RAD: Region-Aware Diffusion Models for Image Inpainting

Supplementary Material

S1. Defining Local Noise Schedules

In this section, we define the local noise schedule b_t. The noise schedule in RAD consists of two phases: one for filling the masked region and another for filling the area outside it. With a total diffusion step of T, the first T_1 steps are allocated to the first phase. To define each phase, we define two scalar sequences β_t $(1 \le t \le T_1)$ and β'_t $(1 \le t \le T_2)$ with $0 < \beta_t, \beta'_t < 1$. Let ν $(0 < \nu < 1)$ be the target accumulated noise level which is very close to one. Then, both schedules must satisfy $\prod_{t=1}^{T_1}(1-\beta_t) = \prod_{t=1}^{T_2}(1-\beta'_t) = 1-\nu$. If this is not satisfied, we can easily normalize β_t and β'_t as

$$\beta_t \leftarrow 1-(1-\beta_t)^{g_t}, \quad g_t = \frac{\log(1-\nu)}{\sum_{s=1}^{T_1}\log(1-\beta_s)},$$
$$\beta'_t \leftarrow 1-(1-\beta'_t)^{g'_t}, \quad g'_t = \frac{\log(1-\nu)}{\sum_{s=1}^{T_2}\log(1-\beta'_s)}.$$

Given a mask m, b_t is defined based on the above pixel-wise schedules:

$$b_t = \begin{cases} \beta_t m & \text{if } 1 \le t \le T_1, \\ \beta'_{t-T_1}(1-m) & \text{otherwise,} \end{cases} \tag{1}$$

which satisfies $\bar{b}_{T_1} = \nu m$ and $\bar{b}_T = \nu \mathbf{1}$.

If one wishes to use a non-binary mask, *i.e.*, one that contains ambiguous regions $(0 < m_i < 1)$, the following equation can be used instead:

$$b_{t,i} = \begin{cases} 1-(1-\nu m_i)^{\gamma_t} & \text{if } 1 \le t \le T_1, \\ 1-\left[\frac{(1-\nu)}{(1-\nu m_i)}\right]^{\gamma'_{t-T_1}} & \text{otherwise,} \end{cases}$$

where $\gamma_t = \frac{\log(1-\beta_t)}{\log(1-\nu)}$ and $\gamma'_t = \frac{\log(1-\beta'_t)}{\log(1-\nu)}$. This equation reduces to (1) if m is binary.

S2. Handling Singularities in Training Loss

The learning objective of RAD (and DDPM) is given as

$$L = \mathbb{E}_q\Bigg[\sum_{t>1} D_{KL}(q(x_{t-1}|x_t,x_0) \,\|\, p_\theta(x_{t-1}|x_t)) - \log p_\theta(x_0|x_1)\Bigg]. \tag{2}$$

This trains $p_\theta(x_{t-1}|x_t)$ to match the target $q(x_{t-1}|x_t,x_0)$, except for $t=1$ where the negative log-likelihood of $p_\theta(x_0|x_1)$ is directly optimized. A troubling fact is that this loss function is potentially ill-conditioned if either of q and p_θ is singular (*e.g.*, a degenerate distribution), inducing an infinite loss.

For DDPM, this does not happen for $t > 1$ in theory because $q(x_{t-1}|x_t,x_0)$ has a non-zero variance r_t by definition, and the same holds for $p_\theta(x_{t-1}|x_t)$ of which the variance s_t is set identical to either b_t or r_t. However, it gets trickier for $t=1$. In this case, r_1 becomes zero because $q(x_0|x_1,x_0)$ is degenerate (*i.e.*, evaluating the probability of x_0 when itself is given as a condition). Even though $q(x_0|x_1,x_0)$ does not appear directly in the above loss function, it can affect $p_\theta(x_0|x_1)$ if we set s_1 to r_1.[1]

In [6], the above loss function is replaced with the following simplified loss, eliminating the above problem:

$$L = \sum_t \mathbb{E}_q\left[\|\epsilon_t - \epsilon_\theta(x_t,t)\|^2\right]. \tag{3}$$

In more advanced models, such as iDDPM [13] and ADM [5], however, (2) is revived and is used in conjunction with (3) to enhance generation quality, by learning s_t based on it. In this case, s_t is defined as a linear combination of b_t and r_t, potentially bringing back the above problem for $t=1$. This is usually fixed by some heuristics in implementations, *e.g.*, clipping the minimum variance or replacing the variance with that in a nearby step.

If we formulate RAD based on DDPM, then the above problem does not matter because we can also use (3). However, we also extend RAD to the above advanced models, as explained in the paper. Hence, resolving this issue is also important for RAD. In RAD, this problem gets a little more complicated because the diffusion process is now divided into two phases. This can be viewed in the following way: The forward process finishes early (at timestep T_1) for the pixels in the mask region, while that is delayed to the second phase for the pixels outside the mask. Accordingly, the above problem appears at different timesteps for different pixels.

Specifically, $r_{t,i}$ can be derived as $r_{t,i} = b_{t,i}\bar{b}_{t-1,i}/\bar{b}_{t,i}$ based on the Bayes rule [6]. Here, when $\bar{b}_{t,i}$ is zero, one can follow an alternate derivation to have $r_{t,i} = 0$, avoiding the explicit division. Including this case, $r_{t,i}$ becomes zero when either of $b_{t,i}$, $\bar{b}_{t-1,i}$, and $\bar{b}_{t,i}$ is zero. These correspond to the following three cases:

1. *Only $b_{t,i}$ is zero:* There is already some accumulated noise in the i-th pixel, but it stopped to add more at t.

[1] Here, we ignored some details, *i.e.*, a discretized Gaussian model is separately proposed for $p_\theta(x_0|x_1)$ in [6]. However, the problem still persists.

2. *Only $\bar{b}_{t-1,i}$ is zero:* There has been no noise so far, and now it has started to add one since t.
3. *All three terms are zero:* There has been no noise so far, and it still does at t.

Note that there are no other cases because $1 - \bar{b}_{t,i} = (1 - b_{t,i})(1 - \bar{b}_{t-1,i})$ holds. Among these, the first and third cases are trivial cases. In both cases, the pixel stays the same in the forward process for the current step t. It is also a good idea to make the corresponding reverse process leave the pixel unchanged. This means that there is no need to learn $s_{t,i}$ for the pixel, which is surely zero. Hence, the corresponding losses can be excluded from (2).

On the other hand, a problem arises when the second case happens. In this case, an appropriate value of $s_{t,i}$ is non-zero because the pixel changes in the forward process even though $r_{t,i}$ is zero. (This is another consequence of $q(x_{t-1}|x_t, x_0)$ being quite different from $q(x_t|x_{t-1})$.) This case actually corresponds to the first step of DDPM, which was explained previously. Accordingly, we find all instances where this happens and apply similar heuristics as in [5, 13]. In reality, this happens on two occasions: (i) pixels in the mask region at $t = 1$, or (ii) pixels outside the mask at $t = T_1 + 1$. The latter case happens because the diffusion process is delayed until the second phase for those pixels, as mentioned earlier. This also means that applying the negative log-likelihood rather than the KL divergence in (2) makes more sense for their individual losses, *i.e.*, $p_\theta(x_{T_1}|x_{T_1+1})$ must be treated similarly to $p_\theta(x_0|x_1)$ in (2).

The above analysis has proven important in our empirical experience, in which the aforementioned problem rendered the training unstable when extending RAD to [5, 13].

S3. Settings for Baseline Methods

We compared our method with various state-of-the-art (SoTA) inpainting techniques. For FFHQ, we referenced results on LaMa, Score-SDE, DDRM, RePaint, and MCG reported in the MCG paper, while for LSUN Bedroom, we evaluated their performance separately on the same validation set for consistency. The hyperparameters of the baseline methods were set according to their respective papers or codebases, as follows:

- **Score-SDE [14]:** The sampling steps were set to 1000.
- **DDRM [8]:** We set $\sigma_y = 0$ to address the noiseless inverse problem. Additionally, we configured $\eta = 0.85$ and $\eta_b = 1$. The number of sampling steps was fixed at 20, utilizing the DDRM sampling procedure.
- **RePaint [12]:** We used pretrained ADM models on FFHQ and LSUN Bedroom. The total diffusion steps were set to 200 and the number of iterations of denoising steps to 10 as the default configurations in [12].
- **MCG [4]:** The sampling steps were set to 1000 as the default configuration in [4].

Figure S1. Unconstrained generation results on FFHQ.

- **DDNM [15]:** We set $\eta = 0.85$ and the sampling steps to 100 as in [15].
- **DeqIR [3]:** The sampling steps were set to 25.

S4. Unconstrained Generation

Here, we provide unconstrained generation results of RAD, i.e., using RAD for full sample generation instead of inpainting. Figure S1 shows the results on the FFHQ dataset. The first and second rows were generated with all-one masks (so all the pixels were generated synchronously in one phase) and random Perlin masks in two phases, respectively. In this figure, we can confirm that RAD is capable of unconstrained generation. Performance-wise, the FID of a pretrained ADM is 28.0, whereas those of one- or two-phase unconstrained generations of RAD are 36.2 and 36.8, respectively. Since the task of RAD is considerably more difficult (handling various inpainting scenarios), it may show lower performance than plain diffusion models in an unconstrained generation. However, RAD still produces plausible results, and it provides state-of-the-art results in its primary goal, *i.e.*, inpainting tasks.

S5. Addtional Results

We provide additional comparison results on CelebA-HQ with more recent works in Table S1. We used the evaluation settings of [10], and the values of other methods were quoted from [10]. Compared with most recent methods, RAD still achieves state-of-the-art performance. In particular, RAD performs significantly better in tasks involving large missing regions, such as "Half," "Completion," and "Expand," while maintaining comparable performance in other cases where performance is already good. In particular, compared to the DDIM+CS [10] that directly learns the target mask, RAD shows an improvement of up to 14 in terms of FID. This demonstrates that the Perlin noise used in RAD training is effective not only in various inpainting scenarios but also in more challenging situations.

To verify the effectiveness of adopting LoRA in RAD, we compared RAD trained with full fine-tuning and LoRA in Figure S2. The solid and dotted lines represent LoRA

Table S1. Quantitative comparison of different methods on CelebA-HQ-256. Lower LPIPS and FID indicate better performance.

Method	Half		Completion		Expand		Thick Line		Medium Line	
	LPIPS	FID	LPIPS	FID	LPIPS	FID	LPIPS	FID	LPIPS	FID
CoModGAN [16]	0.445	37.72	0.406	43.77	0.671	93.48	0.091	5.82	0.105	5.86
LaMa [2]	0.342	33.82	0.315	25.72	0.538	86.21	0.080	5.47	0.077	5.18
CDE [1]	0.344	29.33	0.302	19.07	0.508	71.99	0.079	4.77	0.070	4.33
RePaint [12]	0.435	41.28	0.387	37.96	0.665	92.03	**0.059**	5.08	**0.028**	4.97
MAT [9]	0.331	32.55	0.280	20.63	0.479	82.37	0.080	5.16	0.077	4.95
GLaMa [11]	0.327	30.76	0.289	18.61	0.481	84.01	0.081	5.83	0.080	5.10
FcF [7]	0.305	27.95	0.378	31.14	0.502	73.24	0.086	4.63	0.071	4.42
DDIM+CS [10]	0.272	20.37	0.259	15.33	0.372	39.05	0.079	**4.21**	0.064	**3.58**
RAD (ours)	**0.169**	**10.38**	**0.190**	**11.35**	**0.315**	**25.38**	0.076	5.37	0.049	4.76

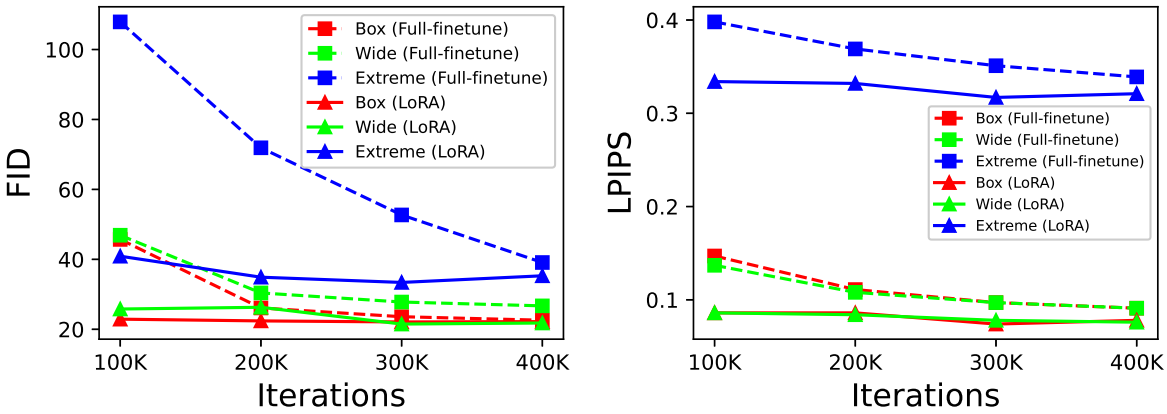

Figure S2. Comparison of fine-tuning methods on FFHQ (batch size 16).

Table S2. RAD with various noise schedules on FFHQ.

Mask	Box		Extreme		Wide	
	FID	LPIPS	FID	LPIPS	FID	LPIPS
Random	125.3	0.232	234.2	0.576	96.0	0.227
Box	**21.9**	**0.071**	206.4	0.859	37.7	0.131
Extreme	170.4	0.409	**27.8**	**0.311**	153.9	0.362
Wide	22.3	0.076	35.2	0.389	**21.4**	**0.072**
Perlin	22.1	0.074	33.4	0.317	21.5	0.078

Table S3. RAD with various noise schedules on CelebA-HQ.

Mask	Half		Completion		Expand		Thick Line		Medium Line	
	FID	LPIPS	FID	LPIPS	FID	LPIPS	FID	LPIPS	FID	LPIPS
Wide	26.5	0.177	28.3	0.200	40.2	0.337	25.0	0.103	23.0	0.063
Perlin	**23.2**	**0.167**	**24.2**	**0.190**	**38.0**	**0.311**	**20.2**	**0.123**	**20.1**	**0.057**

and full fine-tuning, respectively. Here, LoRA converges significantly faster than full fine-tuning across all inpainting scenarios. As shown in the figure, we did not find any performance degradation in LoRA.

In addition, we tested alternative mask types for training RAD, instead of Perlin masks, which are shown in Table S2. The random mask is generated by randomly setting each pixel to zero or one, while the box, extreme, and wide masks are the same as those used in the original RAD evaluation. As expected, the random mask shows poor performance across all cases, as it is far from the actual inpainting scenarios. On the other hand, the box and extreme masks achieve the best performance in their respective evaluation settings, but do not generalize well to other masks. Interestingly, the wide mask also performs well in the box and extreme cases, which we attribute to the fact that the wide mask explicitly includes box-shaped regions, so it somewhat includes the box and extreme masks.

To verify how wide and Perlin masks perform in more general inpainting scenarios, we evaluated them on the CelebA-HQ dataset under the evaluation settings of [10]. In Table S3, we can confirm that the model trained with Perlin noise achieves better performance than the one trained with wide masks for all scenarios.

We provide additional qualitative comparison results of RAD for various datasets and mask types. For FFHQ,

Table S4. Ablation study of RAD on LSUN Bedroom (Cfg. 1: pretrained ADM with RAD reverse steps, Cfg. 2: without spatial noise embedding).

Method	Box		Extreme		Wide	
	FID	LPIPS	FID	LPIPS	FID	LPIPS
Cfg. 1	41.9	0.193	102.4	0.478	40.9	0.175
Cfg. 2	22.2	0.154	25.2	0.405	23.2	0.130
RAD (ours)	**19.2**	**0.131**	**21.6**	**0.399**	**20.8**	**0.107**

the results for box, extreme, and wide masks are demonstrated in Figures S3 to S5, respectively. In general, RePaint tends to produce discontinuous inpainting results, failing to maintain coherence across the boundaries. MCG generally shows good performance; however, it occasionally produces awkward samples such as the blurry hair in the last row of Figure S3, the strange teeth, chin, and cap in the third row of Figure S5, and the asymmetric beard in the fifth rows of Figure S5. For the LSUN Bedroom dataset, similar comparisons are provided in Figures S6 to S8. In Figure S6, the second and fifth rows show that both RePaint and MCG fail to accurately fill in the patterns on the covers and the upper parts of the curtains, respectively, resulting in inconsistencies and a lack of detail in the reconstructed regions. In the first row of Figure S7, RePaint fails to generate the wall region, leaving it incomplete, while MCG produces unrealistic results above the sofa. In the remaining rows, MCG tends to duplicate patterns from the given image, compromising the overall plausibility. Figures S9 to S11 show the corresponding results for ImageNet. In Figure S9, the second, fifth, and last rows reveal that both RePaint and MCG fail to generate the body and head of the animals. Additionally, the first, third, and fourth rows demonstrate that both methods produce discontinuous or unrealistic results. Again, RePaint fails to maintain coherence across the boundaries, while MCG tends to copy patterns from the given image, leading to repetitive and unrealistic outputs.

Additionally, we produced diverse samples to verify the generative capabilities of RAD. For FFHQ, the results for box, extreme, and wide masks are shown in Figures S12 to S14, respectively. Likewise, those for LSUN Bedroom and ImageNet are shown in Figures S15 to S17 and Figures S18 to S20, respectively. All the results demonstrate that RAD can generate diverse outcomes from the same input image and mask.

The ablation study presented in the paper was also performed on the LSUN Bedroom dataset. The results are shown in Table S4, further highlighting the significance of the proposed techniques (*i.e.*, training with spatially variant noise and using spatial noise embedding) in inpainting tasks.

References

[1] Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606*, 2021. 3

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018. 3

[3] Jiezhang Cao, Yue Shi, Kai Zhang, Yulun Zhang, Radu Timofte, and Luc Van Gool. Deep equilibrium diffusion restoration with parallel sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2824–2834, 2024. 2

[4] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. 2022. 2

[5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 1, 2

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[7] Jitesh Jain, Yuqian Zhou, Ning Yu, and Humphrey Shi. Keys to better image inpainting: Structure and texture go hand in hand. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 208–217, 2023. 3

[8] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in Neural Information Processing Systems*, 35:23593–23606, 2022. 2

[9] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10758–10768, 2022. 3

[10] Hui Lu et al. Compensation sampling for improved convergence in diffusion models. In *ECCV*, 2024. 2, 3

[11] Zeyu Lu, Junjun Jiang, Junqin Huang, Gang Wu, and Xianming Liu. Glama: Joint spatial and frequency loss for general image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1301–1310, 2022. 3

[12] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022. 2, 3

[13] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 1, 2

[14] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2

[15] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. *The Eleventh International Conference on Learning Representations*, 2023. 2

[16] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2021. 3

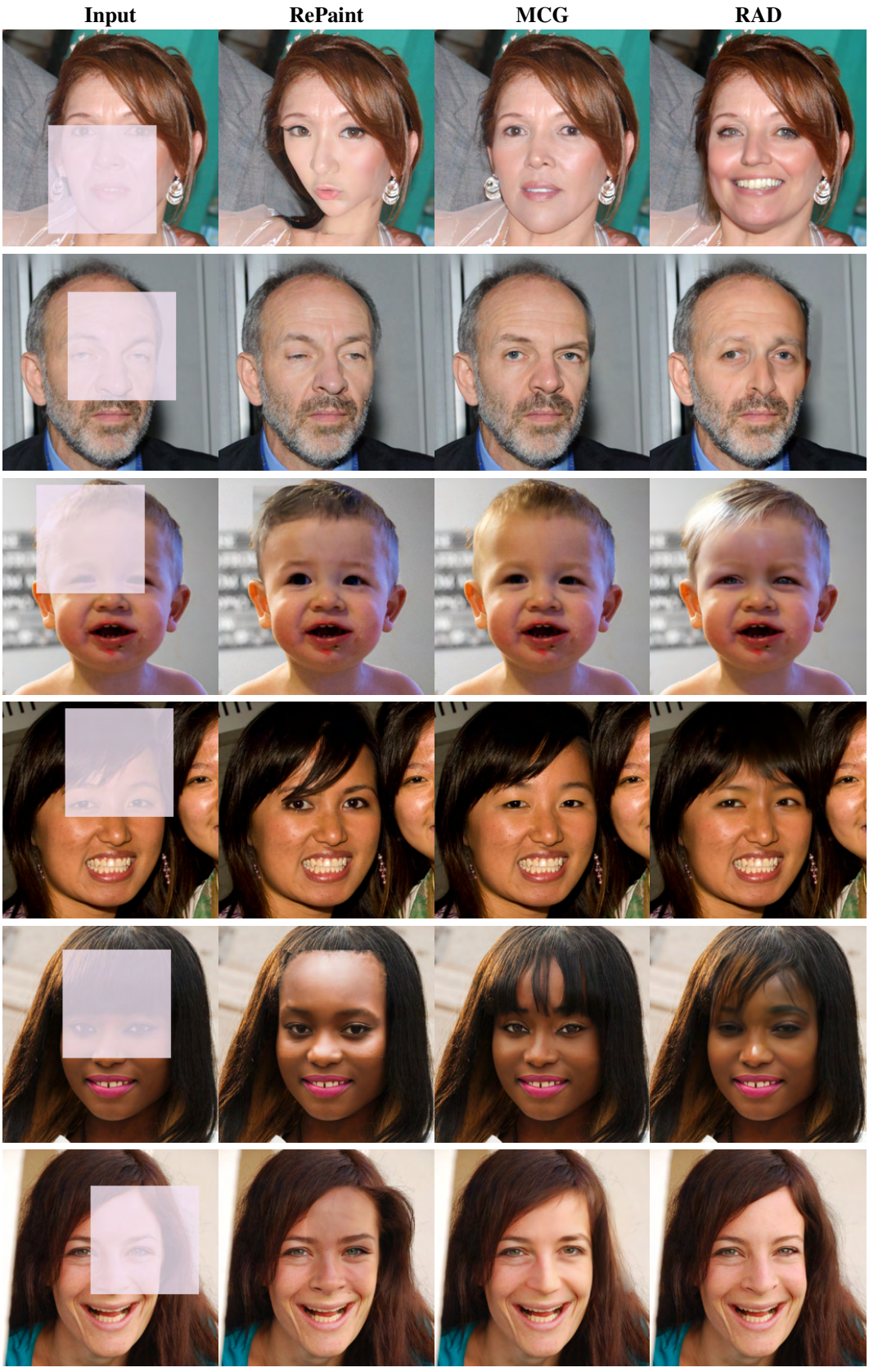

Figure S3. Qualitative comparisons against state-of-the-art inpainting methods on FFHQ for box masks (1st column: Input, 2nd column: RePaint, 3rd column: MCG, 4th column: RAD (ours)).

Figure S4. Qualitative comparisons against state-of-the-art inpainting methods on FFHQ for extreme masks (1st column: Input, 2nd column: RePaint, 3rd column: MCG, 4th column: RAD (ours)).

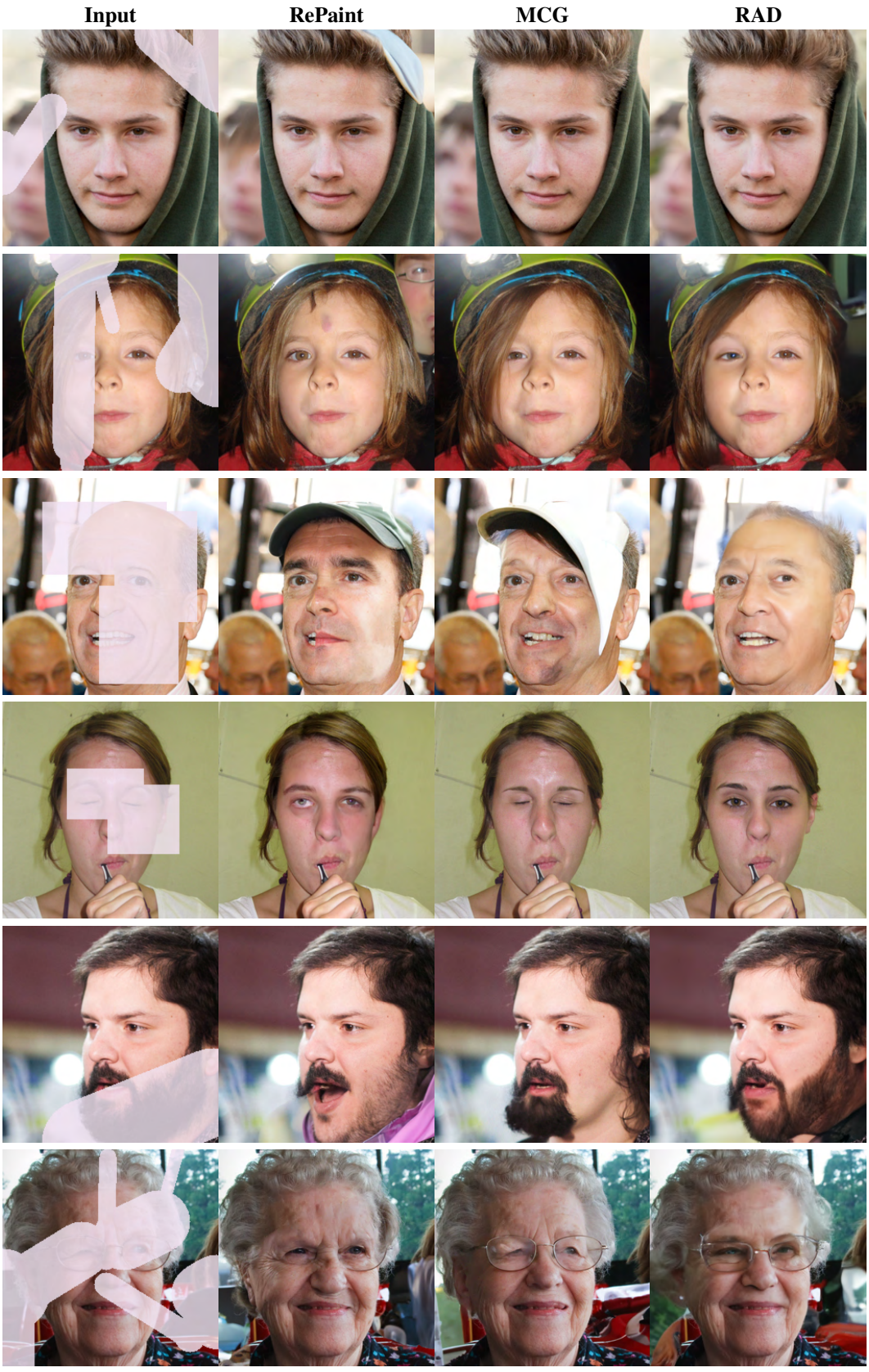

Figure S5. Qualitative comparisons against state-of-the-art inpainting methods on FFHQ for wide masks (1st column: Input, 2nd column: RePaint, 3rd column: MCG, 4th column: RAD (ours)).

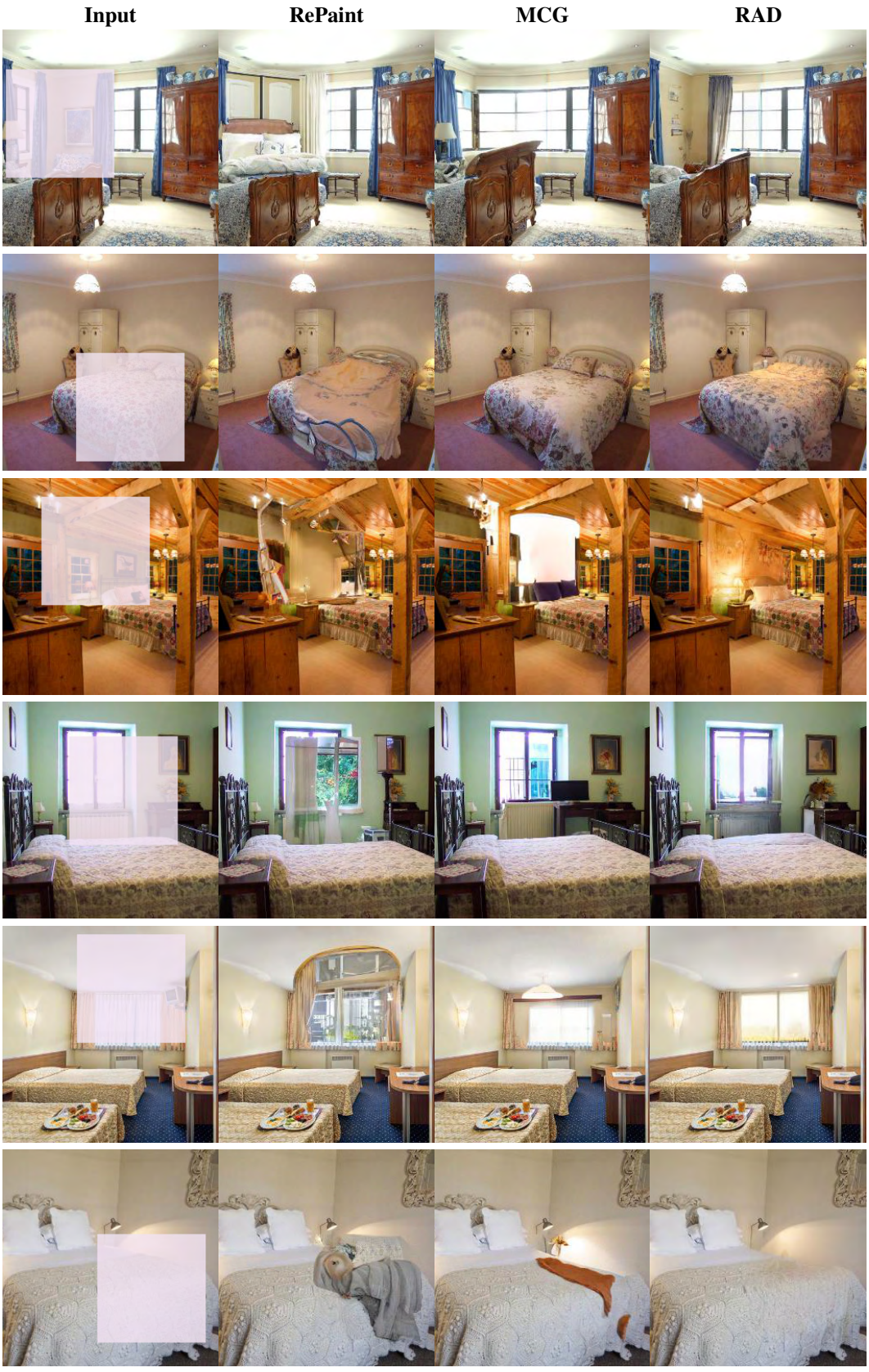

Figure S6. Qualitative comparisons against state-of-the-art inpainting methods on LSUN Bedroom for box masks (1st column: Input, 2nd column: RePaint, 3rd column: MCG, 4th column: RAD (ours)).

Figure S7. Qualitative comparisons against state-of-the-art inpainting methods on LSUN Bedroom for extreme masks (1st column: Input, 2nd column: RePaint, 3rd column: MCG, 4th column: RAD (ours)).

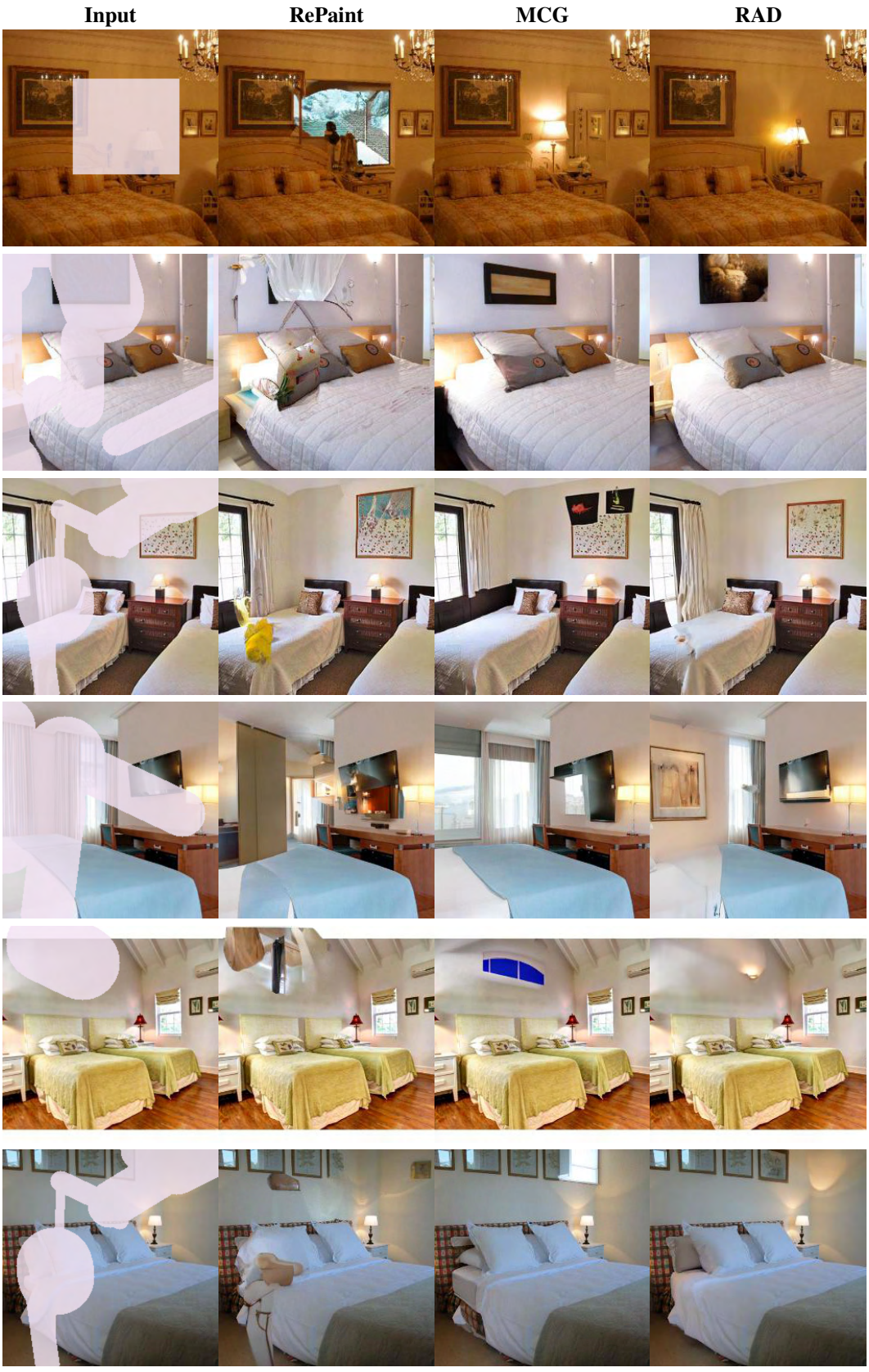

Figure S8. Qualitative comparisons against state-of-the-art inpainting methods on LSUN Bedroom for wide masks (1st column: Input, 2nd column: RePaint, 3rd column: MCG, 4th column: RAD (ours)).

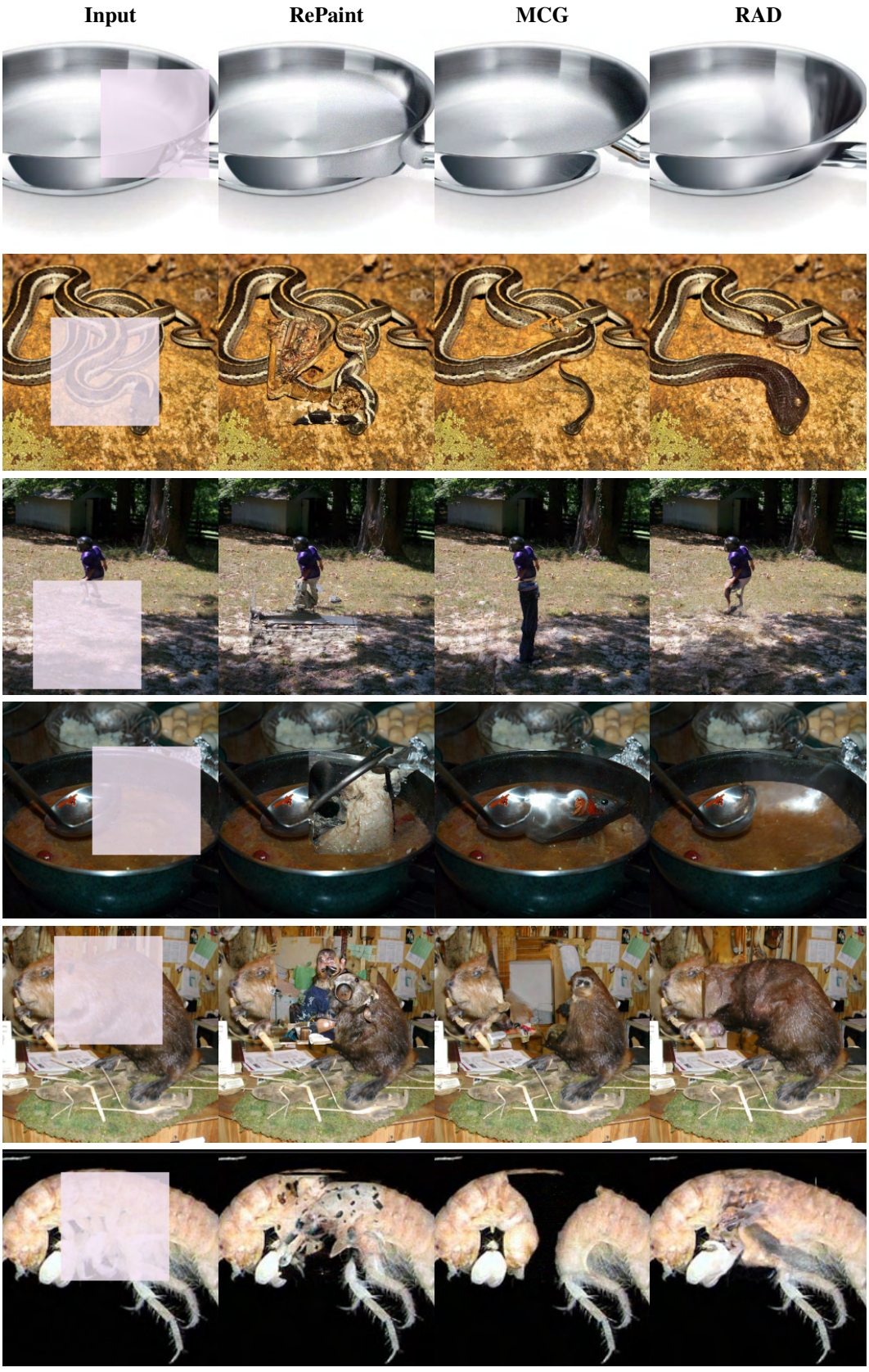

Figure S9. Qualitative comparisons against state-of-the-art inpainting methods on ImageNet for box masks (1st column: Input, 2nd column: RePaint, 3rd column: MCG, 4th column: RAD (ours)).

Figure S10. Qualitative comparisons against state-of-the-art inpainting methods on ImageNet for extreme masks (1st column: Input, 2nd column: RePaint, 3rd column: MCG, 4th column: RAD (ours)).

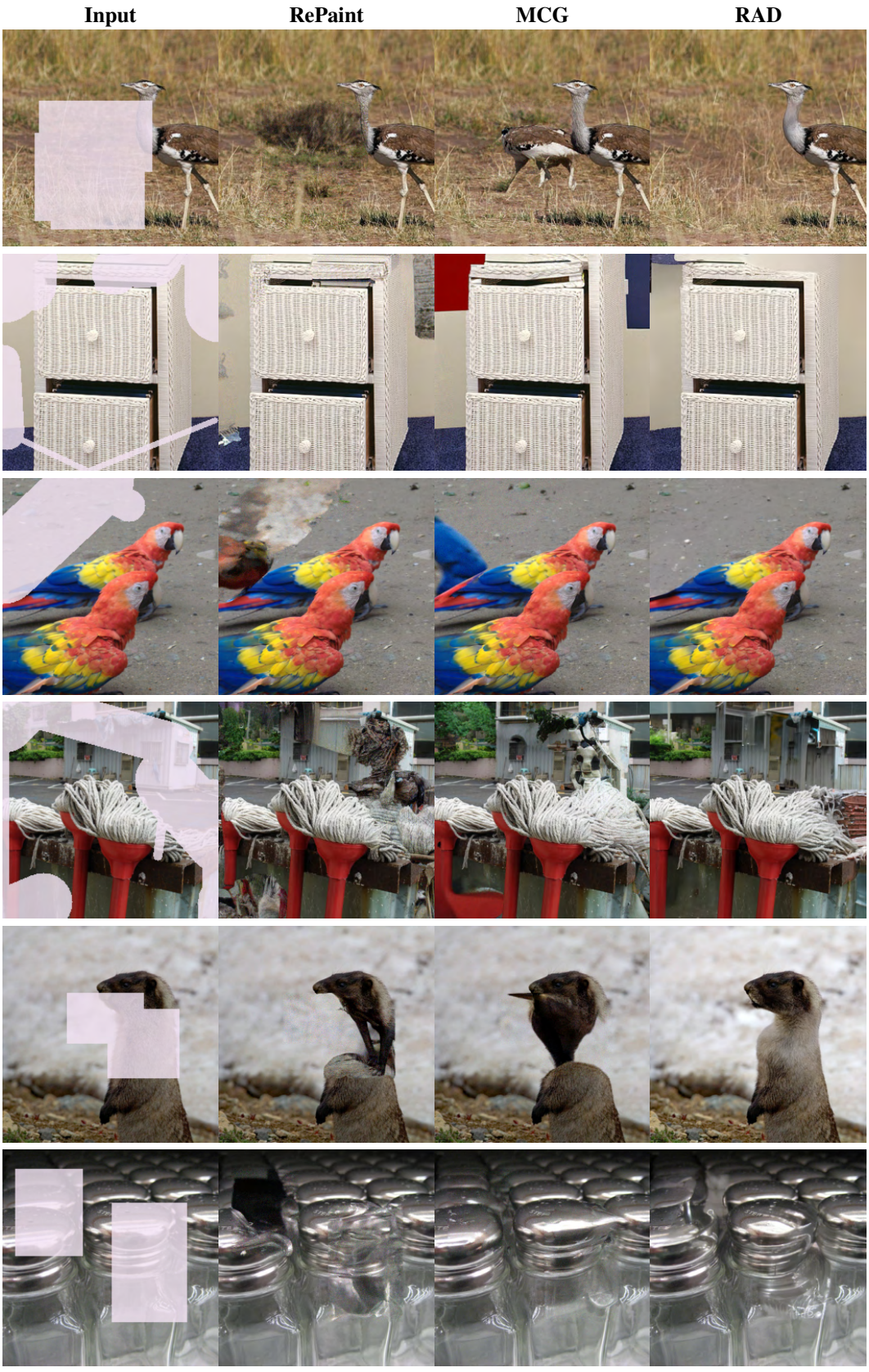

Figure S11. Qualitative comparisons against state-of-the-art inpainting methods on ImageNet for wide mask (1st column: Input, 2nd column: RePaint, 3rd column: MCG, 4th column: RAD (ours)).

Figure S12. Example results of RAD on FFHQ for box masks (1st column: Input, 2nd/3rd/4th/5th columns: inpainting results).

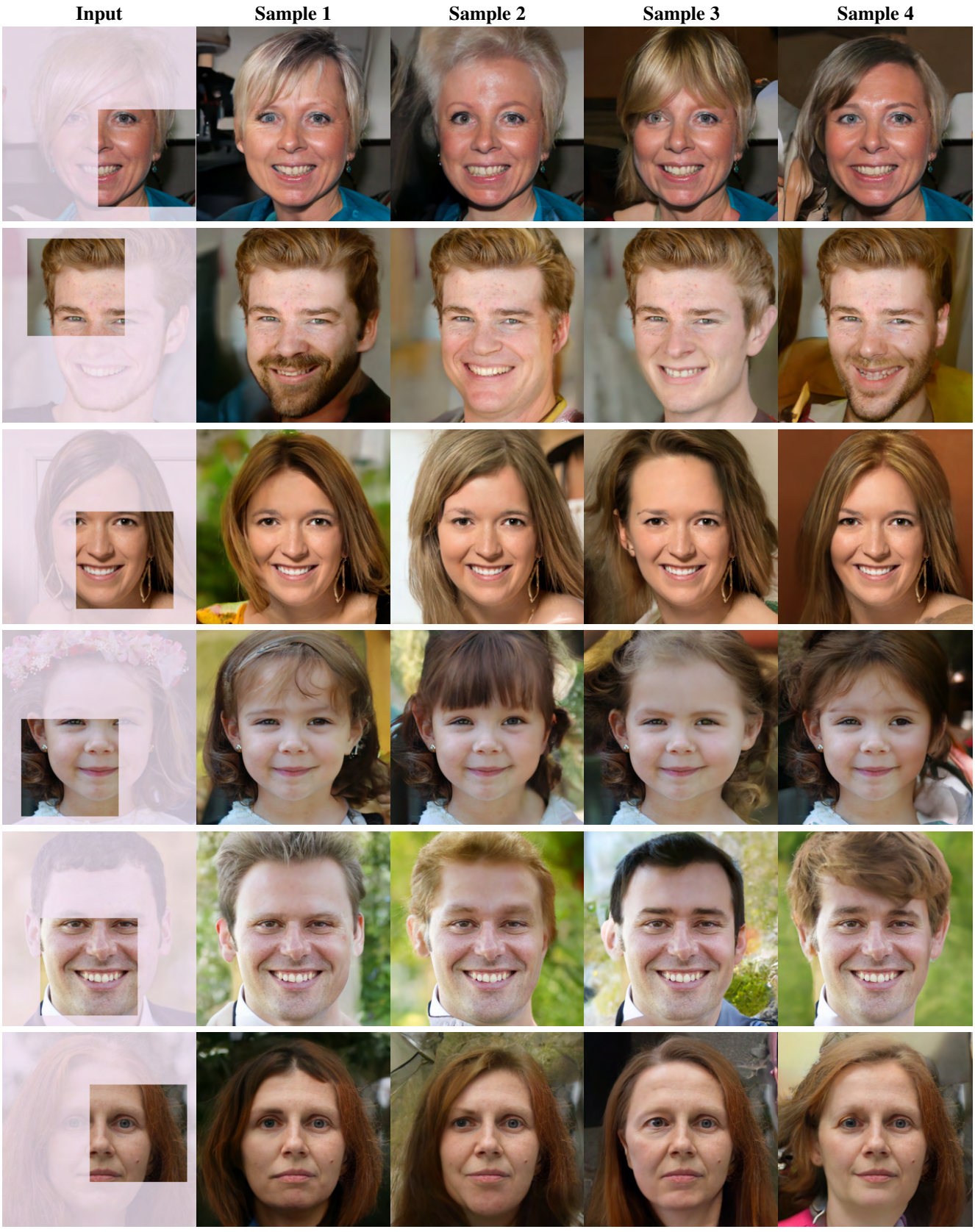

Figure S13. Example results of RAD on FFHQ for extreme masks (1st column: Input, 2nd/3rd/4th/5th columns: inpainting results).

Figure S14. Example results of RAD on FFHQ for wide masks (1st column: Input, 2nd/3rd/4th/5th columns: inpainting results).

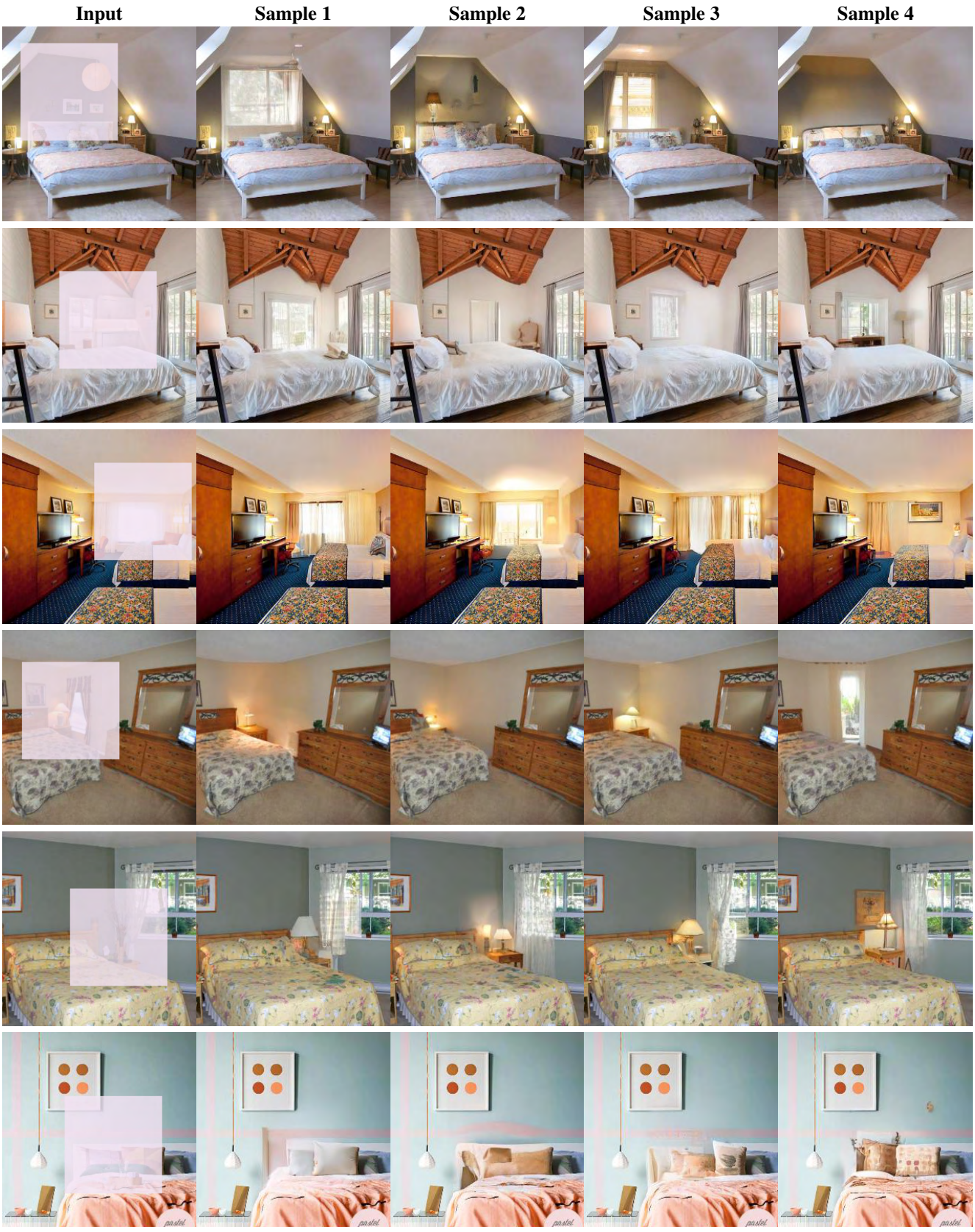

Figure S15. Example results of RAD on LSUN Bedroom for box masks (1st column: Input, 2nd/3rd/4th/5th columns: inpainting results).

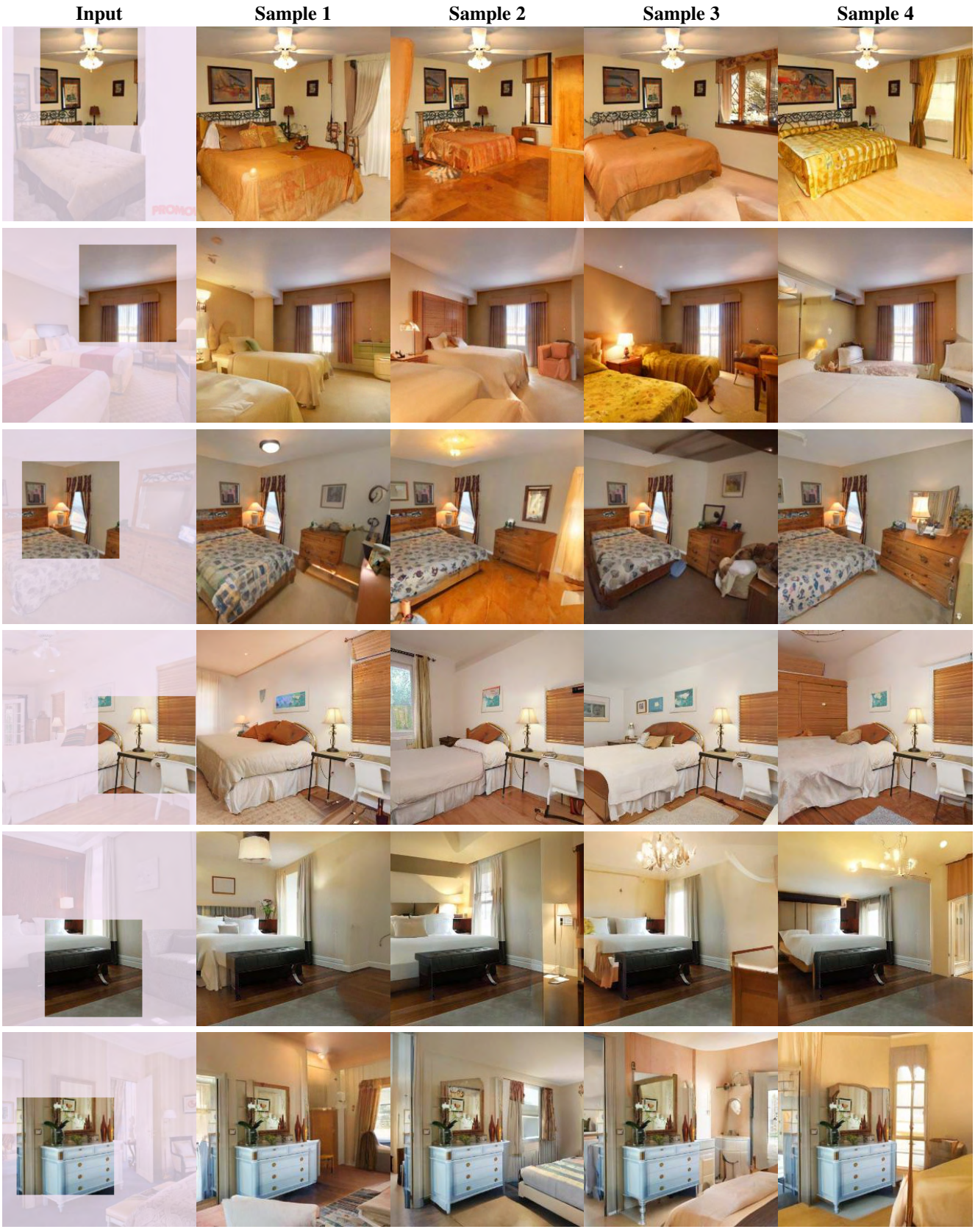

Figure S16. Example results of RAD on LSUN Bedroom for extreme masks (1st column: Input, 2nd/3rd/4th/5th columns: inpainting results).

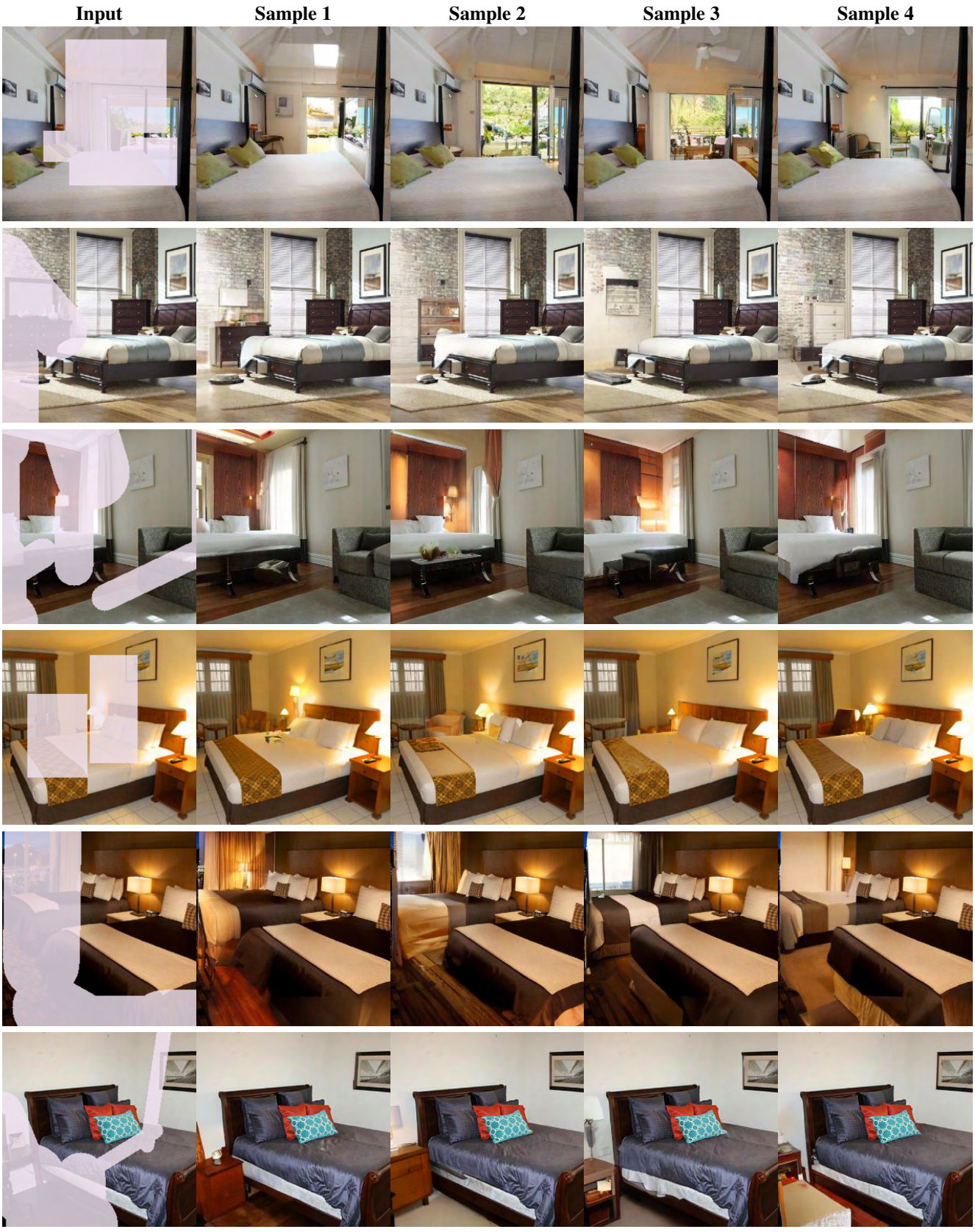

Figure S17. Example results of RAD on LSUN Bedroom for wide masks (1st column: Input, 2nd/3rd/4th/5th columns: inpainting results).

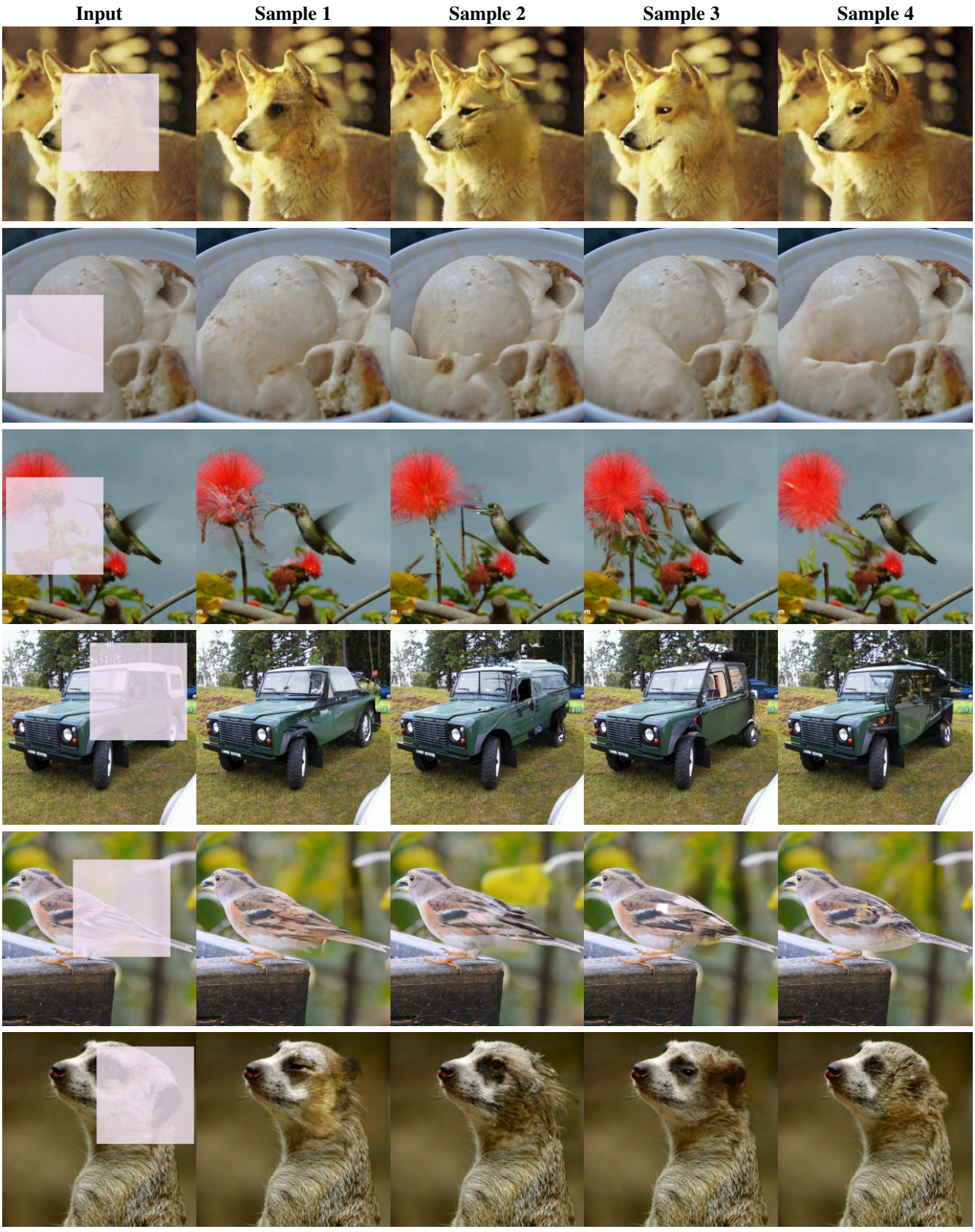

Figure S18. Example results of RAD on LSUN Bedroom for box masks (1st column: Input, 2nd/3rd/4th/5th columns: inpainting results).

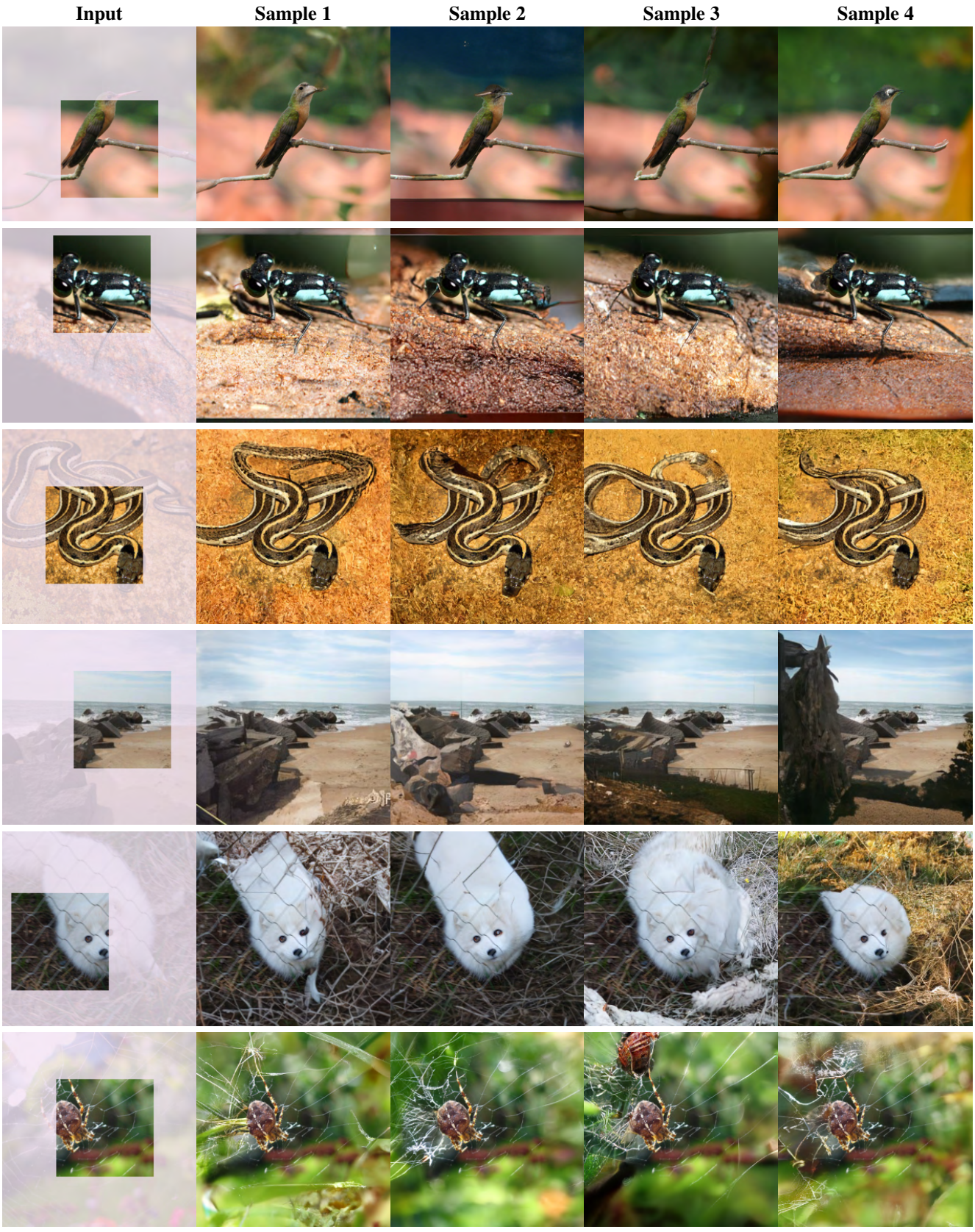

Figure S19. Example results of RAD on LSUN Bedroom for extreme masks (1st column: Input, 2nd/3rd/4th/5th columns: inpainting results).

Figure S20. Example results of RAD on LSUN Bedroom for wide masks (1st column: Input, 2nd/3rd/4th/5th columns: inpainting results).