Denoising for Monte Carlo Renderings

Bing Xu 徐冰
2020.03.19

Contents

Background Recap

camera info, lighting, geometries, textures

Photorealistic Rendering

$$c = \int_{\Omega} f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}$$

[Scene from Kujiale]

Monte Carlo Path Tracing

- Physically based
- Very general: Monte Carlo estimators help to get rid of the high dimensionality of the problem
- Convergence is guaranteed

- Disadvantages:
  - Slow convergence: variance ~1/sqrt(N)
  - sparse sampling => noise

How to reduce variances within time limits

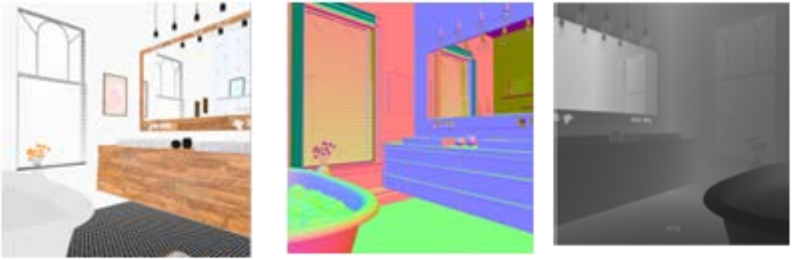
- Sampling
    - Importance sampling
    - Adaptive sampling
    - Various sampling operators….

- Reconstruction (balance between bias & variance)
    - A prior methods: Analyze light transport equations for individual samples, reconstruction filters based on analysis. [Zwicker et.al. 2015]
    - A posterior methods: Ignorant of light transport effects, reconstruction based on empirical statistics.

- Others
    - Control variates
    - MCMC

Primary focus

- “A posterior” method [Zwicker et al. 2015]
- Low sample counts (4spp, 16spp, 32spp ..)
- Guided by per-pixel auxiliary feature buffers (albedo , normal, depth..)
  - Much cheaper!
  - Contain rich information



- CNN based - possible to involve much larger pixel neighbourhoods while improving speed.

Rendered image with 4spp
(MC path tracing)

Keep sampling to convergence
hours/days

Noisy free image

Image-space denoising
seconds

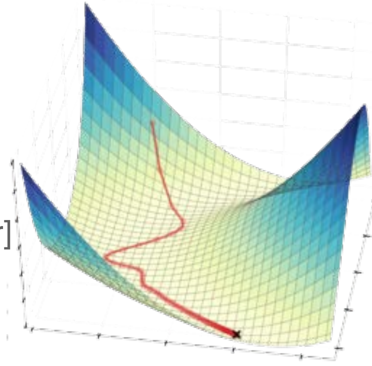
Sample rays for each pixel

Adversarial Monte Carlo Denoising with Conditioned Auxiliary Feature Modulation

BING XU, KooLab, Kujiale, China
JUNFEI ZHANG, KooLab, Kujiale, China
RUI WANG, State Key Laboratory of CAD & CG, Zhejiang University, China
KUN XU, BNRist, Department of Computer Science and Technology, Tsinghua University, China
YONG-LIANG YANG, University of Bath, UK
CHUAN LI, Lambda Labs Inc, USA
RUI TANG, KooLab, Kujiale, China

Motivation & Contribution

Motivation 1 : Loss automation



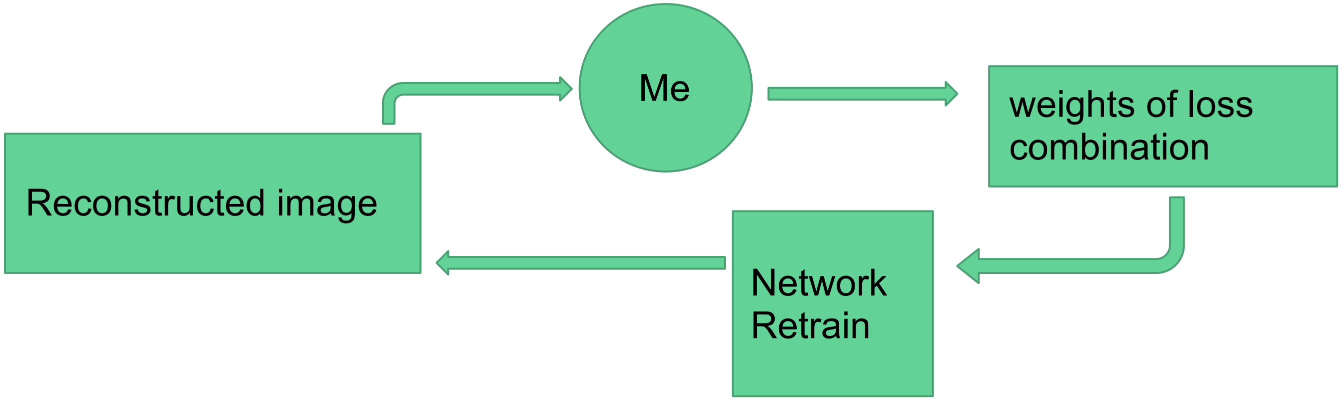
[Interactive Reconstruction of Monte Carlo Image Sequences using a Recurrent Denoising Autoencoder]
Loss function = spatial loss*a + gradient loss*b + temporal loss*c

Original a, b, c

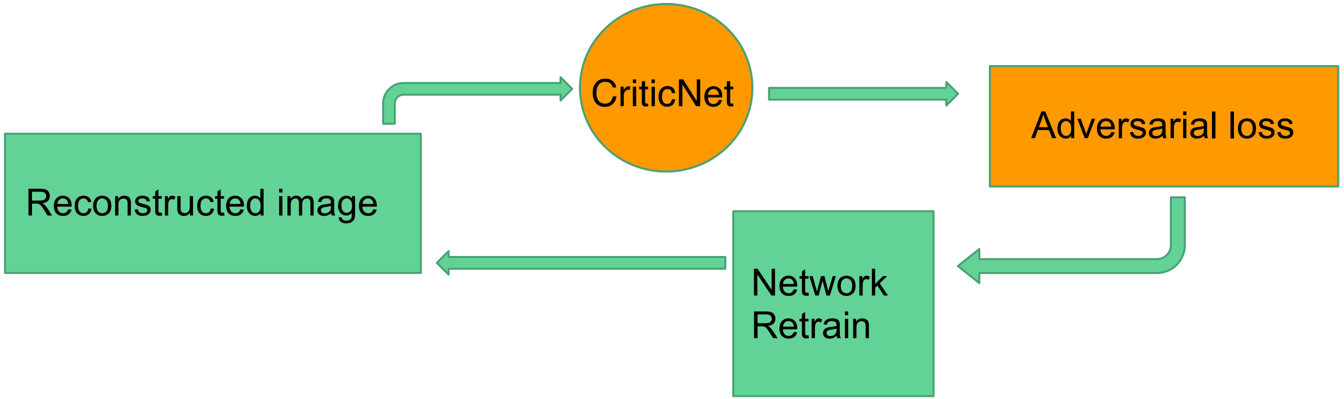
Use larger b at the begining

Better results for high frequency area

Motivation 1 : Loss automation



Motivation 1 : Loss automation



Visual perceptual quality

Lower pixel-wise loss (mostly used) != Better visual perceptual quality

Ideal case:

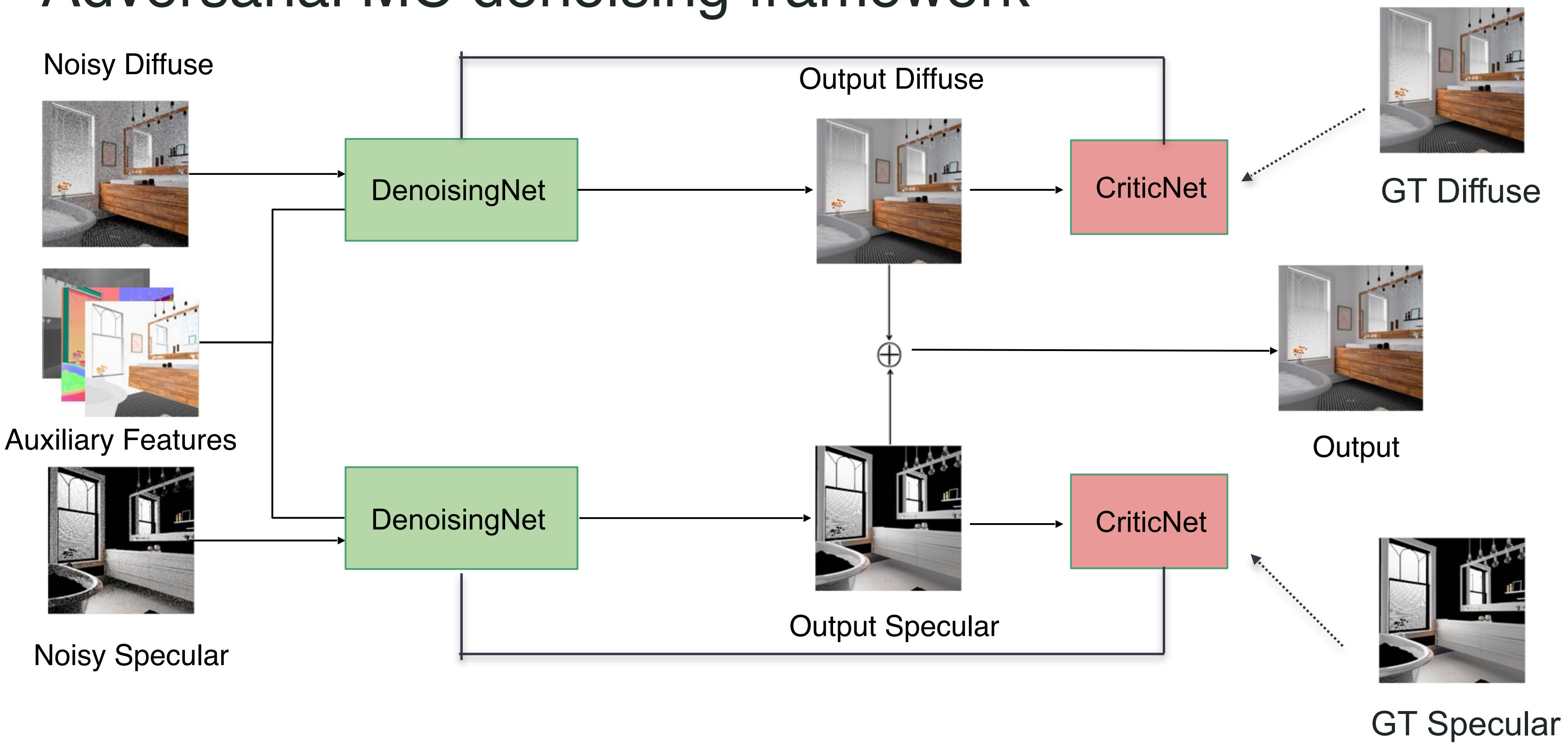
A differentiable metric which naturally reflects human visual system.

Reality:

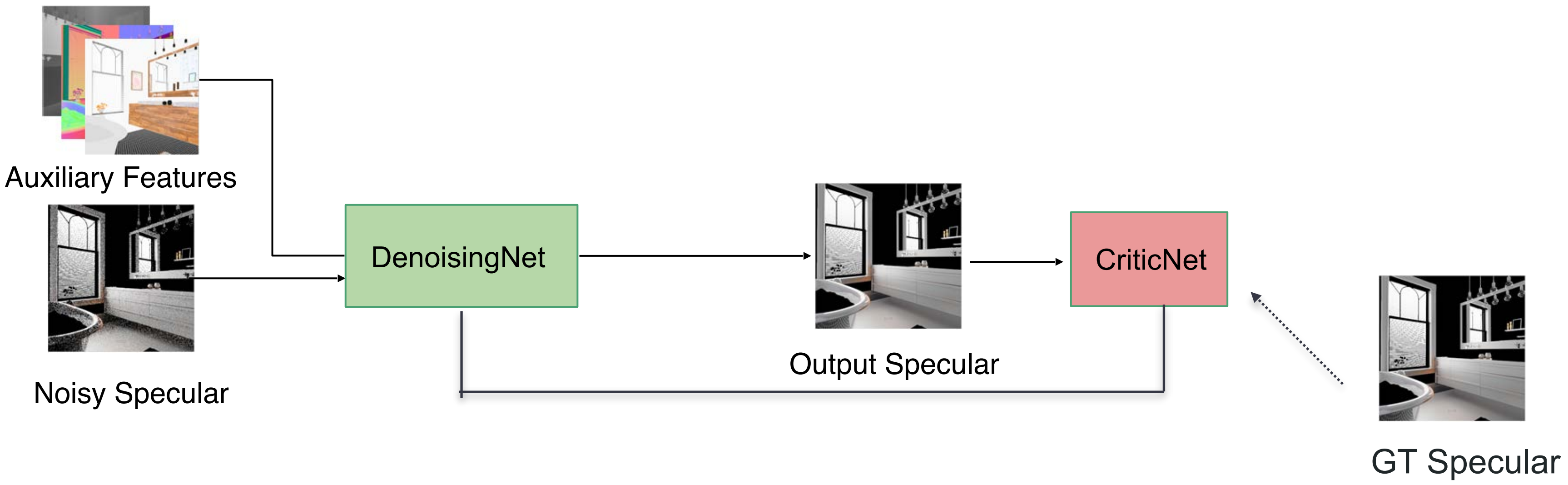
No direct definition or knowledge of the data distribution

Then we can take advantage of implicit models.

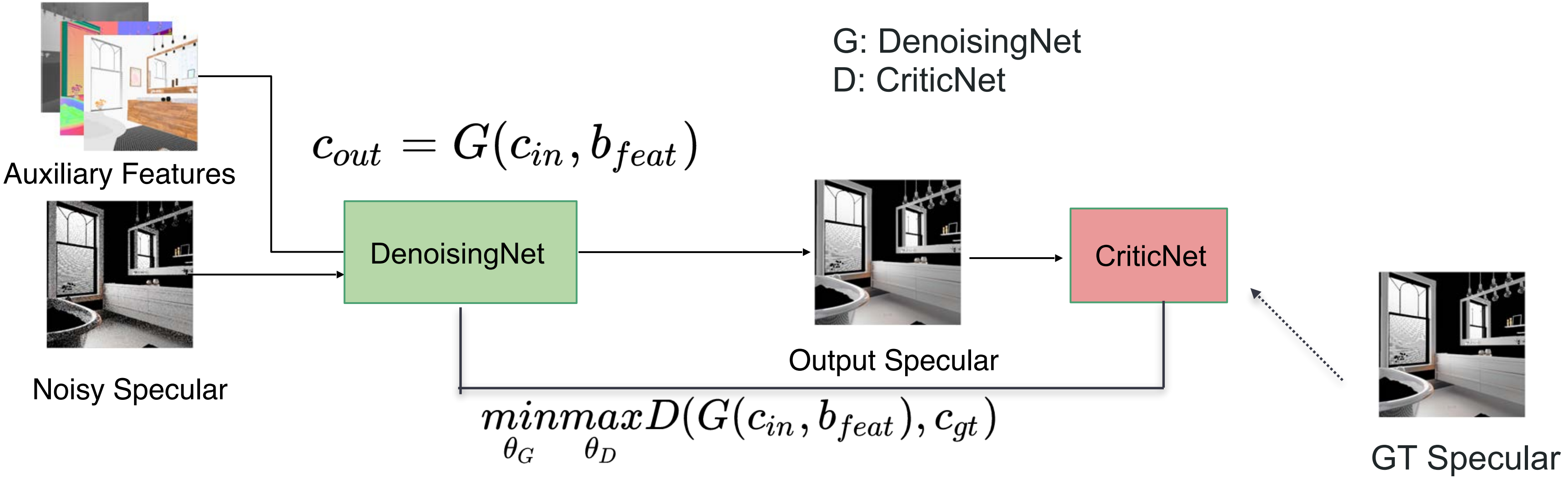
Adversarial MC denoising framework



Adversarial MC denoising framework



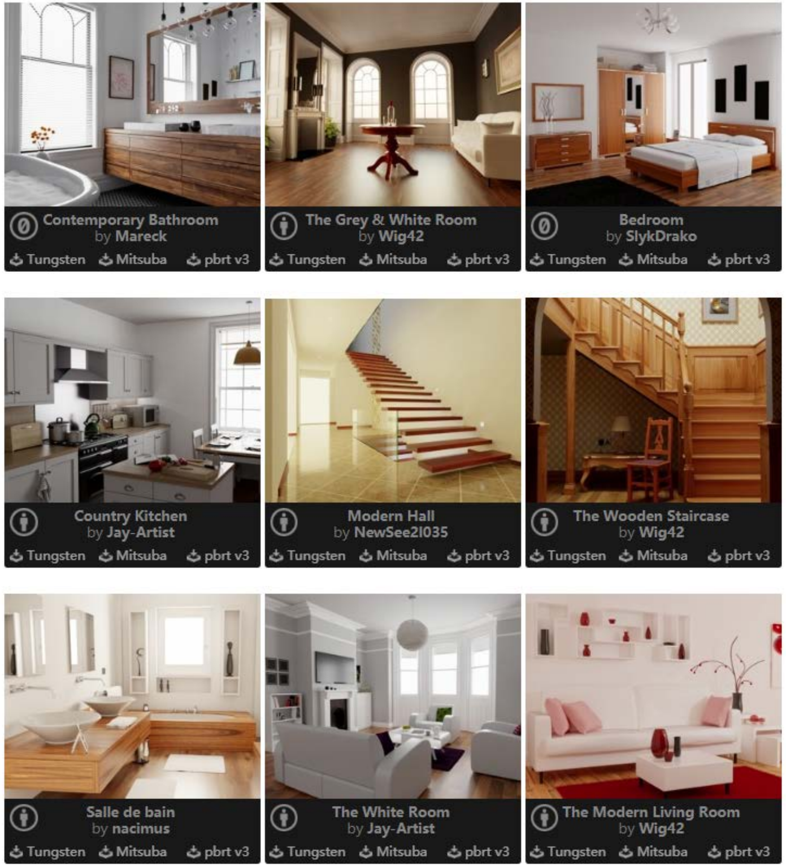
Adversarial MC denoising framework



Training Details & Datasets

- WGAN-GP and auxiliary features help stabilize GAN's training.
- Datasets

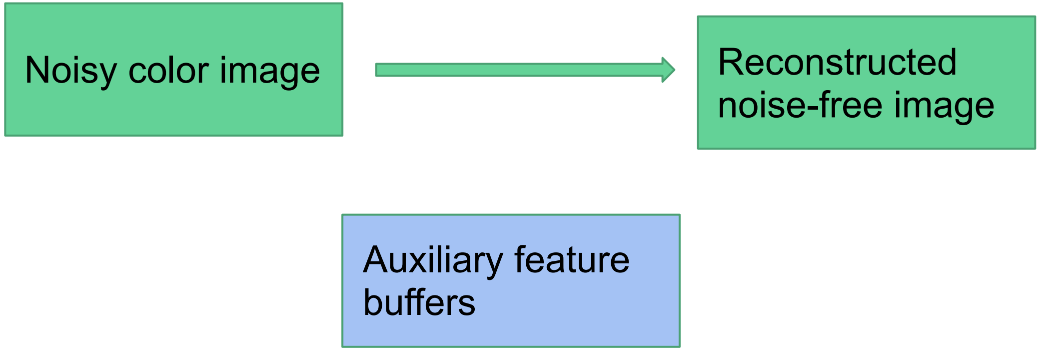
KJL indoor scenes by FF Renderer



Tungsten scenes by **Benedikt Bitterli** https://benedikt-bitterli.me/resources/ released by Disney

Motivation 2: How to use the auxiliary features more wisely?

Image-space denoising: $$\hat{c}_p = \frac{1}{C_p} \sum_{q \in \mathcal{N}_p} c_q\, w(p,q),$$



Motivation 2: How to use the auxiliary features more wisely?

Image-space denoising: $$\hat{c}_p = \frac{1}{C_p} \sum_{q \in \mathcal{N}_p} c_q \, w(p,q),$$

Noisy color image $\longrightarrow$ Reconstructed noise-free image

Conditioning On

Auxiliary feature buffers

Motivation 2: How to use the auxiliary features more wisely?

Image-space denoising: $$\hat{c}_p = \frac{1}{C_p} \sum_{q \in \mathcal{N}_p} c_q \, w(p,q),$$

Noisy color image → Reconstructed noise-free image

Conditioning On

Auxiliary feature buffers

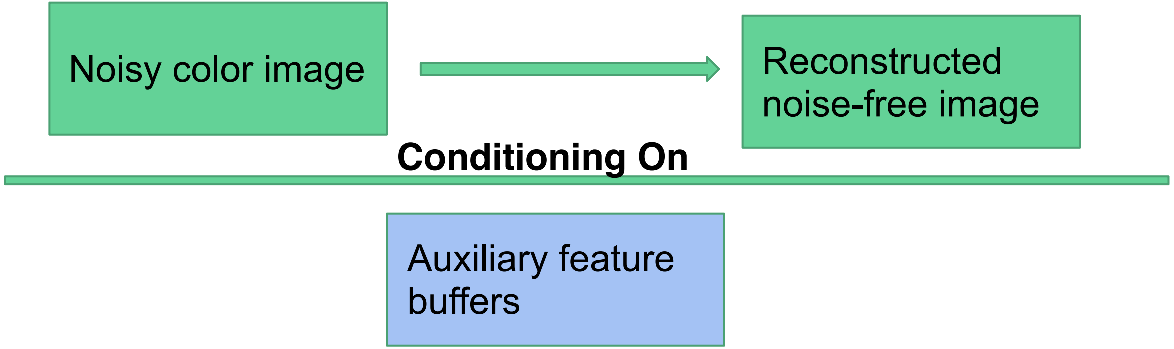
Expectations:

1. To extract **more** clues from auxiliary feature buffers.
2. To explore the correct **relationship** between noisy image and aux features.

Motivation 2: How to use the auxiliary features more wisely?

Image-space denoising:

$$\hat{c}_p = \frac{1}{C_p} \sum_{q \in \mathcal{N}_p} c_q \, w(p,q),$$



Expectations:

1. To extract **more** clues from auxiliary feature buffers.

Extract deep features using NN.

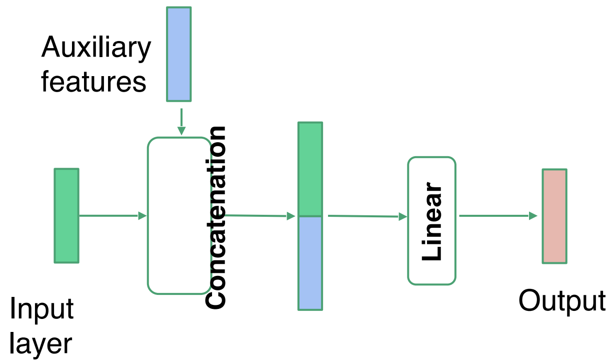
2. To explore the correct **relationship** between noisy image and aux features.

Try more complex interaction to model the relationship.

Motivation 2: How to use the auxiliary features more wisely?

Different Ways of Network Conditioning:
Traditional approach: Concatenation based conditioning. [Bako et al. 2017 ; Chaitanya et al. 2017]

Concatenation on all layers

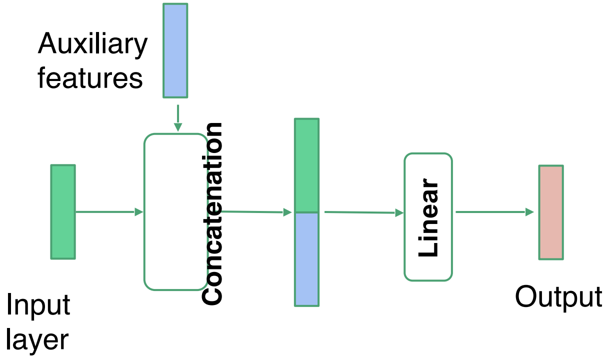


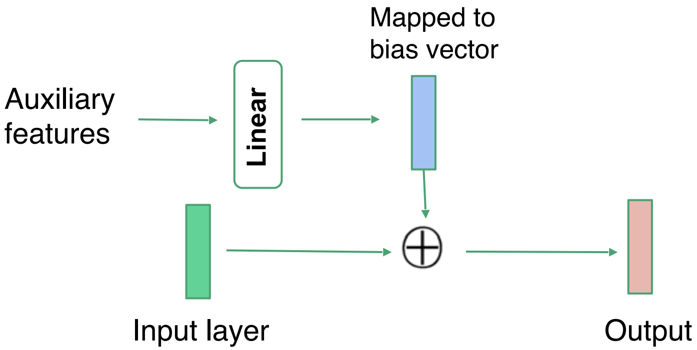
[Dumoulin et al. 2018]

Motivation 2: How to use the auxiliary features more wisely?

Different Ways of Network Conditioning:
Traditional approach: Concatenation based conditioning. [Bako et al. 2017 ; Chaitanya et al. 2017]

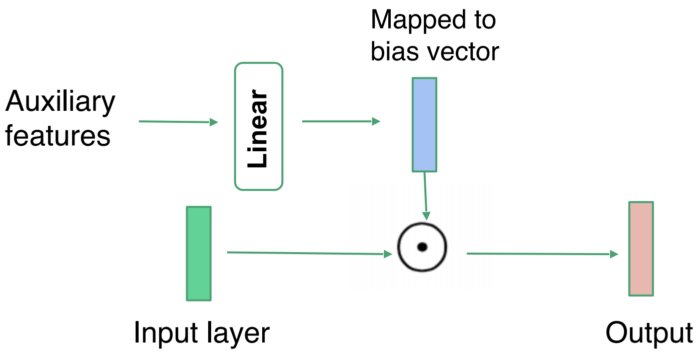
Concatenation on all layers = Conditional biasing





[Dumoulin et al. 2018]

Motivation 2: How to use the auxiliary features more wisely?



Conditional scaling

[Dumoulin et al. 2018]

Motivation 2: How to use the auxiliary features more wisely?

Different Ways of Network Conditioning:

Traditional approach: Concatenation based conditioning. [Bako et al. 2017 ; Chaitanya et al. 2017]

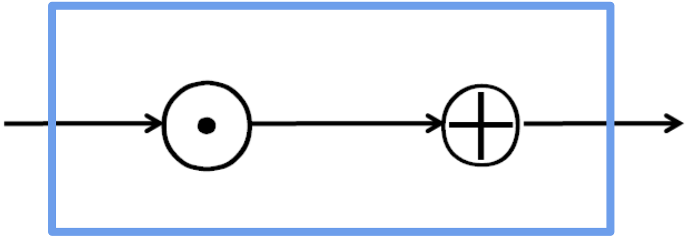
Conditional biasing

Conditional scaling

[Dumoulin et al. 2018]

Motivation 2: How to use the auxiliary features more wisely?

Different Ways of Network Conditioning:
Traditional approach: Concatenation based conditioning. [Bako et al. 2017 ; Chaitanya et al. 2017]



Conditional biasing

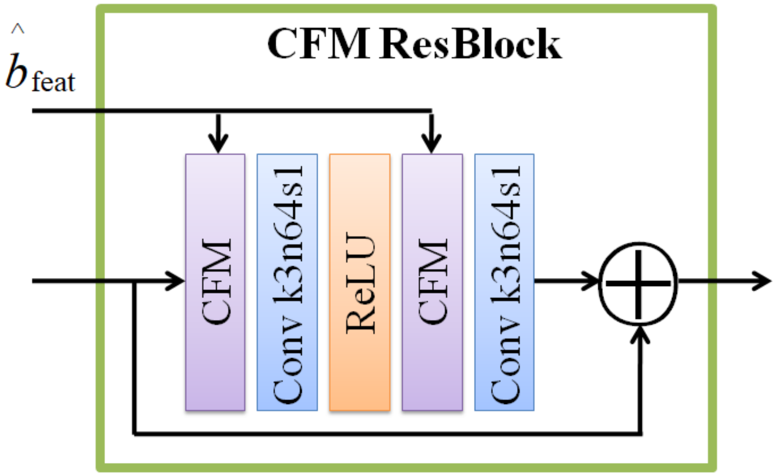
Shazam!

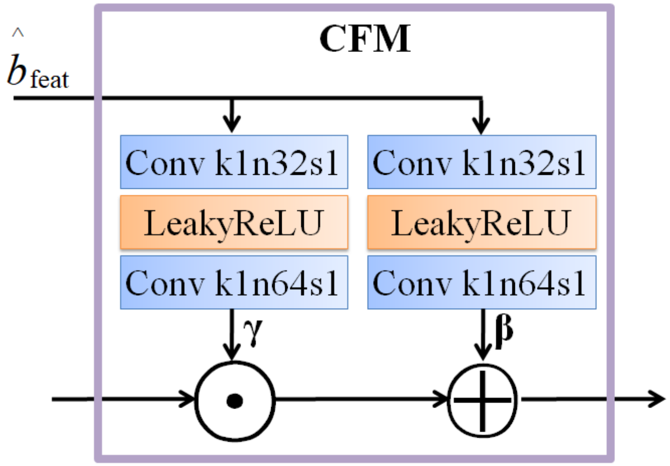
Conditional scaling

[Dumoulin et al. 2018]

Auxiliary Buffer Conditioned Modulation

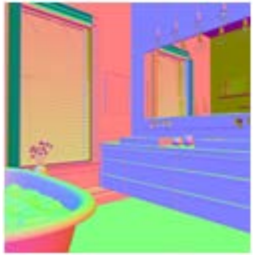
$$CFM(L_{in}) = \gamma(\hat{b_{feat}}) \odot L_{in} \otimes \beta(\hat{b_{feat}})$$





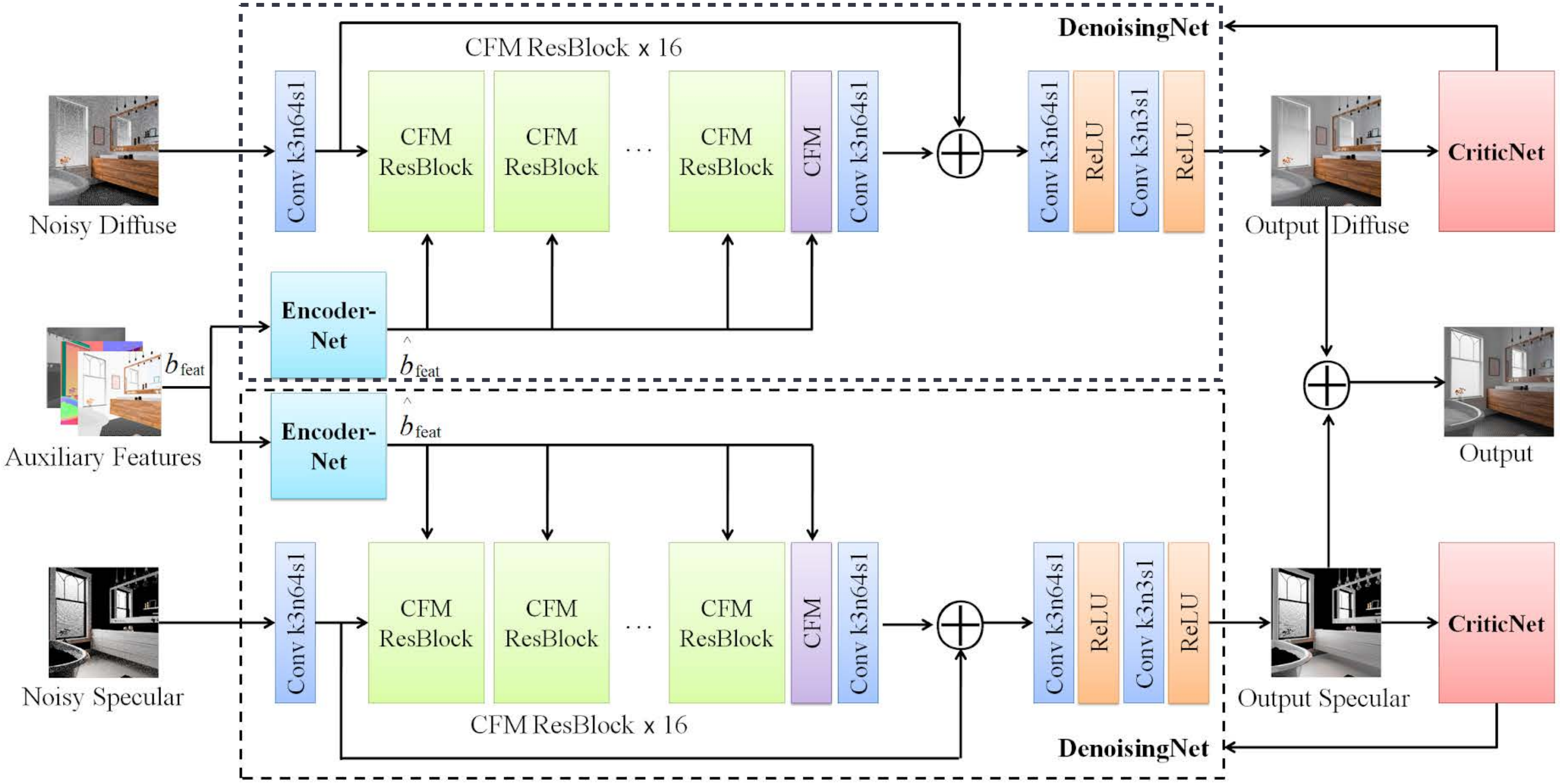
Other details

- Auxiliary feature buffers:
  - Can be obtained from GBuffer or at first bounce of path tracer.
  - Extensible,   you can try more.


- Diffuse/Specular decomposition (same as in KPCN)
  - A simplified light path decomposition.
  - Attention: specular here is not the accurate specular but (color - diffuse)
  - Necessary if calculating an untextured color buffer

Complete Framework



Results & Performance

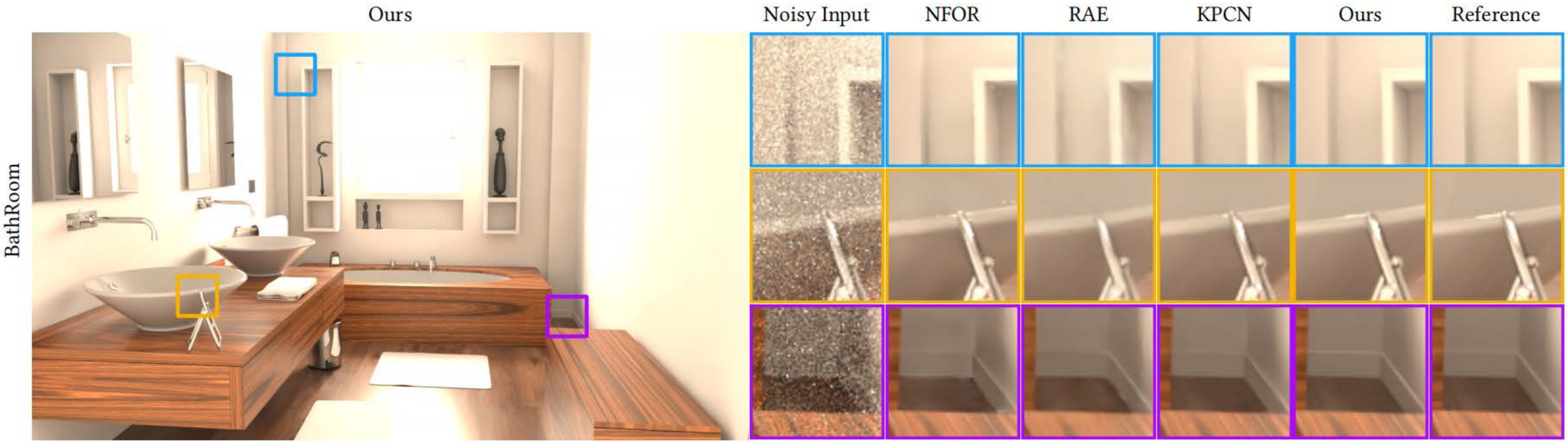
Evaluation

SOTA Baselines:

NFOR [Bitterli et al. 2014],
KPCN [Bako et al. 2017],
RAE [Chaitanya et al. 2017]

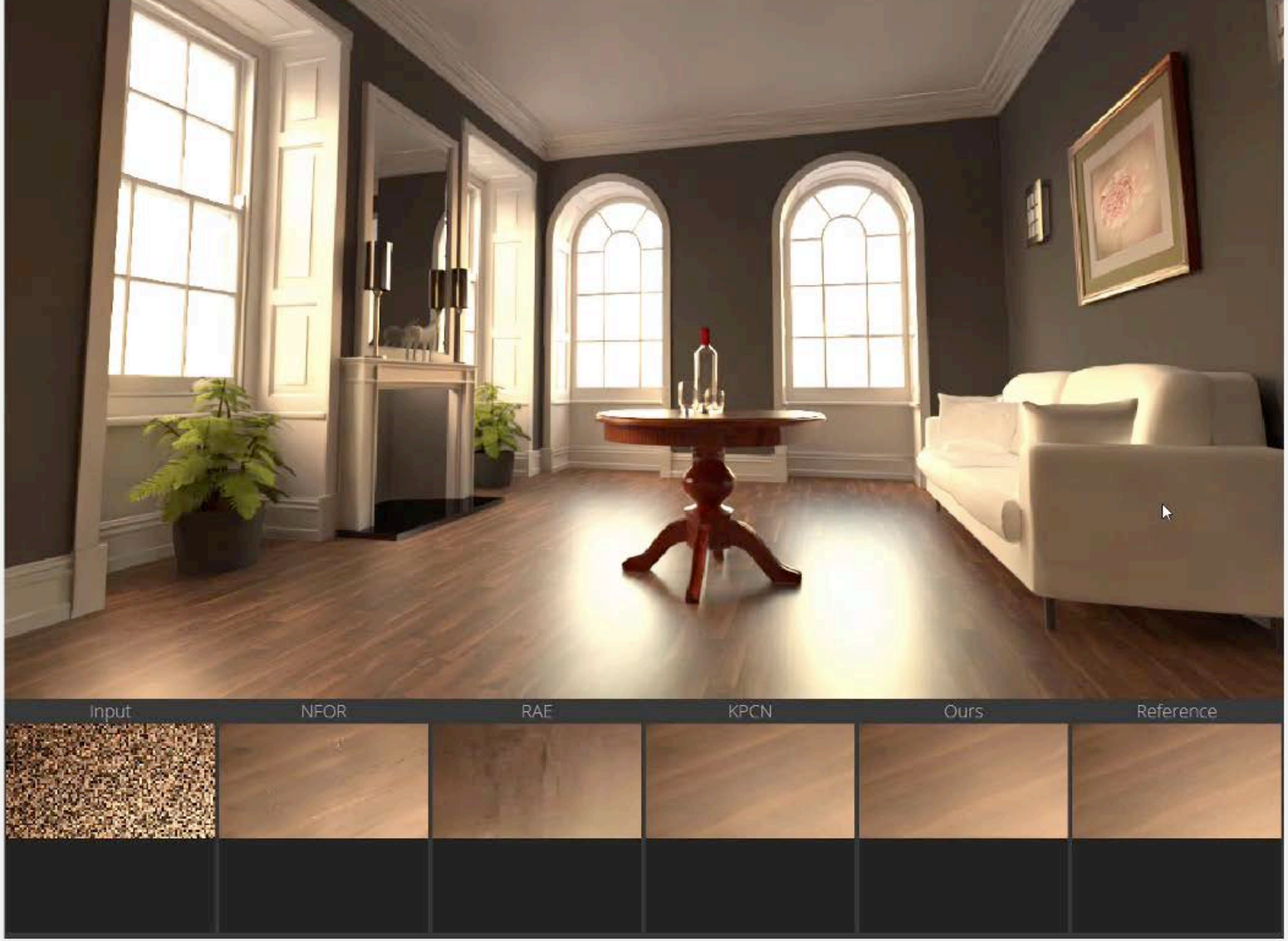
spp	Denoiser	1-SSIM ↓		PSNR ↑		RMSE ↓	
		AVG.	B.P.	AVG.	B.P.	AVG.	B.P.
4	NFOR	0.1614	0.00%	28.3028	0.00%	0.0374	0.00%
	RAE	**0.0751**	37.93%	29.5359	0.00%	0.0080	6.90%
	KPCN	0.0891	13.79%	32.1290	0.00%	0.0059	3.45%
	Ours	0.0773	**48.28%**	**34.3188**	**100.00%**	**0.0038**	**89.66%**
16	NFOR	0.0707	10.34%	32.6832	10.34%	0.0180	3.45%
	RAE	0.0549	3.45%	34.3337	0.00%	0.0033	3.45%
	KPCN	0.0531	20.69%	36.4538	0.00%	0.0024	6.90%
	Ours	**0.0463**	**65.52%**	**37.8608**	**89.66%**	**0.0019**	**86.21%**
32	NFOR	0.0493	10.34%	34.8495	6.90%	0.0118	3.45%
	RAE	0.0482	0.00%	36.0105	0.00%	0.0028	0.00%
	KPCN	0.0426	20.69%	38.4051	3.45%	0.0017	10.34%
	Ours	**0.0366**	**68.97%**	**39.6197**	**89.66%**	**0.0013**	**86.21%**
64	NFOR	0.0389	6.90%	37.3478	6.90%	0.0067	3.45%
	RAE	0.0395	0.00%	38.0982	0.00%	0.0016	0.00%
	KPCN	0.0349	20.69%	40.2623	27.59%	0.0012	34.48%
	Ours	**0.0296**	**72.41%**	**40.9673**	**65.52%**	**0.0009**	**62.07%**
128	NFOR	0.0305	10.34%	39.5127	6.90%	0.0036	3.45%
	RAE	0.0338	0.00%	39.8093	0.00%	0.0011	0.00%
	KPCN	0.0288	27.59%	42.0056	**48.28%**	0.0008	**62.07%**
	Ours	**0.0248**	**62.07%**	**42.0803**	44.83%	**0.0007**	34.48%

Examples of public scenes



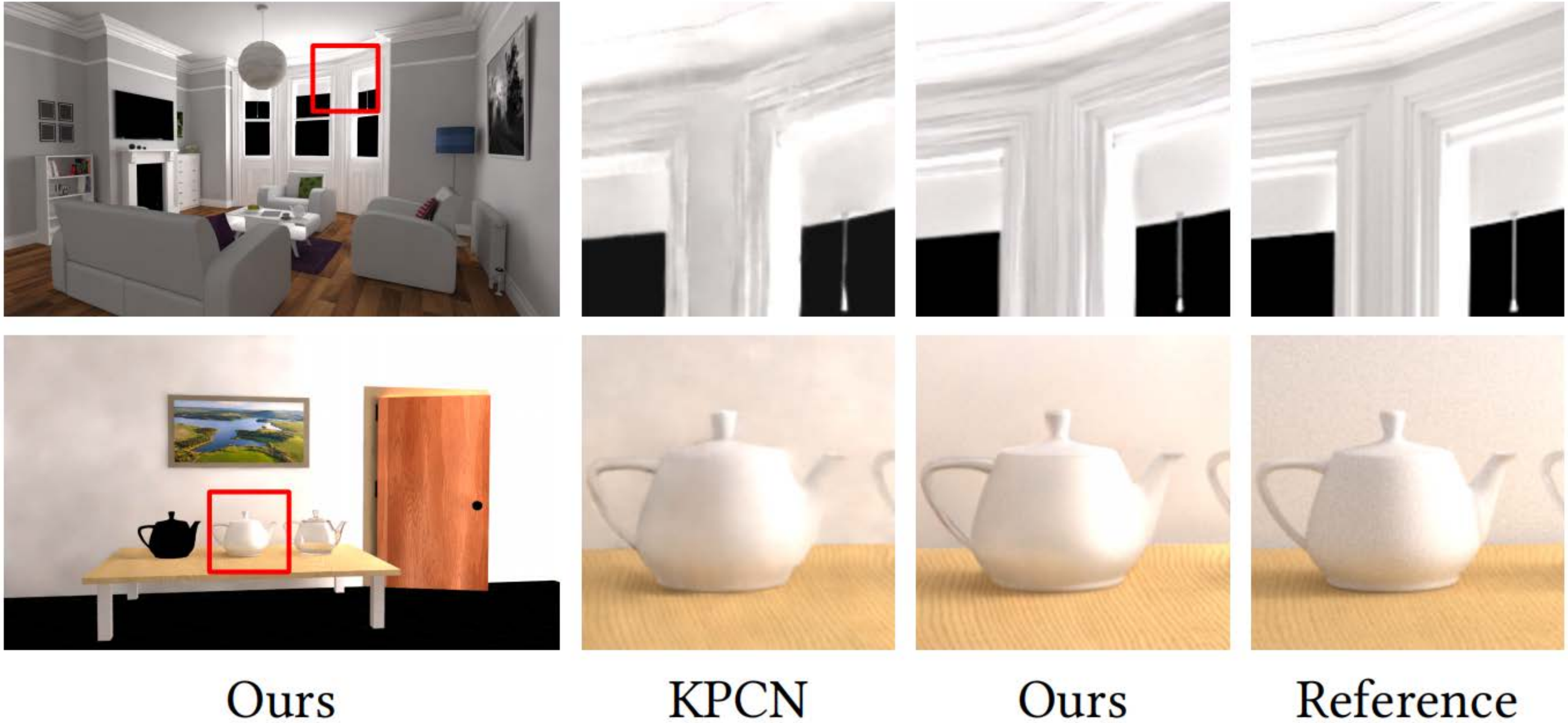
	Noisy Input	NFOR	RAE	KPCN	Ours
1-SSIM	(0.498707)	(0.023738)	(0.036035)	(0.019587)	(0.015482)
RMSE	(0.015005)	(0.000601)	(0.000909)	(0.000411)	(0.000254)
PSNR	(23.433917)	(38.139083)	(36.114381)	(39.167406)	(41.241905)

More results with a html interactive viewer can be seen on http://adversarial.mcdenoising.org/interactive_viewer/viewer.html

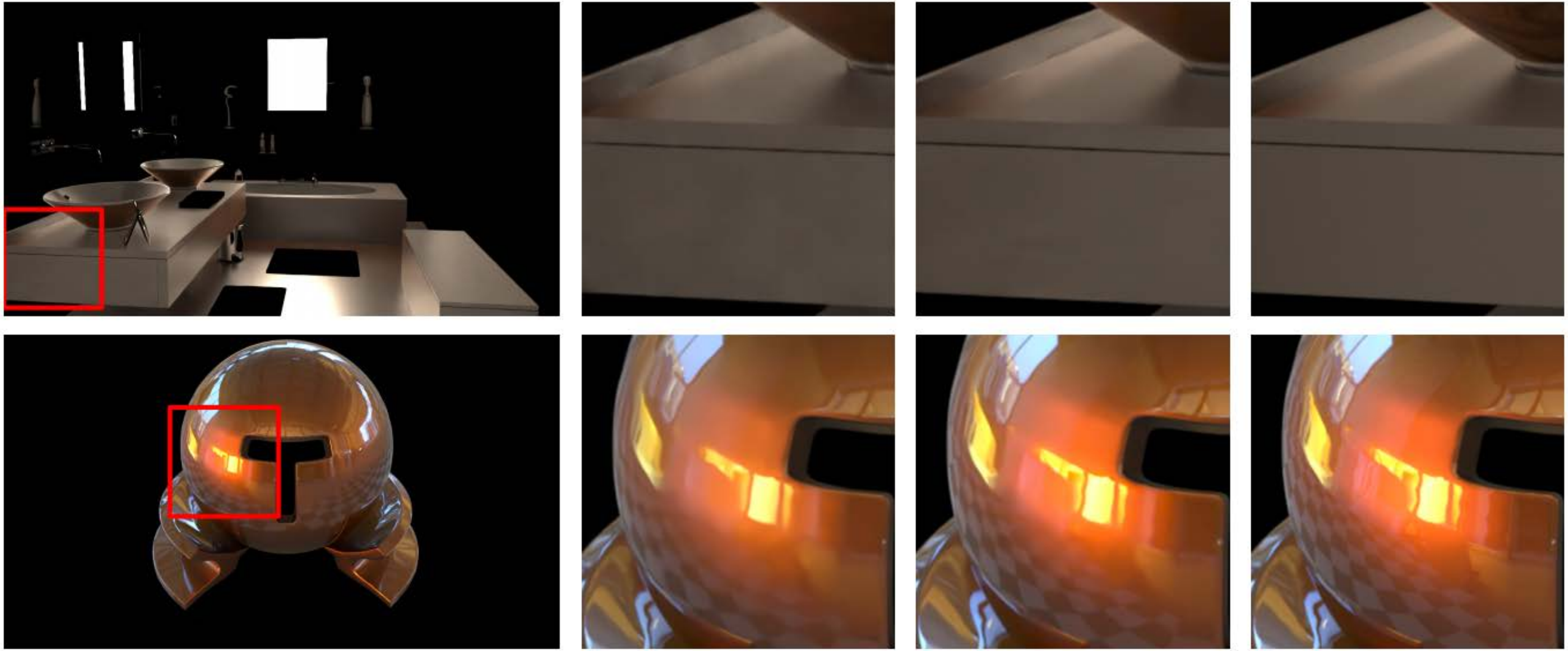


Input
NFOR
RAE
KPCN
Ours
Reference

Reconstructed diffuse results



Reconstructed specular results



Ours KPCN Ours Reference

Reconstruction performance

For 1280x720 image:

Ours: 1.1s (550ms for diffuse/specular) single 2080Ti

KPCN: 3.9s single 2080Ti

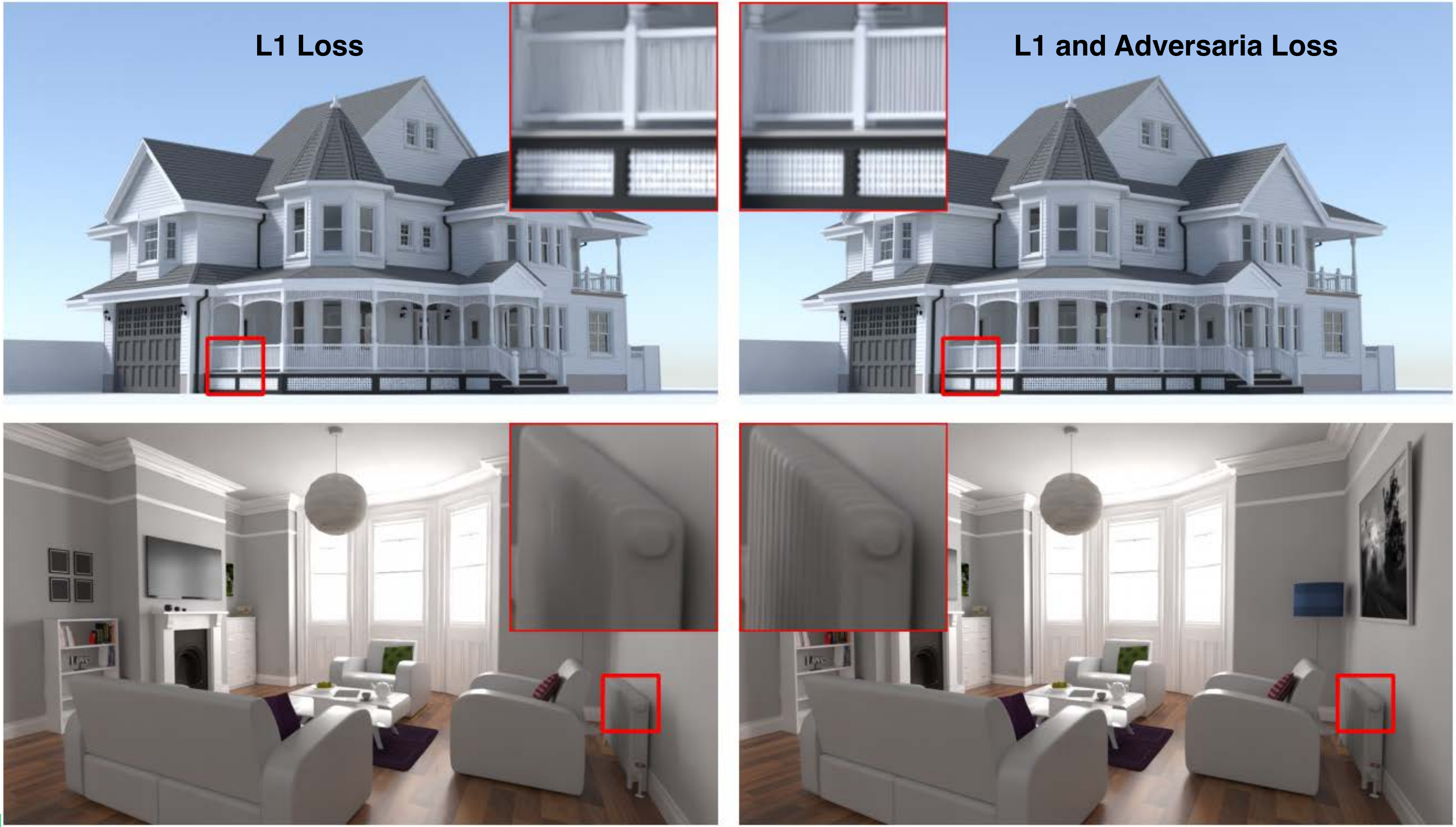
NFOR: more than 10s, 3.4GHz Intel Xeon processor

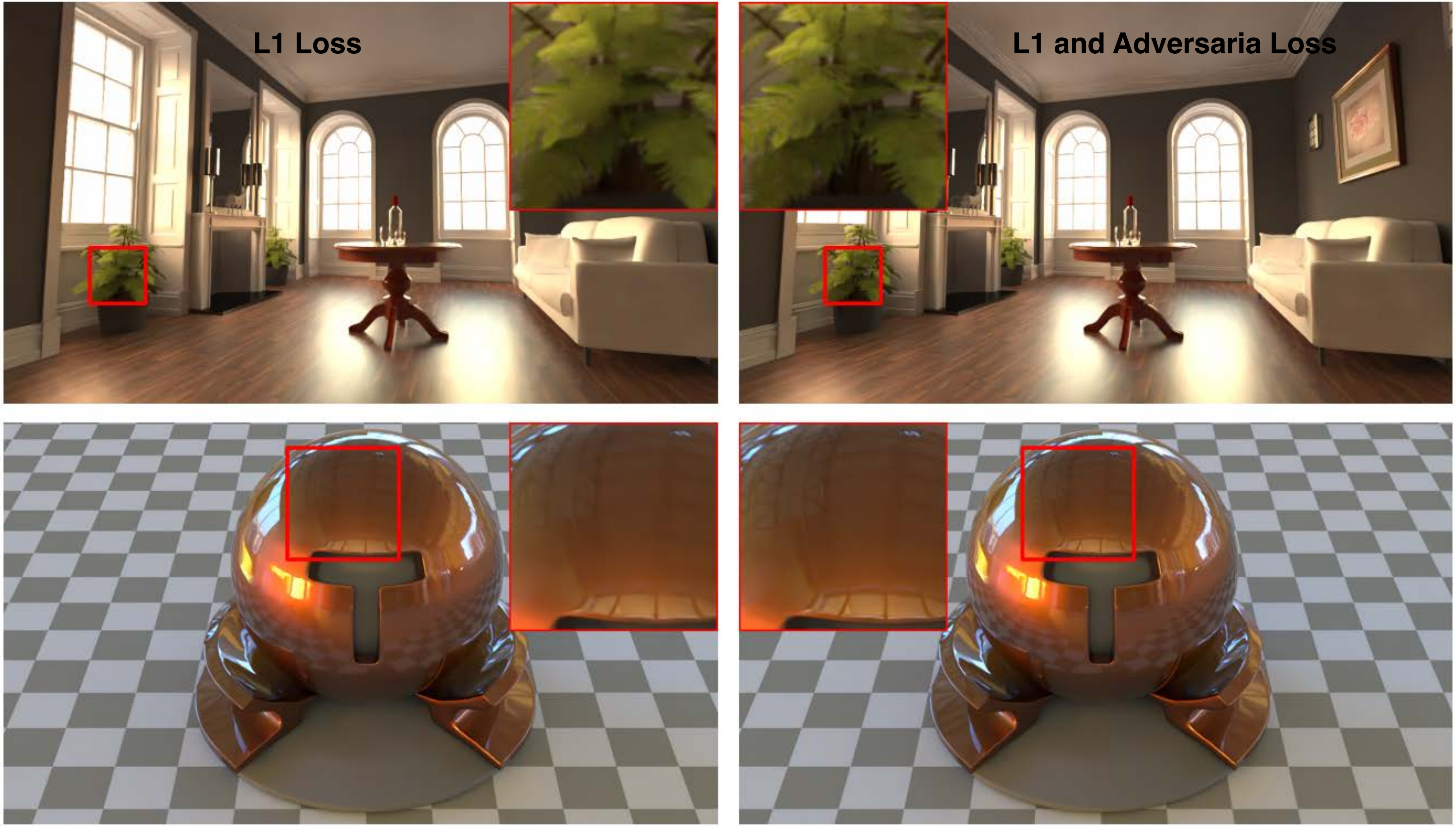
Analysis & Discussion

Effectiveness of the adversarial loss and critic network

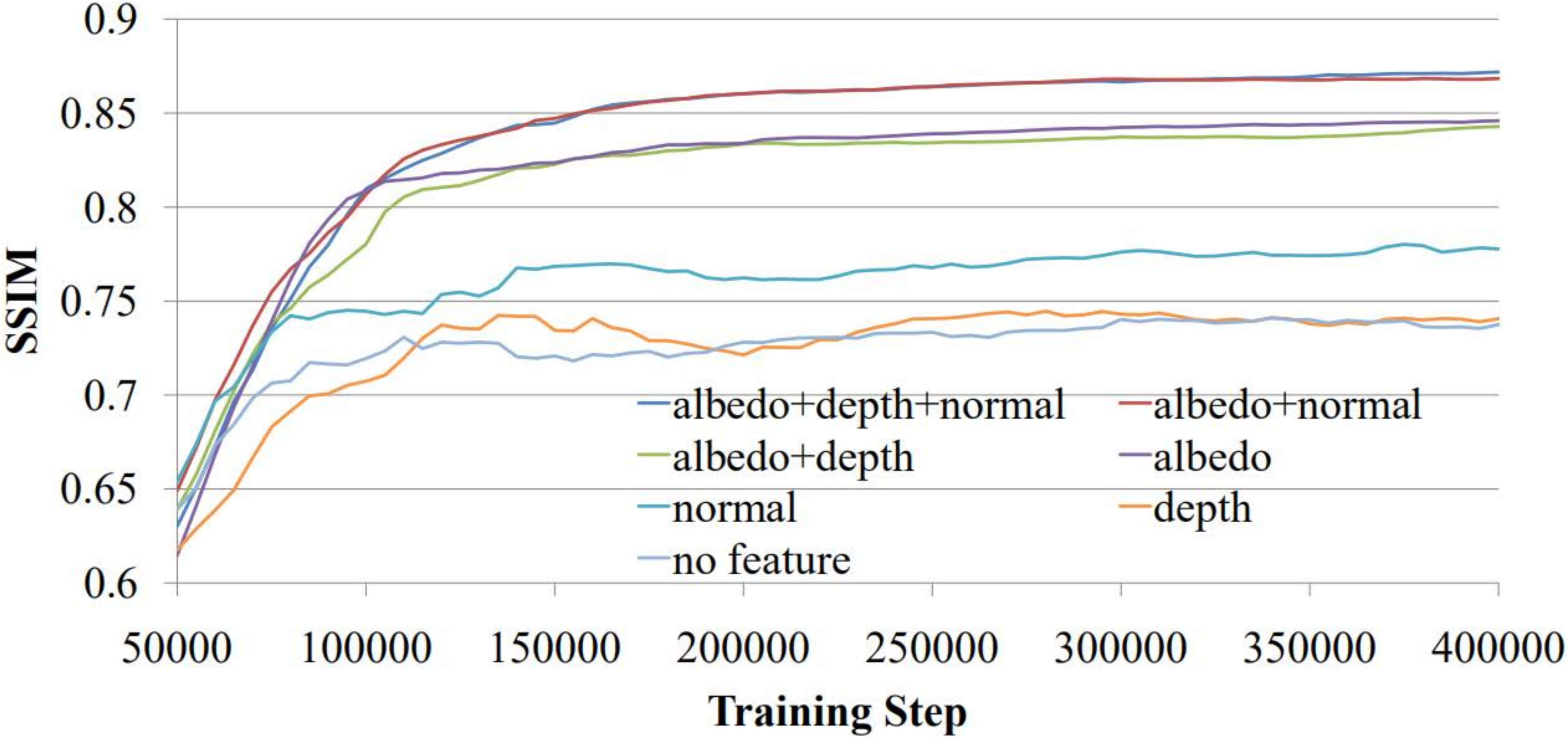
Control groups:

- L1 loss (KPCN tests many loss functions L1, L2, SSIM etc and L1 shown to be the best)
- L1 with adversarial loss


L1 Loss
L1 and Adversaria Loss


L1 Loss
L1 and Adversaria Loss

Effectiveness of auxiliary feature buffers



Effectiveness of feature conditioned modulation

No auxiliary features | Concatenate the auxiliary features & noisy color as fused input | Full model of CFM | Reference

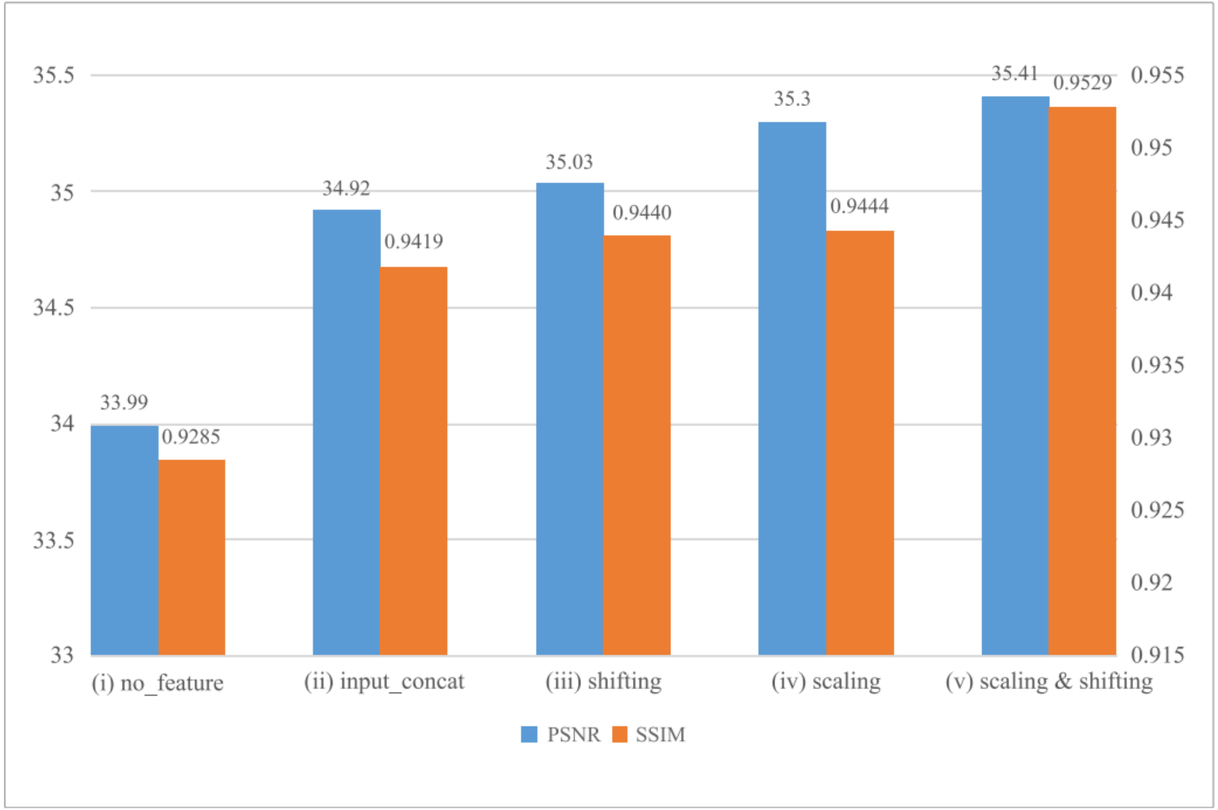
Previous work & Proposed conditioned feature modulation

- Traditional feature-guided filtering:
  - generally based on joint filtering or cross bilateral filtering [Bauszat et al.2011]
  - handcrafted assumption on the correlation between the low-cost auxiliary features and noisy image

- Learning based approaches: concatenation as fused input
  - Limit the effectiveness of auxiliary features to early layers
  - amounts to biasing

- Combination of conditional biasing and scaling:
  - perform scaling and shifting at different scales
  - point-wise shifting modulates the feature activation.
  - point-wise scaling selectively suppresses or highlights feature activation.

Effectiveness of feature conditioned modulation

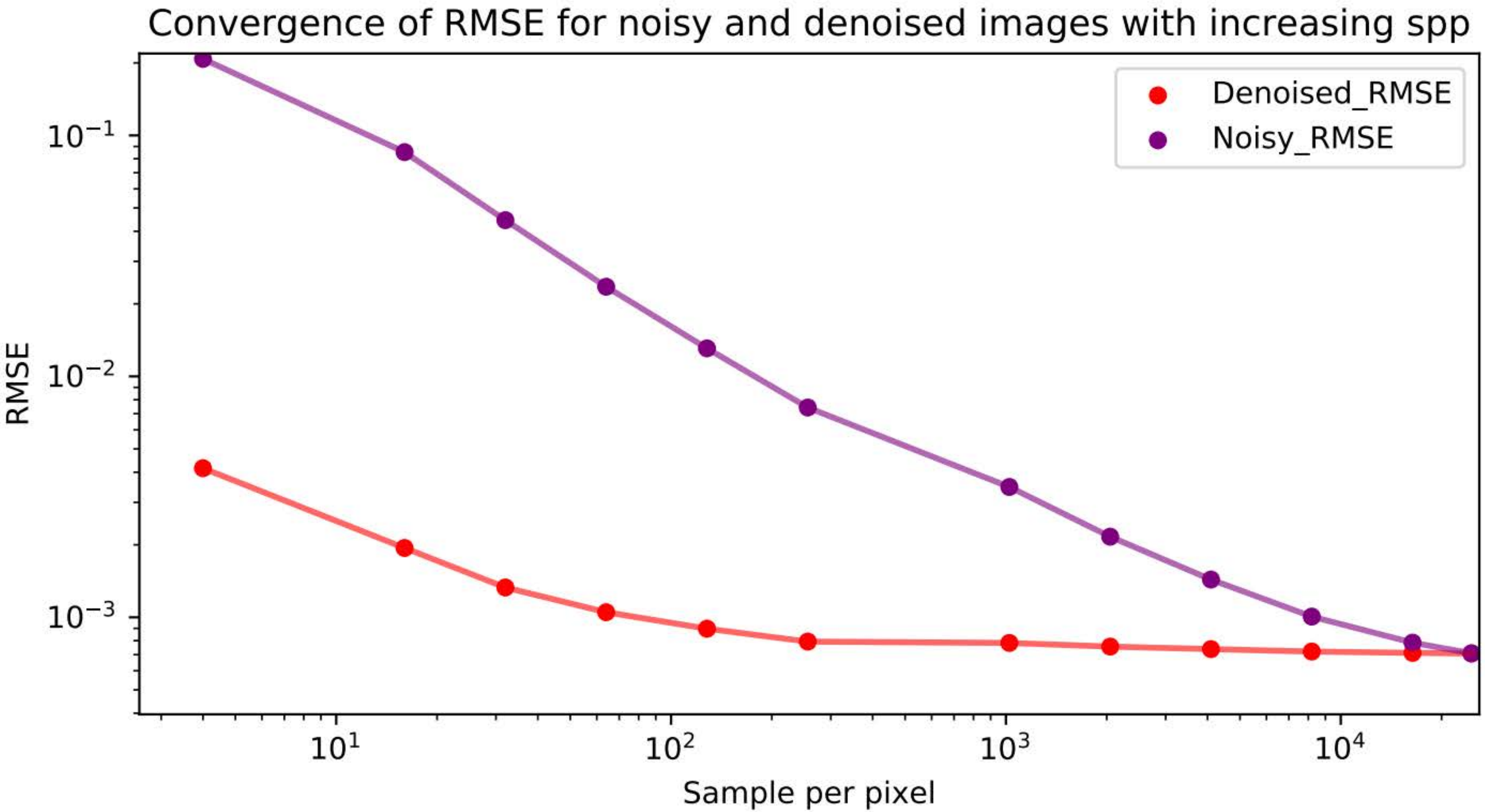


Diffuse and specular decomposition

Reflection is not reconstructed well without separating diffuse and specular components

Reflection is well reconstructed by separating diffuse and specular components

Convergence discussion



Limitation, future work, conclusion

Limitations

Ours | NFOR | KPCN | Ours | Reference

Future work

- Network optimization & speedup
  - Model simplification
  - Custom-precision
  - Model pruning
- Temporal coherence
- Explore more complex relationship between noisy input and auxiliary features
  - Attention mechanism
  - Hypernetworks
- More rendering effects
  - Depth of field
  - Motion blur..
- How to do without large training set? (expensive)

Conclusion

- Adversarial learning framework for MC denoising problem
- Shed light on exploring the relationship between auxiliary features and noisy images by neural networks.
- Open source code and weights released on http://adversarial.mcdenoising.org.

Thank you!

Acknowledgement

We gratefully thank the anonymous reviewers for their constructive suggestions, and Qing Ye, Qi Wu, Junrong Huang for helpful discussions and cluster rendering support. This work is partially funded by National Key R&D Program of China (No. 2017YFB1002605), NSFC (No. 61872319, 61822204, 61521002), Zhejiang Provincial NSFC (No. LR18F020002), CAMERA - the RCUK Centre for the Analysis of Motion, Entertainment Research and Applications (EP/M023281/1), and a gift from Adobe.