LS-01`, `NS-01`, `SYN-01`, `SYN-09`, `VS-01`

Full per-signal provenance, including source files, crop locations, and preprocessing metadata, is provided in the accompanying data manifests.

A.2 INR ARCHITECTURES AND CONFIGURATIONS

We evaluate four INR representations: SIREN, ReLU with Gaussian Fourier features (ReLU-FF), ReLU with log-spaced positional encoding (ReLU-PE), and a dense multiresolution feature grid (Grid). The grid is an unhashed implementation of the multiresolution representation used by

Instant-NGP (Müller et al., 2022). All architectures use the same mean-squared-error objective and Adam (Kingma & Ba, 2015) without weight decay, with architecture-specific constant learning rates. Characterization experiments use full-batch optimization, while adaptive sampling experiments use mini-batches as detailed below.

Table 8 summarizes the configurations. For the MLP-based representations, the reported layer count is the number of hidden layers, followed by a linear output layer. For Grid, width and layer count refer to its ReLU decoder. Hidden widths are chosen to keep parameter counts approximately matched within each modality.

Table 8: Architecture configurations used for images and audio. For Grid, width and layer count refer to the decoder. Learning rates are constant throughout training.

Mod.	Arch.	Input to network	Width	Layers	LR	Params	vs. SIREN
Image	SIREN	$(y, x) \in [-1, 1]^2$ (raw)	256	4	1×10^{-4}	198,915	—
Image	ReLU-FF	256 Fourier features	217	4	5×10^{-4}	198,341	-0.3%
Image	ReLU-PE	258 positional features	217	4	1×10^{-3}	198,775	-0.1%
Image	Grid	6 levels $\times$ 6 features (36)	241	2	1×10^{-3}	199,005	$+0.05\%$
Audio	SIREN	$t \in [-100, 100)$ (raw)	256	4	5×10^{-5}	198,145	—
Audio	ReLU-FF	256 Fourier features	217	4	2×10^{-4}	197,905	-0.1%
Audio	ReLU-PE	257 positional features	217	4	2×10^{-4}	198,122	-0.01%
Audio	Grid	11 levels $\times$ 4 features (44)	224	2	1×10^{-3}	191,713	-3.2%

Image coordinates (y, x) lie in $[-1, 1]^2$, with endpoints included. Audio uses 64,000 uniformly spaced coordinates on $[-100, 100)$. SIREN receives these coordinates directly; for the other three representations, audio coordinates are rescaled to $[-1, 1]$ before encoding.

SIREN. We use sine activations $\sin(\omega_0 Wx + b)$ with $\omega_0 = 30$ in every layer. For audio, the wider coordinate range gives an effective first-layer frequency scale of $\omega_0 = 3000$ when expressed on normalized $[-1, 1]$ coordinates. Following Sitzmann et al. (2020), first-layer weights are drawn from $\mathcal{U}(-1/n_{\text{in}}, 1/n_{\text{in}})$ and subsequent weights from $\mathcal{U}(-\sqrt{6/n_{\text{in}}}/\omega_0, \sqrt{6/n_{\text{in}}}/\omega_0)$; biases use the PyTorch default initialization.

ReLU-FF and ReLU-PE. ReLU-FF maps coordinates through 128 Gaussian random Fourier features (Tancik et al., 2020), $[\sin(2\pi Bx), \cos(2\pi Bx)]$, producing a 256-dimensional input. We use $B \sim \mathcal{N}(0, 16^2)$ for images and $\sigma = 2{,}000$ Hz for audio. The basis is fixed across training runs using a dedicated initialization seed (12345), so variation across training seeds does not change the encoding.

ReLU-PE instead uses a deterministic log-spaced positional encoding (Mildenhall et al., 2020) followed by the raw coordinate. For images, 64 frequencies from 1 to 128 cycles per image are applied independently to each coordinate axis, giving 258 input features. For audio, 128 frequencies span 20–8,000 Hz, giving 257 input features. The Fourier basis and positional-encoding frequencies remain fixed throughout training.

Grid. Grid linearly interpolates learned features at multiple resolution levels, concatenates them, and passes the result through a small ReLU decoder; image features are bilinearly interpolated. Feature tables are initialized from $\mathcal{U}(-10^{-4}, 10^{-4})$. The image model uses six levels spanning resolutions 4–128 with feature dimension 6 and a two-layer decoder of width 241. The audio model uses 11 levels spanning resolutions 16–16,384 with feature dimension 4 and a two-layer decoder of width 224.

Parameter matching. The ReLU-FF and ReLU-PE widths are selected by a reproducible binary search to match the corresponding SIREN parameter count within 0.3%. Grid decoder widths are set separately, resulting in differences of 0.05% for images and 3.2% for audio. This matching controls total model size, but not representational capacity or inductive bias: unlike the MLP-based representations, approximately two thirds of the Grid parameters reside in localized learned feature tables.

Learning rates. Learning rates are architecture-specific and selected for stable convergence. To check whether the substantially higher seed reproducibility of audio SIREN could be explained by its lower learning rate, we additionally train it at 2×10^{-4}, matching ReLU-FF and ReLU-PE. Mean seed reproducibility increases from $\rho = 0.902$ to $\rho = 0.936$, rather than approaching the much lower reproducibility of the ReLU representations. The architecture-level difference therefore persists under this learning-rate substitution.

A.3 Training and Evaluation Protocols

Table 9 summarizes the two training protocols used in the paper.

Table 9: Training protocols used for characterization and adaptive sampling experiments.

	Characterization	**Adaptive sampling**
Optimizer	Adam	Adam
Sampling	Full batch	Mini-batch
Batch size	Full signal	8,192
Image steps	2,000	640
Audio steps	5,000	640
Seeds per setting	5	3
Evaluation	Per-coordinate error	Full-signal quality
Eval. interval	See text	16 steps
Image metric	RGB error / PSNR	PSNR
Audio metric	Amplitude error / SNR	SNR

Characterization protocol. For every signal–architecture pair, we train five independent runs using seeds

$$\{23, 672, 553, 112, 999\}.$$

Training is full batch, allowing the reconstruction error of every coordinate to be observed throughout optimization. Images are trained for 2,000 steps and audio signals for 5,000 steps.

For images, per-coordinate reconstruction error is the Euclidean norm of the RGB residual. For audio, it is the absolute amplitude error. The running sum used to compute coordinate difficulty is updated at every optimization step, so D_i is the exact mean over the complete observed training trajectory rather than an estimate from sparse checkpoints.

For visualization, per-coordinate error snapshots are additionally retained at logarithmically spaced checkpoints. Image checkpoints are

$$0, 1, 2, 4, 5, 8, 13, 19, 29, 45, 68, 104, 159, 242, 369, 563, 860, 1311, 2000,$$

with an analogous 20-checkpoint schedule for audio. Recorded runs use deterministic execution settings with TF32 disabled.

Adaptive-sampling protocol. Adaptive-sampling experiments use mini-batches of 8,192 coordinates for 640 optimization steps. This corresponds to 12.5% of the 65,536 image coordinates and 12.8% of the 64,000 audio coordinates per training step. Image experiments use seeds $\{1, 2, 3\}$ and audio experiments use $\{23, 672, 553\}$.

Full-signal reconstruction quality is evaluated every 16 steps using PSNR for images and SNR for audio. These full-signal evaluations are used only for reporting and are never exposed to the sampler.

When reporting performance on the "hardest 10%" of coordinates, the mask is defined using the top decile of that architecture's canonical coordinate-difficulty map from the characterization experiments. This mask is used only for evaluation and is not available to the adaptive sampler.

A.4 Target-Signal Descriptors and Predictability

To test how much coordinate difficulty can be explained by the target signal itself, we compute 12 local descriptors for each modality. These descriptors use only the target signal and contain no coordinate position, INR output, or training information.

For images, the descriptors are RGB magnitude, Sobel gradient magnitude, Laplacian magnitude, six spatial-frequency band energies, and local RMS contrast at window sizes of 3, 9, and 27 pixels. For audio, they are absolute amplitude, first- and second-difference magnitudes, six frequency-band envelopes spanning 0–8 kHz, and local RMS energy over 2, 8, and 32 ms. The frequency-band envelopes are smoothed over 8 ms.

A separate predictor is fit for every signal–architecture pair. The target is the median log difficulty over the five training seeds, and the inputs are the 12 target descriptors. Images use all pixels, while audio uses a fixed random subset of 8,192 samples. We use a histogram gradient-boosting regressor (Pedregosa et al., 2011) with 200 iterations, learning rate 0.1, and maximum depth 6.

Prediction is evaluated using five-fold blocked cross-validation within each signal. For images, the held-out blocks are contiguous horizontal regions; for audio, they are contiguous temporal regions. Predictions from the five held-out folds are combined, and R^2 is computed over all coordinates of the signal. Table 2 reports the median R^2 across the 50 signals. Thus, this experiment measures whether target properties predict difficulty in unseen regions of the same signal, rather than generalization to entirely unseen signals.

Residual analyses. The two residual analyses in the main paper serve different purposes and use slightly different procedures. For the seed-reproducibility results in Table 2, we subtract the out-of-fold boosted-tree prediction from each seed's log-difficulty map before computing seed correlations.

For the cross-architecture results in Table 3, we instead regress ranked difficulty on the ranked target descriptors using a linear model and compare the resulting residuals across architectures. Using the boosted-tree residual instead gives even lower cross-architecture correlations on images (0.07–0.22, compared with 0.21–0.38 in the reported linear analysis), so the reduction in agreement is not specific to the linear residualization.

The descriptor set is necessarily incomplete, and the blocked folds have no buffer between adjacent training and test regions. Residual difficulty should therefore be interpreted as difficulty not explained by the *measured* target descriptors, rather than as difficulty completely independent of the target signal.

B Temporal Aggregation of Coordinate Difficulty

We examine sensitivity to temporal aggregation on the development set using bounded windows of $K \in \{8, 16, 32, 64, 128, 256, 512\}$ steps, EMA estimators with corresponding half-lives, instantaneous error, and complete-history difficulty. Each setting uses six development signals per modality and three seeds. Results are descriptive; we do not perform significance testing on the development set.

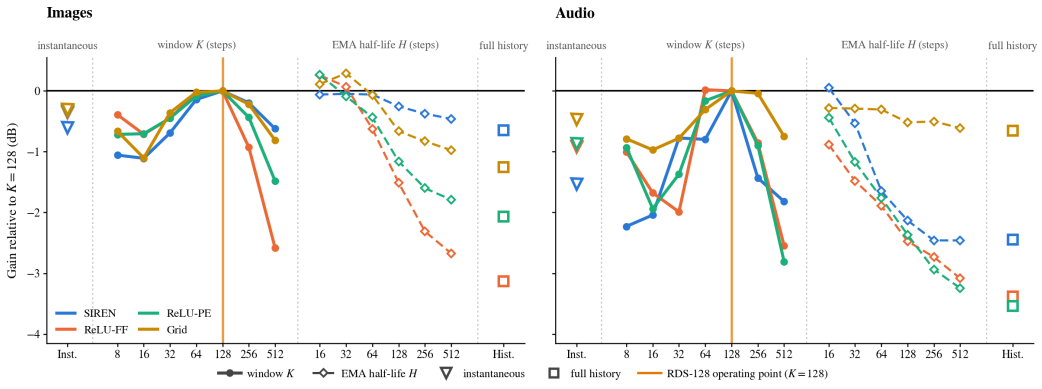


Figure 5: **Sensitivity to temporal aggregation.** Performance across temporal horizons on the development set, shown relative to RDS-128. Among bounded windows, $K = 128$ performs most consistently across settings, while the preferred temporal scale varies by representation and modality.

We use this sweep to select $K = 128$ for the remaining RDS experiments. It is the strongest bounded window in seven of eight settings; the exception is audio ReLU-FF, where $K = 64$ is 0.02 dB higher. The sweep does not identify 128 steps as universally optimal: short-horizon EMA estimators can match or exceed RDS-128 for images, and sensitivity to longer histories varies across representations and modalities.

Overall, the sweep supports the broader finding that recent difficulty is a more useful sampling signal than indefinitely accumulated difficulty, while the preferred degree of recency is setting-dependent. Additional development-set diagnostics support this distinction: recent-history sampling improves over uniform in all 72 tested architecture–modality–regime settings, while complete-history difficulty is a poorer predictor of current error for the ReLU representations despite being a stronger proxy for persistent offline difficulty.

C Recent-Difficulty Sampling Details

Recent-Difficulty Sampling (RDS) maintains a bounded history of reconstruction errors observed during training. For each coordinate, RDS estimates difficulty from errors observed during the previous 128 training steps. Every 32 steps, coordinates are ranked by this recent mean error. If $r_i \in \{1, \dots, N\}$ denotes the ascending rank of coordinate i, the sampling probability is

$$p_i = \frac{(r_i/N)^\alpha}{\sum_{j=1}^{N}(r_j/N)^\alpha}, \qquad \alpha = 2. \tag{5}$$

The 128-step history is implemented using four 32-step blocks. At each refresh, the oldest block is discarded and the sampling distribution is recomputed. Coordinates not observed within the retained history are assigned the mean error of the observed coordinates; approximately 3% of coordinates are unseen at a typical refresh.

Training begins with a 96-step uniform burn-in, after which RDS becomes adaptive. All adaptive-sampling experiments use batches of 8,192 coordinates for 640 optimization steps. Reconstruction errors used to update the sampler are measured before the corresponding parameter update. Sampling is performed with replacement, and duplicate coordinates therefore contribute repeatedly to the mini-batch loss. No importance correction is applied.

RDS uses only errors produced by coordinates already evaluated during optimization and requires no additional model evaluations for scoring. The exponent $\alpha = 2$, 32-step refresh interval, 96-step burn-in, and replacement-sampling protocol are held fixed throughout the main RDS experiments rather than jointly optimized for RDS-128.

C.1 Sampling Without Replacement

The main experiments use uniform sampling with replacement as the reference condition. Because replacement sampling can draw the same coordinate multiple times within a mini-batch, we additionally evaluate uniform sampling without replacement as a stronger baseline.

Relative to this baseline, RDS-128 remains positive in all eight architecture–modality settings, with median whole-signal gains of $+0.39$ to $+1.87$ dB. The margins are generally smaller than against uniform sampling with replacement, but the sign of the RDS effect is unchanged across all eight settings. Thus, the consistent fixed-step improvement of RDS is not explained solely by duplicate draws in the original uniform baseline.

D Adaptive-Sampling Baselines and Computational Cost

We compare RDS-128 with Soft Mining (Kheradmand et al., 2024), INT (Zhang et al., 2024), NINT (Zhang et al., 2026), and EVOS (Zhang et al., 2025). Each method is integrated into the same training and evaluation framework used for RDS. The ports are validated in lockstep against the corresponding released implementations before running the common benchmark.

INT and NINT use their published constants rather than being retuned on our development signals. Soft Mining and EVOS require configuration within the common experimental framework; for the image experiments, their settings are selected using the six image development signals. This selection uses equal total modeled cost as the tuning objective, avoiding configurations that benefit simply from additional scoring computation. Soft Mining additionally uses a predeclared rescaling of its noise constant when transferred to our coordinate and optimization scales. We preserve the behavior of the released NINT implementation, including its squared-output behavior, when validating and reporting our port.

Peer-method configuration is performed only on the development signals. The 44-signal benchmark in the main paper excludes these signals. RDS-128 is evaluated using the temporal horizon selected on the development set, and its results are fixed before comparison with the peer methods.

D.1 Wall-Clock Cost

Equal optimization-step budgets do not imply equal computational cost. RDS requires no additional model evaluations for coordinate scoring, but incurs overhead from maintaining recent-error statistics and periodically ranking coordinates. Across the timing experiments, RDS-128 requires approximately 1.04–$1.28\times$ the wall-clock time of uniform sampling.

The peer methods incur larger overheads in the same implementation. Soft Mining requires approximately 1.2–$1.4\times$ the time of uniform sampling, EVOS approximately 1.4–$1.7\times$, INT and NINT approximately 3.6–$4.5\times$ on the MLP representations and 1.7–$2.3\times$ on the dense grid. These measurements were collected on a shared GPU, and we do not have a corresponding idle-GPU benchmark. They should therefore be interpreted as descriptive runtime measurements rather than precise hardware-independent estimates.

To test whether the fixed-step gains translate to a common runtime budget, we also run an exploratory equal-wall-clock comparison. RDS remains positive for image SIREN ($+1.02$ dB), image Grid ($+1.11$ dB), and audio Grid ($+0.77$ dB); image ReLU-PE ($+0.11$ dB) and audio SIREN ($+0.15$ dB) are approximately neutral, while the remaining three ReLU settings are negative (-0.85 to -1.26 dB). We therefore treat the comparison as exploratory and do not interpret the consistent fixed-step gains as a universal wall-clock advantage.

E Dataset-Size Robustness

We test how strongly the main conclusions depend on the number and particular choice of signals in the 50-signal datasets. No models are retrained. Instead, we repeatedly subsample the outcomes of the existing experiments and recompute the corresponding conclusions. For each subset size $n \in \{3, 5, 8, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$, we draw 5,000 random subsets without replacement.

Table 10: Fraction of random subsets for which selected conclusions are reproduced. Subsets are drawn without replacement from the 50 signals in each modality.

Conclusion	$\mathbf{n = 5}$	$\mathbf{n = 10}$	$\mathbf{n = 20}$	$\mathbf{n = 30}$	$\mathbf{n = 40}$
Window128 $>$ Uniform, all 8 cells	.990	1.000	1.000	1.000	1.000
Window128 $>$ Uniform by ≥ 0.25 dB	.916	.995	1.000	1.000	1.000
Window128 $>$ Instant, all 8 cells	.802	.952	.996	1.000	1.000
Window128 $>$ History, all 8 cells	.974	.999	1.000	1.000	1.000
History $<$ Uniform, all ReLU cells	.981	1.000	1.000	1.000	1.000
Seed-reproducibility architecture split	1.000	1.000	1.000	1.000	1.000
Target-only R^2 architecture ranking	.369	.587	.765	.892	.981
Residual-reproducibility architecture split	1.000	1.000	1.000	1.000	1.000
Raw $>$ residual cross-architecture agreement	.922	.994	1.000	1.000	1.000
ReLU-FF flatter than SIREN	.984	1.000	1.000	1.000	1.000

The principal directional conclusions are stable well below the full dataset size. In particular, Window128 improves over uniform in all eight architecture–modality settings in every one of the 5,000 sampled subsets at $n \geq 10$. The seed-reproducibility architecture split, residual-reproducibility split, and reduction in cross-architecture agreement after residualization are similarly stable.

Not every finer-grained ordering is equally robust. For images, identifying SIREN–Grid as the most similar architecture pair is reproduced in approximately 75% of subsets at $n = 10$ and 87% at $n = 20$. The weaker target-predictability ordering in which SIREN has the highest R^2 and ReLU-FF the lowest is reproduced in approximately 70% and 80% of subsets at $n = 10$ and $n = 20$, respectively. The complete four-architecture target-only R^2 ordering is more sensitive still, reaching a reproduction rate of 98.1% at $n = 40$.

Stable conclusions also do not imply precise effect-size estimation. All eight subset median RDS gains fall within 0.25 dB of their corresponding 50-signal values in only 31% of 20-signal subsets and 66% of 30-signal subsets. Larger signal sets therefore matter more for estimating the magnitude and finer ordering of effects than for recovering the principal directional conclusions.

This analysis measures sensitivity to the number and selection of signals within the datasets considered here. It does not establish generalization to unseen signals or external corpora.