RaFL: An Empirical Study of Random Filter Losses for Image Reconstruction

Anonymous authors
Paper under double-blind review

Abstract

Pixel losses train image reconstruction networks toward a point statistic of the target distribution, often averaging away fine textures under uncertainty. Perceptual losses alleviate this issue using pretrained features, but those features are inherently domain-specific. We study a random filter loss (RaFL) that compares prediction and target through filters with random weights, avoiding both pretrained features and task-specific loss engineering. RaFL combines random convolutions, random Fourier features, and random wavelet filters, automatically balancing their contributions through gradient-norm matching, eliminating manual loss-weight tuning and grid search. We systematically investigate three design choices: filter family, weight resampling schedule, and weight initialization. Across all settings, the best configuration consistently uses all three filter families, redraws filters every optimization step, and employs Gaussian initialization. Using this configuration, we evaluate RaFL on three 2D and two 3D image reconstruction tasks. RaFL outperforms a pixel loss on three tasks and matches it on two, while outperforming prior random filter losses on all 2D tasks. RaFL adds less than 2% to the backbone's forward FLOPs and no inference cost, remaining $182\times$ cheaper than a VGG19 perceptual loss. These findings suggest that RaFL is a simple, tuning-free, and domain-agnostic loss objective for 2D and 3D image reconstruction. Code is available at https://anonymous.4open.science/r/rafl-C10F.

1 Introduction

A pixel loss combined with a perceptual loss is a common setup for image reconstruction tasks, as a solution to the averaging behavior of pixel loss alone. However, perceptual losses require a pretrained network, and the pretrained features are tied to the domain they were trained on. This limits their use on inputs from other domains, e.g., 3D medical images (Pfaehler et al., 2025). Fixed filters that require no learning, such as a gradient penalty (Mathieu et al., 2015), a Laplacian edge term (Jiang et al., 2020), a wavelet-domain discrepancy (Korkmaz et al., 2024), or a Fourier-domain one (Jiang et al., 2021), have demonstrated effectiveness across various image reconstruction tasks. However, their filter responses are fixed, and the selection of filters and weights are typically determined through ablations or expert knowledge, often requiring task-specific hyperparameter tuning.

Untrained network based losses that avoid the designing for filter choices have proven to be effective in image reconstruction tasks (Lempitsky et al., 2018; Heckel & Hand, 2019), image generation (He et al., 2016; Ustyuzhaninov et al., 2017), and image segmentation (Liu et al., 2021). Zhou et al. (2023) and Pfaehler et al. (2025) apply random weight networks as optimization functions, achieving strong results on 2D and 3D image reconstruction tasks, respectively. However, how to design the random filter loss itself remains underexplored. Liu et al. (2021) find that depth matters more than pretraining as a prior, while Pfaehler et al. (2025) find that a small network performs best as a loss function, and since each conclusion comes from experiments conducted in a specific setting, it is unclear which holds more broadly across both 2D and 3D. We use the term random filter loss broadly to encompass objectives based on either explicit random filters or untrained random networks.

In this paper, we propose RaFL (pronounced "raffle"), a random filter loss built not from a convolutional or transformer architecture, but from simple, untrained filter families that complement each other. RaFL uses a fixed loss formulation, balances its filter families automatically by gradient-norm matching, and preconditions each update with MAdam (Liu et al., 2026). Through empirical

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.

experiments, we answer three questions: which filters to use, how often to redraw their weights, and which initialization method to use. Using the resulting configuration, we evaluate RaFL across five tasks: real-noise denoising (2D, SIDD), low-light enhancement (2D, LOL), pan-sharpening (2D, WorldView-2), super-resolution (3D, OASIS3), and MR angiography denoising (3D, IXI). RaFL adds no learnable parameters, is not needed at inference. It costs under two percent of the forward pass of the image reconstruction network it trains, and it needs $182\times$ fewer loss FLOPs than an untrained VGG19 perceptual loss. RaFL achieves PSNR and SSIM that are competitive with or better than those of the baselines across all five tasks (Tables 2 and 3).

2 Related Work

Perceptual losses. Pixel losses, such as ℓ_1 and ℓ_2, are the dominant choice of loss functions for various image reconstruction tasks. Beyond the pixel losses, a perceptual loss compares prediction and target in the pretrained network features on natural images (Johnson et al., 2016), LPIPS calibrates such features against human judgments of similarity (Zhang et al., 2018), and DISTS combines a structure and a texture comparison of them so that the distance tolerates resampled texture (Ding et al., 2020). An adversarial objective trains its critic alongside the model. These trained perceptual metrics carry the data they were trained on. Their features encode the inductive biases of natural images and are not calibrated for other modalities (Pfaehler et al., 2025). In 3D medical imaging, the pretrained encoders that exist are each tied to a specific modality and anatomy (Hatamizadeh et al., 2021), so they do not transfer to new modalities or anatomies. Structural losses such as SSIM and its multi-scale form (Wang et al., 2004; 2003) are often treated as a perceptual alternative in restoration (Zhao et al., 2016), however, they compare only fixed local statistics inside a window at one or a few scales, making them a pixel loss with a larger window rather than a feature loss.

Handcrafted filter losses. Handcrafted filter terms have been proposed as a training-free alternative. Image gradient based filters have been used for video prediction (Mathieu et al., 2015), image synthesis (Yu et al., 2019) and image denoising (Liu et al., 2020). Another line of work includes loss functions based on the Laplacian (Jiang et al., 2020), spectral-spatial structure (Choi et al., 2020), wavelet-domain (Korkmaz et al., 2024), and frequency-domain (Jiang et al., 2021; Li et al., 2022). Since no training is required, the computational cost of these losses is relatively negligible. However, each response is fixed and applied identically regardless of the data. Also, the selection and combination of filters are made a priori and are task-specific, requiring domain expertise to determine which filters to penalize and weight more heavily.

Untrained perceptual/filter losses. The effectiveness of untrained convolutional neural network (CNN) as a prior has been shown in various image reconstruction tasks (Lempitsky et al., 2018; Heckel & Hand, 2019). A randomly weighted CNN also have shown the utility in achieving image representations for generative tasks (He et al., 2016) and synthesizing realistic textures (Ustyuzhaninov et al., 2017). Liu et al. (2021) showed that an untrained CNN used as a perceptual loss performs comparably to a pretrained one on semantic segmentation, depth estimation, and instance segmentation, and reported that network structure matters more than training. Zhou et al. (2023) proposed four random-weight filter prototypes as an image restoration loss constraint a central-difference convolution (CDC) (Yu et al., 2020), zero-order reverse (reverse) filtering (Tao et al., 2017), an invertible network (INN) (Dinh et al., 2014) and a Taylor expansion (Taylor). Each is added to a pixel loss and the weight redrawn per epoch, and showed that all four consistently improve performance over baseline. The most closely related work is an untrained perceptual loss (uPL) Pfaehler et al. (2025), that uses a small CNN with no pixel term as the loss. They apply the loss to 3D MR denoising of brain vessels and plant roots, varying the loss network's depth, kernel size, pooling and initialization. They report that kernel size matters little, pooling helps on one dataset but hurts on the other, and shallow networks win. Three to five layers match or beat an untrained VGG19, contrary to Liu et al. (2021), who found depth mattered.

These related works demonstrate the strong performance of random filter losses, but they remain narrower in scope. Pfaehler et al. (2025) use untrained networks without reweighting as a perceptual loss for 3D MR denoising of line-like structures, and Zhou et al. (2023) use random-weight networks built from a hand-chosen prototypes for image restoration with a pixel loss. In contrast, RaFL works

across dimensionalities (2D and 3D) and modalities (natural and medical images) without requiring per-task tuning of its loss weight.

3 METHOD

In this work, we propose RaFL, a general random filter loss for both 2D and 3D image reconstruction that requires no pretrained weights, modality specific components, or no per-task filter selection. RaFL is motivated by the idea that a signal can be probed through several different measurement filters applied to the same data (Candes et al., 2006). Accordingly, we implement the loss with three filter families, each capturing a different aspect of the image. We characterize RaFL through an empirical study of three factors: (i) choice of filters, (ii) weight resampling schedule, and (iii) weight initialization method. This analysis clarifies which components drive performance and yields practical guidelines. Using the resulting configuration, we then evaluate RaFL on 2D and 3D image reconstruction tasks.

3.1 CHOICE OF FILTERS

The three families, a random convolution, random Fourier features and a random wavelet detail, trade off spatial and frequency resolution differently. Random convolutions are spatially local but broad in frequency, Fourier features compare small patches through random frequencies, and the random wavelet detail sits between them with joint space-frequency localization across scales. Combining them lets the loss penalize errors that any single family would miss, such as fine local texture, patch-level structure, or scale-specific detail. We empirically show this using image denoising on SIDD (Abdelhamed et al., 2018).

Random convolution. Convolutional filters respond to local spatial patterns such as edges and textures. We use a random kernel as a feature extractor for the loss without any training, and following Liu et al. (2021) a single kernel whose weights are drawn from a fixed distribution at a fixed size (e.g., 3×3, 5×5, 7×7), spanning all input channels, with no nonlinearity. A single kernel size, however, operates at one spatial scale and responds broadly across frequencies. Convolution is therefore spatially specific but frequency-unselective, in contrast to the other filters we introduce next.

Random Fourier features. Neural networks are known to fit low-frequency content first (Rahaman et al., 2019), and Fourier features have been shown to help recover high-frequency detail that would otherwise be underweighted (Tancik et al., 2020). Random Fourier features (Rahimi & Recht, 2007) are used to project the input onto randomly sampled sinusoids, so we compare a specific spatial frequency between prediction p and target q. For each patch, D random Fourier features were computed, $\sqrt{2/D}\cos(w^\top p + b)$, where the frequency $w \sim \mathcal{N}(0, 2\gamma I)$ is a random vector and b is a random phase, both redrawn at every randomization step and shared by p and q. The inner product of two feature vectors approximates a Gaussian kernel between the two patches (Rahimi & Recht, 2007), so the term keeps working on the neighborhoods that are nearly correct with a certain tolerance parameter γ, the bandwidth the ablation details in Tables S8 and S7.

Random wavelet detail. The stationary wavelet transform (SWT) applies a wavelet filter bank without downsampling, decomposing the input into subbands that are localized in both space and frequency, at multiple scales and orientations.

Short daubechies wavelets have been shown to help in image tasks: in CNN image classification, db2 improve accuracy, while the longer may reduce it (Li et al., 2020). Considering this and the training patch sizes, we used Daubechies order 2 (db2). At each of the 4 levels, an independent random unit vector is sampled. Ablation can be found in Table S6 and implementation details can be found in the supplementary materials (App. C).

3.2 RANDOMIZATION AND OPTIMIZATION

Weight resampling schedule. We compare three randomization schedules: one draw for the whole run, one per epoch and one per step. Because gradient-norm matching updates the loss weights during training, a per-step redraw introduces a time-varying objective. We therefore use

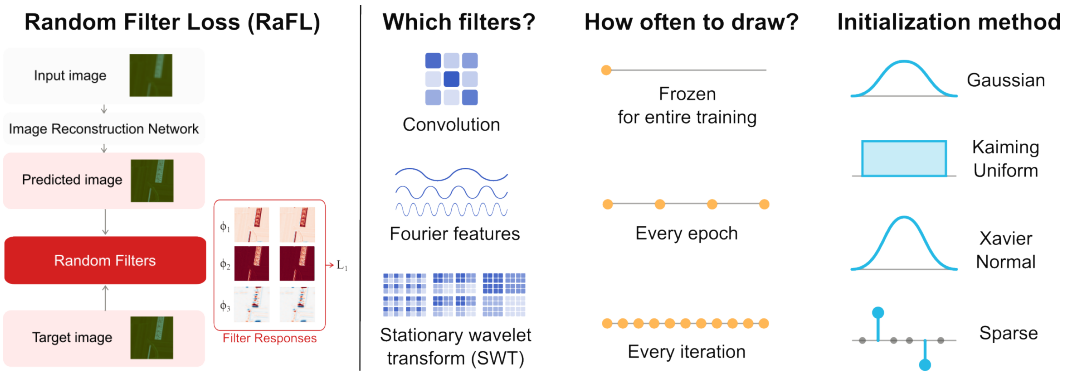

Figure 1: **Overview of RaFL.** Left: training pipeline. Right: the three design choices studied for RaFL: filter family, weight redraw schedule, and initialization.

MAdam (Liu et al., 2026), which is designed for optimization under changing loss weights (§3.3). The effect of the preconditioner is evaluated separately in App. A.3.

Weight initialization. Due to the randomness of untrained networks, the gradient can explode or vanish (Liu et al., 2021). We compared four weight initializations for RaFL: Gaussian, with $\sigma = 0.5$; Kaiming normal (He et al., 2015); Xavier uniform (Glorot & Bengio, 2010); and sparse initialization (Martens et al., 2010).

3.3 Loss Function

Let $\hat{y} = f_\theta(x)$ be the prediction of the reconstruction network f_θ for a degraded input x, and y the target. Let ϕ_1, ϕ_2 and ϕ_3 be the three families of §3.1, the convolution, the Fourier features and the wavelet detail. Our objective is

$$\mathcal{L}(\hat{y}, y) \;=\; \sum_j \lambda_j \left\| \phi_j(\hat{y}) - \phi_j(y) \right\|_1, \tag{1}$$

the ℓ_1 distance between the filtered prediction and the filtered target, summed over the choice of filters. Each ϕ_j carries its own random weights, drawn at every randomization step as described in §3.1, and the same draw is applied to $\hat{y}$ and to y and then discarded.

Unlike conventional multi-term losses, RaFL requires no manual tuning of loss weights. Every 50 steps, we adaptively set λ_j so that $\|\lambda_j \nabla_{\hat{y}} \mathcal{L}_j\|$ is equal across the three filter families, where $\mathcal{L}_j = \|\phi_j(\hat{y}) - \phi_j(y)\|_1$ and $\nabla_{\hat{y}}$ is the gradient with respect to the network output, following gradient-norm matching (Chen et al., 2018). This automatically balances the loss terms and removes the need for task-specific λ selection or grid search. Because the loss weights change during training, Adam's running statistics may become inconsistent with the current objective. MAdam (Liu et al., 2026) fixes this by keeping Fisher estimates for every pair of loss terms, combines them with the current λ_j into a diagonal metric $\hat{M}$, and passes $\hat{M}^{-1}d$ to Adam (Kingma & Ba, 2014) in place of the summed gradient d (App. A.3).

4 Results

4.1 Experimental Settings

To validate RaFL, we adopt the same tasks and datasets from baseline papers(Zhou et al., 2023; Pfaehler et al., 2025) with their best architectures. For **2D image denoising**, the real-world Smartphone Image Denoising Dataset (SIDD) (Abdelhamed et al., 2018), paired with DnCNN (Zhang et al., 2017) was selected. In **2D image enhancement**, we adapted low-light (LOL) dataset (Wei et al., 2018) with DRBN (Yang et al., 2020). For **2D super-resolution**, pan-sharpening on WorldView-2 with SFINet (Zhou et al., 2022) was used. Following the **3D image denoising** setup for untrained perceptual loss (uPL) in Pfaehler et al. (2025), we performed MR angiography denoising on the IXI dataset (Jiang et al., 2018) at 5% Rician noise (Gudbjartsson & Patz, 1995), using a

	Configuration	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
	RaFL: all three, every step, Gaussian	$\mathbf{40.28 \pm 0.13}$	$\mathbf{0.9557 \pm 0.0012}$	0.2488 ± 0.0036	0.1847 ± 0.0034
Q1, which filters	Wavelet only	32.26 ± 3.13	0.8872 ± 0.0659	0.2874 ± 0.0240	0.2077 ± 0.0166
	Convolution only	39.75 ± 0.17	0.9523 ± 0.0014	0.2605 ± 0.0045	0.1907 ± 0.0035
	Fourier only	39.92 ± 0.32	0.9548 ± 0.0008	0.2547 ± 0.0047	0.1878 ± 0.0041
	Convolution + Wavelet	39.88 ± 0.75	0.9538 ± 0.0015	$\mathbf{0.2446 \pm 0.0041}$	$\mathbf{0.1825 \pm 0.0043}$
	Convolution + Fourier	$\underline{39.98 \pm 0.33}$	0.9552 ± 0.0010	0.2573 ± 0.0061	0.1907 ± 0.0052
	Fourier + Wavelet	39.28 ± 1.36	0.9548 ± 0.0010	0.2498 ± 0.0052	0.1894 ± 0.0081
	pixel ℓ_1	39.64 ± 1.00	0.9551 ± 0.0011	0.2478 ± 0.0056	0.1830 ± 0.0057
Q2	Every epoch	39.78 ± 0.67	0.9545 ± 0.0014	0.2503 ± 0.0054	0.1863 ± 0.0068
	Fixed for the entire run	39.67 ± 1.03	0.9546 ± 0.0011	0.2494 ± 0.0052	0.1852 ± 0.0056
Q3	Kaiming normal	39.92 ± 0.60	$\underline{0.9552 \pm 0.0008}$	0.2508 ± 0.0064	0.1867 ± 0.0059
	Xavier uniform	$\underline{39.98 \pm 0.41}$	0.9548 ± 0.0016	0.2529 ± 0.0039	0.1876 ± 0.0042
	Sparse	39.91 ± 0.31	0.9542 ± 0.0011	$\underline{0.2468 \pm 0.0033}$	$\underline{0.1830 \pm 0.0040}$

Table 1: **Ablation study of filter family (Q1), redraw schedule (Q2), and initialization (Q3) on SIDD.** (DnCNN, eight seeds per configuration; mean ± s.d. across seeds; bold indicates the best and underline the second-best value in each column) RaFL uses all three filter families, with filter weights redrawn from a Gaussian distribution at every optimization step (first row).

3D DnCNN. Since uPL was designed for 3D MR denoising, we additionally experimented with **3D image super-resolution** using T1-weighted brain MR images from OASIS-3 (LaMontagne et al., 2019). Images were $4\times$ downsampled and SwinUNETR (Hatamizadeh et al., 2021) was trained to recover the isotropic image.

Baseline Losses. Four prototype losses from Zhou et al. (2023), CDC, reverse, an INN and Taylor, are used for 2D baselines. The pixel loss (ℓ_1) and the uPL of Pfaehler et al. (2025), which compares the features of an untrained CNN and has no pixel loss term, are used for both 2D and 3D baselines. We run uPL with the SimpleNet which paper recommends, on all the datasets so every task compare aginst the same. Also we run an extra untrained VGG19 version of uPL in SIDD dataset, which matches the architecture of the pretrained VGG as a comprehensive compare. Every configuration in a table shares the same backbone, data pipeline, budget, seeds, and optimizer, and is evaluated at its final epoch. Further details are in App. C.

Implementation Details. Across all tasks, the initial learning rate is set to 10^{-3}, except image enhancement on LOL, which uses 10^{-4} and optimized using MAdam (Liu et al., 2026). . Datasets are strictly partitioned into disjoint training, validation, and test splits (approximately $8:1:1$) at the subject or scene level to prevent data leakage. Every design decision in this paper was made on the validation split: which families enter the loss, the support of the convolution, the random weight distribution and the weighting strategy. Details about image preprocessing are in App. C and hyperparameters and further configuration are in App. A.

4.2 Ablation study answering three design questions

We conduct extensive ablation studies on the SIDD image denoising task using DnCNN to systematically address three fundamental design questions for RaFL: which filters to use, how frequently to redraw them, and how to initialize their weights. Results are summarized in Table 1.

Choice of filters RaFL with all three filter families, weights redrawn every iteration in Gaussian initialization, achieves the highest PSNR (40.28 dB) and SSIM (0.9557), outperforming every proper subset. The Convolution+Wavelet pair achieves the best perceptual quality, LPIPS (0.2446) and DISTS (0.1825). Sparse initialization with all three families is a close second on both (LPIPS 0.2468, DISTS 0.1830), and still surpasses RaFL's own Gaussian-initialized result on these two metrics. Across initialization methods (Table 1, Q3), no configuration collapses on any metric, unlike using a partial set of filter families, such as Wavelet only on PSNR. We further evaluate convolution kernel size, additional convolution filters, and with and without handcrafted filters including Sobel and Laplacian (Table S1). A 7^2 convolution achieves the highest PSNR and SSIM with the smallest seed variance, while a 3^2 convolution is lower by 0.23 dB PSNR on seven of eight seeds. Adding additional convolution scales, Sobel filters, or Laplacian filters does not yield consistent improvements, suggesting that the three filter families are sufficient for the tasks studied. Based on the

results, we use all three filter families in RaFL, and evaluate the resulting configuration unchanged across all remaining image reconstruction tasks.

Weight resampling schedule. Redrawing every step outperforms both draw held for an epoch and a draw held for the entire run on all four metrics (Table 1 Q2). Compared to holding weights fixed per epoch or for the entire training run, per-step randomization improves PSNR by 0.50 dB and 0.61 dB, respectively. Per-step redrawing significantly reduces performance variance across random seeds (± 0.13 dB vs. ± 0.67 dB and ± 1.03 dB), preventing optimization from getting trapped into suboptimal configurations. We further evaluate this redraw schedule across all five tasks in App. Table S9. Per-step resampling consistently achieves the highest mean PSNR and generally yields the most stable training across tasks. Additional analyses of loss weighting and preconditioning are reported in App. Tables S3 and S5.

Weight initialization. Weight initialization has comparatively little influence on performance (Table 1). The four initialization strategies differ by at most 0.38 dB in PSNR, while the choice of filter families changes performance by as much as 8.02 dB. Although prior work has highlighted initialization as an important consideration for deep random networks (Liu et al., 2021), our findings are more consistent with Pfaehler et al. (2025), who observed only small differences across initialization schemes. We therefore use Gaussian initialization, which achieves the highest mean performance and the smallest variance across seeds.

4.3 Experimental Results on Image Reconstruction Tasks

2D tasks: image denoising, enhancement, and super-resolution. Across the three 2D tasks, RaFL matches or improves upon the pixel loss and is competitive with or better than prior random filter losses (Table 2). Notably, the same configuration selected in Sec. 4.2 is transferred unchanged across all tasks.

On SIDD, RaFL achieves the highest mean PSNR and is statistically indistinguishable from the strongest baseline, CDC (40.28 vs. 40.26 dB; paired interval $[-0.12, +0.15]$). Both methods outperform the pixel loss by more than 0.6 dB, whereas the perceptual metrics differ only within the observed seed variability. Qualitative examples (Fig. 2) show that RaFL preserves fine image texture while avoiding the oversmoothing visible in several competing methods.

On LOL, PSNR differences among RaFL, the pixel loss, and uPL are small (21.17, 21.16, and 21.33 dB). However, RaFL consistently achieves significantly lower LPIPS and DISTS and improves upon all four random filter prototypes. This suggests that, despite similar reconstruction fidelity, RaFL better preserves perceptual image quality. Fig. 2 (top) shows that RaFL better preserves the wood grain texture and local contrast, whereas several competing methods introduce color artifacts and blur fine structural details.

The largest improvement is observed on WorldView-2, where RaFL outperforms the pixel loss by 1.51 dB and the strongest prototype (CDC) by 0.87 dB. Furthermore, RaFL achieves the best value on all four evaluation metrics. While uPL remains competitive, it falls behind RaFL across every metric. Together, these results indicate that the proposed filter combination is particularly beneficial for reconstruction tasks requiring recovery of high-frequency spatial and spectral information.

3D tasks: super-resolution on OASIS3 and denoising on IXI. On $4\times$ super-resolution of OASIS3, RaFL consistently outperforms both the pixel loss and uPL (Table 3). It achieves the best value on all four metrics and the highest PSNR on every seed, significantly improving upon the pixel loss by 0.33 dB and uPL by 3.74 dB. The small variance across seeds (± 0.03 dB) further indicates stable optimization. Representative examples in Fig. 3 show improved recovery of fine anatomical structures relative to both baselines. The pixel loss gives the cerebellar folia as straight striping and lost sulcus overall, RaFL follows their curve as the ground truth does, and the untrained network introduces a regular lattice-like texture that is not present in the ground truth image.

On MR angiography denoising with a 3D DnCNN on IXI (Jiang et al., 2018) at 5% Rician noise (Gudbjartsson & Patz, 1995), RaFL, the pixel loss, and uPL achieve similar performance (Table 3). RaFL attains the highest mean PSNR (29.66 dB), although the difference relative to both baselines

Image denoising, SIDD Loss	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
RaFL	$\mathbf{40.28 \pm 0.13}$	0.9557 ± 0.0012	0.2488 ± 0.0036	0.1847 ± 0.0034
pixel ℓ_1	39.64 ± 1.00	0.9551 ± 0.0011	$\mathbf{0.2478 \pm 0.0056}$	$\mathbf{0.1830 \pm 0.0057}$
CDC	40.26 ± 0.06	$\mathbf{0.9568 \pm 0.0004}^*$	0.2681 ± 0.0057	0.2012 ± 0.0049
reverse	39.33 ± 0.71	0.9550 ± 0.0015	0.2560 ± 0.0061	0.1905 ± 0.0054
Taylor	38.20 ± 1.87	0.9523 ± 0.0021	0.2655 ± 0.0118	0.2027 ± 0.0199
INN	33.92 ± 5.21	0.9447 ± 0.0194	0.2822 ± 0.0329	0.2338 ± 0.0386
uPL, SimpleNet	40.09 ± 0.07	0.9541 ± 0.0005	0.2648 ± 0.0021	0.1957 ± 0.0026
uPL, VGG19	39.18 ± 0.16	0.9410 ± 0.0020	0.2631 ± 0.0058	0.1903 ± 0.0032
Image enhancement, LOL Loss	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
RaFL	21.17 ± 0.33	0.8035 ± 0.0032	$\mathbf{0.2568 \pm 0.0043}^*$	$\mathbf{0.1995 \pm 0.0032}^*$
pixel ℓ_1	21.16 ± 0.46	0.7799 ± 0.0031	0.3055 ± 0.0075	0.2273 ± 0.0049
CDC	20.55 ± 0.31	$\mathbf{0.8046 \pm 0.0047}$	0.2863 ± 0.0061	0.2141 ± 0.0052
reverse	21.08 ± 0.38	0.7998 ± 0.0018	0.2827 ± 0.0036	0.2146 ± 0.0028
Taylor	20.58 ± 0.32	0.7893 ± 0.0038	0.2904 ± 0.0084	0.2199 ± 0.0041
INN	20.74 ± 0.35	0.8000 ± 0.0053	0.2804 ± 0.0065	0.2129 ± 0.0045
uPL	$\mathbf{21.33 \pm 0.43}$	0.7880 ± 0.0111	0.2951 ± 0.0169	0.2220 ± 0.0094
Super-resolution, WorldView-2 Loss	PSNR ↑	SSIM ↑	ERGAS ↓	SAM ↓
RaFL	$\mathbf{39.62 \pm 0.08}^*$	$\mathbf{0.9631 \pm 0.0005}^*$	$\mathbf{1.318 \pm 0.011}^*$	$\mathbf{0.0297 \pm 0.0002}^*$
pixel ℓ_1	38.10 ± 0.13	0.9531 ± 0.0011	1.532 ± 0.026	0.0349 ± 0.0005
CDC	38.75 ± 0.14	0.9594 ± 0.0011	1.437 ± 0.024	0.0326 ± 0.0005
reverse	38.70 ± 0.12	0.9591 ± 0.0009	1.434 ± 0.020	0.0326 ± 0.0005
Taylor	37.90 ± 0.71	0.9493 ± 0.0086	1.670 ± 0.206	0.0367 ± 0.0030
INN	37.49 ± 0.67	0.9457 ± 0.0076	1.783 ± 0.214	0.0389 ± 0.0033
uPL	39.15 ± 0.20	0.9603 ± 0.0016	1.401 ± 0.040	0.0314 ± 0.0008

Table 2: **Results on 2D image reconstruction tasks.** uPL follows (Pfaehler et al., 2025); CDC, reverse, Taylor, and INN follow (Zhou et al., 2023). The symbol * denotes significance against the second-best method (paired t-test, $p < 0.05$); mean ± s.d. across 8 seeds.

Loss	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
Super-resolution, OASIS3				
RaFL	$\mathbf{22.22 \pm 0.03}^*$	$\mathbf{0.8249 \pm 0.0017}$	$\mathbf{0.2962 \pm 0.0010}^*$	$\mathbf{0.2547 \pm 0.0010}$
pixel ℓ_1	21.89 ± 0.06	0.7782 ± 0.0768	0.3029 ± 0.0060	0.2592 ± 0.0069
uPL	18.48 ± 0.24	0.5294 ± 0.0521	0.4104 ± 0.0169	0.3158 ± 0.0080
Image denoising, IXI				
RaFL	$\mathbf{29.66 \pm 0.19}$	0.9627 ± 0.0007	0.0646 ± 0.0012	0.0785 ± 0.0013
pixel ℓ_1	29.56 ± 0.33	0.9622 ± 0.0014	0.0645 ± 0.0021	0.0788 ± 0.0029
uPL	29.63 ± 0.35	$\mathbf{0.9628 \pm 0.0014}$	$\mathbf{0.0632 \pm 0.0016}$	$\mathbf{0.0775 \pm 0.0025}$

Table 3: **Results on 3D image reconstruction tasks.** LPIPS and DISTS for super-resolution task are averaged over all slices of the three orthogonal planes. The symbol * denotes significance against the second-best method (paired t-test, $p < 0.05$); mean ± s.d. across 8 seeds.

falls within the observed seed variability. While uPL achieves the best mean SSIM, LPIPS, and DISTS, the differences among the three methods are small.

Optimizer. The adaptive weighting strategy is held fixed across all tasks by evaluated once. Table S3 shows that gradient-norm matching consistently outperforms fixed uniform weighting, uncertainty weighting (Kendall et al., 2018), and randomly sampled Dirichlet weights. On SIDD, gradient-norm matching improves PSNR by 0.23–0.77 dB relative to these alternatives, with every competing weighting strategy falling behind on all eight seeds. It needs each term's gradient with respect to the network output once every 50 steps, which does not pass back through the network, and the weights it settled are listed in Table S4. As shown in Table S5, MAdam consistently improves or matches Adam alone. The largest benefit is observed on WorldView-2, where PSNR increases by 0.87 dB, while LOL improves by 0.26 dB and SIDD remains statistically similar (+0.09 dB). These

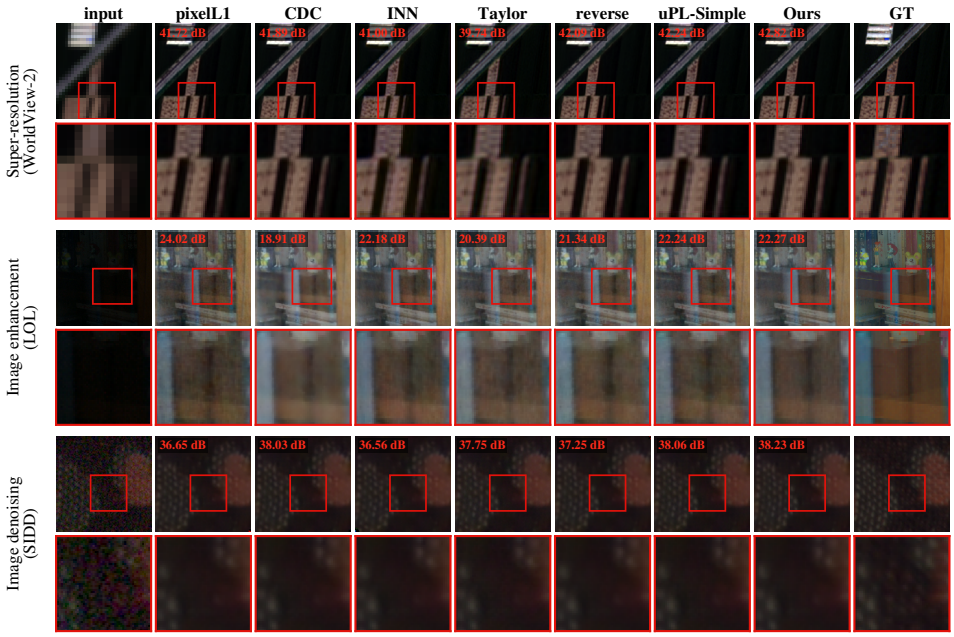

Figure 2: **Representative examples for 2D image reconstruction tasks.** Columns compare competing baseline losses (pixel loss, four prototypes from Zhou et al. (2023), uPL Pfaehler et al. (2025), RaFL (Ours), and the ground truth. PSNR is shown on each panel.

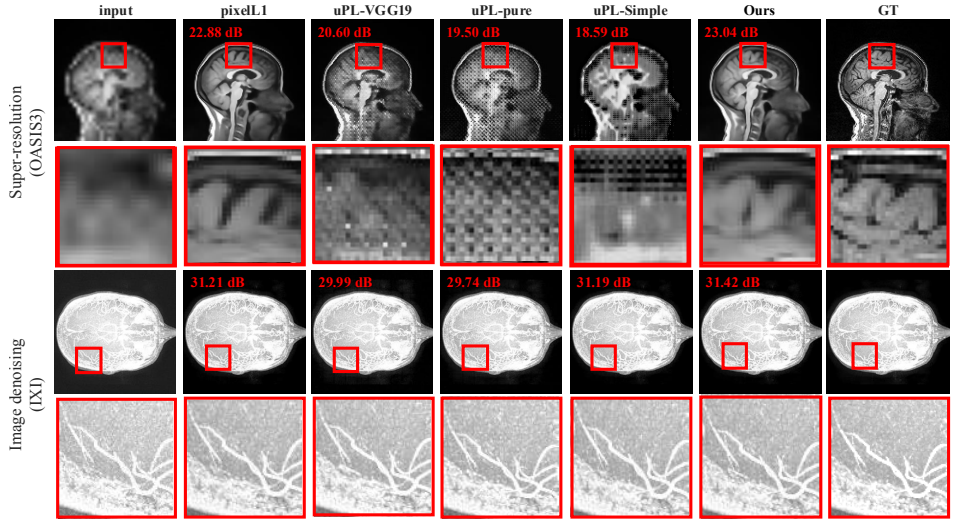

Figure 3: **Representative examples for 3D image reconstruction tasks.** Columns compare competing losses (from left to right): pixel loss, uPL-VGG19, uPL-pure (untrained VGG19), uPL-Simple (3D SimpleNet of (Pfaehler et al., 2025)), and RaFL.PSNR is shown on each panel.

results suggest that accounting for changing objective weights becomes increasingly important on more challenging reconstruction tasks. Additionally, the weights learned by gradient-norm matching are stable across both seeds and tasks (App. Table S4), with the Fourier family consistently receiving the largest weight and the convolution family the smallest.

Overall, RaFL either improves upon or matches the strongest baseline on all five tasks while using the same filter families, redraw schedule, and initialization, without task-specific selection of loss weights. This suggests that the resulting objective transfers across various reconstruction problems.

Image denoising, SIDD, DnCNN				*Super-resolution, OASIS3, SwinUNETR*			
Loss	GFLOPs	of fwd	ms/it	Loss	GFLOPs	of fwd	ms/it
RaFL	1.12	1.9%	17.8	**RaFL**	11.8	1.0%	296.0
RaFL, Adam only	1.12	1.9%	12.1	-	-	-	-
pixel ℓ_1	0.00	0.0%	3.3	pixel ℓ_1	0.00	0.0%	128.4
CDC	5.00	7.8%	4.2	uPL, VGG19	1821.0	60.7%	138.4
reverse	0.31	0.5%	3.6	-	-	-	-
Taylor	0.21	0.4%	3.6	-	-	-	-
INN	0.05	0.1%	3.7	-	-	-	-
uPL, SimpleNet	10.12	14.7%	3.6	-	-	-	-
uPL, VGG19	203.8	77.6%	5.4	-	-	-	-

Table 4: Computational cost of the loss functions on SIDD and OASIS3. GFLOPs measures the loss term only. fwd denotes the fraction of backbone forward pass FLOPs, and ms/it is the time of one training step, measured in one job with Adam for every baseline (App. C).

4.4 Loss Complexity

Table 4 compares the computational cost of the loss functions. On SIDD, the RaFL loss costs 1.12 GFLOPs, corresponding to only 1.9% of the DnCNN forward pass. This is substantially lower than both the CDC prototype (5.00 GFLOPs) and the VGG19-based perceptual loss (203.8 GFLOPs), while remaining competitive with filter based losses. On OASIS3, the relative overhead is even smaller, accounting for only 1.0% of the SwinUNETR forward pass. In contrast, the VGG19 form of uPL requires 1821 GFLOPs, corresponding to 60.7% of the backbone forward-pass cost. Because the random filters are discarded after training, RaFL adds nothing at inference. FLOPs are measured using `FlopCounterMode` and implementation details are provided in App. C. Wall-clock time does not scale directly with FLOPs, as the small random filter operations in RaFL are less hardware-efficient than large convolutional kernels (see §C for details). However, this overhead is incurred only during training and does not affect inference.

5 Discussion and Conclusion

In this work, we investigated which design choices matter most in random-filter losses, focusing on three factors: filter family, weight resampling schedule, and initialization. Building on these findings, we introduced RaFL, a random-filter loss that we evaluated on five image reconstruction tasks spanning natural and medical images in both 2D and 3D.

The most consistent finding is that filter-family selection dominates the other design choices. Combining convolutional, Fourier, and wavelet filters outperforms every proper subset, whereas initialization has only a modest effect on performance. Redraw frequency plays an intermediate role: per-step resampling consistently improves mean performance and reduces seed-to-seed variability. Together, these results suggest that representation diversity is the primary factor governing random-filter loss design. Using the configuration selected on SIDD unchanged across all remaining tasks, RaFL improves upon pixel loss on denoising, pan-sharpening, and super-resolution, while remaining competitive on low-light enhancement and MR angiography denoising. The same filter families, redraw schedule, initialization, and adaptive weighting strategy were applied throughout, without manual loss-weight selection, demonstrating that a single random-filter objective can transfer across reconstruction problems spanning natural and medical images in both 2D and 3D.

Several limitations remain. Although RaFL adds little computational overhead in terms of FLOPs and no inference cost, per-step resampling and gradient balancing increase wall-clock training time. In addition, while the same configuration transferred across all five tasks, the design study was conducted on a single dataset. Finally, understanding why the learned loss weights exhibit similar patterns across datasets and modalities remains an interesting direction for future work.

Overall, this work moves beyond proposing another random-filter loss and provides empirical evidence for which design choices matter most. We hope these findings provide practical guidance for future random-filter objectives and encourage further study of lightweight, domain-agnostic alternatives to pretrained perceptual losses.