Robust Unsupervised StyleGAN Image Restoration Supplementary Materials

Anonymous CVPR submission

Paper ID 8084

This supplementary material presents the following additional information to complement the paper:

- Sec. 1: a set of ablation experiments on our multi-resolution loss function as well as the normalized gradient descent optimizer;
- Sec. 3: more details on the entire FFHQ-X test set (see sec. 6.1 of the main paper);
- Sec. 4: illustration of the degradation levels used in our benchmark (sec. 5.1 of the main paper);
- Sec. 5: details on the selection of hyperparameters for the PULSE [4] and L-BRGM [1] baselines;
- Sec. 6: quantification of the impact of our 3-phase latent extension approach;
- Sec. 7: compares our method to the baselines under the Natural Image Quality Evaluator (NIQE) [5] score;
- Sec. 8: additional qualitative results to complement figs. 4 and 5 from the main paper;
- Sec. 9: results on other datasets;
- Sec. 10: high resolution results to better see the differences;

1. Ablations

	Accur. (LPIPS) ↓			Fidelity (LPIPS) ↓			Realism (pFID) ↓		
	MR	256	1024	MR	256	1024	MR	256	1024
Upsampling (bilinear, bicubic or Lanczos)									
XS	.414	.432	.397	.313	.323	.278	17.0	28.8	16.1
M	.472	.507	.454	.172	.214	.101	22.3	41.3	25.1
XL	.514	.534	.492	.090	.080	.023	21.3	62.0	21.9
Denoising (clamped Poisson and Bernoulli mixture)									
XS	.425	.430	.433	.156	.167	.147	18.5	26.4	21.0
M	.446	.476	.466	.130	.154	.128	19.8	41.8	27.8
XL	.474	.529	.519	.084	.103	.102	17.9	55.8	26.9
Deartifacting (JPEG compression)									
XS	.432	.428	.478	.349	.344	.392	14.8	21.2	28.6
M	.445	.450	.504	.357	.360	.413	15.4	24.8	35.1
XL	.490	.535	.584	.412	.446	.510	18.7	63.4	72.6
Inpainting (random strokes)									
XS	.378	.385	.356	.348	.354	.326	12.9	17.7	17.0
M	.396	.406	.387	.206	.213	.195	14.5	19.6	17.6
XL	.422	.437	.410	.132	.141	.125	15.9	22.5	17.2

Table 1. Ablations comparing performances on all metrics for single degradations, comparing our multi-resolution loss function (columns labeled MR) against an LPIPS loss applied at resolution 256 × 256 (columns labeled 256) and 1024 × 1024 (columns labeled 1024). In all cases, an L1 loss with a weight of 0.1 was also used. For upsampling tasks, column 1024 instead shows LPIPS applied at the maximal resolution after downsampling, and column 256 after another 4x downsampling when applicable. The multi-resolution loss function is critical for performance on denoising and deartifacting. For upsampling and inpainting, its use results in a small loss in accuracy in exchange for a (sometimes large) gain in realism. Results are color-coded as best and second best .

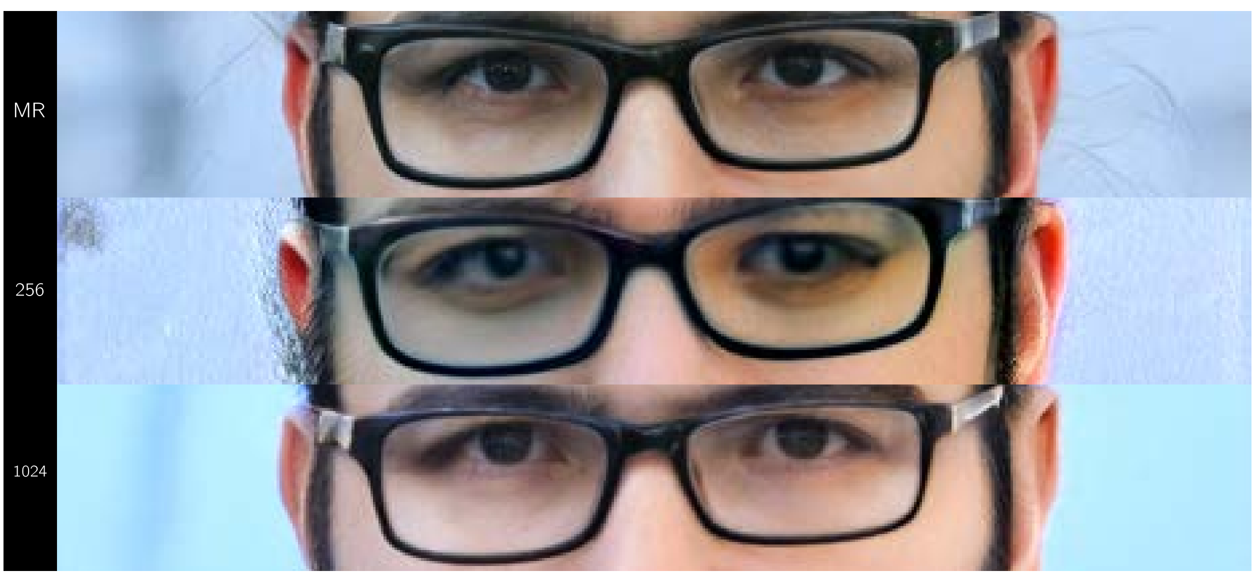


Figure 1. Final predictions during XL denoising with the LPIPS loss applied at different resolutions (from top to bottom: MR, 256, 0.9; MR is the multi-resolution loss function proposed in our method). While the prediction with our multi-resolution loss function is not artifact-free, it is much less blurry then the prediction at 1024.

Table 1 shows ablations on our multi-resolution loss function. We compare our proposed loss single degradations at (XL, M, XS) compared to a more standard (VGG) perceptual loss applied at a full resolution and at a fixed (256) resolution (the latter is commonly performed because it is close to VGG's native resolution 224 × 224). It shows that our multi-resolution loss function is critical for denoising and deartifacting: this is somewhat unsurprising because these degradations damage the high frequencies while mostly preserving the low frequencies; JPEG compression was *explicitly designed* with this intent. But, it also shows that such losses retain good performance for other degradations: it is robust.

	Accur. (LPIPS) ↓			Fidelity (LPIPS) ↓			Realism (pFID) ↓		
	0.0	0.5	0.9	0.0	0.5	0.9	0.0	0.5	0.9
Upsampling (bilinear, bicubic or Lanczos)									
XS	.414	.397	.375	.313	.282	.236	17.0	19.8	22.6
M	.472	.473	.484	.172	.135	.106	22.3	33.2	50.2
XL	.514	.492	.495	.090	.074	.055	21.3	18.8	20.9
Denoising (clamped Poisson and Bernoulli mixture)									
XS	.425	.413	.409	.156	.143	.131	18.5	21.6	26.9
M	.446	.444	.452	.130	.120	.114	19.8	25.4	37.2
XL	.474	.471	.489	.084	.077	.076	17.9	21.8	38.8
Deartifacting (JPEG compression)									
XS	.432	.414	.391	.349	.324	.291	14.8	15.7	18.1
M	.445	.432	.417	.357	.332	.300	15.4	17.3	21.5
XL	.490	.491	.500	.412	.392	.381	18.7	26.6	45.1
Inpainting (random strokes)									
XS	.378	.353	.319	.348	.325	.293	12.9	13.0	13.8
M	.396	.382	.355	.206	.190	.170	14.5	15.3	17.6
XL	.422	.405	.392	.132	.124	.105	15.9	17.3	20.9

Table 2. Ablations comparing performances on all metrics for single degradations, comparing our optimizer without (columns labeled 0.0) and with momentum at values 0.5 and 0.9 (columns labeled 0.5 and 0.9, respectively). Results are color-coded as best and second best. While momentum improves fidelity and, in many cases, accuracy, it results in a severe degradation in realism, especially for non-linear tasks such as inpainting and denoising.

Table 2 shows ablations for our normalized gradient descent optimizer. Recall that the Adam optimizer takes steps proportional to the value obtained by normalizing an elementwise running average the gradient's first moment (momentum) with an elementwise running average of its second moment. We show show that including just momentum (at values 0.5 and 0.9) leads to a degradation in realism. In other words, even a minimal attempts at moving our optimizer closer to Adam results in a severe degradation in realism.

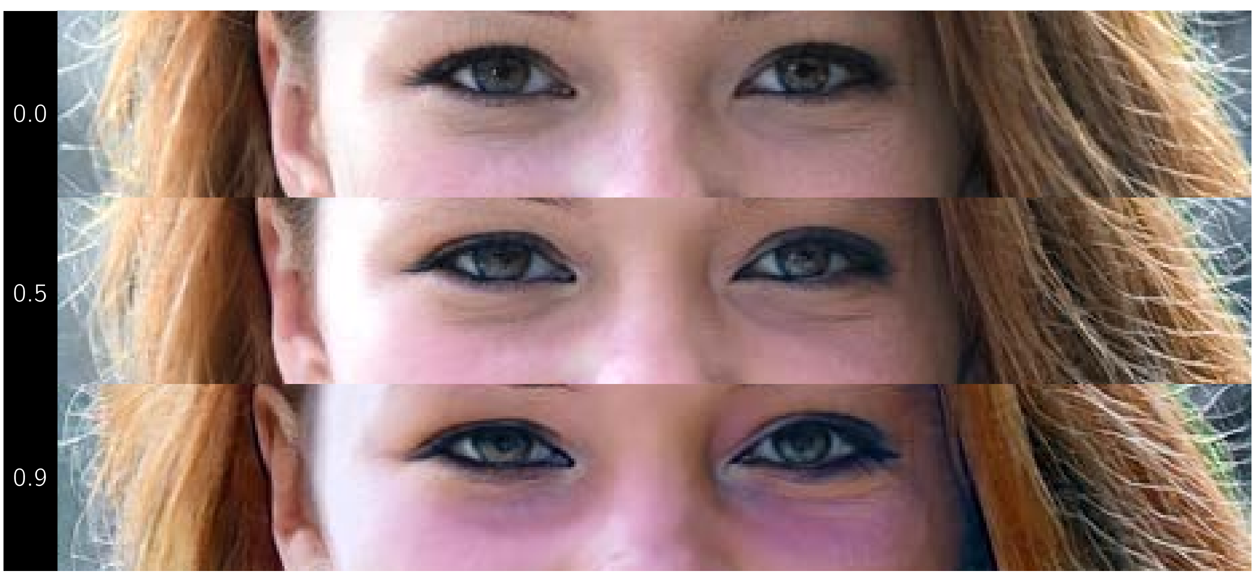


Figure 2. Final predictions during XL deartifacting for different momentum levels (from top to bottom: 0.0, 0.5, 0.9; 0.0 is the value proposed in our method). Adding momentum results in a severe degradation in realism.

2. Additional comparison to denoising diffusion restoration models (DDRM)

The main advantage of image restoration with StyleGAN inversion is that, since the latter is gradient based, it can handle non-linear degradation easily while DDRM cannot. Besides, this section gives some additional comparisons to DDRM for uncropping, and attempts to explain its high accuracy scores. For inpainting we can decompose the accuracy into two parts, provided that it is measured with an L1 norm: the fidelity (measuring if the degraded prediction matches the target), and its complement that we will dub the authenticity (measuring if what is absent from the target is correctly inferred by the prediction). More concretely, authenticity measures the distance between the masked part of the ground truth and the prediction. Recalling that $\boldsymbol{x}_{clean}$ denotes the prediction and that $\boldsymbol{y}_{clean}$ denotes the ground truth, for a (linear) masking degradation H we have:

$$\text{accuracy}_{\text{L1}}(\boldsymbol{x}_{clean}, \boldsymbol{y}_{clean}) = |\boldsymbol{x}_{clean} - \boldsymbol{y}_{clean}| = \text{fidelity}_{\text{L1}}(\boldsymbol{x}_{clean}, \boldsymbol{y}_{clean}) + \text{authenticity}_{\text{L1}}(\boldsymbol{x}_{clean}, \boldsymbol{y}_{clean})$$

where

$$\text{fidelity}_{\text{L1}}(\boldsymbol{x}_{clean}, \boldsymbol{y}_{clean}) = |H\boldsymbol{x}_{clean} - H\boldsymbol{y}_{clean}|$$
$$\text{authenticity}_{\text{L1}}(\boldsymbol{x}_{clean}, \boldsymbol{y}_{clean}) = |(I-H)\boldsymbol{x}_{clean} - (I-H)\boldsymbol{y}_{clean}| \, .$$

Results for uncropping are shown in table X with these new metrics. These results show that DDRM obtains high accuracy because it fidelity is excellent, but that it struggles with authenticity, meaning that it fails to infer beyond this. In figure Y, we give more results on uncropping, which shows this well: DDRM perfectly reproduces the provided crop, but struggles to infer the missing part. In contrast, StyleGAN-based image restoration is much better at doing so.

Accuracy (L1) ↓		Fidelity (L1) ↓		Authenticity (L1) ↓	
DDRM	Ours	DDRM	Ours	DDRM	Ours
Uncropping (top left corner unmasked)					
0.16721	0.12120	0.00101	0.00826	0.16620	0.11343

Table 3. Quantitative comparison to DDRM [3] on L1 accuracy, fidelity, and authenticity. Results are color-coded as best .

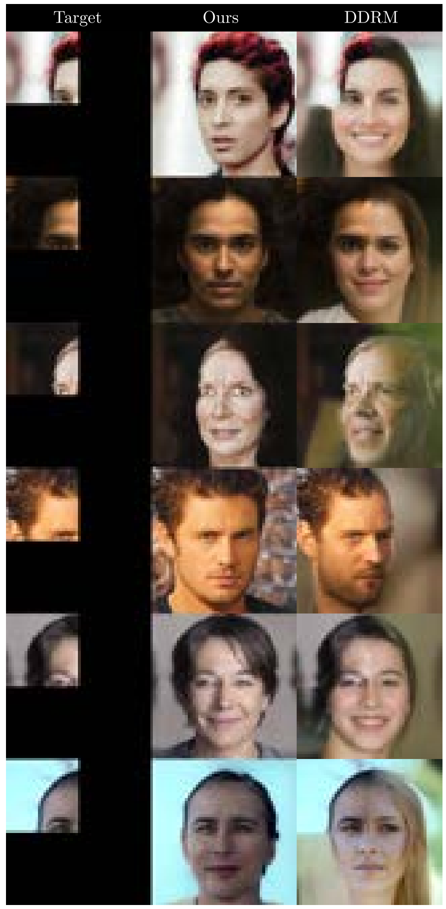


Figure 3. DDRM struggles to generalize when the signal is very faint.

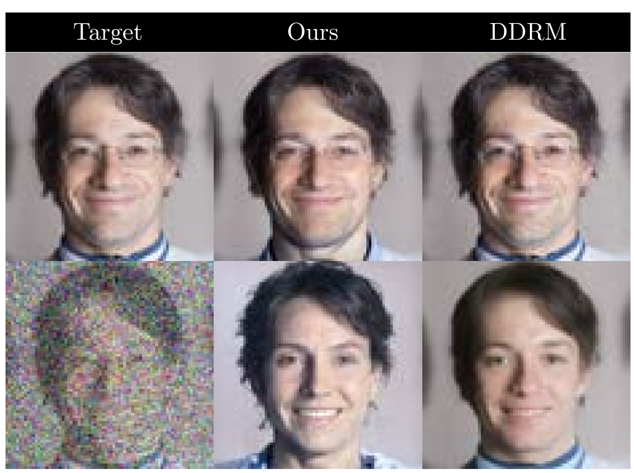


Figure 4. DDRM has near-perfect fidelity when degradations are low but yields unrealistic (blurry) predictions under high degradations (top: XS additive gaussian noise, bottom: XL addititive gaussian noise). StyleGAN generates a realistic image but poorly fitting image.

3. Dataset collection

The pretrained FFHQ networks are trained on all 70k FFHQ images, leaving none for testing. For this reason, we collect an additional 100 images (50 men and 50 women) by scraping Flickr for portrait images uploaded after 2020 (ensuring that they were not part of the original training set). In detail, we scraped 1k images using the query "portrait man" and 1k images using the query "portrait woman". After running the facial alignment script, we discarded all images with: low resolution, noise, blur or visible JPEG compression; extremely unnatural lighting; pure white or pure black background; facial obstructions; black and white coloration; nudity; too little padding around the head. Of the remaining images, we kept 50 for each class by manually selecting for diversity.

Diversity As with the original FFHQ dataset, where 69.2% of all images are of white people [2], our dataset is biased by the distribution naturally present in Flickr images. Our method also inherits the bias present in StyleGAN; refer to PULSE [4] and for a more in-depth discussion.

Figure 5. All 100 test images of our FFHQ-X test dataset. All images were scraped from Flickr and have a resolution 1024×1024.

4. Degradation levels

Fig. 6 below illustrates the degradation levels used in our benchmark.

Figure 6. All degradation levels used in our benchmark (from left to right: XS, S, M, L, XL; from top to bottom: upsampling, denoising, deartifacting, and inpainting). Please note that all images have native resolution 1024×1024; inlays (bottom-left of each image) magnify $4\times$ a small 32×32 crop near the left eye. Please zoom in to better appreciate the levels.

5. Hyperparameter search

5.1. PULSE

PULSE's main regularization hyperparameter is the geocross term λ_{geo} which constrains the geodisc distance between all 18 latent codes in $\mathcal{W}^+$. To account for our change of the network and the introduction of new tasks we first increased the number of steps from 100 to 250 and adjusted the learning rate to 0.8 after manual inspection on validation data. We then ran our benchmark on a range of 8 hyperparamter values $\lambda_{reg} = \{0.0, 0.001, 0.01, 0.05, 0.15, 0.3, 0.5, 1.5\}$ directly on the test set, retaining the parameter values with the best accuracy for each (task, level) pair. Note that the original paper proposes the value 0.05 and but that most retained values were in $\{0.001, 0.01\}$.

λ_{geo}	0.0	0.001	0.01	0.05	0.15	0.3	0.5	1.5
upsampling/XS	0.5059	0.5011	0.4926	0.4958	0.5049	0.5079	0.5119	0.5222
upsampling/S	0.5041	0.4966	0.4920	0.4972	0.5043	0.5104	0.5132	0.5240
upsampling/M	0.5036	0.4979	0.4953	0.4999	0.5063	0.5124	0.5192	0.5251
upsampling/L	0.5102	0.5058	0.5028	0.5015	0.5103	0.5182	0.5206	0.5274
upsampling/XL	0.5223	0.5153	0.5121	0.5151	0.5199	0.5208	0.5209	0.5306
denoising/XS	0.5083	0.5008	0.5051	0.5113	0.5191	0.5244	0.5313	0.5615
denoising/S	0.5085	0.4989	0.5025	0.5129	0.5198	0.5241	0.5326	0.5621
denoising/M	0.5100	0.5000	0.5033	0.5141	0.5210	0.5251	0.5339	0.5645
denoising/L	0.5103	0.5015	0.5043	0.5101	0.5217	0.5316	0.5391	0.5743
denoising/XL	0.5208	0.5044	0.5110	0.5209	0.5293	0.5487	0.5587	0.5991
deartifacting/XS	0.5057	0.4984	0.4994	0.5041	0.5145	0.5187	0.5235	0.5427
deartifacting/S	0.5055	0.4975	0.4996	0.5059	0.5173	0.5206	0.5245	0.5402
deartifacting/M	0.5096	0.5003	0.4981	0.5087	0.5176	0.5215	0.5239	0.5443
deartifacting/L	0.5131	0.5005	0.5018	0.5113	0.5176	0.5225	0.5276	0.5419
deartifacting/XL	0.5233	0.5129	0.5085	0.5166	0.5246	0.5289	0.5302	0.5457
inpainting/XS	0.5079	0.5048	0.4981	0.4997	0.5077	0.5123	0.5155	0.5230
inpainting/S	0.5145	0.5057	0.5009	0.5034	0.5120	0.5143	0.5203	0.5303
inpainting/M	0.5231	0.5125	0.5088	0.5122	0.5143	0.5204	0.5245	0.5380
inpainting/L	0.5278	0.5182	0.5128	0.5148	0.5234	0.5286	0.5266	0.5452
inpainting/XL	0.5378	0.5238	0.5246	0.5253	0.5301	0.5331	0.5345	0.5519

Table 4. Accuracies (LPIPS) in single tasks for PULSE at all hyperparameter values. Best accuracies are marked as best and default hyperparameter values as default.

λ_{geo}	0.0	0.001	0.01	0.05	0.15	0.3	0.5	1.5
NA	0.5289	0.5167	0.5225	0.5311	0.5402	0.5536	0.5704	0.6038
AP	0.5227	0.5129	0.5108	0.5186	0.5278	0.5335	0.5380	0.5624
UA	0.5254	0.5151	0.5097	0.5165	0.5239	0.5269	0.5302	0.5488
NP	0.5234	0.5109	0.5110	0.5220	0.5311	0.5406	0.5509	0.5922
UN	0.5148	0.5006	0.5018	0.5132	0.5195	0.5303	0.5407	0.5690
UP	0.5275	0.5136	0.5105	0.5135	0.5185	0.5239	0.5276	0.5405
UNP	0.5290	0.5098	0.5163	0.5273	0.5312	0.5414	0.5535	0.5867
UAP	0.5380	0.5279	0.5253	0.5299	0.5387	0.5389	0.5418	0.5672
UNA	0.5427	0.5212	0.5284	0.5392	0.5484	0.5592	0.5744	0.6059
NAP	0.5384	0.5258	0.5329	0.5422	0.5553	0.5719	0.5878	0.6228
UNAP	0.5573	0.5330	0.5384	0.5496	0.5614	0.5799	0.5939	0.6245

Table 5. Accuracies (LPIPS) in composed tasks for PULSE at all hyperparameter values (composed tasks). Best accuracies are marked as best and default hyperparameter values as default.

5.2. L-BRGM

L-BRGM main regularization hyperparameters are λ_{lat} and λ_{map}, which respectively constrain the optimization to stay near the mean in $\mathcal{Z}$ space and the $\mathcal{W}^+$ extension to be as minimal as possible. To account for our change of the network and the introduction of new tasks we first adjust the learning rate to 0.05, again after manual inspection on validation data. L-BRGM uses best-of-k random initialization; because initialization is difficult under high degradations, we doubled the number of samples from 100 to 200 and added a 0.8 truncation to avoid outliers. L-BRGM also performs early-stopping by inspecting the ground truth; for a fair comparison, we instead fixed the number of steps to 500, the value used in BRGM. We then ran our benchmark on the full test set for 12 hyperparameter value pairs

$$\begin{pmatrix}\lambda_{map}\\ \lambda_{lat}\end{pmatrix} = \{\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0.01\\0.0001\end{pmatrix}, \begin{pmatrix}0.1\\0.001\end{pmatrix}, \begin{pmatrix}1.0\\0.001\end{pmatrix}, \begin{pmatrix}1.0\\0.01\end{pmatrix}, \begin{pmatrix}5\\0.05\end{pmatrix}, \begin{pmatrix}5\\0.01\end{pmatrix}, \begin{pmatrix}15\\0.2\end{pmatrix}, \begin{pmatrix}30\\0.2\end{pmatrix}, \begin{pmatrix}30\\0.4\end{pmatrix}, \begin{pmatrix}60\\0.8\end{pmatrix}, \begin{pmatrix}90\\1.6\end{pmatrix}\},$$

retaining the parameter values with the best accuracy for each (task, level) pair.

λ_{map} λ_{lat}	0.0 0.0	0.01 0.0001	0.1 0.001	1 .001	1 0.01	5 0.01	5 0.05	15 0.2	30 0.2	30 0.4	60 0.8	90 1.6
upsampling/XS	0.4123	0.4097	0.4074	0.4122	0.4112	0.4269	0.4241	0.4430	0.4518	0.4538	0.4644	0.4718
upsampling/S	0.4157	0.4157	0.4125	0.4149	0.4155	0.4268	0.4303	0.4442	0.4577	0.4547	0.4683	0.4735
upsampling/M	0.4662	0.4621	0.4607	0.4591	0.4583	0.4596	0.4613	0.4719	0.4792	0.4782	0.4876	0.4902
upsampling/L	0.5160	0.5107	0.5089	0.4925	0.4978	0.4885	0.4892	0.4875	0.4932	0.4922	0.4956	0.5014
upsampling/XL	0.5596	0.5582	0.5437	0.5287	0.5324	0.5196	0.5125	0.5095	0.5060	0.5106	0.5144	0.5172
denoising/XS	0.4429	0.4434	0.4402	0.4488	0.4496	0.4651	0.4652	0.4827	0.4942	0.4975	0.5072	0.5157
denoising/S	0.4561	0.4538	0.4501	0.4572	0.4554	0.4736	0.4734	0.4892	0.5007	0.5017	0.5120	0.5213
denoising/M	0.4696	0.4666	0.4653	0.4660	0.4648	0.4803	0.4828	0.4977	0.5084	0.5089	0.5195	0.5273
denoising/L	0.4931	0.4897	0.4853	0.4815	0.4814	0.4940	0.4928	0.5066	0.5198	0.5235	0.5327	0.5444
denoising/XL	0.5252	0.5212	0.5173	0.5108	0.5107	0.5255	0.5245	0.5381	0.5519	0.5490	0.5669	0.5756
deartifacting/XS	0.4417	0.4429	0.4422	0.4513	0.4520	0.4729	0.4763	0.4864	0.4959	0.4974	0.5066	0.5101
deartifacting/S	0.4510	0.4500	0.4482	0.4569	0.4577	0.4781	0.4777	0.4915	0.4954	0.5008	0.5098	0.5151
deartifacting/M	0.4658	0.4616	0.4612	0.4636	0.4662	0.4822	0.4829	0.4959	0.5058	0.5026	0.5118	0.5191
deartifacting/L	0.4877	0.4824	0.4755	0.4770	0.4776	0.4922	0.4925	0.5023	0.5078	0.5105	0.5179	0.5219
deartifacting/XL	0.5265	0.5283	0.5179	0.5044	0.5032	0.5074	0.5046	0.5114	0.5237	0.5250	0.5303	0.5351
inpainting/XS	0.4130	0.4121	0.4088	0.4136	0.4112	0.4278	0.4251	0.4408	0.4533	0.4545	0.4670	0.4727
inpainting/S	0.4268	0.4249	0.4249	0.4278	0.4299	0.4378	0.4387	0.4518	0.4606	0.4633	0.4735	0.4794
inpainting/M	0.4400	0.4389	0.4384	0.4402	0.4412	0.4506	0.4480	0.4609	0.4670	0.4706	0.4810	0.4847
inpainting/L	0.4551	0.4564	0.4522	0.4552	0.4522	0.4594	0.4584	0.4712	0.4763	0.4781	0.4867	0.4925
inpainting/XL	0.4697	0.4691	0.4626	0.4653	0.4601	0.4693	0.4673	0.4779	0.4870	0.4847	0.4930	0.5006

Table 6. Accuracies (LPIPS) in composed tasks for L-BRGM at all hyperparameter values (single tasks). Best accuracies are marked as best and default hyperparameter values as default .

λ_{map} λ_{lat}	0.0 0.0	0.01 0.0001	0.1 0.001	1 .001	1 0.01	5 0.01	5 0.05	15 0.2	30 0.2	30 0.4	60 0.8	90 1.6
NA	0.5068	0.4990	0.4848	0.4875	0.4876	0.5068	0.5072	0.5217	0.5357	0.5354	0.5504	0.5560
AP	0.4903	0.4871	0.4807	0.4784	0.4796	0.4913	0.4916	0.5077	0.5168	0.5132	0.5212	0.5292
UA	0.6237	0.6235	0.5893	0.5316	0.5301	0.5185	0.5207	0.5206	0.5248	0.5265	0.5328	0.5374
NP	0.4933	0.4908	0.4856	0.4798	0.4816	0.4898	0.4946	0.5100	0.5206	0.5188	0.5363	0.5397
UN	0.5589	0.5531	0.5407	0.5195	0.5190	0.5250	0.5246	0.5378	0.5471	0.5480	0.5638	0.5753
UP	0.4958	0.4922	0.4870	0.4814	0.4778	0.4857	0.4847	0.4917	0.5013	0.4951	0.5040	0.5094
UNP	0.5823	0.5744	0.5511	0.5318	0.5258	0.5347	0.5338	0.5520	0.5676	0.5683	0.5765	0.5876
UAP	0.6305	0.6219	0.5834	0.5364	0.5346	0.5265	0.5231	0.5307	0.5348	0.5374	0.5451	0.5546
UNA	0.6235	0.5999	0.5662	0.5353	0.5351	0.5498	0.5490	0.5643	0.5759	0.5760	0.5927	0.6022
NAP	0.5257	0.5128	0.5029	0.5016	0.5018	0.5178	0.5201	0.5407	0.5523	0.5499	0.5655	0.5717
UNAP	0.6372	0.6026	0.5668	0.5463	0.5488	0.5651	0.5712	0.5847	0.5987	0.5972	0.6167	0.6188

Table 7. Accuracies (LPIPS) in composed tasks for L-BRGM at all hyperparameter values (composed tasks). Best accuracies are marked as best and default hyperparameter values as default .

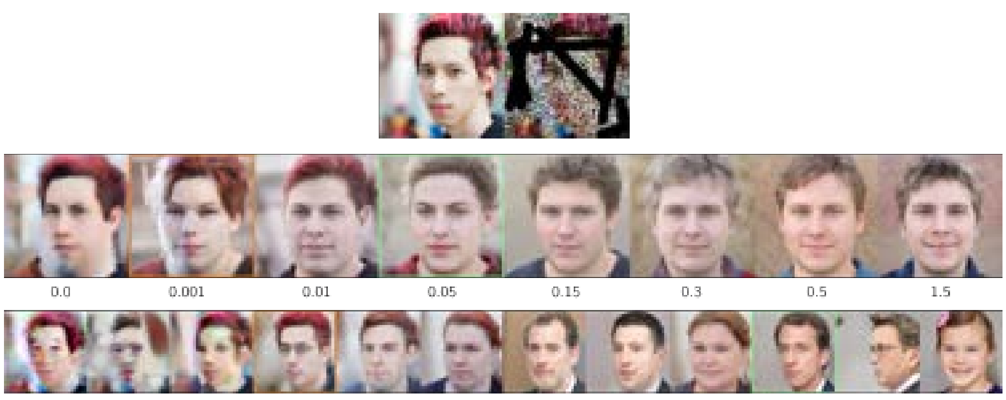


Figure 7. Visualization of the hyperparameter search: the first row shows the ground truth and the target; the second row gives the predictions for PULSE at all hyperparameter values; the third row gives the same for L-BRGM. The predictions with the best accuracies for both methods are bordered in orange while the predictions obtained with the default hyperparameters are bordered in green. Note that, for both methods, obtaining high accuracy results in poor realism.

6. Robustness of latent extension

See fig. 8 for qualitative results on denoising.

	Accur. (LPIPS) ↓			Fidelity (LPIPS) ↓			Realism (pFID) ↓		
	$\mathcal{W}$	$\mathcal{W}^+$	$\mathcal{W}^{++}$	$\mathcal{W}$	$\mathcal{W}^+$	$\mathcal{W}^{++}$	$\mathcal{W}$	$\mathcal{W}^+$	$\mathcal{W}^{++}$
Upsampling (bilinear, bicubic or Lanczos)									
XS	.480	.457	.414	.423	.387	.313	18.4	17.1	17.0
S	.496	.479	.449	.360	.318	.239	18.6	19.4	22.0
M	.507	.493	.472	.291	.246	.172	18.2	19.1	22.3
L	.523	.508	.490	.230	.186	.127	19.8	18.7	20.9
XL	.540	.525	.514	.168	.130	.090	22.8	20.4	21.3
Denoising (clamped Poisson and Bernoulli mixture)									
XS	.484	.464	.425	.221	.199	.156	18.1	17.7	18.5
S	.488	.469	.434	.201	.179	.140	18.2	17.9	19.1
M	.492	.476	.446	.189	.168	.130	18.3	18.2	19.8
L	.500	.484	.457	.164	.143	.110	18.6	17.8	19.2
XL	.515	.498	.474	.121	.107	.084	19.1	17.9	17.9
Deartifacting (JPEG compression)									
XS	.489	.470	.432	.421	.398	.349	17.9	16.5	14.8
S	.491	.474	.437	.421	.400	.350	17.7	16.0	15.4
M	.495	.478	.445	.427	.407	.357	17.5	15.9	15.4
L	.500	.485	.460	.434	.415	.374	18.6	16.6	16.0
XL	.519	.507	.490	.469	.453	.412	18.8	17.1	18.7
Inpainting (random strokes)									
XS	.462	.433	.378	.430	.402	.348	16.5	14.2	12.9
S	.464	.436	.387	.332	.308	.264	17.3	15.4	14.2
M	.468	.442	.396	.264	.243	.206	17.5	15.1	14.5
L	.475	.450	.409	.212	.194	.163	17.5	15.6	15.3
XL	.483	.460	.422	.173	.159	.132	18.0	16.3	15.9

Table 8. All metrics for all single degradations for each phase. Results color-coded as best and second best. The $\mathcal{W}^{++}$ phase consistently obtains the best accuracy and fidelity, although it leads to a slight degradation in realism in certain cases (namely denoising and upsampling). Interestingly, in inpainting, realism consistently improves across phases.

	Accur. (LPIPS) ↓			Fidelity (LPIPS) ↓			Realism (pFID) ↓		
	$\mathcal{W}$	$\mathcal{W}^+$	$\mathcal{W}^{++}$	$\mathcal{W}$	$\mathcal{W}^+$	$\mathcal{W}^{++}$	$\mathcal{W}$	$\mathcal{W}^+$	$\mathcal{W}^{++}$
2 degradations									
NA	.500	.486	.459	.362	.338	.290	18.2	16.6	17.3
AP	.504	.487	.457	.253	.238	.204	20.1	18.1	17.0
UA	.534	.523	.508	.392	.356	.287	21.2	19.4	19.7
NP	.502	.484	.458	.097	.085	.062	19.4	18.0	19.2
UN	.536	.524	.511	.218	.192	.153	22.3	20.8	21.1
UP	.515	.502	.485	.157	.134	.089	19.8	18.3	20.7
3 degradations									
UNP	.532	.519	.507	.084	.070	.051	21.8	20.0	20.1
UAP	.534	.524	.513	.178	.158	.119	22.2	20.3	20.5
UNA	.548	.540	.533	.360	.333	.290	25.3	22.7	22.8
NAP	.509	.493	.470	.212	.194	.160	20.4	18.4	18.5
4 degradations									
UNAP	.543	.533	.525	.180	.160	.131	24.3	22.1	21.8

Table 9. All metrics for all composed degradations for each phase. Results color-coded as best and second best. The $\mathcal{W}^{++}$ phase consistently obtains the best accuracy and fidelity. Realism is mostly preserved across all tasks (the $\mathcal{W}^{++}$ scores are very close to the $\mathcal{W}^+$ scores), except for composed upsampling and denoising.

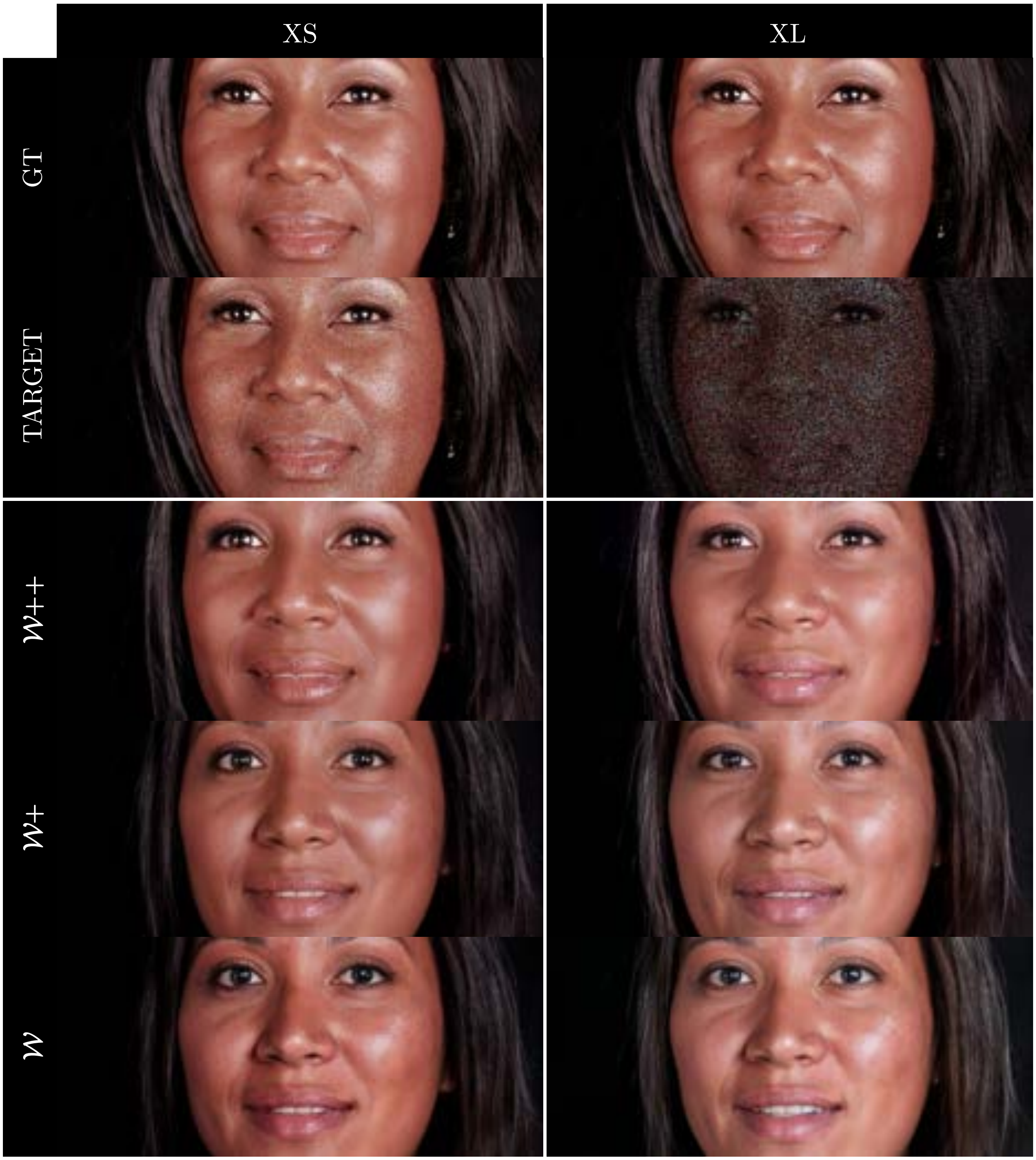


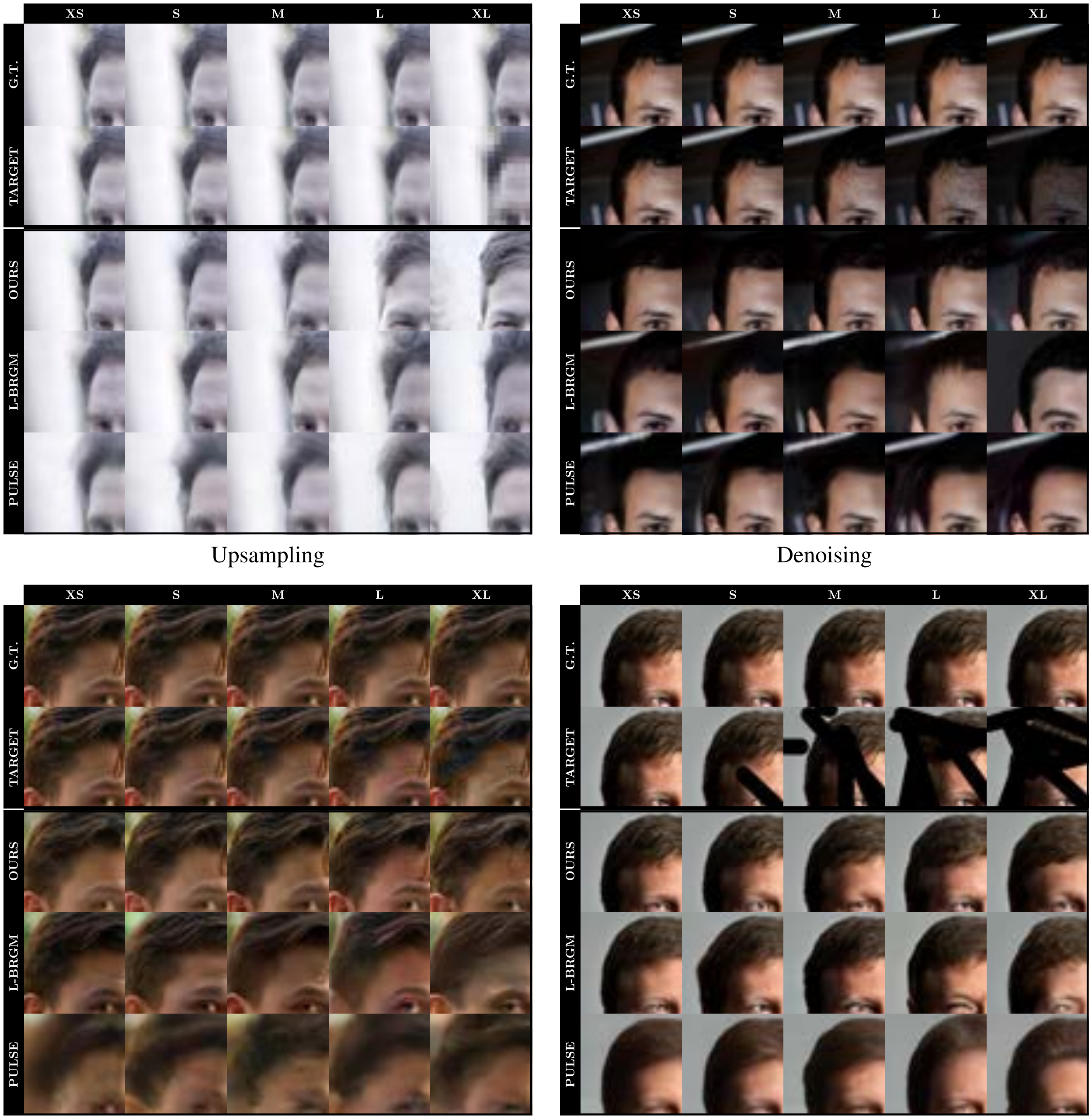
Figure 8. Predictions for denoising (left column: XS, right column: XL) at the end of each phase, for a hand-picked image. The first and second rows give the ground truth and target images, while the last three rows give the predictions for each phase (top to bottom: $\mathcal{W}^{++}$, $\mathcal{W}^{+}$, $\mathcal{W}$). As shown in this figure, the earlier phases give realistic but inaccurate predictions—they are unable to closely match the target even when the degradation levels are very low, while the $\mathcal{W}^{++}$ extension finds a close albeit imperfect match. With our proposed optimizer and loss function, $\mathcal{W}^{++}$ extension remains realistic even when the levels are very high (third row, second column).

7. NIQE scores

	NIQE ↓		
	PULS	**L-BRG**	**OURS**
Upsampling			
XL	3.46975	2.99718	3.07367
L	3.09940	3.13561	3.06486
M	2.96611	3.08266	2.98017
S	2.87300	3.12542	2.89456
XS	2.73831	3.05527	2.84786
Denoising			
XL	3.65306	3.38110	3.03496
L	3.66355	3.38293	3.00178
M	3.68114	3.25477	3.05595
S	3.60475	3.21669	3.02591
XS	3.50100	3.10698	2.99727
Deartifacting			
XL	3.31883	3.21646	3.01557
L	3.25669	3.21580	3.01792
M	3.11149	3.14421	3.01310
S	3.20801	3.08838	2.98830
XS	3.05744	2.98699	2.89791
Inpainting			
XL	3.61890	3.01630	3.14573
L	3.52034	3.00967	3.14987
M	3.48374	3.05570	3.15512
S	3.32579	2.96624	3.16195
XS	3.38315	3.05658	3.16246

Table 10. Quantitative comparison of the NIQE [5] scores on individual tasks against baselines (“PULS” [4] and “L-BRG” [1]).

8. Additional results



Upsampling

Denoising

Deartifacting

Inpainting

Figure 9. Additional results for all single restoration tasks showing all degradation levels. Each figure shows the same 128×128 crop at the top left of a randomly selected FFHQX images image.

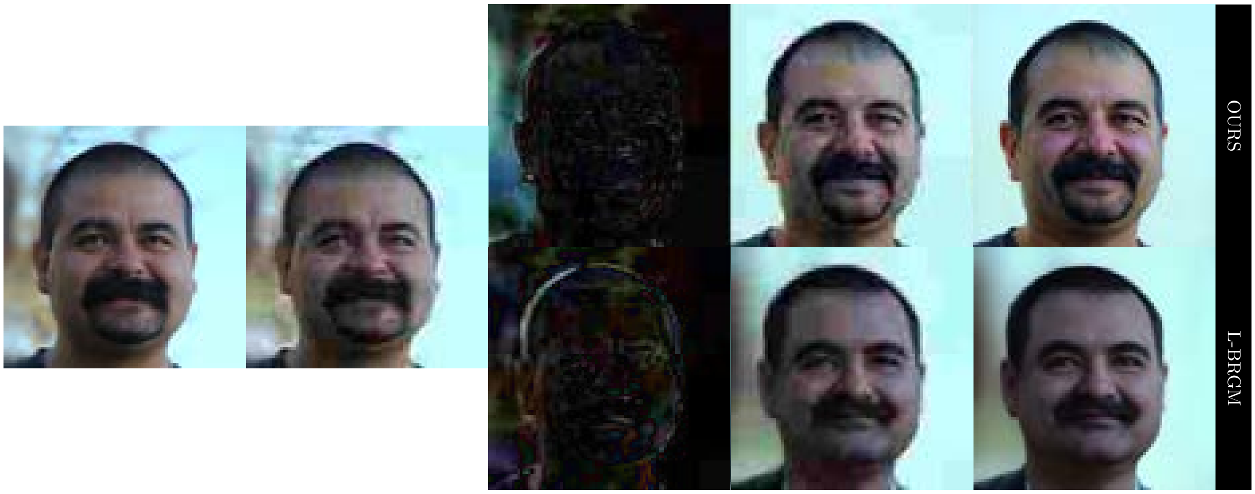


Figure 10. Overview of StyleGAN image restoration on composed upsampling and deartifacting with a hand-picked example. From left to right: ground truth $\boldsymbol{y}_{clean}$, target $\boldsymbol{y}$, error $|\boldsymbol{x} - \boldsymbol{y}|$, degraded prediction $\boldsymbol{x}$, prediction $\boldsymbol{x}_{clean}$.

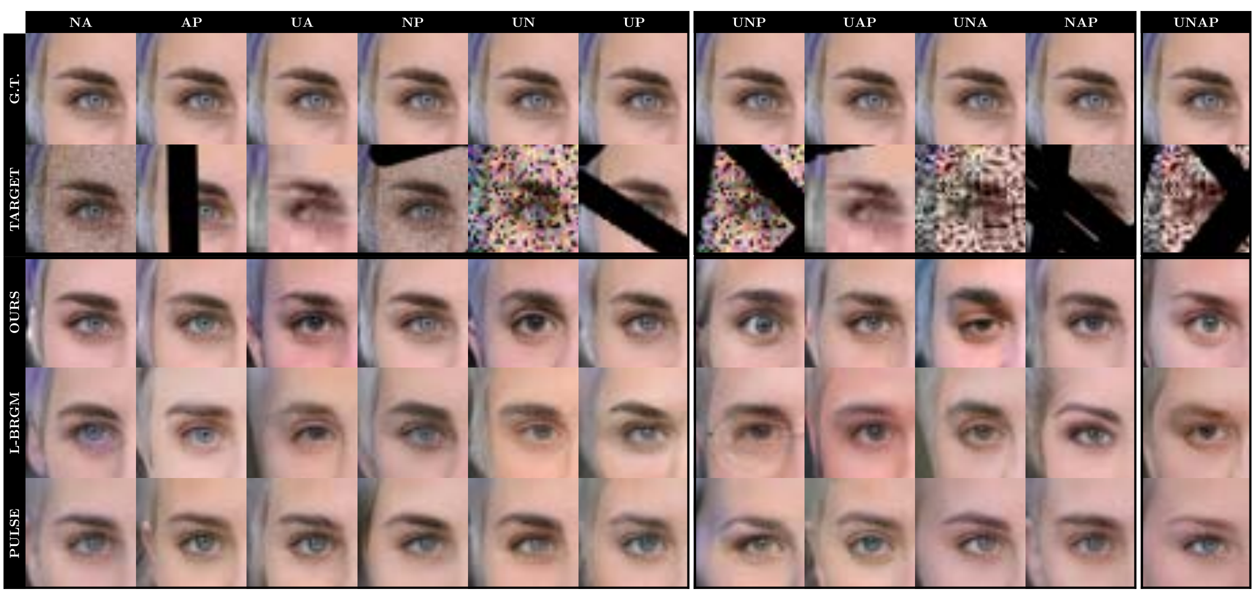


Figure 11. Additional results for composed degradations, where the same ground truth was used for all targets.

9. Other datasets

Figure 12. Qualitative results on all composed degradations (”UNAP”) for five images of AFHQ cats (left), AFHQ dogs (middle), and AFHQ wild (right). In all cases, we used the *StyleGAN2-ada* networks pretrained on each respective dataset with *no hyperparameter adjustment*. Please note that these generative models were trained on *both the training and the test sets*; as such the ground truths for these targets were seen during (pre)training.

10. High resolution

See the figures below for results shown in high resolution to better show details.

CVPR
#8084

Figure 13. Additional results for upsampling shown in high resolution (images were restored normally then sliced horizontally) on random FFHQX images. Top: extra-high (XL), middle: medium (M), bottom: extra-small (XS).

Figure 14. Additional results for denoising shown in high resolution (images were restored normally then sliced horizontally) on random FFHQX images. Top: extra-high (XL), middle: medium (M), bottom: extra-small (XS).

Figure 15. Additional results for deartifacting shown in high resolution (images were restored normally then sliced horizontally) on random FFHQX images. Top: extra-high (XL), middle: medium (M), bottom: extra-small (XS).

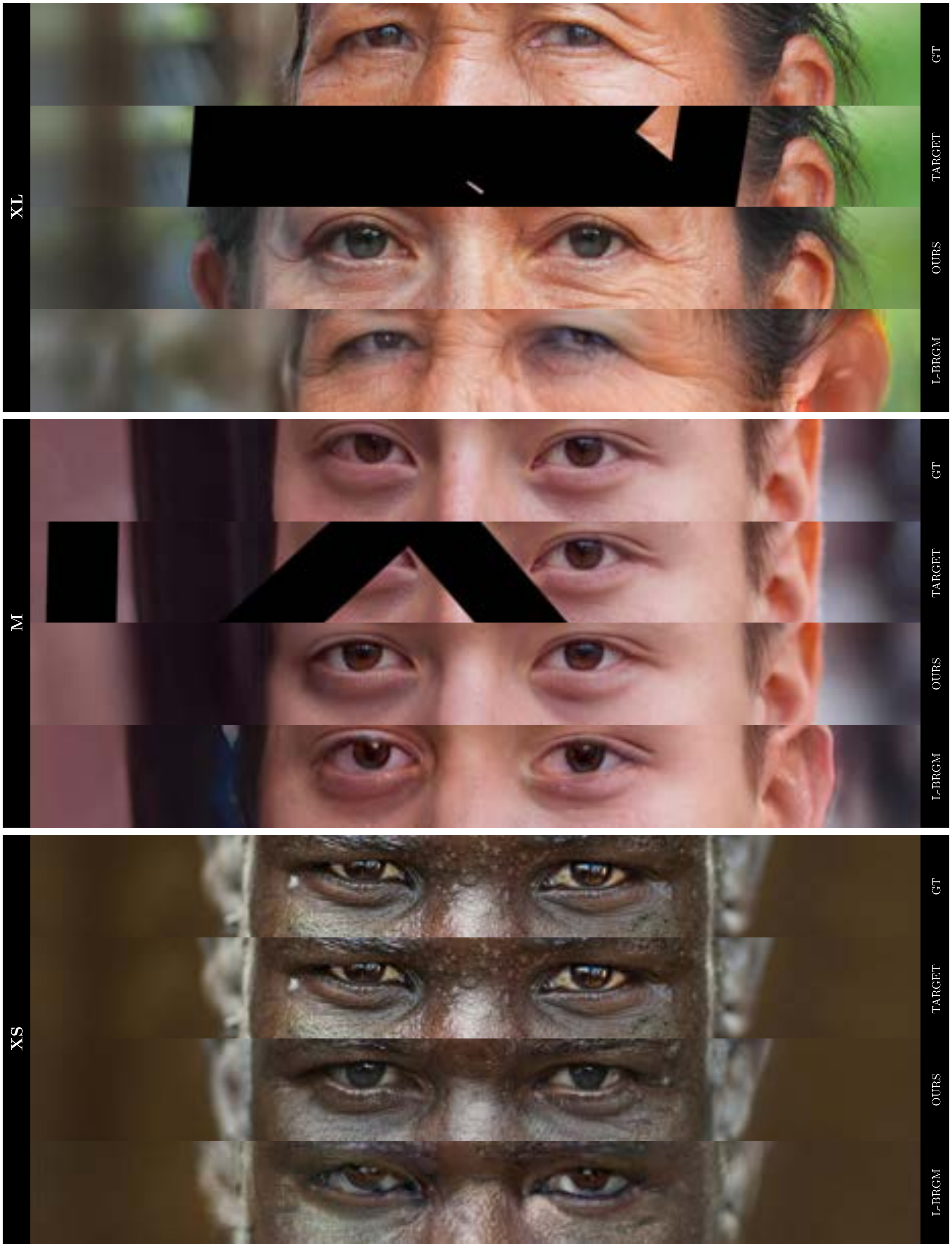


Figure 16. Additional results for inpainting shown in high resolution (images were restored normally then sliced horizontally) on random FFHQX images. Top: extra-high (XL), middle: medium (M), bottom: extra-small (XS).

Figure 17. Additional, high resolution results for three composed tasks (images were restored normally then sliced horizontally) on random FFHQX images. From top to bottom: denoising with deartifacting; deartifacting with inpainting; upsampling with deartifacting.

Figure 18. Additional, high resolution results for three composed tasks (images were restored normally then sliced horizontally) on random FFHQX images. From top to bottom: denoising with inpainting; upsampling with denoising; upsampling with inpainting.

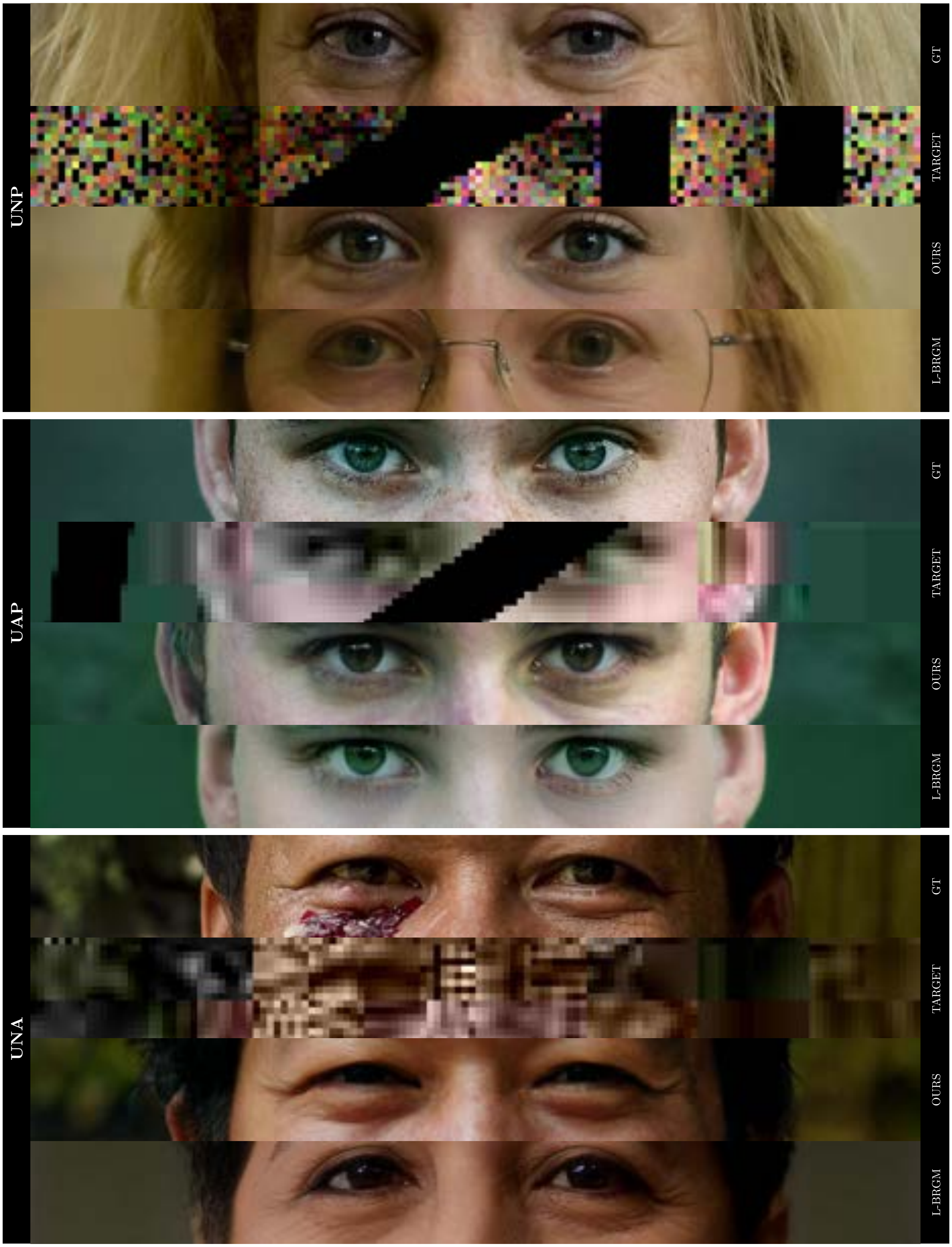


Figure 19. Additional, high resolution results for three composed tasks (images were restored normally then sliced horizontally) on random FFHQX images. From top to bottom: upsampling with denoising and inpainting; upsampling with deartifacting and inpainting; upsampling with denoising and deartifacting.

Figure 20. Additional, high resolution results for two composed tasks (images were restored normally then sliced horizontally) on random FFHQX images. From top to bottom: denoising with deartifacting and inpainting; all four tasks (upsampling, denoising, deartifacting, and inpainting).

Figure 21. Restoration for selected examples (left half: target, right half: prediction) show in high resolution (in reading order: upsampling, denoising, deartifacting, and inpainting).

Figure 22. Full resolution composed restoration of all four degradations (upsampling, denoising, deartifacting, and inpainting) for a selected example (left half: target, right half: prediction).

References

[1] Arthur Conmy, Subhadip Mukherjee, and Carola-Bibiane Schönlieb. Stylegan-induced data-driven regularization for inverse problems. In *Int. Conf. Acous. Sp. Sig. Proc.*, 2022. 1, 12

[2] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 6

[3] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Adv. Neural Inform. Process. Syst.*, 2022. 4

[4] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 1, 6, 12

[5] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013. 1, 12