WHICH DENOISER? BLIND SELECTION FROM NOISE STRUCTURE

Anonymous authors
Paper under double-blind review

ABSTRACT

Denoising research has produced candidates of increasingly narrow competence, each strong on the noise it was designed or trained for and weak elsewhere. On an arbitrary image, nothing indicates which candidate to run: the scene and the corruption are both unknown, and no clean reference is available. Practitioners therefore apply the denoiser that performs best on average, which in the pools we study falls 1 to 2.5 dB short of the best choice for each image. Closing that gap requires knowing which denoiser suits the image, which is usually approached either by identifying the corruption first or by running every candidate and judging the results. We show that the right denoiser can be found without identifying the corruption exactly, since corruptions that differ often call for the same treatment. In a space of thirteen statistics of the corrupted image, each measuring one physical property of the process that produced the noise, the nearest neighbor of an image has the same best denoiser in 83 to 89% of cases, even though it has the same corruption in only 60%. Matching a new image to its nearest calibration image in this space and running the denoiser that won there recovers 43 to 66% of the available improvement on corruptions withheld from calibration, in one denoising call and with no retraining. Routing falls below the best overall denoiser on noise far from any calibration condition, such as real camera noise, but adding labeled references of that noise restores the gain without retraining. A descriptor space organized by how noise is best treated therefore turns a pool of denoisers with different strengths into a better denoiser than any of its members.

1 INTRODUCTION

Denoisers differ in the noise they are designed for: a fixed variance (Dabov et al., 2007), counts that scale with intensity (Foi et al., 2008), impulses at isolated pixels (Hwang & Haddad, 1995), or, for a trained network, the degradations it was shown during training (Zhang et al., 2017). A denoiser run on noise outside its intended range either erases detail that belongs to the scene or leaves the corruption in place. Choosing among them requires knowing which corruption is present in a given image. But scene and corruption are both unknown, and no clean reference separates them. How, then, is one to denoise an arbitrary image?

Although noise is unpredictable, the process that produced it is not. For example, photon counting yields a variance that grows with intensity; a periodic readout artifact concentrates energy at a few frequencies. Regularities of this kind persist across realizations and across scenes, so they can be estimated from the corrupted image alone. We collect thirteen such statistics, covering amplitude, marginal law, dependence on the signal, and spatial structure. Each image therefore yields a descriptor of the process that corrupted it, a point in a thirteen-dimensional space. Similar processes should occupy nearby regions of this space. Since only a finite number of corruptions can be tested, we evaluate these claims on conditions withheld from the descriptor's design, on mixtures of two conditions, and at severities not used in calibration.

Given such a space, blind denoiser selection becomes a retrieval problem. We take a pool of frozen denoisers, classical or pretrained, and label a set of calibration corruptions with the candidate that performs best on each. Each labeled point defines the region of the space nearest to it, so the labels partition noise space by which denoiser is preferred. A new image is mapped into this space, matched to its nearest reference, and denoised by the candidate that won there, at a cost of one denoising call

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.

and no retraining.[1] A query therefore need not land in a region labeled with its own corruption, only in one whose denoiser serves it well. To separate what the descriptor contributes from what the pool allows, we measure the geometry and the outcome: whether descriptor distance tracks the quality lost by substituting one denoiser for another, and whether routing this way beats the alternative of picking the best overall denoiser on the calibration set and running it on every image.

On corruptions withheld from calibration, routing recovers 43 to 66% of the improvement available to a perfect per-image choice, using a single denoising call, in the three pools where that improvement exceeds 2 dB, and it keeps that gain at severities above the calibrated range. In the fourth pool, where the available improvement is 0.93 dB, routing matches the best overall denoiser. On real camera noise, which no calibration condition reproduces, routing falls below the best overall denoiser, but adding labeled references of that noise restores the gain without retraining. The established alternative of running every candidate and ranking the outputs by a quality score computed without the clean image (Zhu & Milanfar, 2010; Lu, 2019) runs one denoising call per candidate, and in every pool it recovers less of the available improvement than routing.

2 Related work

Prior methods for choosing a denoiser under an unknown corruption make the choice either after running the candidates or before, from a description of the corruption. Methods of the first kind score each output without a clean reference, using measures designed for denoising (Zhu & Milanfar, 2010; Lu, 2019; Rakhshanfar & Amer, 2018), general no-reference quality scores (Mittal et al., 2013; Ke et al., 2021), or Monte-Carlo SURE (Ramani et al., 2008), and they must produce and score every candidate output before deciding. Methods of the second kind classify the noise type and dispatch to a matched network (Liu et al., 2020; Lemarchand et al., 2020), estimate the parameters of a sensor noise model (Foi et al., 2008; Huang & Huang, 2025), or route each input to a degradation-specific expert through a learned representation (Potlapalli et al., 2023; Ashrafee & Krawczyk, 2026). In each case the corruption is described by its identity or by model parameters, and the denoiser follows from that description. A third line trains one network to cover many corruptions (Zhang et al., 2021; Zamir et al., 2022; Zhang et al., 2023); such networks form our general-purpose pool. We label calibration references by the candidate that denoises them best rather than by their corruption, so a query need only be matched to references served by the same denoiser, and our descriptor is designed rather than learned. Appendix A discusses each line of work in more detail.

3 Method

Routing chooses a candidate denoiser for a new image by comparing its descriptor with the descriptors of a set of calibration images whose best performing denoiser is known. Its design therefore depends on which corruptions the descriptor must distinguish and on which candidates are available to choose from. We first define the corruption families and the properties along which they vary (Section 3.1), then the descriptor built to measure those properties (Section 3.2) and the pools of candidate denoisers (Section 3.3). Section 3.4 describes the routing procedure, Section 3.5 the methods we compare it with, and Section 3.6 the evaluation protocol.

3.1 Corruptions

We consider ten families of corruption, each with one to three physical parameters (Table 1 in Appendix B). Every family has a parameter that sets the magnitude of the corruption. Beyond amplitude, the families are chosen to vary the other three properties the descriptor is meant to capture: the marginal law of the noise (Gaussian, Laplacian, uniform, and impulse), its dependence on the signal (Poisson and speckle), and its spatial structure (correlated, anisotropic, banding, and annular). Appendix B defines each family and gives its physical motivation. A *condition* fixes a family and its parameter values, for example Gaussian noise with $\sigma = 25$ or banding with amplitude 20 and a period of 10 pixels. A *mixture* assigns each pixel at random to one of two conditions, so the image contains both noises but no pixel receives both (Appendix B).

[1] Code: `https://anonymous.4open.science/r/blind-denoiser-selection-8D40`.

3.2 DESCRIPTOR

The descriptor, which we call D13, maps a corrupted image to thirteen statistics, each chosen to measure one of the properties along which the corruption families of Section 3.1 vary. The statistics are computed from the corrupted image alone and are designed to respond more strongly to the corruption than to the scene. Figure 6 in Appendix C shows, for each statistic, the range of its values as one corruption parameter is varied, relative to its variation across photographs and noise realizations.

Statistics. Table 2 in Appendix C groups the thirteen statistics by the property each measures. All are computed on a high-pass residual of the image, formed by splitting it into the four interleaved phases of a 2×2 decomposition and subtracting adjacent phases, giving two horizontal and two vertical differences. The decomposition follows the pixel-shuffle downsampling of Lee et al. (2022). Differencing adjacent phases suppresses image content that is shared locally while retaining the statistics of the corruption. The spectral statistics, except for the energy in four frequency bands, describe how concentrated and how directional the spectrum is without recording the frequency at which the energy lies. A periodic corruption whose frequency changes therefore stays close in descriptor space, and such a change leaves the denoising problem largely the same.

Calibration set and standardization. Because the variance and the kurtosis of the residual range over several orders of magnitude, we replace the variance by its logarithm and the kurtosis by its inverse hyperbolic sine (Appendix C). We then standardize each statistic to zero mean and unit variance over a *calibration set*, consisting of clean photographs corrupted under a fixed collection of calibration conditions (Table 4 in Appendix E).

3.3 POOLS OF DENOISERS

A *candidate* is a denoiser that is applied without retraining or tuning, in one call per image, and a *pool* is a fixed set of candidates. Routing can improve on a single denoiser only when the candidates in a pool are strong on different corruptions. We therefore evaluated four pools that differ in how strongly their candidates specialize.

The *classical* pool contains 10 fixed operators, including Gaussian, median, bilateral, and Wiener filters, total-variation and a notch filter for periodic corruption. The *specialist* pool contains 14 networks of a common architecture, a reduced U-Net (Ronneberger et al., 2015), each trained on one calibration condition with photographs from DIV2K (Agustsson & Timofte, 2017), none of which appear in our benchmarks. The *general-purpose* pool contains 8 pretrained candidates, among them DRUNet (Zhang et al., 2021), SwinIR (Liang et al., 2021), Restormer (Zamir et al., 2022), and SCUNet (Zhang et al., 2023). Their training covered only Gaussian noise or real camera noise, and they are expected to perform more alike across corruptions. The *hybrid* pool contains 17 candidates, the eight general-purpose networks together with three further pretrained networks and six classical operators. Appendix D lists the candidates in each pool and gives implementation details.

3.4 ROUTING

Our routing procedure has a calibration stage, carried out once for each pool, and a test stage, carried out for each new image (Figure 1). In the calibration stage, every candidate m in the pool $\mathcal{A}$ is run on every image of the calibration set. Writing y for a corrupted image, x for its clean photograph, and $\mathbb{E}_c$ for the mean over the images of condition c, we label each calibration condition with the candidate of highest mean PSNR, $m_c^* = \arg\max_{m \in \mathcal{A}} \mathbb{E}_c[\mathrm{PSNR}(m(y), x)]$. Each calibration image i then serves as a reference (d_i, c_i), consisting of its standardized descriptor and its condition.

In the test stage, the standardized descriptor d of a new image y is computed from $D_{13}(y)$, the nearest reference $i^* = \arg\min_i \|d - d_i\|_2$ is retrieved, and the candidate $m_{\mathrm{route}} = m^*_{c_{i^*}}$ is run on the image. Routing therefore costs one descriptor computation and one denoising call. Computing the descriptor takes 23 ms for a 256×256 image on one core of an Apple M2 Max, comparable to a median or total-variation filter. Each calibration image serves as its own reference. We do not average the descriptors of a condition into one prototype per condition, because images of the same condition can fall in separate regions of the space.

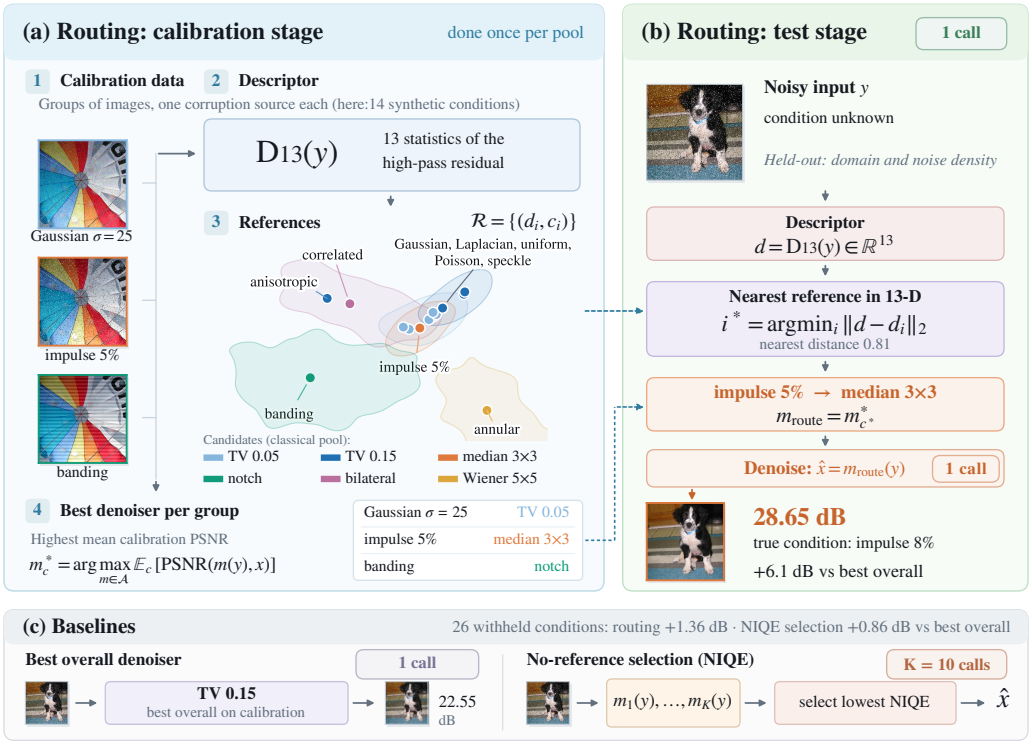

Figure 1: **Routing chooses a denoiser from the corrupted image before running any.** (a) During calibration, candidates are run on groups of photographs that share a source corruption, and each calibration image becomes a reference in descriptor space (drawn schematically), paired with the candidate that denoised its group best. The descriptor, D13, consists of thirteen statistics that respond to the corruption more than to the scene. (b) During test, a new image arrives without information about its corruption. It is matched to its nearest reference, and only that reference's candidate is run. The example is corrupted by impulse noise at a density and in a content domain not used in calibration; routing matches it to the reference for 5% impulse noise and runs the median filter, which improves on the best overall denoiser by 6.1 dB. (c) We compare routing with two baselines: the best overall denoiser is chosen during calibration and run on every image; no-reference selection runs every candidate and keeps the output with the best quality score. With the classical pool, routing gains 1.36 dB over the best overall denoiser on withheld conditions, more than the 0.86 dB gained by selection with NIQE, which runs ten candidates instead of one. TV denotes total-variation denoising.

Because labels record which candidate performs best and not which corruption is present, the procedure does not need to identify the corruption of a new image. As noted in the introduction, an image matched to the wrong condition is still denoised well when that condition has the same or a similarly effective best candidate.

3.5 Comparisons

We compare routing with three other ways of choosing a candidate denoiser for each test image. All comparisons are made within a pool, using only its candidates, and PSNR is always measured against the clean image. The three differ in how many candidates they run and in what information they use to choose among them.

The *best overall* denoiser is the candidate with the highest mean PSNR on the calibration set. It is chosen once and run on every test image, so, like routing, it costs one call per image. It is the alternative named in the introduction, and we report all gains as the difference in PSNR from it.

The *oracle* runs every candidate on a test image and keeps the output with the highest PSNR. Because it needs the clean image, it cannot be used in a realistic scenario, and we use it only to measure the largest gain that any choice among a pool's candidates can achieve. Its difference from the best overall denoiser is the *oracle gap*, and we report the fraction of this gap that routing recovers.

No-reference selection also runs every candidate, but it keeps the output that scores best under a no-reference image quality metric, computed on the output alone. Unlike the oracle, it can be used in the absence of ground truth, but it chooses correctly to the extent that the metric orders the outputs as PSNR would. It costs at least one call per candidate and one quality-score evaluation per output, whereas routing costs one call in total.

3.6 Evaluation protocol

All photographs came from ten content domains, the ten classes of ImageNette (Howard, 2019), with thirty photographs per domain in each benchmark (Appendix E). To keep calibration and test images separate, we held out one domain at a time, calibrated on photographs from the other nine, and tested on photographs from the held-out domain. Results were pooled over the ten held-out domains. For each held-out domain, the references, the standardization statistics, the labels of the calibration conditions, and the best overall denoiser were all recomputed from the other nine domains.

We evaluated routing on two benchmarks, one testing transfer to conditions and mixtures not used in calibration, the other testing transfer to severities not used in calibration within a family.The *discrete* benchmark used 40 conditions, of which 14 formed the calibration set (Section 3.2) and the other 26 were withheld. The withheld conditions included parameter values not used in calibration, variants of the structured families, and mixtures. Routing was tested only on the withheld conditions.

The *parametric* benchmark varied one physical parameter per family, holding the others fixed, at six severity levels numbered 1 to 6 from lowest to highest, and used its own calibration set. Three splits divided the levels between calibration and test: S1 calibrated on levels 1, 3, and 5 and tested on levels 2 and 4, S2 calibrated on levels 1 to 3 and tested on levels 4 to 6, and S3 calibrated on levels 4 to 6 and tested on levels 1 to 3 (Appendix E).

We report the mean gain in PSNR over test images, where a test image is one photograph corrupted under one condition, and every test image has equal weight. Confidence intervals are 95% intervals from a paired bootstrap over content domains and photographs, with 5,000 replicates (Appendix E). Each photograph was corrupted once under each condition, with an independent noise realization, so every condition is evaluated over 300 realizations and none is repeated under a second seed.

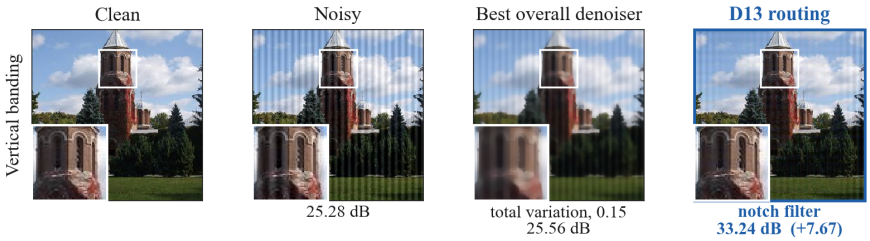

Figure 2: **Example of routing.** Vertical banding noise, denoised by the best overall denoiser of the classical pool (total variation) and by the candidate that routing selected (a notch filter), with PSNR under each panel. Appendix I gives examples from families with smaller gains (Section 4.4).

4 Results

The gain of routing over the best overall denoiser depends on the descriptor, which must place conditions that share a best candidate near each other, and on the pool, whose candidates must differ enough on those conditions for the choice to matter. We first test whether images that are close in descriptor space call for the same denoiser (Section 4.1). We then measure the gain of routing on the conditions withheld from calibration (Section 4.2) and break it down by corruption family (Section 4.4). Finally, we test routing at severities not used in calibration (Section 4.3) and on real camera noise, which no calibration condition reproduces (Section 4.5).

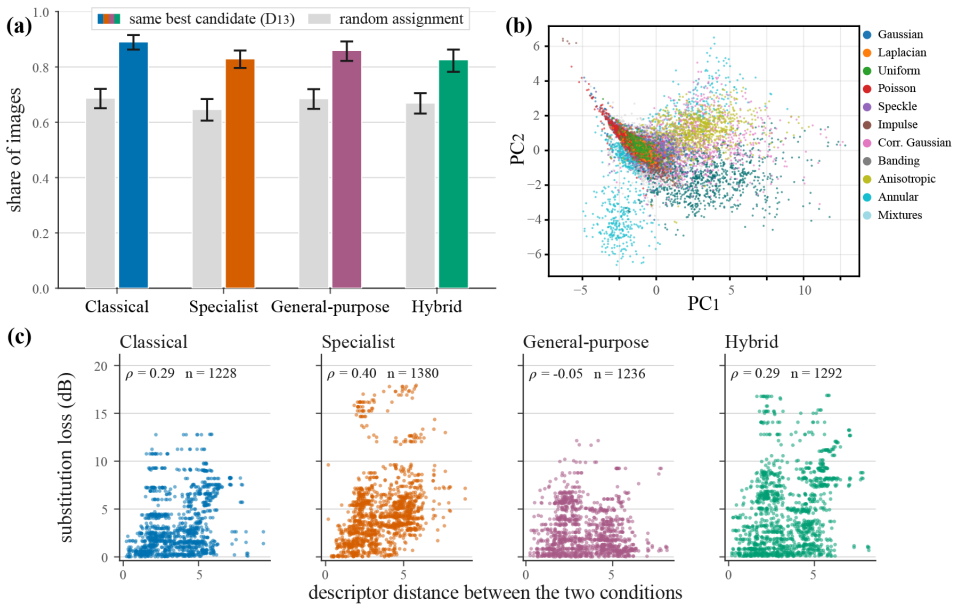

Figure 3: **D13 places noise conditions that call for the same denoiser near each other.** (a) Fraction of images whose nearest neighbor under D13, retrieved over all 40 conditions, had the same best candidate as the query (colored bars). Gray bars show the expected fraction when best candidates are assigned to conditions at random, with each candidate remaining best on the same number of conditions. Error bars are 95% paired-bootstrap intervals. (b) Descriptors of 12,000 images of the discrete benchmark projected onto their first two principal components, which account for 39% and 12% of the variance, colored by noise family. Mixtures of two conditions are shown in gray. Families that overlap in this projection may be separated along the remaining components. (c) Substitution loss against descriptor distance for each ordered pair of conditions. The substitution loss is the drop in PSNR when images of the first condition are denoised by the best candidate of the second. Pairs that share a best candidate have zero loss and are omitted from the plot and the correlations.

4.1 The descriptor space

Routing can only be as good as the descriptor space it relies on. Before measuring its gain, we therefore tested whether images that are close in this space call for the same denoiser. We tested this on the discrete benchmark by retrieving, for each image, its nearest neighbor among the images of all 40 conditions from the other nine content domains. The neighbor had the same best candidate denoiser as the query in 83 to 89% of images, depending on the pool, even though it had the same noise condition in only 60% (Figure 3a). Because a few candidates are best on many conditions, neighbors agree to some extent by chance: assigning best candidates to conditions at random, keeping the number of conditions each candidate wins, gave 65 to 69%. The descriptor therefore agrees far more often than the random assignment does. Routing can send an image to the right denoiser even when it matches the image to the wrong condition.

When the neighbor's best candidate differed from the query's, the loss depended on how well the neighbor's denoiser served the query. For each ordered pair of conditions, we denoised the images of the first condition with the best candidate of the second and measured the mean drop in PSNR relative to the first condition's own best candidate, which we call the substitution loss. We measured the distance between two conditions as the Euclidean distance between the means of their descriptors. Among pairs whose best candidates differed, this substitution loss was positively rank-correlated with descriptor distance in the classical, specialist, and hybrid pools (Spearman correlations of 0.29, 0.40, and 0.29; Figure 3b). Routing sends a query to its nearest reference, so its errors fall between conditions that are close in this space and tend to lose less PSNR than errors between distant conditions. In the general-purpose pool there was no relation (-0.05), a case we return to in Section 4.2.

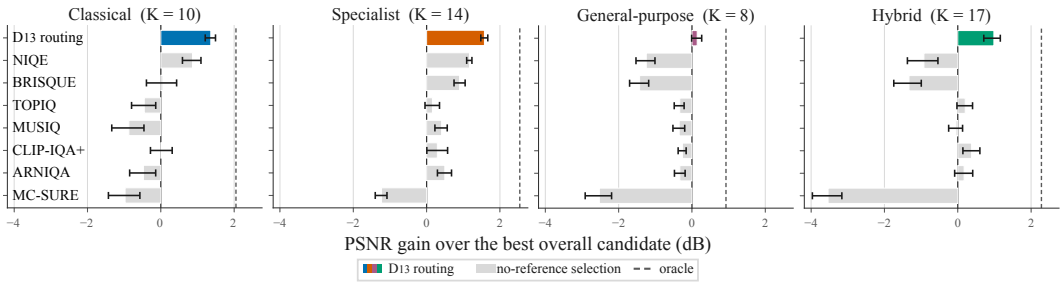

Figure 4: **Selection from the noisy image versus selection from denoised outputs.** Gain in PSNR over the best overall candidate for each pool. Colored bars show routing with D13, which runs one candidate. Gray bars show selection among the outputs of all K candidates by a no-reference quality score, or by MC-SURE, which requires $2K$ calls. Dashed lines mark the oracle gap (gain available to a per-image choice with access to the clean target). Error bars are 95% paired-bootstrap intervals. D13 routing exceeds every no-reference selector in every pool.

4.2 Routing on withheld conditions

Choosing the right denoiser for an image is not enough to improve on the best overall denoiser, because on many images the two are the same candidate and routing then runs it. A gain can arise only on images where the two choices differ. We measured it on the 26 conditions of the discrete benchmark withheld from calibration, 14 of which are mixtures. The gain of routing over the best overall denoiser was 1.36 dB for the classical pool, 1.58 dB for the specialist pool, and 0.99 dB for the hybrid pool, with 95% intervals lying entirely above zero (Figure 4). These gains recovered 66, 62, and 43% of the oracle gap. Figure 2 shows one image on which routing chose a better candidate than the best overall denoiser. For the general-purpose pool the gain was 0.14 dB, recovering 15% of a gap of 0.93 dB. The 95% interval runs from -0.01 to 0.27 dB, so routing matched the best overall denoiser but did not measurably improve on it. Both the pool and the descriptor limit routing here: the oracle gap is the smallest of the four pools, and descriptor distance did not predict substitution loss (Section 4.1).

Comparison with no-reference selection. Routing chooses a denoiser before running any. The alternative is to run every candidate and choose among the outputs, using a score computed without the clean image. We compared routing with seven such selectors (Appendix F). Six are no-reference quality scores: NIQE (Mittal et al., 2013), BRISQUE (Mittal et al., 2012), and four recent learned scores, MUSIQ (Ke et al., 2021), CLIP-IQA+ (Wang et al., 2023), TOPIQ (Chen et al., 2024), and ARNIQA (Agnolucci et al., 2024). The seventh, Monte-Carlo SURE (Ramani et al., 2008), estimates the mean squared error of each output under the assumption of additive white Gaussian noise, with the noise level estimated from the corrupted image, and requires a second denoising call per candidate. Routing exceeded every selector in every pool, with 95% intervals of the difference lying entirely above zero, at one denoising call against K for the quality scores and $2K$ for SURE. The closest selector, NIQE, fell short of routing by 0.41 dB in the specialist pool and 0.50 dB in the classical pool. In the general-purpose pool, where the candidates differ least, every selector did worse than running the best overall denoiser on every image, whereas routing did not. SURE estimates the error that PSNR measures, but only under its Gaussian assumption; on impulse and correlated noise its ranking of the candidates was negatively correlated with their true error (Appendix F).

Comparison with noise classification. The other alternative is to identify the corruption first. We reimplemented the classifier of Liu et al. (2020), which assigns an image to one of four noise types, and compared it with routing restricted to the same types, on the eight withheld conditions composed only of them (Appendix J). With the published preprocessing, routing gave higher PSNR than the classifier in every pool, by 0.47 to 0.70 dB. At full resolution, the classifier still gave lower PSNR in three pools and did not differ in the fourth. The difference arose on mixtures of two types, for which the classifier has no label.

4.3 Routing at unseen severities

The withheld conditions of the discrete benchmark differ from the calibration conditions in several parameters at once. The parametric benchmark varies one parameter per family and tests severities between calibrated ones (S1), above the calibrated range (S2), and below it (S3) (Section 3.6). Routing kept its gain in S1 and S2, recovering about three quarters of the oracle gap in the classical and hybrid pools and about half in the general-purpose pool. In S3 it recovered only a quarter to a third, although the oracle gap was largest there (Figure 7 in Appendix G). Mild noise changes the statistics the descriptor measures little and is hard to distinguish from image content, so severities below the calibrated range are matched to the mildest calibrated reference, whose best candidate over-smooths them. The specialist pool is not evaluated on this benchmark, because its networks were each trained on one calibration condition of the discrete benchmark (Appendix G).

4.4 Gain by corruption family

The gains of Section 4.2 are averages over corruption families whose oracle gaps differ by more than two orders of magnitude, from 0.04 to 15.8 dB (Table 9 in Appendix H). Routing recovered nearly all of the gap where it was largest. On impulse noise it recovered 99% in every pool, and on banding 81 and 80% in the classical and hybrid pools, where the gap was about 8 dB. These two families form 3 of the 26 withheld conditions but accounted for 0.76 of the 1.36 dB gained by the classical pool, and for all of the gain of the hybrid and general-purpose pools. On the other 23 conditions, routing gained 0.69 dB with the classical pool (95% interval $[0.51, 0.84]$) and 1.09 dB with the specialist pool ($[0.99, 1.17]$), whereas with the hybrid and general-purpose pools the gain was -0.06 dB, with intervals that included zero. Where the oracle gap was small, routing could lose PSNR. In the general-purpose and hybrid pools, the gap on correlated, annular, and signal-dependent noise was below 0.2 dB, yet routing lost 0.47 to 1.13 dB on these families by selecting candidates worse than the best overall denoiser. The classical pool did the same on additive noise, losing 0.76 dB where the gap was 0.20 dB. The largest shortfall on a family with a large gap was banding in the specialist pool, where routing recovered 7% of a 3.0 dB gap. This pool contains a single banding specialist, trained on horizontal stripes, and the withheld banding condition has vertical ones. Mixtures, which form 14 of the 26 withheld conditions, combine two families in one image. The classical and specialist pools recovered 59 and 58% of their gap on mixtures, which was about 1.6 dB, and the general-purpose and hybrid pools recovered 13 and 8%.

4.5 Noise outside calibration

Routing runs the candidate that was best at the nearest reference, so it can be no better than the references near a query. When a query's noise resembles no calibration condition, its nearest reference does not describe it, and routing should lose its advantage over the best overall denoiser and may fall below it. The remedy the method suggests is to add references of the new noise, labeled with their best candidate, which requires clean images of that noise but no retraining. We tested both expectations on real camera noise, which combines dependence on the signal, spatial correlation, and the effects of the camera's processing, and which no calibration condition reproduces.

We routed the 1,280 sRGB validation blocks of SIDD (Abdelhamed et al., 2018) with all four pools. With the synthetic calibration set alone, the median distance from a SIDD block to its nearest reference was 3.42, against 1.02 for images of withheld synthetic conditions and 0.81 for images of calibration conditions. As expected, routing did worse than the best overall denoiser of the synthetic calibration set, by 1.01 dB with the specialist pool, 3.29 dB with the classical pool, and about 8 dB with the general-purpose and hybrid pools, where most of the loss was the cost of the fixed choice itself rather than of routing (Figure 5, Appendix K).

The descriptor also indicated, before any denoiser was run, when a query lay outside the calibration conditions. We flagged the images whose descriptor had a negative log-likelihood, under the best-fitting calibration condition, above the 99th percentile of the calibration images. The flag fired on 90% of the SIDD blocks and on 7% of the images of withheld synthetic conditions. On those synthetic images, routing recovered a smaller share of the oracle gap when the image was flagged, in all four pools (Appendix K). The flag therefore warns a user, before any denoiser is run, that an image lies outside the calibration conditions and that routing is less reliable on it.

We then added the 320 SIDD Small patches as references, each labeled with the best candidate for its ISO band. Routing then recovered most of the loss in every pool, and changed the results on withheld synthetic conditions by at most 0.04 dB. Part of the recovery was a change of default rather than a per-image choice, since the real references point to a network trained on real camera noise; in the general-purpose and hybrid pools that network accounted for the whole of the gain, and a per-image choice could have added at most 0.12 and 0.27 dB, respectively. In the specialist pool, whose networks had never seen real noise, routing gained 1.84 dB [1.11, 2.59] over the best candidate on the real references, out of a gap of 3.48 dB (Appendix K).

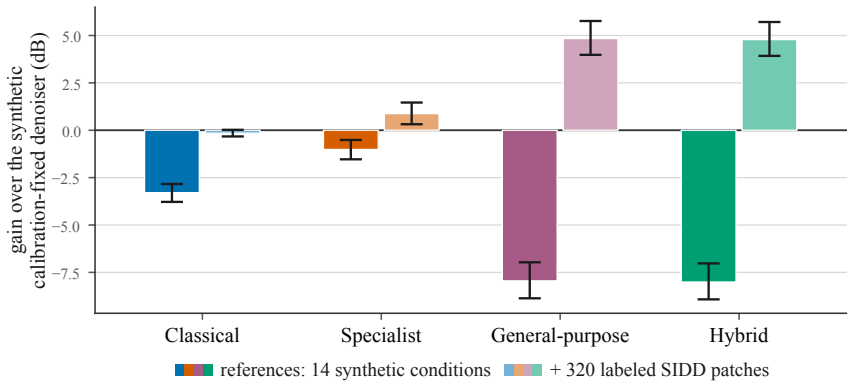

Figure 5: **Adding references repairs routing on real camera noise.** Gain in PSNR on the SIDD validation images over the best overall denoiser of the synthetic calibration set, with references from that set alone and with 320 labeled SIDD Small patches added to it. Error bars are 95% intervals.

5 Discussion

We showed that a pool of existing denoisers can be combined into a single procedure that adapts to the noise of each image, without retraining any candidate and at the cost of one denoising call. When the candidates of a pool were strong on different corruptions, routing recovered much of the improvement available to a per-image choice, including on conditions withheld from calibration and on mixtures of two conditions. In every pool, it gave higher PSNR than selection among the outputs of all candidates, while running one candidate instead of all of them. When the candidates were all trained for the same noise, there was little to recover, and routing did on average as well as the best overall denoiser. Within a family, routing transferred to severities above the calibrated range nearly as well as between calibrated severities, but less so to milder noise, so the calibration conditions should include the mildest noise a user expects.

Limitations. Routing depends on references that resemble the noise of a new image, and extending it to a new kind of noise requires clean images of that noise for labeling. The gain also depends on how differently the candidates perform on the noise at hand. Where little gain was available from a per-image choice, routing sometimes selected a worse candidate than the best overall denoiser; this can be checked on the calibration set before routing is used, and a fallback to the best overall denoiser in such cases is a natural extension. The descriptor measures only the properties it was designed to capture, so corruptions that differ in another respect, such as the orientation of stripes, fall close together even when a pool's candidates treat them differently. A statistic for such a property can be added to the descriptor at the cost of recomputing the references but without retraining any candidate. Both the labels and the evaluation used PSNR, and a perceptual metric may prefer a different candidate. Labeling the references by another metric requires no change to the procedure.

Extension to new noise. The procedure can be extended to a kind of noise it was not calibrated for by adding labeled images of that noise as references. For real camera noise, which lay far from the synthetic references, a few hundred such images removed most of the loss, without retraining any candidate and without affecting results on synthetic noise. Whether a pool is worth routing over can be assessed in advance from the gain available on the calibration set, and whether a new image is covered by the references from the flag of Section 4.5.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology and experiments and to implement methods. We have not used generative AI tools to propose or refine hypotheses, design research methodology or experiments, assist with translation, clean or reformat datasets, support qualitative and thematic data analysis, or interpret results. Generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving them, and writing proofs are not applicable to this work. Additionally, we used generative AI tools to create or edit software code, create or edit scientific figures, draft parts of the paper, edit the paper to improve readability, and identify relevant literature. We have reviewed all AI-assisted work. Two authors verified and tested all AI-generated code for correctness, and all authors checked every figure and passage of text produced or edited with AI assistance. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All data are public. The photographs come from ImageNette and DIV2K, and the real camera noise from the SIDD validation set (Appendices E and K). Appendix B defines the corruption families and their parameters, and Appendix C defines the thirteen statistics of the descriptor. Appendix D lists every candidate with its implementation, checkpoint, and software version, and describes the training of the specialist networks. Appendix E gives the benchmarks and the bootstrap procedure, and Appendices F and J give the baselines. Code is available at `https://anonymous.4open.science/r/blind-denoiser-selection-8D40`.

REFERENCES

Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1692–1700. IEEE, 2018.

Lorenzo Agnolucci, Leonardo Galteri, Marco Bertini, and Alberto Del Bimbo. ARNIQA: Learning distortion manifold for image quality assessment. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 188–197. IEEE, 2024.

Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1122–1131. IEEE, 2017.

Alif Ashrafee and Bartosz Krawczyk. Restoring without forgetting: Continual learning across image degradations. *arXiv preprint arXiv:2608.23799*, 2026.

Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1):89–97, 2004.

Chaofeng Chen and Jiadi Mo. IQA-PyTorch: Pytorch toolbox for image quality assessment. [Online]. Available: `https://github.com/chaofengc/IQA-PyTorch`, 2022.

Chaofeng Chen, Jiadi Mo, Jingwen Hou, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. TOPIQ: A top-down approach from semantics to distortions for image quality assessment. *IEEE Transactions on Image Processing*, 33:2404–2418, 2024.

Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European Conference on Computer Vision*, pp. 17–33. Springer, 2022.

Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16 (8):2080–2095, 2007.

David L Donoho and Iain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

Claudio Fantasia, Luca Calatroni, Xavier Descombes, and Rim Rekik. Patch-based learning of adaptive total variation parameter maps for blind image denoising. In *33rd European Signal Processing Conference*, pp. 1712–1716. IEEE, 2025.

Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008.

Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Pearson, 4th edition, 2018.

Joseph W Goodman. *Speckle phenomena in optics: theory and applications*. Roberts and Company Publishers, 2007.

Jeremy Howard. Imagenette: A smaller subset of 10 easily classified classes from imagenet, March 2019. URL `https://github.com/fastai/imagenette`.

JiaKui Hu, Lujia Jin, Zhengjian Yao, and Yanye Lu. Universal image restoration pre-training via degradation classification. In *International Conference on Learning Representations*, volume 2025, pp. 64647–64668, 2025.

Yuanfei Huang and Hua Huang. Beyond image prior: Embedding noise prior into latent space of conditional denoising transformer. *International Journal of Computer Vision*, 133(11):7591–7611, 2025.

Humor Hwang and Richard A Haddad. Adaptive median filters: new algorithms and results. *IEEE Transactions on Image Processing*, 4(4):499–502, 1995.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: Multi-scale image quality transformer. In *2021 IEEE/CVF International Conference on Computer Vision*, pp. 5128–5137. IEEE, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Wooseok Lee, Sanghyun Son, and Kyoung Mu Lee. AP-BNS: Self-supervised denoising for real-world images via asymmetric PD and blind-spot network. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17704–17713. IEEE, 2022.

Florian Lemarchand, Thomas Findeli, Erwan Nogues, and Maxime Pelcat. NoiseBreaker: Gradual image denoising guided by noise analysis. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing*, pp. 1–6. IEEE, 2020.

Boyun Li, Xiao Liu, Peng Hu, Zhongqin Wu, Jiancheng Lv, and Xi Peng. All-in-one image restoration for unknown corruption. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17431–17441. IEEE, 2022.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using Swin Transformer. In *2021 IEEE/CVF International Conference on Computer Vision Workshops*, pp. 1833–1844. IEEE, 2021.

Fan Liu, Qingzeng Song, and Guanghao Jin. The classification and denoising of image noise based on deep neural networks. *Applied Intelligence*, 50(7):2194–2207, 2020.

Si Lu. No-reference image denoising quality assessment. In *Science and Information Conference*, pp. 416–433. Springer, 2019.

Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012.

Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

Vaishnav Potlapalli, Syed Waqas Zamir, Salman H Khan, and Fahad Shahbaz Khan. Promptir: Prompting for all-in-one image restoration. *Advances in Neural Information Processing Systems*, 36:71275–71293, 2023.

Meisam Rakhshanfar and Maria A Amer. Sparsity-based no-reference image quality assessment for automatic denoising. *Signal, Image and Video Processing*, 12(4):739–747, 2018.

Sathish Ramani, Thierry Blu, and Michael Unser. Monte-Carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms. *IEEE Transactions on Image Processing*, 17(9):1540–1554, 2008.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image Computing and Computer-assisted Intervention*, pp. 234–241. Springer, 2015.

Julián Tachella, Matthieu Terris, Samuel Hurault, Andrew Wang, Leo Davy, Jérémy Scanvic, et al. DeepInverse: A python package for solving imaging inverse problems with deep learning. *Journal of Open Source Software*, 10(115):8923, 2025. doi: 10.21105/joss.08923. URL `https://doi.org/10.21105/joss.08923`.

C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pp. 839–846, 1998. doi: 10.1109/ICCV.1998.710815.

Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, et al. scikit-image: image processing in Python. *PeerJ*, 2:e453, 2014.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, et al. SciPy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17 (3):261–272, 2020.

Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring CLIP for assessing the look and feel of images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 2555–2563, 2023.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5728–5739. IEEE, 2022.

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26 (7):3142–3155, 2017.

Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6360–6376, 2021.

Kai Zhang, Yawei Li, Jingyun Liang, Jiezhang Cao, Yulun Zhang, Hao Tang, Deng-Ping Fan, Radu Timofte, and Luc Van Gool. Practical blind image denoising via Swin-Conv-UNet and data synthesis. *Machine Intelligence Research*, 20(6):822–836, 2023.

Yuqian Zhou, Jianbo Jiao, Haibin Huang, Yang Wang, Jue Wang, Honghui Shi, and Thomas Huang. When AWGN-based denoiser meets real noises. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 13074–13081, 2020.

Xiang Zhu and Peyman Milanfar. Automatic parameter selection for denoising algorithms using a no-reference measure of image content. *IEEE Transactions on Image Processing*, 19(12):3116–3132, 2010.

A Extended related work

This appendix expands Section 2. Prior work either defers the choice of denoiser until the candidates have been run and their outputs can be compared, or makes it in advance by describing the corruption, as a type, as the parameters of a noise model, or as a learned embedding. A third line avoids the choice by training one network to cover every corruption. None of these methods asks what a description of the noise must preserve in order to choose among given denoisers. Our descriptor is designed around that question.

Selecting a denoiser from its output. One line of work runs the candidates and judges the results. Zhu & Milanfar (2010) score a denoised image by the coherence of its gradients, Lu (2019) learns a regression from features of the denoised image and of the removed noise, and Rakhshanfar & Amer (2018) score a result by the structure left in the residual. A second group scores the outputs with a general-purpose no-reference quality measure rather than one designed for denoising, from the natural-scene statistics of NIQE (Mittal et al., 2013) and BRISQUE (Mittal et al., 2012) to learned scores such as MUSIQ (Ke et al., 2021), CLIP-IQA+ (Wang et al., 2023), TOPIQ (Chen et al., 2024), and ARNIQA (Agnolucci et al., 2024). Monte-Carlo SURE (Ramani et al., 2008) instead estimates the mean squared error of each output, at the cost of a second denoising call per candidate and under an assumption of additive white Gaussian noise. All of these methods need no clean image and can tune a parameter as well as choose an algorithm, but they must produce every candidate output before deciding, and each output must then be scored; Lu (2019) reports about five seconds per image for this step.

Recognizing the corruption. A second line identifies the noise and applies a denoiser matched to it. Liu et al. (2020) classify the noise type with one network and denoise with another, and Lemarchand et al. (2020) classify the noise into five types, each with its own trained network, restoring the image in several passes. Classification is also used within a single method, to select the data term of a variational reconstruction (Fantasia et al., 2025) or to pretrain a restoration network (Hu et al., 2025). These methods assume that the corruption must be identified for the denoiser to be chosen, and they train a classifier to identify it. Our descriptor measures properties of the corruption without training and uses them only to choose the denoiser.

One network for many corruptions. A different response to the same problem is to train one network to cover a range of corruptions, whether by conditioning on an estimated noise level (Zhang et al., 2021), by training blind over a range of severities (Zamir et al., 2022), or by synthesizing a wide degradation model at training time (Zhang et al., 2023). These networks are the candidates of our general-purpose pool. Training of this kind makes the candidates of a pool perform alike, which is the case in which a choice among them has least to offer, and it requires the corruptions to be anticipated during training, whereas a pool can be extended by adding a denoiser and its calibration references.

Estimating noise parameters. Rather than a discrete type, the noise can be described by the parameters of a sensor model. Foi et al. (2008) fit the variance of smooth patches against their intensity, and Huang & Huang (2025) estimate the two parameters of a Poisson–Gaussian model from a single raw image and condition a trained transformer on them. Our signal-dependence statistic is an estimate of this kind, computed on pixel differences rather than on patches selected for smoothness. Two scalars of a sensor model cannot express impulse, periodic, or spatially correlated corruption, which is where the largest part of our gain arises.

Routing to trained experts. All-in-one restoration methods adapt a shared model to unknown degradations using learned degradation representations or prompts (Li et al., 2022; Potlapalli et al., 2023). Closest to our setting, Ashrafee & Krawczyk (2026) freeze a pretrained backbone, train one adapter per degradation, and route each input to the nearest degradation prototype, separating degradations as distinct as noise, rain, and haze and scoring the routing against the degradation label. These approaches organize routing around degradation identity. We instead label references by the candidate that denoises them best, and we score routing by the PSNR of its output, using the corruption label only to analyze the descriptor space. The corruptions we distinguish are all noise, and two different corruptions can call for the same denoiser.

B CORRUPTION FAMILIES

Standard treatments of additive, impulse, and periodic noise are given by Gonzalez & Woods (2018); signal-dependent noise in camera sensors is modeled by Foi et al. (2008), and speckle by Goodman (2007). Intensities are on $[0, 255]$ unless stated otherwise. We write x for the clean image, y for the corrupted image, and n for a noise field.

Table 1: Corruption families and their parameters.

Property varied	Family	Parameters
Marginal law	Gaussian, Laplacian, uniform	σ
	Impulse	fraction of pixels p
Signal dependence	Poisson	peak count
	Speckle	σ
Spatial structure	Correlated	σ, correlation length
	Anisotropic	σ, correlation length in x and in y
	Banding	amplitude, period, orientation
	Annular	σ, center frequency f_0, bandwidth

Marginal law. Gaussian, Laplacian, and uniform noise add independent samples of standard deviation σ to every pixel, $y = x + n$. Gaussian noise is the standard model of thermal and readout noise. Laplacian noise has heavier tails, and uniform noise, which approximates quantization error, has lighter ones. Impulse noise replaces a fraction p of pixels, chosen independently, with extreme values,across the three RGB channels, as dead sensor elements or transmission errors do.

Signal dependence. Poisson noise models photon counting. Intensity is mapped to a photon count by the fixed scale $\lambda = x\,\text{peak}/255$, so intensity 255 corresponds to the peak count in every photograph, each pixel is drawn from a Poisson distribution with mean λ, and the count is mapped back by the same scale; the noise variance is therefore proportional to intensity, and a lower peak gives noisier images. Speckle models coherent imaging, as in ultrasound and radar, and multiplies the image by a random factor, $y = x(1 + n)$ with n of standard deviation σ, so the noise standard deviation grows in proportion to intensity.

Spatial structure. Correlated noise is white Gaussian noise smoothed by a Gaussian kernel of standard deviation $\ell/2$ pixels, where ℓ is the correlation length, and then rescaled to standard deviation σ. It models the effect of interpolation and filtering in a camera's processing. Anisotropic noise uses the same filter with standard deviations $\ell_x/2$ and $\ell_y/2$ along the horizontal and vertical axes. Banding adds a sinusoidal pattern of horizontal or vertical stripes with a given amplitude and period, as produced by row- or column-dependent readout offsets and electrical interference. The phase of the pattern is drawn uniformly at random for each realization. Annular noise multiplies the spectrum of white noise by a Gaussian ring, $\exp\big(-(\|f\| - f_0)^2/(2\,\mathrm{bw}^2)\big)$, and rescales the result to standard deviation σ. Here f is the two-dimensional spatial frequency in cycles per pixel, with each component in $[-1/2, 1/2)$, and f_0 is the center frequency of the ring; the calibration value $f_0 = 0.25$ is half the Nyquist frequency along each axis, a period of four pixels. The bandwidth bw is the standard deviation of the ring in amplitude, so the ring in power is narrower by $\sqrt{2}$. The ring depends on $\|f\|$ alone, which gives structure at a characteristic scale with no preferred direction.

Mixtures. A mixture independently applies two conditions to the same clean image and subsequently selects between the two results on a pixel-by-pixel basis, using a random binary mask generated independently for each pixel and shared across the three color channels. Consequently, each pixel is subject to only one of the two types of noise and never both. The fourteen mixtures consist of seven pairs of conditions, each applied using two mixing fractions: 0.3 (where 30% of the pixels derive from the first condition of the pair and 70% from the second) and 0.6 (where the split is 60% and 40%).

Table 2: The thirteen statistics of D13, grouped by the property each one measures. The first nine are computed on the four-phase residual of the color channels, the last four on an estimate of the power spectrum of the luminance. The names are those used in Appendix C.1.

Property	Statistic	Name
Amplitude	variance	`marg_var`
Marginal law	skewness	`marg_skew`
	excess kurtosis	`marg_kurt`
Signal dependence	slope of the log variance across intensity bins	`sig_slope`
Spatial correlation	correlation at a lag of one pixel along rows	`spat_rlag01`
Radial spectrum	energy in the four innermost of eight radial bands	`freq_band0-3`
Spectral structure	prominence of the largest peak	`peak_prom`
	energy in the strongest half percent of bins	`line_frac`
	angular concentration at the peak radius	`ang_conc`
	anisotropy	`anisotropy`

C STATISTICS

Properties measured. The four properties of Table 2 are those along which the corruption families of Appendix B vary, and each statistic is designed to respond to one of them. The variance of the residual measures amplitude. The skewness and the excess kurtosis measure the shape of the marginal law, separating Laplacian and impulse noise, which have heavy tails, from uniform noise, which has light ones. The slope of the log variance across intensity bins is zero for additive noise and grows with the rate at which the noise variance follows the signal, as it does for Poisson and speckle noise. The nine remaining statistics describe spatial structure: the lag-one correlation and the four radial bands respond to correlated and anisotropic noise, whose energy is concentrated at low frequencies, and the four structural statistics respond to banding and annular noise, which place their energy on a few frequencies or on a ring. Of the four, `peak_prom` and `line_frac` record how concentrated the spectrum is and `ang_conc` and `anisotropy` how directional it is, none of them recording the frequency at which the energy lies.

Scaling. Two of the thirteen statistics are transformed before standardization. The variance v of the residual is replaced by $\log v$, and the excess kurtosis κ by $\operatorname{asinh}\kappa = \ln\bigl(\kappa + \sqrt{\kappa^2+1}\bigr)$. Both range over several orders of magnitude across the corruption families, and in raw form a few conditions would dominate the Euclidean distance between descriptors. A logarithm cannot be used on the kurtosis, which is negative for light-tailed noise such as uniform noise; the inverse hyperbolic sine is nearly linear near zero and logarithmic for large values of either sign. Neither transformation has a fitted parameter. The remaining eleven statistics enter the descriptor as defined.

C.1 D13 IMPLEMENTATION

This section specifies each statistic in enough detail for D13 to be reimplemented. Throughout, y denotes the corrupted image, with channels y_c, $c = 1, \dots, C$ ($C = 3$ for sRGB), on the intensity scale $[0, 255]$. The first nine statistics are computed from a residual formed by differencing the four polyphase components of each channel, and the last four from an estimate of the noise power spectrum of the luminance. Each division, logarithm, and radius normalization is protected by a small positive constant. These constants do not change the statistics of a corrupted image, and we omit them below except where a reported value depends on one.

The residual. Each channel is cropped to even side lengths, $H' = 2\lfloor H/2 \rfloor$ and $W' = 2\lfloor W/2 \rfloor$, and split into its four polyphase components,

$$p^c_{ab}(i,j) = y_c(2i+a,\, 2j+b), \qquad a,b \in \{0,1\}, \quad 0 \le i < H'/2,\ 0 \le j < W'/2.$$

This is the pixel-shuffle downsampling that Zhou et al. (2020) and Lee et al. (2022) use to break the spatial correlation of real sensor noise before it enters a network; here it is used to form differences. Of the six pairs of components, we difference the four that are adjacent, two along rows and two

along columns,

$$\mathcal{P} = \big\{(p_{00}, p_{01}),\ (p_{10}, p_{11}),\ (p_{00}, p_{10}),\ (p_{01}, p_{11})\big\},$$

and for each channel and each pair $(p, p') \in \mathcal{P}$ we form a residual plane and a local-mean plane,

$$r_k = \frac{p - p'}{\sqrt{2}}, \qquad m_k = \frac{p + p'}{2},$$

where k indexes the $4C$ combinations of channel and pair.

The factor $1/\sqrt{2}$ gives the residual the variance of the noise. If $y = x + \eta$ with η white of variance σ^2, the noise component of r_k has variance σ^2, and so does r_k wherever x is locally constant. No other rescaling is applied, and intensities remain on $[0, 255]$. A difference along rows has zero response at zero horizontal frequency, and a difference along columns at zero vertical frequency, so with both orientations the residual responds to noise at every frequency except the origin. Using both orientations also treats horizontal and vertical image edges alike. Cropping is the only treatment of the borders: each component has exactly $H'/2 \times W'/2$ samples, and nothing is padded. Each pixel enters one difference along rows and one along columns, so for white noise the samples of a residual plane are independent. Statistics 1 to 9 treat the channels identically and do not combine them. The $4C$ planes are concatenated for the marginal and signal-dependence statistics (1–4), and the statistic is averaged over planes for the correlation and the radial bands (5–9). Only the four structural statistics (10–13) combine the channels, through the unweighted luminance $\ell = C^{-1} \sum_c y_c$.

Amplitude and marginal law (1–3). Let r denote the $4C$ residual planes concatenated into a single vector of length N, with mean $\bar{r}$ and variance $s^2 = N^{-1} \sum_n (r_n - \bar{r})^2$. The first three statistics are its variance, skewness, and excess kurtosis,

$$\texttt{marg_var} = s^2, \texttt{marg_skew} = \frac{1}{N} \sum_n \Big(\frac{r_n - \bar{r}}{s}\Big)^3, \qquad \texttt{marg_kurt} = \frac{1}{N} \sum_n \Big(\frac{r_n - \bar{r}}{s}\Big)^4 - 3.$$

The excess kurtosis is zero when the noise is Gaussian, positive for heavier-tailed noise such as Laplacian and impulse noise, and negative for uniform noise.

Signal dependence (4). The local-mean planes are concatenated in the same order as the residual, and their range $[\min m, \max m]$ is divided into $B = 8$ bins of equal width. Let $\mathcal{B}_b$ be the set of residual samples whose local mean falls in bin b. The variance in each bin is

$$v_b = \begin{cases} \widehat{\mathrm{Var}}\big(r_n : n \in \mathcal{B}_b\big) & \text{if } |\mathcal{B}_b| \geq 8, \\ s^2 & \text{otherwise,} \end{cases}$$

and the statistic is the least-squares slope of $\log v_b$ against the bin index,

$$\texttt{sig_slope} = \frac{\sum_b (b - \bar{b})\,(\log v_b - \overline{\log v})}{\sum_b (b - \bar{b})^2},$$

where $\bar{b}$ and $\overline{\log v}$ are means over the B bins. Each residual sample is conditioned on the mean of the two pixels from which it was computed, so no window larger than the pair is involved. Because the bins span the intensity range of the image, the slope per unit intensity equals the statistic divided by the bin width, which differs between images.

The statistic is zero for additive noise and increases with the rate at which the noise variance grows with intensity. If the conditional variance follows a power law, $\mathrm{Var}(r \mid x) \propto x^\gamma$, then $\log v_b \approx \gamma \log x_b + \text{const}$, where x_b is the center of bin b, and the fitted slope is γ times a positive factor determined by the bin centers. For a given image the statistic is therefore proportional to γ, and a linear fit suffices without estimating the noise-level function. Poisson noise has $\gamma = 1$ and speckle $\gamma = 2$.

Spatial correlation (5). For each residual plane, with $\tilde{r}_k = r_k - \bar{r}_k$, the correlation at a lag of one sample along rows is

$$\rho_k = \frac{\big\langle \tilde{r}_k(i,j)\, \tilde{r}_k(i,j+1) \big\rangle}{\big\langle \tilde{r}_k(i,j)^2 \big\rangle}, \qquad \texttt{spat_rlag01} = \frac{1}{4C} \sum_k \rho_k,$$

where $\langle\cdot\rangle$ averages over the positions at which both terms are defined. One sample of the residual lattice is two pixels of the image, and the same lag is taken on all four planes, so the coordinate is a fixed linear functional of the noise autocovariance along rows at that separation. It is zero for white noise, where every lag involved vanishes, and departs from zero when the noise carries horizontal dependence at that scale.

Radial bands (6–9). For each residual plane, the periodogram of the mean-removed plane is $P_k = \big|\mathcal{F}\{\tilde{r}_k\}\big|^2$, with zero frequency shifted to the center of the array. Write $\rho(u, v)$ for the distance of a bin from the center and $\rho_{\max}$ for the largest such distance on the grid, which is attained at a corner. The grid is divided into eight annuli of equal width in radius,

$$\beta(u, v) = \min\Big(\big\lfloor 8\,\rho(u, v)/\rho_{\max}\big\rfloor,\ 7\Big),$$

the floor function being clipped so that the corner bins themselves fall in band 7. The fraction of the energy of plane k in band j is

$$e_j^{(k)} = \frac{\sum_{\beta(u,v)=j} P_k(u, v)}{\sum_{u,v} P_k(u, v)}, \qquad j = 0, \ldots, 7,$$

and the four statistics are the averages of the four innermost bands over planes, $\texttt{freq_band}_j = (4C)^{-1} \sum_k e_j^{(k)}$ for $j = 0, 1, 2, 3$. The band edges follow from the grid alone and have no free parameter.

Why four bands. The normalizing radius is the corner of the sampling square, $\sqrt{2}$ times the Nyquist radius, so band j spans $[\,0.088\,j,\ 0.088\,(j + 1))$ cycles per sample of the residual lattice, against a Nyquist frequency of 0.5. Bands 0 to 4 lie entirely inside the Nyquist disc, band 5 is cut by it, and bands 6 and 7 consist only of the corners of the square, so their content reflects the sampling geometry rather than the noise. That leaves five complete annuli, of which we keep the first four. Nothing is lost about how much energy sits above the fourth edge: since the eight fractions sum to one, the pooled share of bands 4 to 7 is one minus the sum of the four we keep, an affine function of them. The statistics of the following paragraph already describe narrow-band content at any frequency up to Nyquist, so the radial bands need only carry the coarse low-frequency profile, which is where the spatially structured families place their energy. Keeping four also matches the four structural statistics: with all eight bands, the radial block would carry close to half of the squared distance between standardized descriptors, the imbalance that the logarithm and the inverse hyperbolic sine of Section 3.2 are there to prevent.

The noise spectrum. The last four statistics are computed from an estimate of the noise power spectrum of the luminance (not to be understood as a photometric luminance, as it is unweighted) $\ell = C^{-1} \sum_c y_c$, at full resolution rather than on the residual lattice. Three first differences are formed,

$$d_1 = \frac{\ell(i, j{+}1) - \ell(i, j)}{\sqrt{2}}, \qquad d_2 = \frac{\ell(i{+}1, j) - \ell(i, j)}{\sqrt{2}}, \qquad d_3 = \frac{\ell(i{+}1, j{+}1) - \ell(i, j)}{\sqrt{2}},$$

each of which passes the noise spectrum S through a known filter, with squared transfer function

$$|H_1(f)|^2 = 2\sin^2(\pi f_x), \qquad |H_2(f)|^2 = 2\sin^2(\pi f_y), \qquad |H_3(f)|^2 = 2\sin^2\big(\pi(f_x + f_y)\big).$$

Each of the three difference planes d_m, $m = 1, 2, 3$, is winsorized on its own scale, with $\hat{\sigma}_m = 1.4826\ \mathrm{med}\big|d_m - \mathrm{med}\, d_m\big|$ and $d_m \leftarrow \mathrm{clip}(d_m, \pm 4\hat{\sigma}_m)$, the median and the deviation being taken over all pixels of that plane. Its periodogram P_d is a Welch average over $L \times L$ patches on a stride of $L/2$, each patch having its own mean removed, multiplied by a separable Hann window w, and normalized by $L^2\langle w^2\rangle$. We use $L = 128$. If either side of the plane is shorter than 128, L becomes the largest even number not exceeding the shorter side, with a floor of 16, and the stride becomes half of that, with a floor of 8. Patches that would extend past the last row or column are dropped. If no patch fits, which happens when the shorter side is below 16, the four structural statistics are set to zero.

Since $\mathbb{E}P_d = |H_d|^2 S$, the three periodograms are combined by least squares, $\hat{S} = \arg\min_S \sum_d \big(P_d - |H_d|^2 S\big)^2$, with a ridge term in the denominator,

$$\hat{S}(f) = \frac{\sum_{d=1}^{3} |H_d(f)|^2 \, P_d(f)}{\sum_{d=1}^{3} |H_d(f)|^4 + 10^{-6}}, \qquad \hat{S}(0) = 0.$$

Each direction therefore contributes with weight $|H_d|^2$, which vanishes on the line of frequencies its own difference does not transmit: $f_x = 0$ for d_1, $f_y = 0$ for d_2, and $f_x + f_y = 0$ for d_3. No single difference determines S on its own blind line, and any two of the three have blind lines that meet only at the origin, where $\hat{S}$ is set to zero.

Structural statistics (10–13). All four are computed on the annulus $\mathcal{M} = \{f : 0.02 < \|f\| \le 0.5\}$, which excludes the lowest frequencies, where image content dominates, and the corners of the square, which lie beyond Nyquist. Write $S = \hat{S}|_{\mathcal{M}}$, $|\mathcal{M}|$ for the number of bins in the annulus, and $S_{(1)} \le \cdots \le S_{(|\mathcal{M}|)}$ for the sorted values of S. The first two statistics measure how concentrated the spectrum is,

$$\texttt{peak_prom} = \log \frac{\max S}{\operatorname{med} S}, \qquad \texttt{line_frac} = \log \frac{\sum_{i > |\mathcal{M}| - \kappa} S_{(i)}}{\sum_i S_{(i)}}, \kappa = \max\big(\lfloor 0.005\,|\mathcal{M}| \rfloor, 1\big),$$

the height of the largest bin above the bulk of the spectrum, measured against the median, and the share of the energy carried by the strongest half percent of the bins. Neither involves detecting a peak or a line, and neither refers to a frequency, so a periodic corruption keeps its value when its frequency changes.

The remaining two measure how the energy is distributed in direction. Let $f^\star = \arg\max_{\mathcal{M}} S$ and let $\mathcal{R} = \big\{ f \in \mathcal{M} : 0.85\,\|f^\star\| < \|f\| < 1.15\,\|f^\star\| \big\}$ be the annulus at the radius of the maximum. Then

$$\texttt{ang_conc} = \log \frac{\max_{\mathcal{R}} S}{\operatorname{mean}_{\mathcal{R}} S}, \qquad \texttt{anisotropy} = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} = \frac{\sqrt{(M_{xx} - M_{yy})^2 + 4M_{xy}^2}}{M_{xx} + M_{yy}},$$

where `ang_conc` is set to zero if $|\mathcal{R}| \le 8$, and $\lambda_1 \ge \lambda_2$ are the eigenvalues of the second-moment matrix of the spectrum, $M_{pq} = \sum_{\mathcal{M}} S\, f_p f_q \big/ \sum_{\mathcal{M}} S$ for $p, q \in \{x, y\}$. No centring appears because the spectrum of a real field satisfies $S(-f) = S(f)$, so its first moment vanishes and the second moment about the origin is the covariance. A ring and a pair of isolated peaks have the same radial profile, and `ang_conc` separates them; `anisotropy` reports the strength of a preferred direction without reporting which direction it is. That is a deliberate omission and it has a cost: horizontal and vertical banding differ in no coordinate by as much as its noise floor (Figure 6), which is why routing sends vertical banding to a specialist trained on horizontal stripes (Section 4.4). Both depend on where in the spectrum the structure lies, `ang_conc` through the number of bins in $\mathcal{R}$ and `anisotropy` through the weighting by $f_p f_q$, so both read a low-frequency line as weaker than a high-frequency one of the same contrast.

All four are ratios and are therefore unchanged when the noise is scaled: multiplying the image noise by a constant multiplies $\hat{S}$ by its square and leaves every one of them at its previous value. None of the four is zero on white noise, however: the residual fluctuation of a Welch average sets a floor, on a 256×256 plane 1.33, -4.25, 1.24 and 0.02 respectively, against 9.70, -0.16, 4.80 and 0.49 once a sinusoid is added. What the statistics report is the excursion above that floor.

Robustness choices. Several of the steps above guard against outliers and against leakage between frequencies. Winsorization at four median absolute deviations, applied before every periodogram, prevents impulsive pixels from spreading energy across the whole spectrum; removing the mean of each patch before the window stops a difference plane with a nonzero mean from leaking its offset in the same way; and the median, rather than the mean, serves as the reference level in `peak_prom` so that the peak itself does not raise the level it is measured against. The Welch average reduces the variance of the periodogram at the cost of resolution, and the ridge of 10^{-6} bounds the estimate near the blind lines. The three concentration statistics are reported as logarithms because their raw ratios span several orders of magnitude. A bin of the intensity sweep with fewer than eight samples is replaced by the global variance, and any coordinate that is not finite is replaced by zero.

Cost. The descriptor is computed on the CPU, with no GPU and no trained component. On a 256×256 RGB image it takes 23.7 ms on one core of an Apple M2 Max, the median over 50 photographs of the benchmark.

C.2 Sensitivity to the corruption parameters

To check that each statistic responds to the property it is meant to measure, we varied one parameter of one family at a time, holding the others at the family's reference values, for 17 parameters of the ten families. Each parameter took 13 values over the range given in Appendix B, and each value was applied to 300 photographs, 30 from each content domain, with independent noise realizations. For each statistic and parameter we measured the displacement of the statistic across the sweep, the range between its largest and smallest value over the 13 parameter values, each averaged over the 300 photographs. We divided this by the standard deviation of the statistic over the 300 photographs and five independent noise realizations at the family's reference values, which is its variation at a fixed corruption (Figure 6). The ratio exceeded 1 for 16 of the 17 parameters.

The variance of the residual responded to every amplitude parameter except those of correlated and banding noise, and the structural statistics to the parameters of the spatially structured families. The response was weakest for the orientation of banding and the bandwidth of annular noise, and it was low for the amplitude and the correlation length of correlated noise, whose energy lies mostly at the low frequencies that the residual attenuates. The structural statistics measure how concentrated and how directional the spectrum is, not in which direction, so they do not distinguish horizontal from vertical stripes. The classical pool is unaffected, since its notch filter removes stripes of either orientation, but the specialist pool is not (Section 4.4).

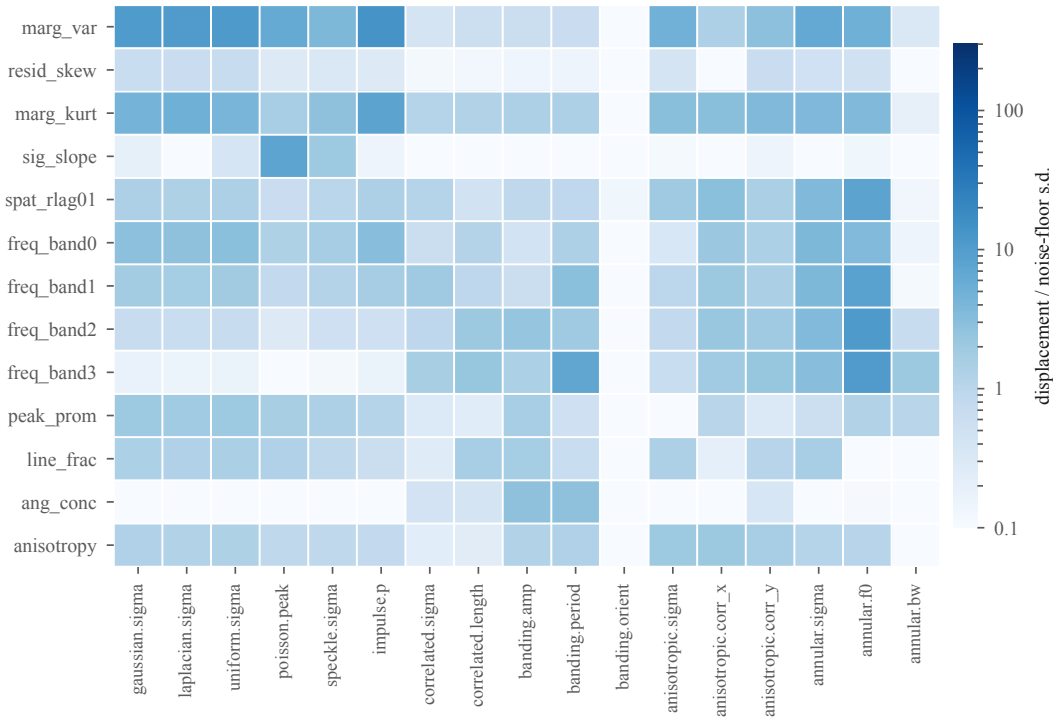

Figure 6: Sensitivity of the D13 statistics to the corruption parameters. Each column varies one parameter of one corruption family while the others are held fixed, and each row is one statistic. Each cell reports how far the statistic moves across the sweep, divided by its standard deviation across photographs and noise realizations. Values above one indicate that the statistic moves more with the parameter than with the photograph and the noise realization; darker cells indicate larger values. The statistics are nearly insensitive to the orientation of banding (see Appendix C).

Table 3: Candidates in each pool.

Pool	Candidate	Target family	Source	Settings
Classical	gauss_s0.6	broadband additive	SciPy	Gaussian blur, $\sigma = 0.6$ px
	gauss_s1.2	broadband additive	SciPy	Gaussian blur, $\sigma = 1.2$ px
	gauss_s2.4	broadband additive	SciPy	Gaussian blur, $\sigma = 2.4$ px
	median_3	impulse	SciPy	median filter, 3×3
	median_5	impulse	SciPy	median filter, 5×5
	tv_0.05	none	scikit-image Chambolle (2004)	TV-Chambolle, weight 0.05
	tv_0.15	none	scikit-image Chambolle (2004)	TV-Chambolle, weight 0.15
	bilateral	none	scikit-image, Tomasi & Manduchi (1998)	$\sigma_{\mathrm{col}} = 0.10$, $\sigma_{\mathrm{sp}} = 3$
	wiener_5	none	SciPy	local Wiener, 5×5
	notch	periodic	ours	FFT bins 3.5 sd above a 7×7 median background, $f > 0.06$
Specialist	M_gaussian_15	Gaussian	ours	$\sigma = 15$
	M_gaussian_25	Gaussian	ours	$\sigma = 25$
	M_gaussian_50	Gaussian	ours	$\sigma = 50$
	M_laplacian_25	Laplacian	ours	$\sigma = 25$
	M_uniform_25	uniform	ours	$\sigma = 25$
	M_poisson_30	Poisson	ours	peak $= 30$
	M_poisson_10	Poisson	ours	peak $= 10$
	M_speckle_10	speckle	ours	$\sigma = 0.10$
	M_speckle_20	speckle	ours	$\sigma = 0.20$
	M_sp_5pct	impulse	ours	$p = 0.05$
	M_corr_gauss	correlated	ours	$\sigma = 25$, length $= 3$
	M_banding	banding	ours	amp $= 20$, period $= 10$
	M_aniso_h	anisotropic	ours	$\sigma = 25$, $(8, 1)$
	M_annular_mid	annular	ours	$\sigma = 25$, $f_0 = 0.25$
General-purpose	S_DRUNet_s15	Gaussian	Zhang et al. (2021)	noise-level input 15
	S_DRUNet_s25	Gaussian	Zhang et al. (2021)	noise-level input 25
	S_DRUNet_s50	Gaussian	Zhang et al. (2021)	noise-level input 50
	S_SwinIR	Gaussian	Liang et al. (2021)	colour, $\sigma = 15$
	S_Restormer_gau	Gaussian	Zamir et al. (2022)	blind over σ, colour
	S_Restormer_real	real camera noise (SIDD)	Zamir et al. (2022)	real-noise checkpoint
	S_SCUNet_real	real-world degradations, synthesised	Zhang et al. (2023)	real-noise checkpoint
	S_PromptIR	blind, all-in-one	Potlapalli et al. (2023)	authors' checkpoint: denoising, deraining, dehazing
Hybrid	S_SwinIR_s25	Gaussian	Liang et al. (2021)	colour, $\sigma = 25$
	S_SwinIR_s50	Gaussian	Liang et al. (2021)	colour, $\sigma = 50$
	S_NAFNet_SIDD_w32	real camera noise	Chen et al. (2022)	width 32, SIDD
	adaptive_median_w7	impulse	Gonzalez & Woods (2018)	max window 7

Implementation details. Classical operators use scikit-image 0.26.0 (van der Walt et al., 2014) and SciPy 1.17.1 (Virtanen et al., 2020); learned models use PyTorch 2.12.1 (Paszke et al., 2019). Every candidate is applied per RGB channel. The eight general-purpose networks are loaded through DeepInverse (Tachella et al., 2025) (deepinv 0.4.1), using its default released checkpoints: drunet_deepinv_color_finetune_22k, 005_colorDN_DFWB_s128w8_SwinIR-M_noise{15,25,50}, gaussian_color_denoising_blind, real_denoising, scunet_color_real_psnr, and promptir.ckpt.
Hybrid pool. It reuses unchanged the eight general-purpose networks, with S_SwinIR renamed S_SwinIR_s15, plus median_3, median_5, tv_0.05, tv_0.15, and notch; the four additional candidates are listed above.

D POOLS OF DENOISERS

Table 3 lists every candidate of the four pools of Section 3.3, with the corruption family each is meant for, the implementation or the publication it comes from, and the settings at which it is run. The notes below the table give the library versions, the checkpoints used for the pretrained networks, and the composition of the hybrid pool, whose candidates are shared with the other pools apart from the four listed separately. A candidate is run at these settings on every image, since no candidate is retrained or tuned during routing.

Specialist networks. The fourteen specialists share one architecture and one training procedure, and differ only in the corruption applied during training. Each is a U-Net (Ronneberger et al., 2015) with three resolution levels and 32 channels at the first level, doubled at each downsampling, for 468k parameters in total. It differs from the original in four ways: three levels instead of five; padded 3×3 convolutions, so the output has the size of the input and the skip connections need no cropping; batch normalization after every convolution; and a 1×1 convolution to three channels followed by a sigmoid in place of the two-class output. The network predicts the clean image directly, on the unit intensity scale. Each specialist was trained on 128×128 patches cropped from the 800 training photographs of DIV2K (Agustsson & Timofte, 2017), corrupted with a new realization of its calibration condition at every step by the benchmark's own generator from a disjoint seed stream, with random flips and multiples of ninety degrees, the ℓ_2 loss and Adam (Kingma & Ba, 2014) at learning rate 2×10^{-4} with cosine decay for 30,000 iterations and batch size 32, in mixed precision. Validation PSNR was measured every 2,000 iterations on fixed 256×256 crops of the 100 DIV2K validation photographs with a fixed noise realization, and the best checkpoint was kept. DIV2K shares no photograph and no content domain with the benchmark. Training one specialist took about twelve minutes on one NVIDIA A100 80GB, and under three hours for all fourteen.

E BENCHMARKS

We describe the discrete and parametric evaluation suites, including their calibration conditions, held-out corruptions, content splits, and evaluation protocols.

The discrete suite tests whether routing transfers to unseen corruption conditions and compositions, including mixtures. The parametric suite isolates a different question: within each primitive corruption family, how does routing behave as one physical parameter moves within or beyond the calibrated range?

Photographs. The ten content domains are the ImageNette classes: tench, English springer, cassette player, chain saw, church, French horn, garbage truck, gas pump, golf ball, and parachute (Howard, 2019). Each photograph was resized to a short side of 320 pixels and center-cropped to 256×256 RGB pixels. Each benchmark used thirty photographs per domain, and the photographs of the parametric benchmark were not used in the discrete benchmark.

Discrete benchmark. The discrete benchmark has 40 conditions, of which 14 form the calibration set (Table 4). The calibration set contains at least one condition from each of the ten families of Appendix B and no mixture. The other 26 conditions are withheld: twelve single conditions that change the severity or the orientation of a calibrated family, and the fourteen mixtures. None of the withheld conditions introduces a family absent from calibration. Each network of the specialist pool was trained on one calibration condition (Appendix D).

Parametric benchmark. For each family, the six severity levels were drawn from equal strata of the parameter's range. The thirty photographs of each domain were split by identity into fifteen used for references and fifteen used for queries. Each split was calibrated separately, and results are reported for each split. Table 5 lists the parameter ranges and levels.

Confidence intervals. Each photograph was corrupted once under each condition, with an independent noise realization, so a condition is evaluated over 300 realizations; no condition is repeated under a second seed. Each bootstrap replicate draws ten content domains with replacement and, within each drawn domain, draws its photographs with replacement. Conditions are not resampled, since they define the benchmark rather than sample from a population. All methods are evaluated on the same resampled images, so their differences remain paired in every resample. Intervals are percentile intervals over $5{,}000$ replicates.

F NO-REFERENCE SELECTORS

Each selector ran every candidate of the pool on a test image, scored each output without the clean image, and kept the output with the best score. The six perceptual scores were computed with

Table 4: The 14 calibration conditions of the discrete benchmark. The correlation length ℓ is given in pixels, and ℓ_x and ℓ_y are the correlation lengths along the horizontal and vertical axes (Appendix B).

#	Condition	Family	Parameters
1	`gaussian_15`	Gaussian	$\sigma = 15$
2	`gaussian_25`	Gaussian	$\sigma = 25$
3	`gaussian_50`	Gaussian	$\sigma = 50$
4	`laplacian_25`	Laplacian	$\sigma = 25$
5	`uniform_25`	Uniform	$\sigma = 25$
6	`sp_5pct`	Impulse	$p = 0.05$
7	`poisson_30`	Poisson	peak $= 30$
8	`poisson_10`	Poisson	peak $= 10$
9	`speckle_10`	Speckle	$\sigma = 0.10$
10	`speckle_20`	Speckle	$\sigma = 0.20$
11	`corr_gauss`	Correlated	$\sigma = 25$, $\ell = 3$
12	`aniso_h`	Anisotropic	$\sigma = 25$, $\ell_x = 8$, $\ell_y = 1$
13	`banding`	Banding	amplitude 20, period 10, horizontal
14	`annular_mid`	Annular	$\sigma = 25$, $f_0 = 0.25$, bw $= 0.04$

Table 5: Parameter levels used in the parametric benchmark.

Family	Parameter	Range	1	2	3	4	5	6
Gaussian	σ	10–50	10.0	22.7	29.9	35.2	36.7	45.5
Laplacian	σ	10–50	15.5	22.0	25.2	31.6	41.4	48.1
Uniform	σ	10–50	13.2	21.5	23.6	32.8	41.7	46.2
Poisson	peak count	8–60	58.3	49.0	42.0	33.2	23.4	13.4
Speckle	σ	0.04–0.25	0.060	0.079	0.112	0.157	0.189	0.220
Impulse	p	0.01–0.12	0.018	0.030	0.055	0.071	0.091	0.119
Correlated	correlation length	1.5–8	2.19	3.00	4.06	5.13	6.83	7.51
Banding	period	5–30	7.89	9.23	15.8	20.5	25.3	29.2
Anisotropic	horizontal correlation length	1–16	2.41	3.57	6.97	10.1	11.9	14.9
Annular	center frequency f_0	0.08–0.42	0.407	0.362	0.260	0.226	0.156	0.092

Notes. Parameters held fixed: $\sigma = 25$ for correlated, anisotropic, and annular noise; vertical correlation length 1 for anisotropic noise; bandwidth 0.04 for annular noise; amplitude 20 and orientation 0 for banding. The six levels are sampled from equal strata of each stated range. The level index increases with severity rather than parameter value: for Poisson and annular noise, severity increases as the parameter decreases, so their values decrease from left to right.

IQA-PyTorch (Chen & Mo, 2022), version 0.1.16, each through `pyiqa.create_metric` with the authors' default checkpoints; the direction is read from the metric itself rather than assumed, so the lowest score is kept for NIQE and BRISQUE and the highest for TOPIQ, MUSIQ, CLIP-IQA+ and ARNIQA. All selectors were evaluated on the same test images as routing, and the PSNR of each selected output was taken from the same recorded outputs.

Monte-Carlo SURE. For an image y corrupted by Gaussian noise of variance σ^2, and a denoiser f, Stein's unbiased risk estimate of the mean squared error is

$$\mathrm{SURE}(f, y) = \frac{1}{N} \|y - f(y)\|^2 - \sigma^2 + \frac{2\sigma^2}{N} \operatorname{div} f(y), \tag{1}$$

where N is the number of pixels. Following Ramani et al. (2008), we estimated the divergence with one random probe b of independent standard normal entries,

$$\operatorname{div} f(y) \approx \frac{1}{\epsilon} b^\top \big(f(y + \epsilon b) - f(y)\big), \tag{2}$$

with $\epsilon = 1$ in intensity units on the $[0, 255]$ scale so each candidate requires two denoising calls. The probe seed is fixed and depends deterministically on the candidate, so the ranking is reproducible from a single probe. The divergence is estimated for the clipped map, which is the map whose risk is

Table 6: Routing with D13 on the withheld conditions of the discrete benchmark. Gain in PSNR (dB) over the best overall denoiser, with 95% paired-bootstrap intervals, and the oracle gap for the same pool. *Recovered* is the fraction of the oracle gap that routing recovers, and *optimal* the fraction of images on which routing selects the candidate with the highest PSNR.

Pool	K	D13 routing	Oracle	Recovered	Optimal
Classical	10	$+1.363$ $[+1.212, +1.492]$	$+2.050$	0.665	0.563
Specialist	14	$+1.577$ $[+1.479, +1.669]$	$+2.542$	0.621	0.456
General-purpose	8	$+0.142$ $[-0.009, +0.265]$	$+0.928$	0.153	0.485
Hybrid	17	$+0.986$ $[+0.710, +1.161]$	$+2.282$	0.432	0.404

being ranked. The variance σ^2 was estimated from the corrupted image, as the square of the median absolute deviation of the finest-scale diagonal Haar wavelet coefficients divided by 0.6745 (Donoho & Johnstone, 1994); one estimate was computed per test image and shared by all candidates. We checked the implementation in three steps. The Monte-Carlo divergence reproduced the analytic trace of a linear shift-invariant filter within 0.2%. On an unclipped Gaussian perturbation with known σ, where SURE is exactly unbiased, it reproduced the true mean squared error within 0.04%. Clipping the output to the intensity range, which every candidate does, biases the estimate downward by 11%, and on the trained candidates at `gaussian_35` it sits 17 to 21% below the true error while still ranking them in the order the true error does; the selector uses only that ranking.

Routing selected the oracle candidate, the one with the highest PSNR on that image, in 56, 46, 49, and 40% of images for the classical, specialist, general-purpose, and hybrid pools (Table 6), and the selectors in 2 to 39% (Table 7).

Table 7: Gain in PSNR (dB) over the best overall denoiser on the withheld conditions of the discrete benchmark. Calls denote number of denoising calls per image for a pool of K candidates.

Selector	Calls	Classical	Specialist	General-purpose	Hybrid
Routing (D13)	1	$+1.36$	$+1.58$	$+0.14$	$+0.99$
NIQE	K	$+0.86$	$+1.17$	-1.25	-0.92
BRISQUE	K	$+0.05$	$+0.90$	-1.43	-1.33
MUSIQ	K	-0.87	$+0.41$	-0.35	-0.05
CLIP-IQA+	K	$+0.02$	$+0.29$	-0.26	$+0.38$
TOPIQ	K	-0.45	$+0.16$	-0.33	$+0.20$
ARNIQA	K	-0.47	$+0.50$	-0.33	$+0.17$
MC-SURE	$2K$	-0.97	-1.23	-2.53	-3.54
Oracle	—	$+2.05$	$+2.54$	$+0.93$	$+2.28$

SURE by condition. Table 8 gives, for each withheld condition, the rank correlation between the SURE estimate and the true mean squared error across the candidates of the specialist pool, averaged over test images. The mean over the 26 conditions was 0.050; the correlation was positive for 13 of them and zero to two decimals for two more. It was close to one on Gaussian noise and close to minus one on impulse noise, and it was negative on the anisotropic and correlated conditions, whose noise is not independent across pixels, and on annular noise at low center frequency, though not at high center frequency. Nothing in the estimate itself indicates which of these cases applies to a given image.

G TRANSFER ACROSS NOISE SEVERITIES

Figure 7 gives the gains of Section 4.3 for each split. The loss of gain outside the calibrated range, measured as the gain on S1 minus the mean gain on S2 and S3, was 0.30 dB for the classical pool and 0.61 dB for the hybrid pool, against 0.19 dB for the general-purpose pool, with 95% intervals lying entirely above zero. In the classical and hybrid pools, strong noise dominates the statistics the

Table 8: Rank correlation between the SURE estimate and the true mean squared error for each of the 26 withheld conditions. For each image, the Spearman correlation is computed across the candidates of the specialist pool; each entry is its mean over the 300 images of the condition.

Condition	Family	Correlation
Gaussian, $\sigma = 35$	additive	$+0.89$
30% anisotropic + 70% Gaussian $\sigma = 25$	mixture	$+0.88$
60% Gaussian $\sigma = 25$ + 40% Poisson 30	mixture	$+0.87$
30% correlated + 70% Gaussian $\sigma = 25$	mixture	$+0.86$
Annular, $f_0 = 0.40$	annular	$+0.84$
30% Gaussian $\sigma = 25$ + 70% Poisson 30	mixture	$+0.77$
30% banding + 70% Poisson 30	mixture	$+0.62$
Speckle, $\sigma = 0.15$	signal-dep.	$+0.59$
Poisson, peak 20	signal-dep.	$+0.58$
60% banding + 40% Poisson 30	mixture	$+0.33$
Banding, vertical	banding	$+0.18$
Banding, period 14	banding	$+0.16$
60% anisotropic + 40% Gaussian $\sigma = 25$	mixture	$+0.12$
60% correlated + 40% Gaussian $\sigma = 25$	mixture	$+0.00$
30% correlated + 70% banding	mixture	-0.00
60% correlated + 40% banding	mixture	-0.12
Correlated, length 8	correlated	-0.31
60% Gaussian $\sigma = 25$ + 40% impulse 5%	mixture	-0.34
Correlated, length 5	correlated	-0.51
Anisotropic, horizontal, length 16	anisotropic	-0.56
Anisotropic, vertical	anisotropic	-0.61
Annular, $f_0 = 0.10$	annular	-0.71
60% annular + 40% impulse 5%	mixture	-0.72
30% Gaussian $\sigma = 25$ + 70% impulse 5%	mixture	-0.78
30% annular + 70% impulse 5%	mixture	-0.83
Impulse, $p = 0.08$	impulse	-0.93

descriptor measures and is estimated accurately, whereas mild noise changes them little, so severities below the calibrated range crowd together near the mildest calibrated reference.

The specialist pool is omitted because each of its networks was trained on one calibration condition of the discrete benchmark. The correspondence between candidates and calibration conditions, which this pool was built to test, does not hold for the calibration sets of the parametric benchmark unless the networks are retrained for each split.

H Gains by corruption family

Figure 8 gives the gains of Table 9 with their 95% intervals, computed by the paired bootstrap of Section 3.6. Tables 10 and 11 give the gains and oracle gaps by family on the parametric benchmark, at severities above (S2) and below (S3) the calibrated range.

I Examples of routing

Figure 9 shows images from the withheld conditions of the discrete benchmark, each with a different corruption, denoised by the best overall denoiser of the classical pool and by the candidate that routing selected. The examples illustrate the errors that routing avoids; Section 4.2 and Appendix H give the average gains, including the families on which routing lost PSNR.

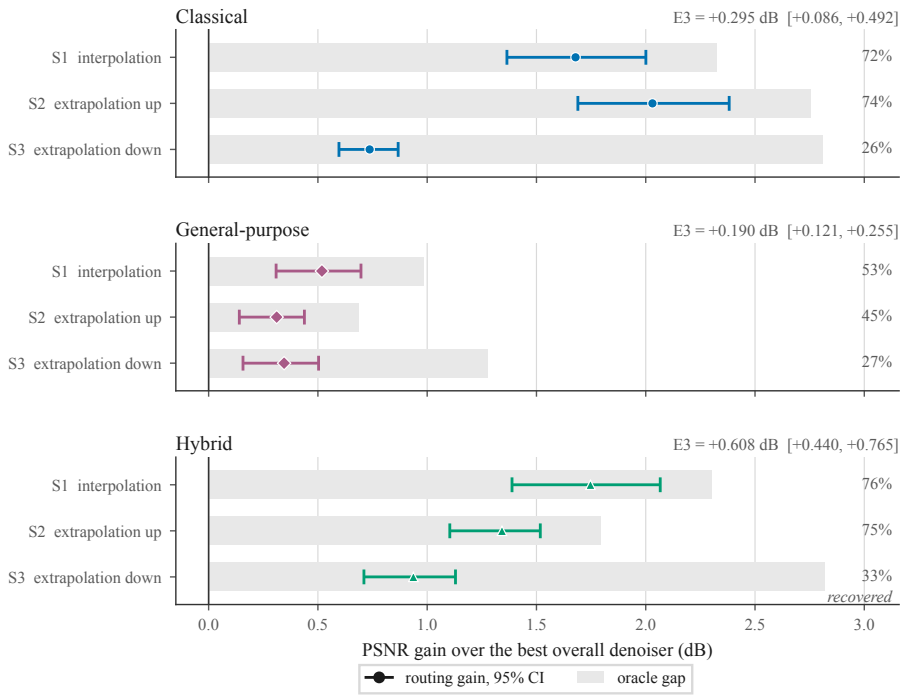

Figure 7: **Transfer across noise severities.** Gain in PSNR over the best overall denoiser on the parametric benchmark, at severities between the calibrated ones (S1), above them (S2), and below them (S3). The percentages are the routing gain as a share of the oracle gap (the gray bar). Markers and whiskers show routing gain and its 95% interval, gray bars show the oracle gap, and percentages give the fraction of it recovered. The specialist pool is not evaluated on this benchmark.

Table 9: Gain of routing over the best overall denoiser, and oracle gap, by corruption family on the withheld conditions of the discrete benchmark, in dB. Each entry gives gain / oracle gap. The number after each family is its number of withheld conditions; families are weighted by it in the gains of Section 4.2. Figure 8 gives the same gains with 95% confidence intervals.

Family	Classical	Specialist	General-purpose	Hybrid
Impulse (1)	+6.83 / 6.88	+15.62 / 15.83	+3.91 / 3.96	+13.91 / 14.07
Banding (2)	+6.41 / 7.95	+0.20 / 2.99	+0.55 / 0.89	+6.53 / 8.19
Correlated (2)	+0.76 / 1.46	+1.94 / 3.81	−0.73 / 0.06	−0.72 / 0.08
Annular (2)	+0.71 / 1.19	+1.70 / 2.15	−1.11 / 0.04	−1.13 / 0.04
Signal-dependent (2)	+0.23 / 1.08	+1.41 / 2.11	−0.47 / 0.17	−0.52 / 0.18
Anisotropic (2)	+0.11 / 0.44	+0.49 / 1.81	+0.59 / 1.01	+0.58 / 1.05
Additive (1)	−0.76 / 0.20	+0.50 / 1.33	+0.18 / 0.41	+0.14 / 0.41
Mixture (14)	+0.92 / 1.57	+0.96 / 1.65	+0.14 / 1.10	+0.15 / 1.84

J CNN Comparison

We used the network of Liu et al. (2020) as published, with the SAME padding the paper declares. It has four blocks, each a 3×3 convolution, a LeakyReLU with slope 0.1, a second 3×3 convolution, a second LeakyReLU, a 2×2 max pooling, and batch normalization, at 32, 64, 128, and 256 channels. The $256 \times 4 \times 4$ output is flattened and passed through two fully connected layers of 1024 units to four logits. Training used Adam at a learning rate of 10^{-4}, batches of 128, random flips, and class-balanced epochs, leave-one-content-domain-out over the same ten folds as routing.

The classifier was trained on the eight calibration conditions that fall within its taxonomy, grouped into its four types: Gaussian at $\sigma = 15$, 25, and 50; Poisson at peaks 30 and 10; speckle at $\sigma = 10$ and 20; and salt-and-pepper at 5% density. Routing was restricted to the same four types. Both

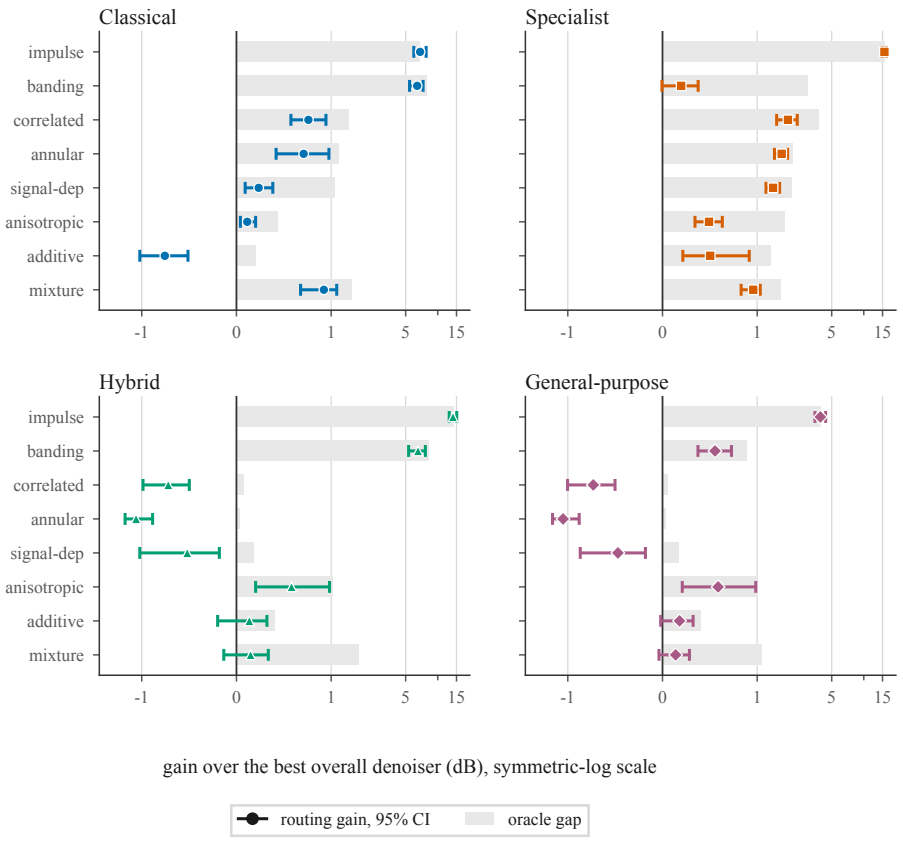

Figure 8: Gain of routing over the best overall denoiser, by corruption family and pool, on the withheld conditions of the discrete benchmark. Points and whiskers show the gain and its 95% interval; gray bars show the oracle gap. The horizontal axis is linear between -1 and 1 dB and logarithmic beyond, so that the gains on impulse noise and on the other families can be read on one axis.

Table 10: Gain of routing over the best overall denoiser and oracle gap (dB), by corruption family, at severities above the calibrated range (S2) of the parametric benchmark.

	Classical		General-purpose		Hybrid	
Family	gain	gap	gain	gap	gain	gap
Gaussian	2.39	2.59	0.14	0.31	0.10	0.31
Laplacian	2.18	3.15	0.37	0.53	0.30	0.53
Uniform	2.42	2.59	0.18	0.34	0.14	0.34
Poisson	0.86	2.47	-0.16	0.12	-0.17	0.13
Speckle	-0.07	1.05	-0.39	0.14	-0.40	0.14
Impulse	11.42	11.49	3.66	3.76	14.44	14.58
Correlated	0.38	1.20	-0.90	0.04	-0.91	0.05
Banding	-0.12	0.67	0.45	0.68	0.25	0.87
Anisotropic	0.27	0.63	0.67	0.85	0.57	0.87
Annular	0.57	1.71	-0.91	0.10	-0.89	0.12
All ten	2.03	2.75	0.31	0.69	1.34	1.79

methods dispatch through the same table of four entries, one denoiser per type, and both run the single candidate the table returns, so they differ only in how the type of an image is chosen.

Table 11: Gain of routing over the best overall denoiser and oracle gap (dB), by corruption family, at severities below the calibrated range (S3) of the parametric benchmark.

	Classical		General-purpose		Hybrid	
Family	gain	gap	gain	gap	gain	gap
Gaussian	0.43	2.18	0.69	1.79	0.69	1.79
Laplacian	-0.07	1.85	0.71	1.39	0.69	1.39
Uniform	0.16	2.28	0.75	1.59	0.75	1.59
Poisson	0.13	1.16	-0.01	0.41	-0.01	0.41
Speckle	2.12	4.35	-0.10	1.98	-0.10	1.98
Impulse	3.14	5.89	2.22	3.35	8.23	11.49
Correlated	0.45	1.09	-1.04	0.02	-1.04	0.03
Banding	1.10	7.52	0.36	0.96	0.32	8.12
Anisotropic	-0.04	0.41	0.67	1.26	0.66	1.35
Annular	-0.04	1.38	-0.81	0.05	-0.81	0.05
All ten	0.74	2.81	0.35	1.28	0.94	2.82

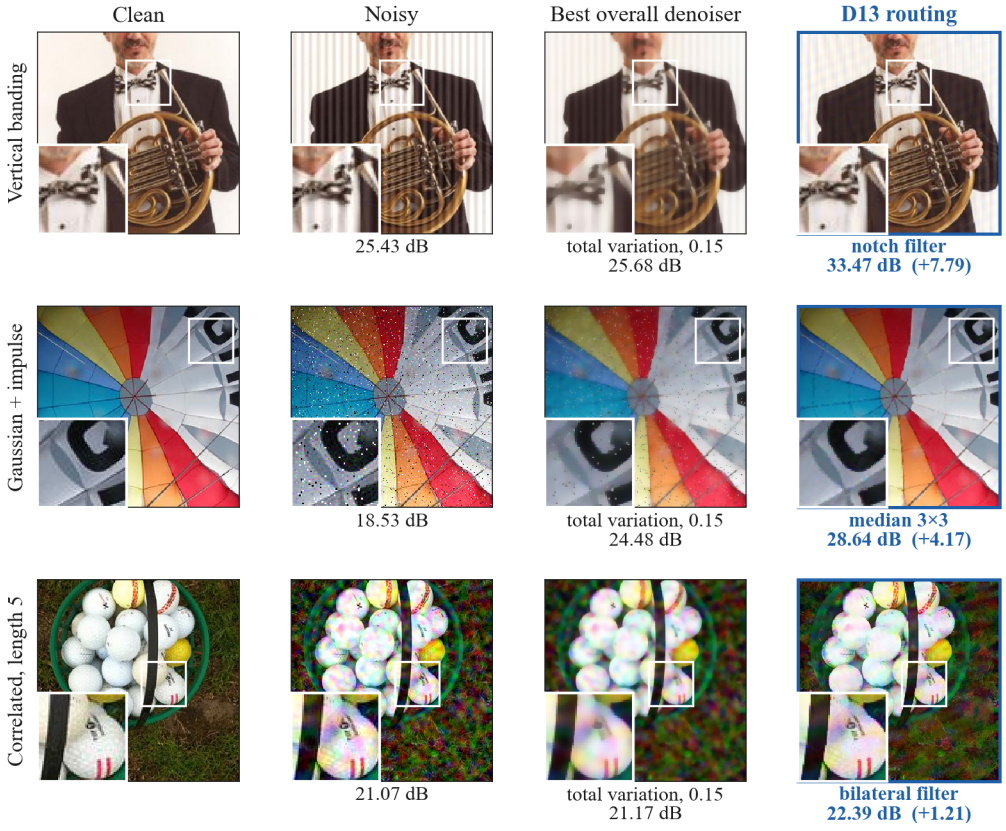

Figure 9: **Examples of routing.** Each row shows one image and corruption. Columns show the clean image, the noisy image, the output of the best overall denoiser of the classical pool, and the output of the candidate that routing selected. PSNR is given under each panel, and the insets enlarge one region.

Conditions compared. The comparison used the eight withheld conditions composed only of the classifier's four types: four at severities not used in calibration (Gaussian $\sigma = 35$, Poisson peak 20, speckle $\sigma = 0.15$, salt-and-pepper at 8%) and four that combine two of them (Gaussian $\sigma = 25$ with Poisson peak 30, and Gaussian $\sigma = 25$ with salt-and-pepper at 5%, each at mixing fractions 0.3 and

Table 12: Gain of routing, restricted to the classifier's four types, over the classifier, on the eight withheld conditions composed only of those types, in dB. Positive favors routing. Bold marks a 95% interval excluding zero.

Arm	Classical	Specialist	General-purpose	Hybrid
Area average, 1× (published)	**+0.70**	**+0.47**	**+0.62**	**+0.63**
Area average, 9×, 2508 steps	**+0.81**	**+0.56**	**+0.81**	**+0.87**
Area average, 9×, 7752 steps	**+0.84**	**+0.66**	**+0.88**	**+0.99**
Native resolution, 1×	**+0.33**	−0.01	**+0.42**	**+0.29**
Native resolution, 9×, 2508 steps	**+0.38**	+0.08	**+0.47**	**+0.35**
Native resolution, 9×, 7752 steps	**+0.38**	+0.01	**+0.44**	**+0.33**

Table 13: The same comparison split into its two halves: the four conditions that are pure instances of the classifier's types, and the four that combine two of them. Gain of routing over the classifier, in dB; positive favors routing, and bold marks a 95% interval excluding zero.

	Four unseen severities				Four mixtures			
Arm	Clas.	Spec.	Gen.	Hyb.	Clas.	Spec.	Gen.	Hyb.
Area average, 1× (published)	+0.05	−0.04	+0.06	+0.03	**+1.35**	**+0.99**	**+1.17**	**+1.22**
Area average, 9×, 2508 steps	−0.07	**−0.19**	+0.01	−0.05	**+1.70**	**+1.32**	**+1.60**	**+1.80**
Area average, 9×, 7752 steps	−0.06	−0.19	−0.01	−0.06	**+1.75**	**+1.51**	**+1.77**	**+2.04**
Native resolution, 1×	**−0.20**	**−0.31**	−0.02	**−0.11**	**+0.86**	**+0.30**	**+0.86**	**+0.69**
Native resolution, 9×, 2508 steps	**−0.17**	**−0.28**	−0.01	−0.10	**+0.92**	**+0.45**	**+0.95**	**+0.81**
Native resolution, 9×, 7752 steps	**−0.16**	**−0.34**	−0.02	−0.11	**+0.92**	**+0.37**	**+0.89**	**+0.77**

0.6). The other 18 withheld conditions contain correlated, banding, anisotropic, or annular noise, for which the classifier has no label; they are reported at the end of this appendix.

Preprocessing of the input. The published method averages the 256×256 cell down to 64×64 with a 4×4 area average. We also report a variant that leaves the amplitude of the noise unchanged: the central 128×128 at native resolution, from which training draws random 64×64 windows and testing averages the four quadrant tiles in softmax space. On Gaussian noise at $\sigma = 25$, the standard deviation of the noise component of the four-phase residual is 25.0 at native resolution and 6.25 after the area average, the factor of four that averaging sixteen samples predicts, and the severities within a family differ only in that amplitude. We report both variants and treat the published one as the faithful reimplementation.

Training data and budget. We also trained the classifier on nine times the data, so that its accuracy would not be limited by the size of the training set. The 1× arm ran 100 epochs of 26 steps per fold, 2600 steps in all. At nine times the data an epoch is 228 steps, so the arm matched in optimizer steps ran eleven epochs, 2508 steps, 3.5% short of the 1× budget. A second 9× arm ran 7752 steps, three times the 1× budget. Routing is not trained.

Gain on the eight conditions. With the published preprocessing, routing gave higher PSNR than the classifier in every pool, by 0.47 to 0.70 dB, with intervals $[0.46, 1.03]$, $[0.26, 0.73]$, $[0.49, 0.75]$, and $[0.46, 0.80]$. Reading the noise at native resolution narrowed the difference without changing its sign in three of the four pools, and in the specialist pool the two drew level, with interval $[-0.13, 0.11]$.

Where the difference comes from. On the four conditions that are pure instances of the classifier's types the two methods are close. With the published preprocessing at the standard budget they do not differ in any pool; nine times the data puts the classifier 0.19 dB ahead in the specialist pool, and at native resolution it leads there by up to 0.34 dB. All of routing's advantage on the eight conditions comes from the four mixtures, where it gains 0.99 to 1.35 dB with the published preprocessing and 0.30 to 0.86 dB at native resolution. The specialist pool separates the two effects: at native resolution with the largest budget the classifier leads by 0.34 dB on the four severities and trails by 0.37 dB on the four mixtures, and the two cancel to the parity recorded in Table 12. A mixture of two of the classifier's types is nameable component by component and has no name as a whole, and the output

Table 14: Noise-type accuracy on the four withheld conditions that are pure instances of the classifier's types. A mixture has no single true type, so accuracy is not defined on the other four. With four classes, chance is 0.25.

Arm	Gaussian 35	Poisson 20	Speckle 0.15	S&P 8%	All four
Routing, four types	0.967	0.913	0.850	0.950	0.920
Area average, 1× (published)	0.683	0.803	0.753	0.977	0.804
Area average, 9×, 2508 steps	0.700	0.900	0.883	0.993	0.869
Area average, 9×, 7752 steps	0.710	0.863	0.930	0.990	0.873
Native resolution, 1×	0.397	0.950	0.970	1.000	0.829
Native resolution, 9×, 2508 steps	0.457	0.933	0.967	1.000	0.839
Native resolution, 9×, 7752 steps	0.603	0.983	0.970	0.997	0.888

space has four labels and none for a combination of types, so the classifier must discard one of the two mechanisms present in the image. Routing carries no such restriction, because its references are labeled by the denoiser that gives the highest PSNR rather than by the corruption.

Effect of more training data. Nine times the training data raised the classifier's type accuracy on unseen severities from 80.4 to 86.9% at the matched budget, and it widened routing's advantage on the eight conditions in every pool and under both input variants (Table 12). The improvement in recognition falls on the conditions that have a name in the classifier's taxonomy; the mixtures, which have none, are where the two methods differ, and they are unaffected. Identifying the corruption more accurately therefore did not lead to a better choice of denoiser.

Accuracy on interpolated severities. Gaussian noise at $\sigma = 35$ lies between the $\sigma = 25$ and $\sigma = 50$ the classifier trained on, and it is the condition it identifies least often, the lowest of its four columns in every arm. The published arm names it correctly for 68.3% of images and the descriptor for 96.7%. The native-resolution arm, whose accuracy on held-out images of the eight calibration conditions exceeds 99%, falls to 39.7% here, so reading the noise at full amplitude let it learn the severities it was shown without letting it interpolate between them. More training raises this column to 45.7% at the matched budget and to 60.3% when the optimizer budget is tripled as well, and it remains the lowest of the four throughout.

Images outside the four types. On the 18 withheld conditions that contain correlated, banding, anisotropic, or annular noise, the classifier must still return one of its four labels. The published arm calls 55.8% of them Gaussian and 21.4% Poisson, and splits the remaining 22.8% almost evenly between speckle and salt-and-pepper. It assigns these labels with high confidence: the mean probability of its top class on these images is 0.929, and 77.7% of them exceed 0.9. A fixed taxonomy gives no indication that an image falls outside it. The out-of-support score of Appendix K provides that indication for the descriptor.

K NOISE OUTSIDE CALIBRATION

Setup. The queries are the 1,280 sRGB validation blocks of SIDD (Abdelhamed et al., 2018), taken from 40 photographs, 32 blocks each. Blocks of one photograph share a scene, a camera, and an ISO setting, so intervals come from a cluster bootstrap over the 40 photographs, 5,000 replicates. The real references are 320 patches of the SIDD Small release, from 160 scene instances over 10 scenes and 5 cameras, with ISO values from 50 to 10,000, labeled with the best candidate for their ISO band. The two sets are disjoint, and once the patches are present 99.6% of the blocks retrieve a real reference.

Gain with and without real references. Table 15 reports both experiments. The first two columns are the gain over the best overall denoiser of the synthetic calibration set, the baseline used in the main text. The last two measure the per-image choice alone, against the candidate with the highest mean PSNR on the added real references, which is what a system would run once those references are available. In the general-purpose and hybrid pools that candidate is a network trained on the SIDD training split, disjoint from the validation blocks; the oracle gap above it is small, and routing does

Table 15: Routing on the SIDD validation blocks, in dB, with 95% cluster-bootstrap intervals over the 40 photographs.

	Over the synthetic baseline		Over the real-reference baseline	
Pool	synthetic refs.	with real refs.	Gain	Oracle gap
Classical	-3.29 [-3.78, -2.84]	-0.15	-0.15 [-0.33, 0.02]	0.31
Specialist	-1.01 [-1.53, -0.51]	$+0.88$	$+1.84$ [1.11, 2.59]	3.48
General-purpose	-7.95 [-8.87, -6.97]	$+4.84$	-0.04 [-0.08, -0.00]	0.09
Hybrid	-8.01 [-8.92, -7.03]	$+4.78$	-0.04 [-0.08, -0.01]	0.23

not improve on it. The specialist networks were trained on the synthetic families and none had seen real noise. Removing the synthetic conditions from the library, and routing the blocks against the 320 real patches alone, leaves the gain of every pool within its 95% interval. The synthetic references are almost never the nearest ones once the real patches are present, and they are still required for synthetic queries, on which the mixed library changes the gain by at most 0.04 dB.

The out-of-support score. Let z be the standardized descriptor of a query. We score it by the negative log-likelihood under the best-fitting calibration condition,

$$s_{\text{out}}(z) = -\max_{c} \log \mathcal{N}(z; \mu_c, \Sigma_c),$$

where μ_c and Σ_c are the mean and the Ledoit–Wolf shrunk covariance of the references of condition c; the shrinkage is needed because each covariance is estimated from a few hundred references in thirteen dimensions. Scores are computed leave-one-domain-out, and the threshold is the 99th percentile of the calibration images, so 1% of them are flagged by construction. The flag fires on 90.2% of the SIDD blocks and 7.3% of the withheld synthetic images, and within the latter on 15.5% of the single families but 0.2% of the mixtures, which superpose families already in the calibration set. The distance to the nearest reference, which routing already computes, gives the same ordering but flags less: 54.3%, 5.1%, 10.9%, and 0.1%.

Routing on flagged images. Table 16 splits the withheld synthetic images at that threshold. The images here are the 7,800 withheld synthetic ones, not the SIDD blocks, so the intervals are those of the discrete benchmark, from the paired bootstrap over content domains and photographs of Appendix E; the cluster bootstrap declared above applies only to Table 15.The comparison is made in recovered share rather than in decibels, because a flagged image lies farther from the references and its oracle gap is larger; in the specialist pool routing gains more decibels on the flagged images while recovering less of what is available. Splitting on the distance instead gives the same sign in three pools.

Table 16: Routing on the withheld synthetic images, split by the flag. *Recovered* is the gain as a share of the oracle gap. The last column gives the difference in recovered share between unflagged and flagged images, in percentage points, with a 95% interval. It is computed from the unrounded shares, so it does not reproduce the difference of the two rounded columns exactly.

	Flagged (566)		Unflagged (7234)		
Pool	Gain / gap (dB)	Recovered	Gain / gap (dB)	Recovered	Difference
Classical	$+1.12$ / 2.38	47%	$+1.38$ / 2.02	68%	$+22$ [6, 38]
Specialist	$+1.80$ / 3.63	50%	$+1.56$ / 2.46	64%	$+14$ [2, 27]
General-purpose	-0.32 / 0.41	-78%	$+0.18$ / 0.97	18%	$+96$ [36, 180]
Hybrid	$+0.92$ / 2.53	36%	$+0.99$ / 2.26	44%	$+7$ [-12, 36]