High-Quality Self-Supervised Deep Image Denoising

Samuli Laine — NVIDIA

Tero Karras — NVIDIA

Jaakko Lehtinen — NVIDIA, Aalto University

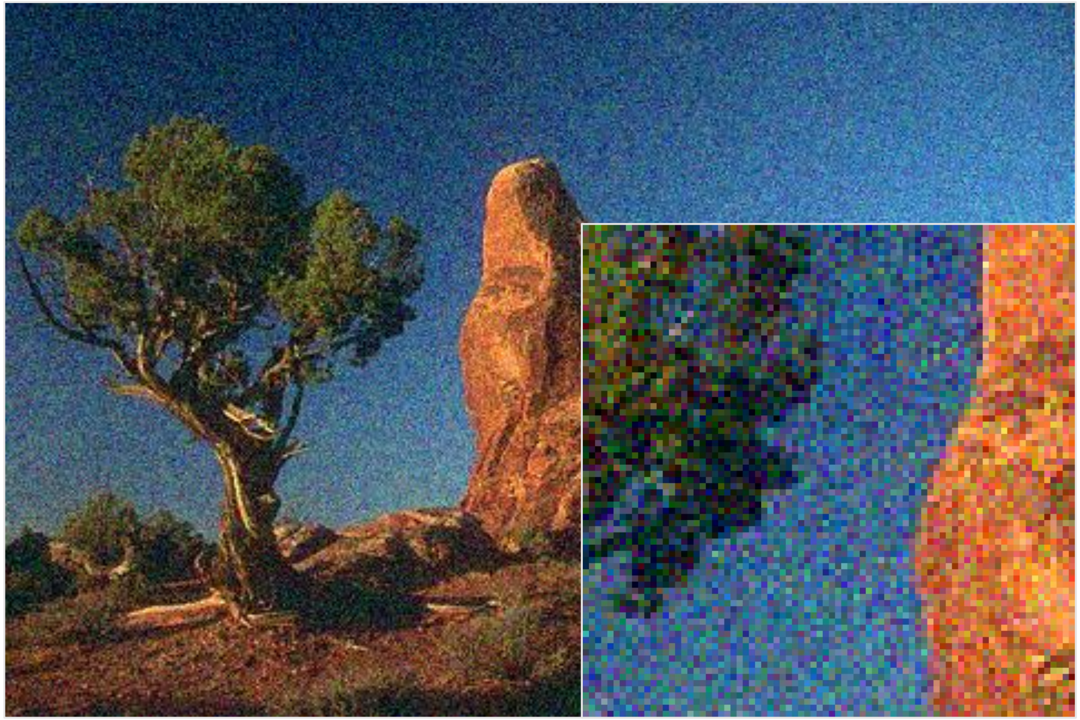
Timo Aila — NVIDIA

NeurIPS | 2019

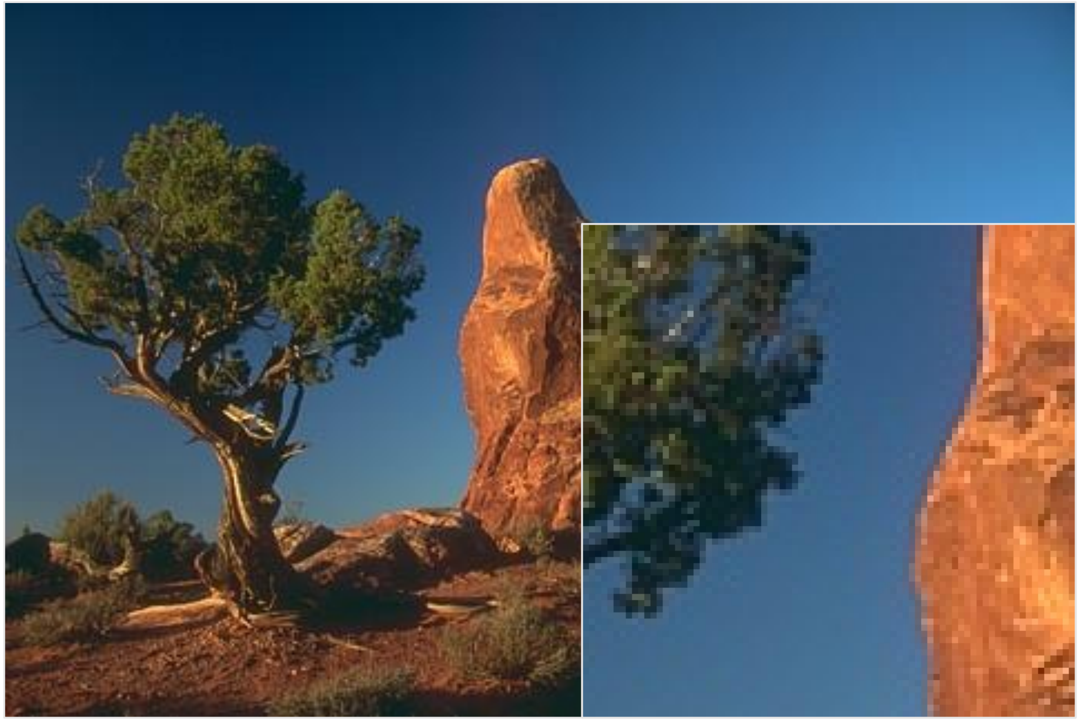
Goals and contributions

- Train a deep denoiser network
    - Why is deep learning needed? So that denoiser adapts to underlying data!
- Remove the need for separate training data with self-supervision
- **Contribution 1**: Bayesian approach for high-quality denoising results
    - Target is to match a denoiser trained with clean reference data
- **Contribution 2**: Improve training performance with an efficient blind-spot network architecture

Background: Traditional training



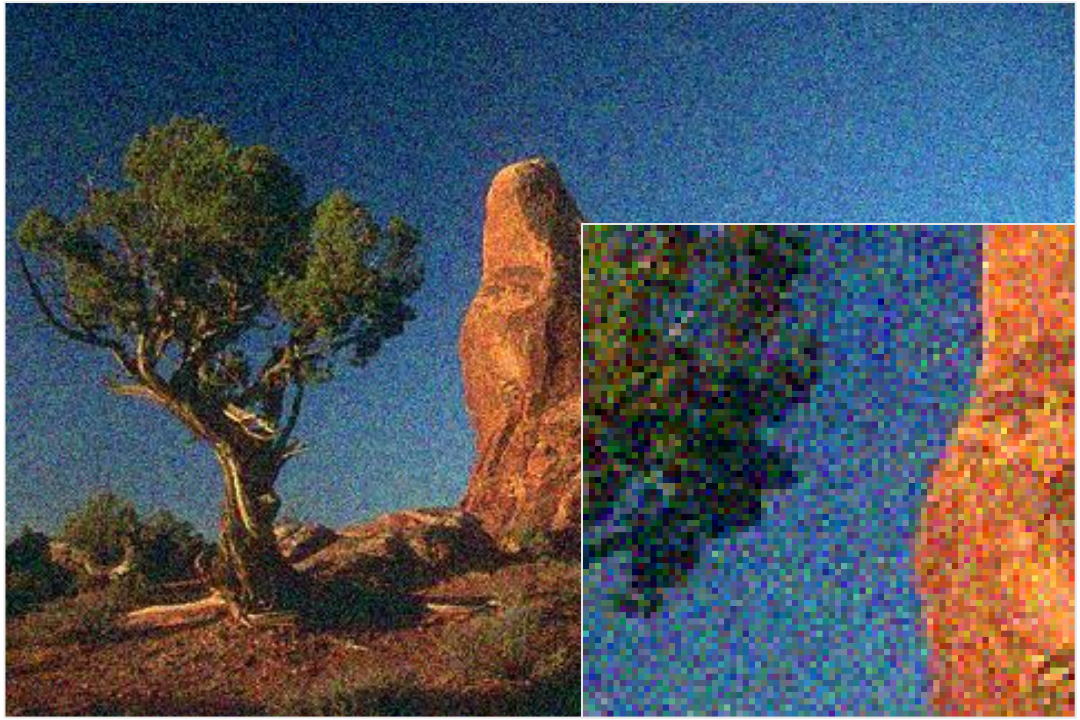
Input: **Noisy image**



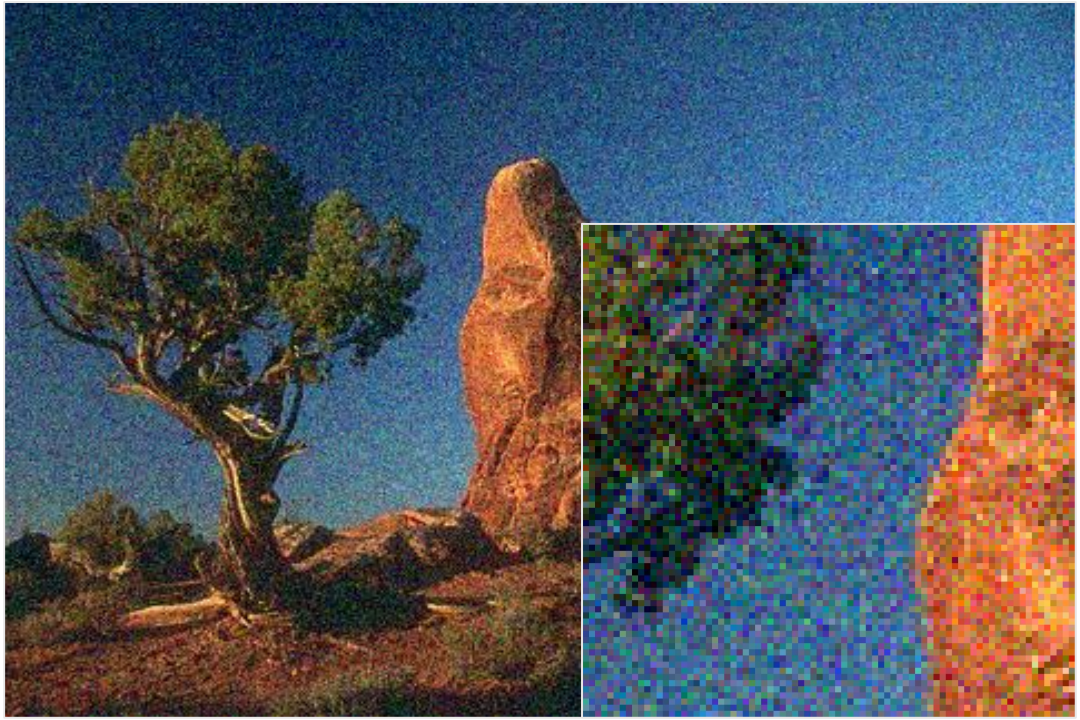
Target: **Clean image**

Background: Noise2Noise training

[Lehtinen et al., 2018]



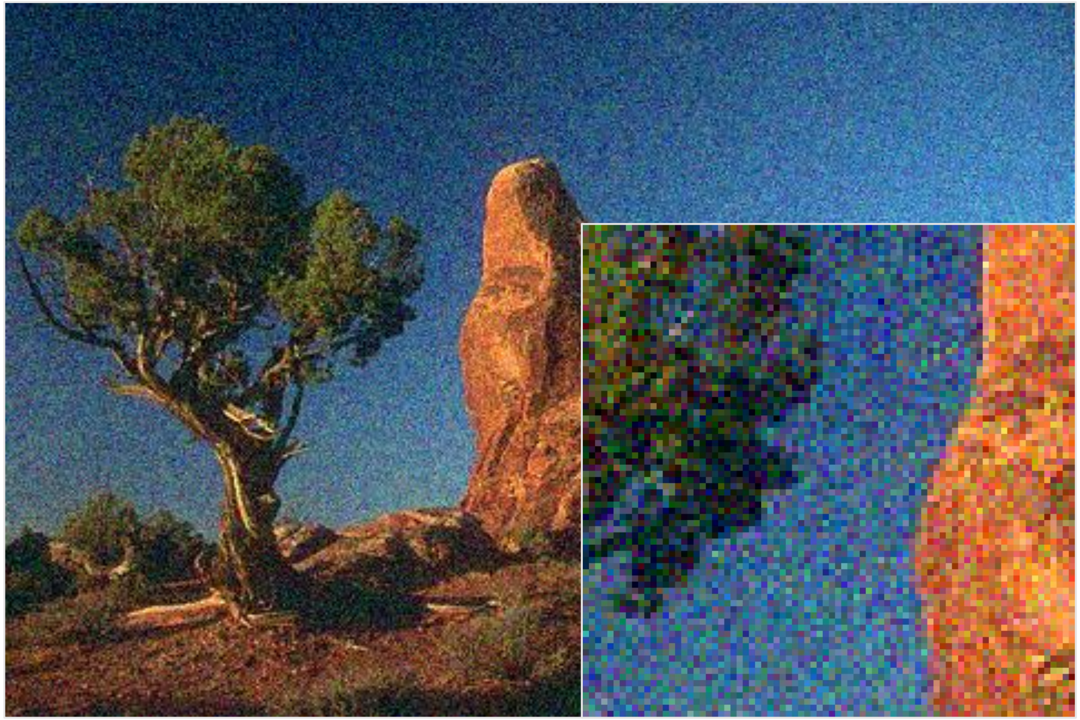
Input: **Noisy image**



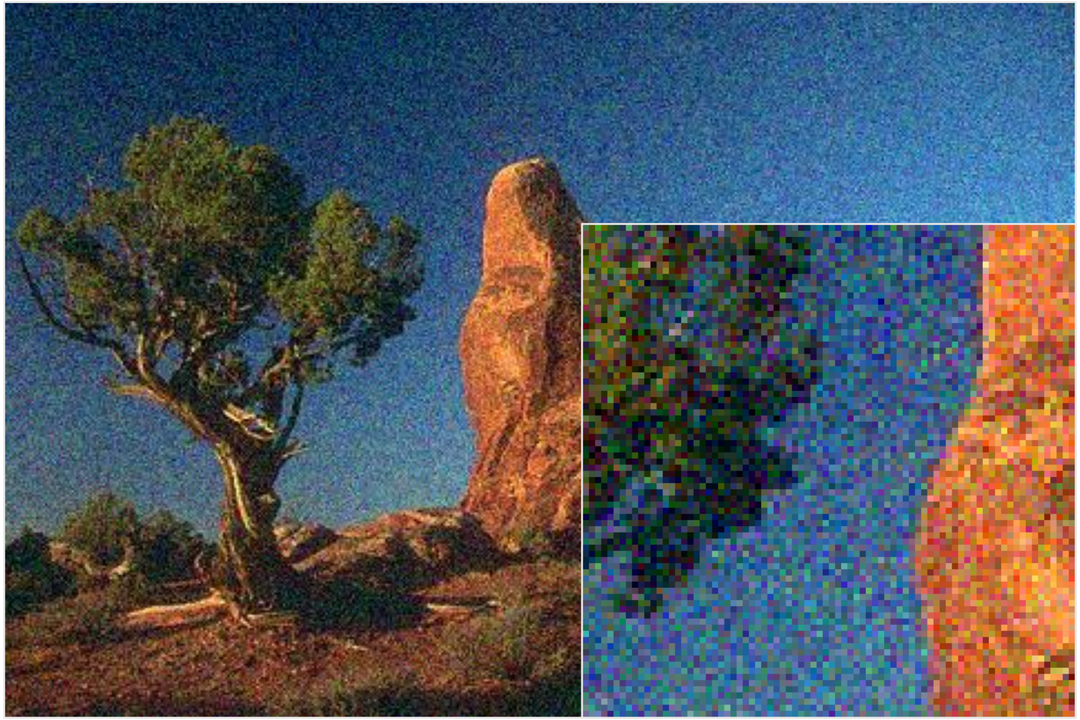
Target: **A different noisy image**

Background: Noise2Void training

[Krull et al., 2018]



Input: **Noisy image**



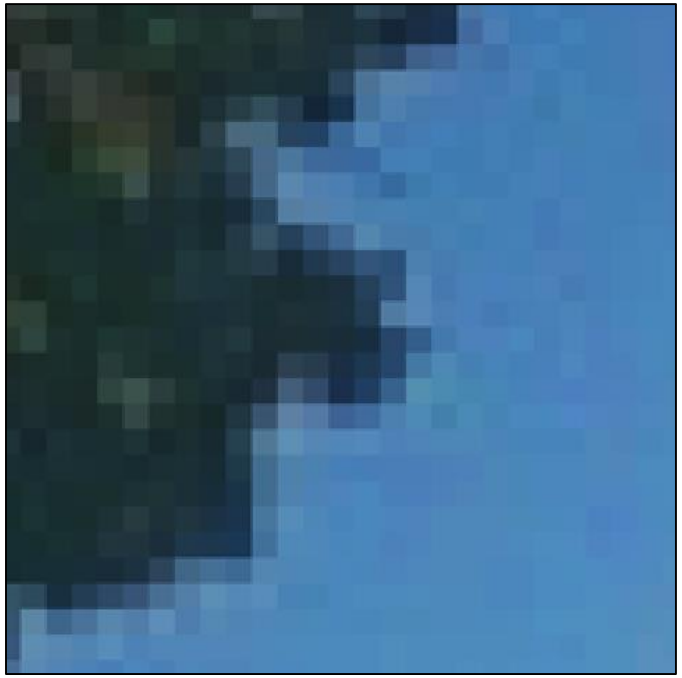
Target: **The <u>same</u> noisy image**

Concepts and notation

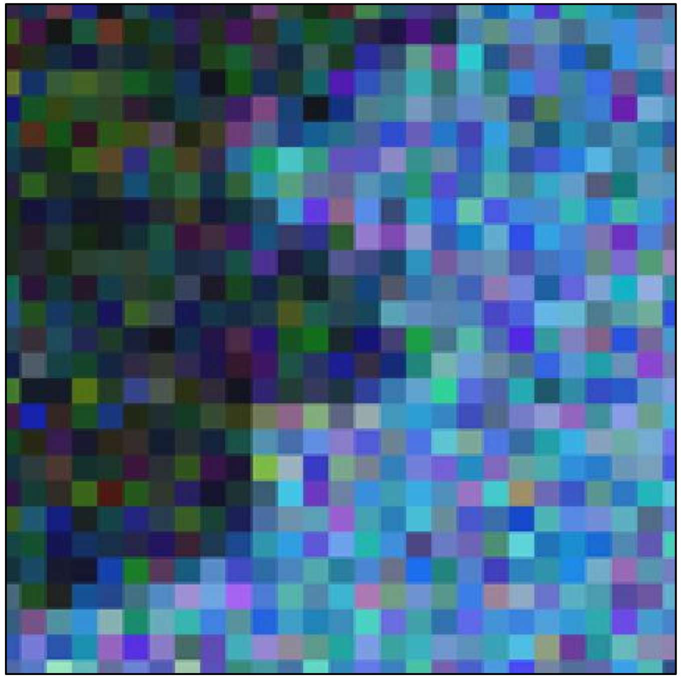
$\boldsymbol{x}$ clean pixel value

$\boldsymbol{y}$ noisy pixel value

$\boldsymbol{\Omega_y}$ noisy *context*, i.e., noisy image except pixel $\boldsymbol{y}$



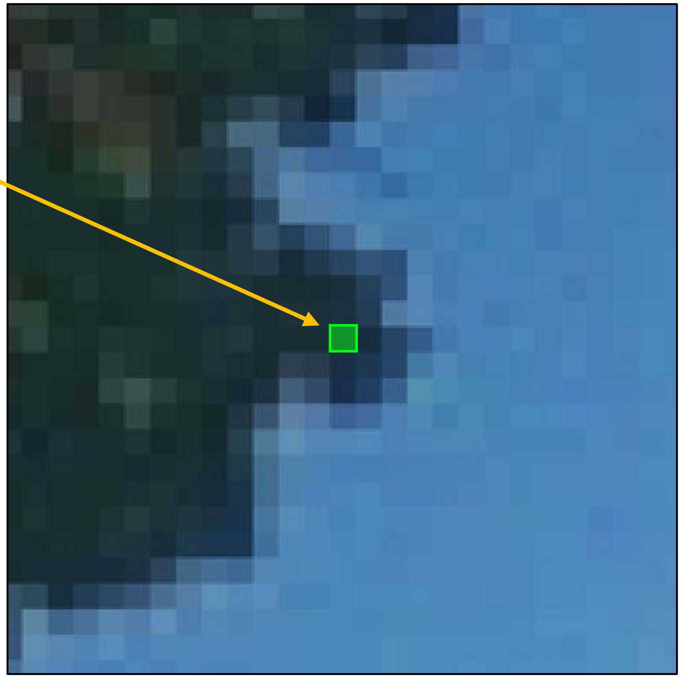
Clean image



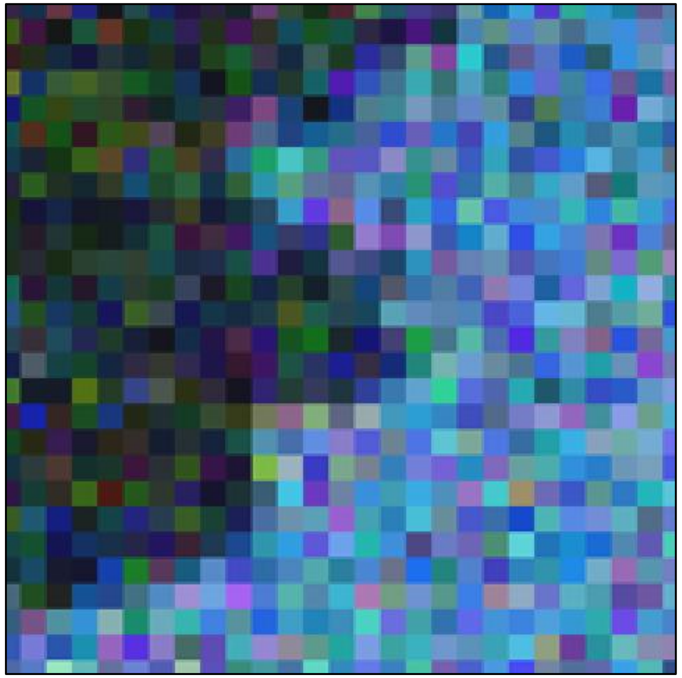
Noisy image

Concepts and notation

- $\boldsymbol{x}$ **clean pixel value**
- $\boldsymbol{y}$ noisy pixel value
- $\boldsymbol{\Omega_y}$ noisy *context*, i.e., noisy image except pixel $\boldsymbol{y}$



Clean image



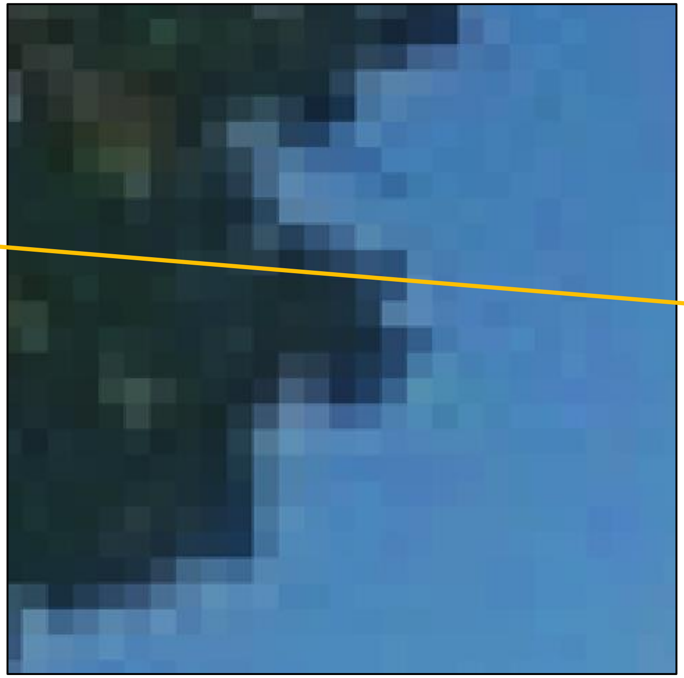
Noisy image

Concepts and notation

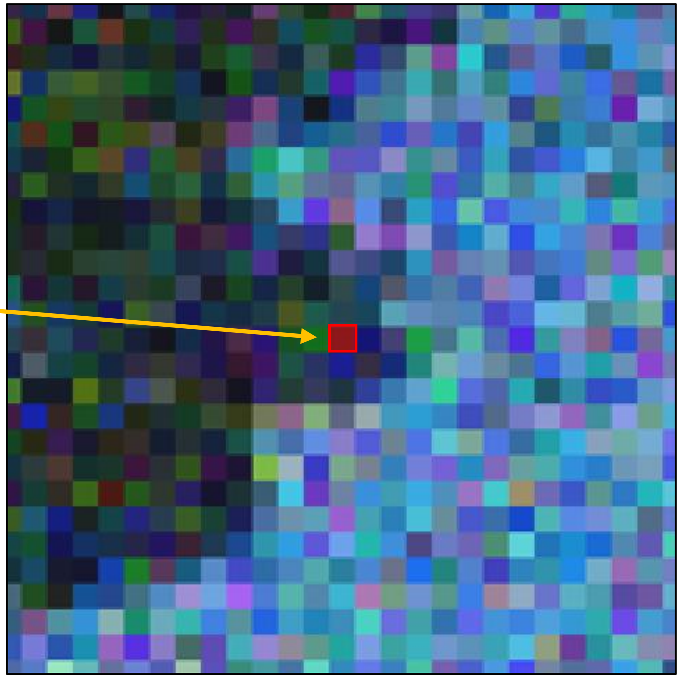
$\boldsymbol{x}$ clean pixel value

$\boldsymbol{y}$ **noisy pixel value**

$\boldsymbol{\Omega}_{y}$ noisy *context*, i.e., noisy image except pixel $\boldsymbol{y}$



Clean image



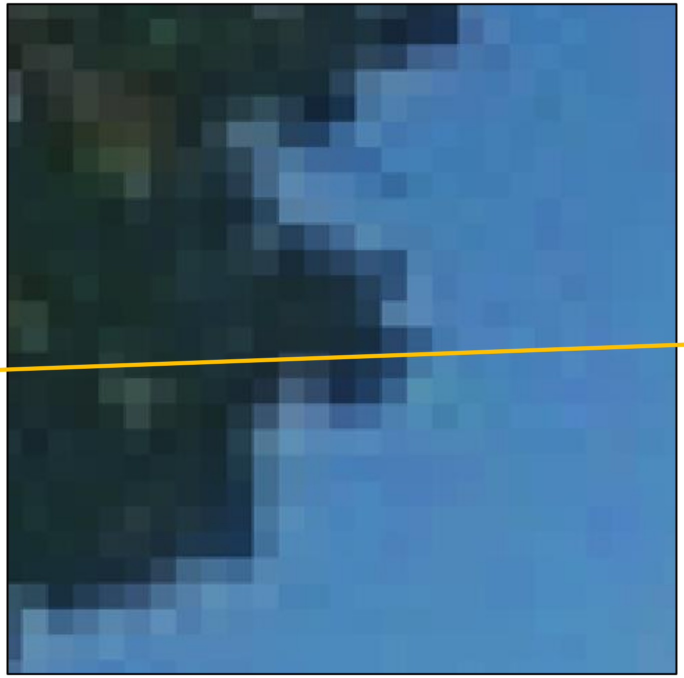
Noisy image

Concepts and notation

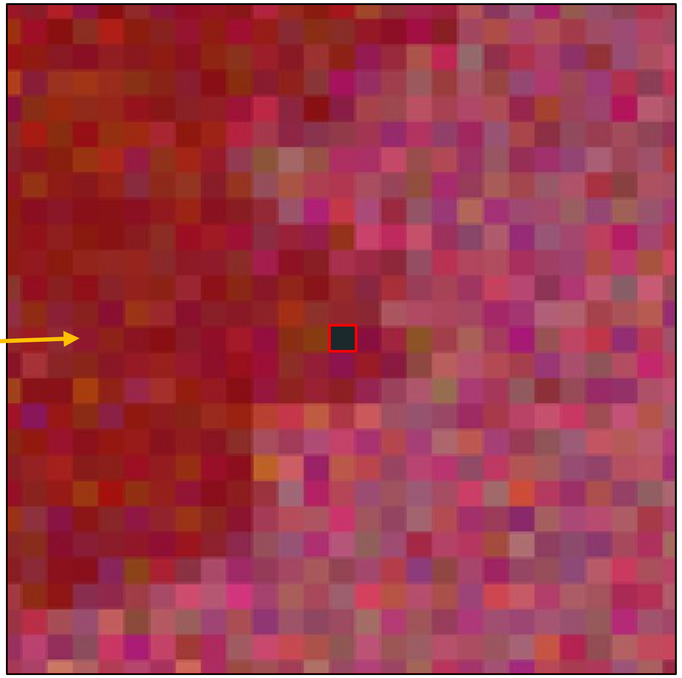
$\boldsymbol{x}$ clean pixel value

$\boldsymbol{y}$ noisy pixel value

$\boldsymbol{\Omega_y}$ **noisy *context***, i.e., noisy image except pixel $\boldsymbol{y}$



Clean image



Noisy image

What is supervised training?

- Lump noisy pixel $\boldsymbol{y}$ and context $\boldsymbol{\Omega_y}$ together
- Learn to infer clean pixel $\boldsymbol{x}$ as $\mathbb{E}_x\left[p\left(\boldsymbol{x} \middle| \boldsymbol{y}, \boldsymbol{\Omega_y}\right)\right]$
- I.e., train $f_\theta: \boldsymbol{y}, \boldsymbol{\Omega_y} \rightarrow \boldsymbol{x}$ by optimizing $\underset{\theta}{\operatorname{argmin}} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{\Omega_y}}\left[L\left(f_\theta\left(\boldsymbol{y}, \boldsymbol{\Omega_y}\right), \boldsymbol{x}\right)\right]$
  - Simplifying assumptions made here: L2 loss, zero-mean noise

What is Noise2Void training?

- Only use context $\boldsymbol{\Omega}_{\boldsymbol{y}}$ for inference [Krull et al., 2018]
  - Thus, approximate clean pixel $\boldsymbol{x}$ as $\mathbb{E}_{\boldsymbol{x}}\left[p\left(\boldsymbol{x} \middle| \boldsymbol{\Omega}_{\boldsymbol{y}}\right)\right]$
- Can replace $\boldsymbol{x}$ with $\boldsymbol{y}$ if noise is zero-mean [Lehtinen et al., 2018]
  - Optimize $\operatorname*{argmin}_{\theta} \mathbb{E}_{\boldsymbol{y}}\left[L\left(f_\theta\left(\boldsymbol{\Omega}_{\boldsymbol{y}}\right), \boldsymbol{y}\right)\right]$ – no clean $\boldsymbol{x}$ is needed
- This is equivalent if corruption is independent between pixels!
  - See [Batson and Royer, 2019] for further analysis

Limitations of Noise2Void

- Ignoring $\boldsymbol{y}$ when denoising clearly leaves useful information unused
- While we can regress $f_\theta: \boldsymbol{\Omega_y} \to \boldsymbol{y}$, we cannot regress $f_\theta: \boldsymbol{y}, \boldsymbol{\Omega_y} \to \boldsymbol{y}$
  - Trivial solution is to pass pixel value through as-is → no denoising
  - Hence, at training time we cannot use $\boldsymbol{y}$ as an input
- Our solution is to bring in $\boldsymbol{y}$ via Bayesian inference at test time
  - Concurrent work by [Krull et al., 2019]

A more complete view

- Assume a known noise model $p(\boldsymbol{y}|\boldsymbol{x})$ that is independent of $\boldsymbol{\Omega}_y$
- Observed noisy data (training data) now relates to clean data as

$$\underbrace{p\left(\boldsymbol{y}\middle|\boldsymbol{\Omega}_y\right)}_{\text{Training data}} = \int \underbrace{p(\boldsymbol{y}|\boldsymbol{x})}_{\text{Noise model}}\, \underbrace{p\left(\boldsymbol{x}\middle|\boldsymbol{\Omega}_y\right)}_{\text{Unobserved}}\, \mathrm{d}\boldsymbol{x}$$

- This lets us learn to predict a parametric model for $p\left(\boldsymbol{x}\middle|\boldsymbol{\Omega}_y\right)$ that we represent as a multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$ over color components

Test-time inference

- The (unnormalized) posterior probability of $\boldsymbol{x}$, given observations of $\boldsymbol{y}$ and $\boldsymbol{\Omega}_y$, is given by Bayes' rule as

$$\underbrace{p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\Omega}_y)}_{\text{Posterior}} \propto \underbrace{p(\boldsymbol{y}|\boldsymbol{x})}_{\text{Noise model}} \underbrace{p(\boldsymbol{x}|\boldsymbol{\Omega}_y)}_{\text{Prior}}$$

Known

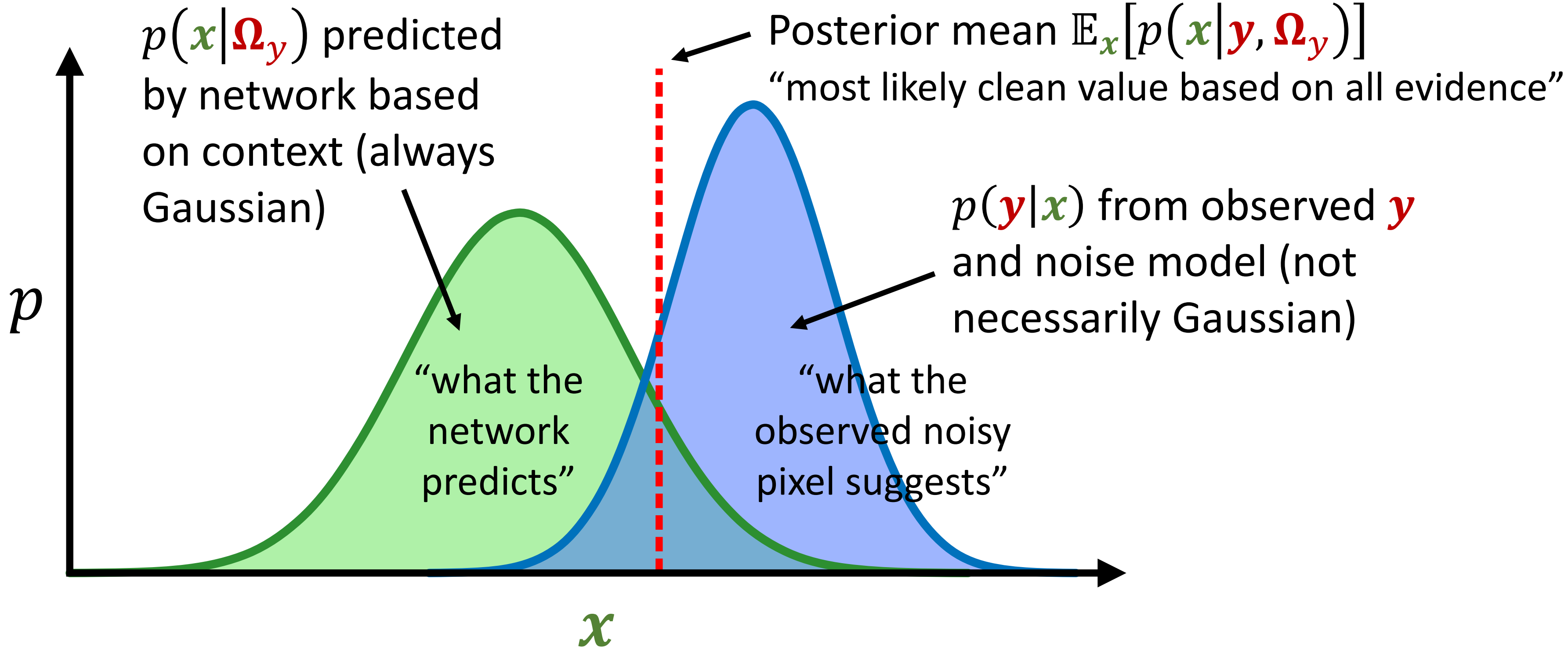
Predicted by the network

- We can make our best guess of $\boldsymbol{x}$ based on the posterior distribution

- Concretely, we output the posterior mean $\mathbb{E}_x[p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\Omega}_y)]$ because it minimizes MSE and therefore maximizes PSNR

Test-time inference — a sketch

- Simplified view of a 1D (monochromatic) case

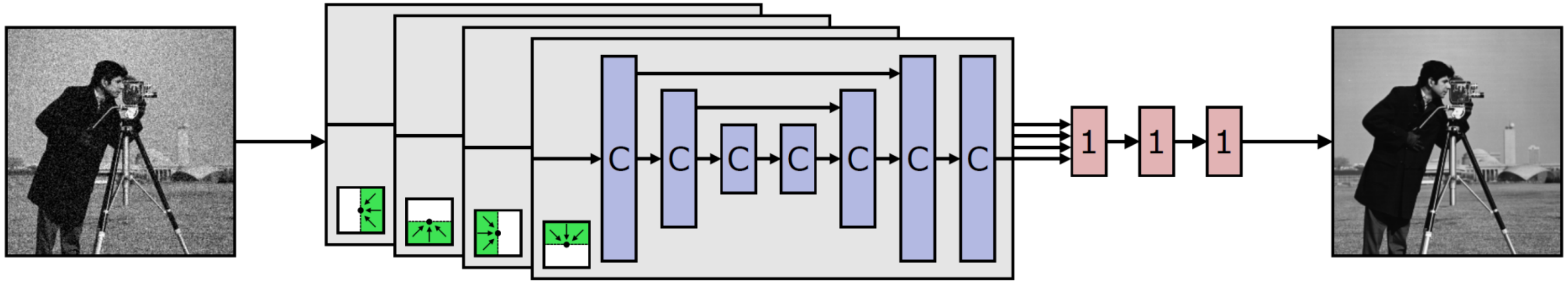


Summary of our approach

- In training phase, train neural network f_θ to map context $\boldsymbol{\Omega}_y$ to mean $\boldsymbol{\mu}_x$ and variance $\boldsymbol{\Sigma}_x$ to approximate prior $p(\boldsymbol{x}|\boldsymbol{\Omega}_y)$
  - Known noise model maps $\mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x) \to \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$ so training can be done using standard Gaussian process regression (see e.g., [Nix and Weigend, 1994])
- At test time, evaluate $f_\theta(\boldsymbol{\Omega}_y)$ and compute posterior mean $\mathbb{E}_x[p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\Omega}_y)]$ by closed-form integration

Implementing blind-spot network efficiently

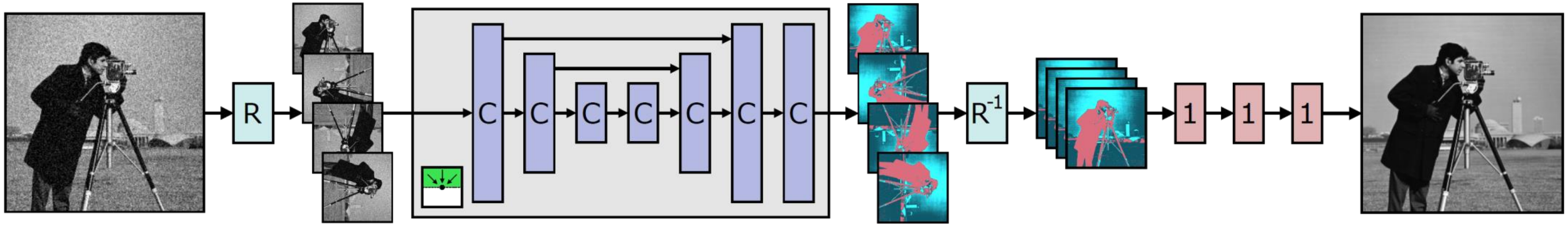
- Our solution: Combine information from four branches, each having its receptive field restricted to one direction only



- Restricting the receptive field to one half-space is easier than removing just one pixel

Optimizing a bit

- Roll the four branches into one, rotate image data instead



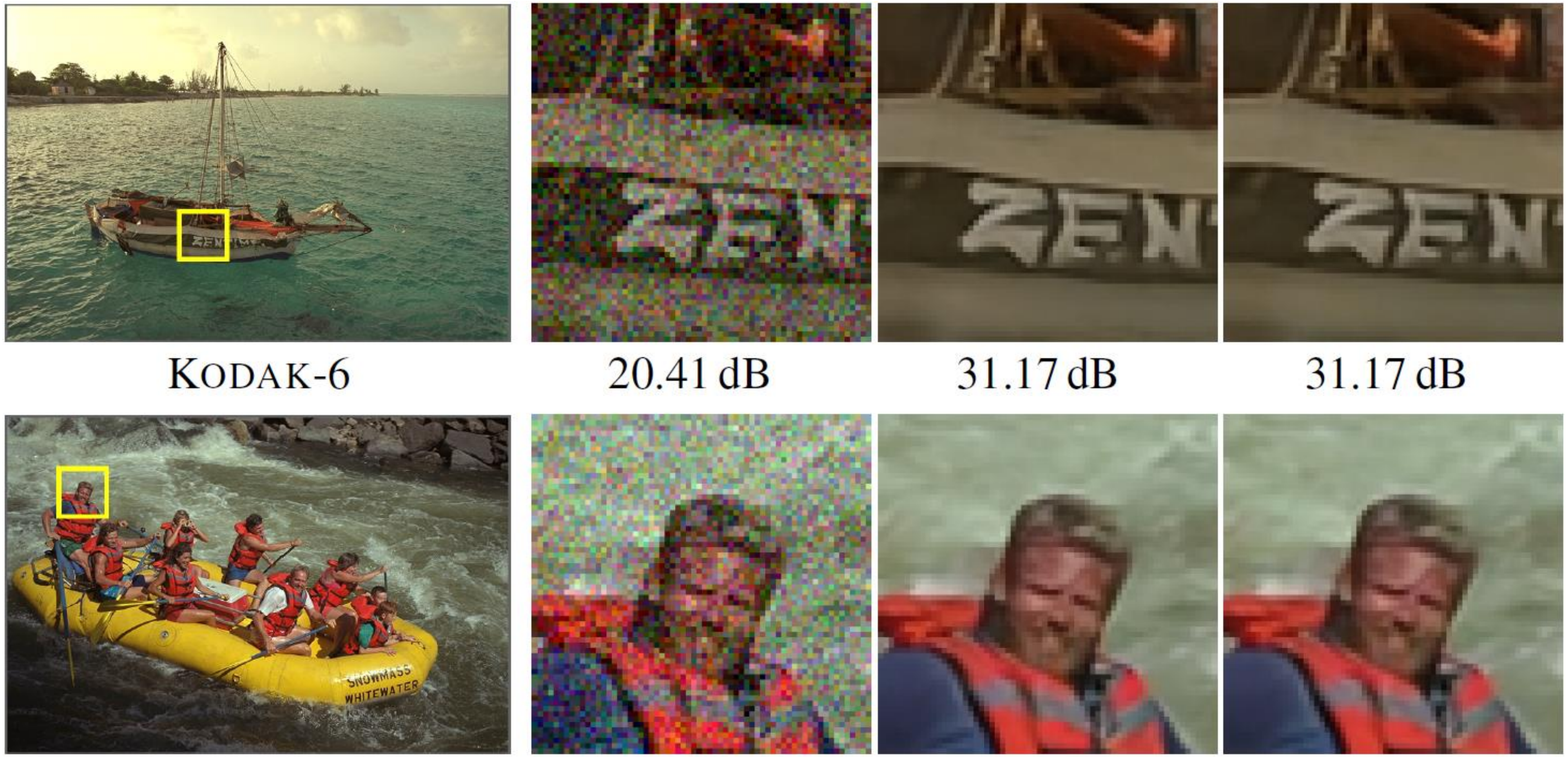
- Implicitly shares weights between branches
- Implementation details in the paper

Unknown noise parameters

- What if the noise model has an unknown parameter? What if the parameter varies for every image?
  - E.g., standard deviation σ in Gaussian noise $\mathcal{N}(\mathbf{0}, \sigma^2 I)$
- We show that these can be estimated from the data as well, so that each image in training and test data can have a different, unknown amount of noise
  - Requires regularization in certain cases to break ambiguity (is the image actually noisy vs. is the clean signal hard to predict) — see paper for details

Results: Gaussian noise ($\sigma = 25$)

Noisy input | Supervised training, clean training data | Our method, noisy data only



KODAK-6 | 20.41 dB | 31.17 dB | 31.17 dB

KODAK-14 | 20.42 dB | 30.88 dB | 30.80 dB

Table 1: Image quality results for Gaussian noise. Values of σ are shown in 8-bit units.

Noise type	Method	σ known?	KODAK	BSD300	SET14	Average
Gaussian $\sigma = 25$	Baseline, N2C	no	32.46	31.08	31.26	31.60
	Baseline, N2N	no	32.45	31.07	31.23	31.58
	Our	yes	32.45	31.03	31.25	31.57
	Our	no	32.44	31.02	31.22	31.56
	Our ablated, diag. $\mathbf{\Sigma}$	yes	31.60	29.91	30.58	30.70
	Our ablated, diag. $\mathbf{\Sigma}$	no	31.55	29.87	30.53	30.65
	Our ablated, $\boldsymbol{\mu}$ only	no	30.64	28.65	29.57	29.62
	CBM3D	yes	31.82	30.40	30.68	30.96
	CBM3D	no	31.81	30.40	30.66	30.96
Gaussian $\sigma \in [5, 50]$	Baseline, N2C	no	32.57	31.29	31.27	31.71
	Baseline, N2N	no	32.57	31.29	31.26	31.70
	Our	yes	32.47	31.19	31.21	31.62
	Our	no	32.46	31.18	31.13	31.59
	Our ablated, diag. $\mathbf{\Sigma}$	yes	31.59	30.06	30.54	30.73
	Our ablated, diag. $\mathbf{\Sigma}$	no	31.58	30.05	30.45	30.69
	Our ablated, $\boldsymbol{\mu}$ only	no	30.54	28.56	29.41	29.50
	CBM3D	yes	31.99	30.67	30.78	31.15
	CBM3D	no	31.99	30.67	30.72	31.13

Supervised training with clean targets

Our result when σ is known vs. estimated from data

Noise2Void: Ignore $\boldsymbol{y}$ and predict based on context only

Our results are within 0.04 dB from supervised training

Table 1: Image quality results for Gaussian noise. Values of σ are shown in 8-bit units.

Noise type	Method	σ known?	KODAK	BSD300	SET14	Average
Gaussian $\sigma = 25$	Baseline, N2C	no	32.46	31.08	31.26	31.60
	Baseline, N2N	no	32.45	31.07	31.23	31.58
	Our	yes	32.45	31.03	31.25	31.57
	Our	no	32.44	31.02	31.22	31.56
	Our ablated, diag. Σ	yes	31.60	29.91	30.58	30.70
	Our ablated, diag. Σ	no	31.55	29.87	30.53	30.65
	Our ablated, μ only	no	30.64	28.65	29.57	29.62
	CBM3D	yes	31.82	30.40	30.68	30.96
	CBM3D	no	31.81	30.40	30.66	30.96
Gaussian $\sigma \in [5, 50]$	Baseline, N2C	no	32.57	31.29	31.27	31.71
	Baseline, N2N	no	32.57	31.29	31.26	31.70
	Our	yes	32.47	31.19	31.21	31.62
	Our	no	32.46	31.18	31.13	31.59
	Our ablated, diag. Σ	yes	31.59	30.06	30.54	30.73
	Our ablated, diag. Σ	no	31.58	30.05	30.45	30.69
	Our ablated, μ only	no	30.54	28.56	29.41	29.50
	CBM3D	yes	31.99	30.67	30.78	31.15
	CBM3D	no	31.99	30.67	30.72	31.13

Supervised training with clean targets

Our result when σ is known vs. estimated from data

Noise2Void: Ignore $\boldsymbol{y}$ and predict based on context only

Our results are within 0.04 dB from supervised training

Table 1: Image quality results for Gaussian noise. Values of σ are shown in 8-bit units.

Noise type	Method	σ known?	KODAK	BSD300	SET14	Average
Gaussian $\sigma = 25$	Baseline, N2C	no	32.46	31.08	31.26	31.60
	Baseline, N2N	no	32.45	31.07	31.23	31.58
	Our	yes	32.45	31.03	31.25	31.57
	Our	no	32.44	31.02	31.22	31.56
	Our ablated, diag. $\mathbf{\Sigma}$	yes	31.60	29.91	30.58	30.70
	Our ablated, diag. $\mathbf{\Sigma}$	no	31.55	29.87	30.53	30.65
	Our ablated, $\boldsymbol{\mu}$ only	no	30.64	28.65	29.57	29.62
	CBM3D	yes	31.82	30.40	30.68	30.96
	CBM3D	no	31.81	30.40	30.66	30.96
Gaussian $\sigma \in [5, 50]$	Baseline, N2C	no	32.57	31.29	31.27	31.71
	Baseline, N2N	no	32.57	31.29	31.26	31.70
	Our	yes	32.47	31.19	31.21	31.62
	Our	no	32.46	31.18	31.13	31.59
	Our ablated, diag. $\mathbf{\Sigma}$	yes	31.59	30.06	30.54	30.73
	Our ablated, diag. $\mathbf{\Sigma}$	no	31.58	30.05	30.45	30.69
	Our ablated, $\boldsymbol{\mu}$ only	no	30.54	28.56	29.41	29.50
	CBM3D	yes	31.99	30.67	30.78	31.15
	CBM3D	no	31.99	30.67	30.72	31.13

Supervised training with clean targets

Our result when σ is known vs. estimated from data

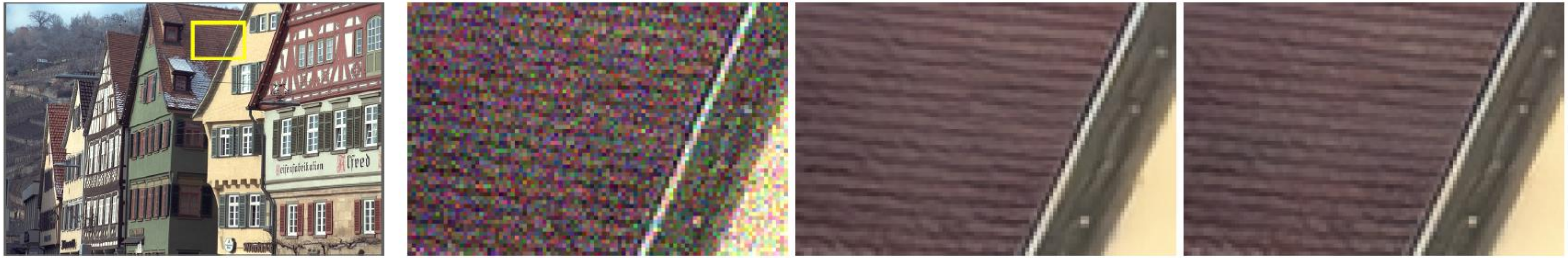
Noise2Void: Ignore $\boldsymbol{y}$ and predict based on context only

Close to baseline with variable noise ($\sigma \in [5,50]$) as well

Results: Poisson noise ($\lambda = 30$)

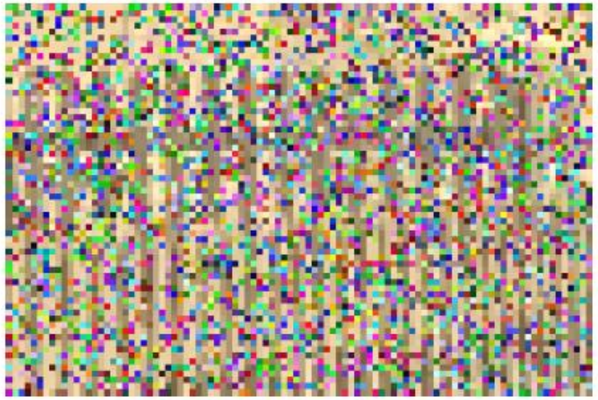
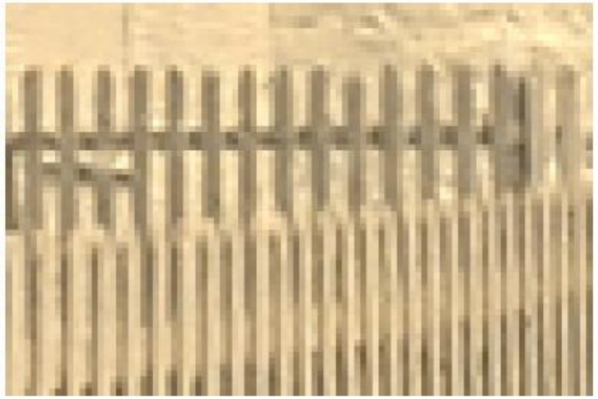
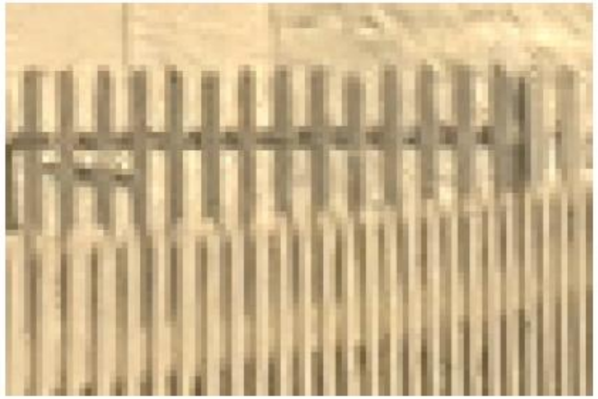
	Noisy input	Supervised training, clean training data	Our method, noisy data only

KODAK-23 | 19.13 dB | 34.83 dB | 34.63 dB



KODAK-8 | 18.63 dB | 29.12 dB | 29.11 dB

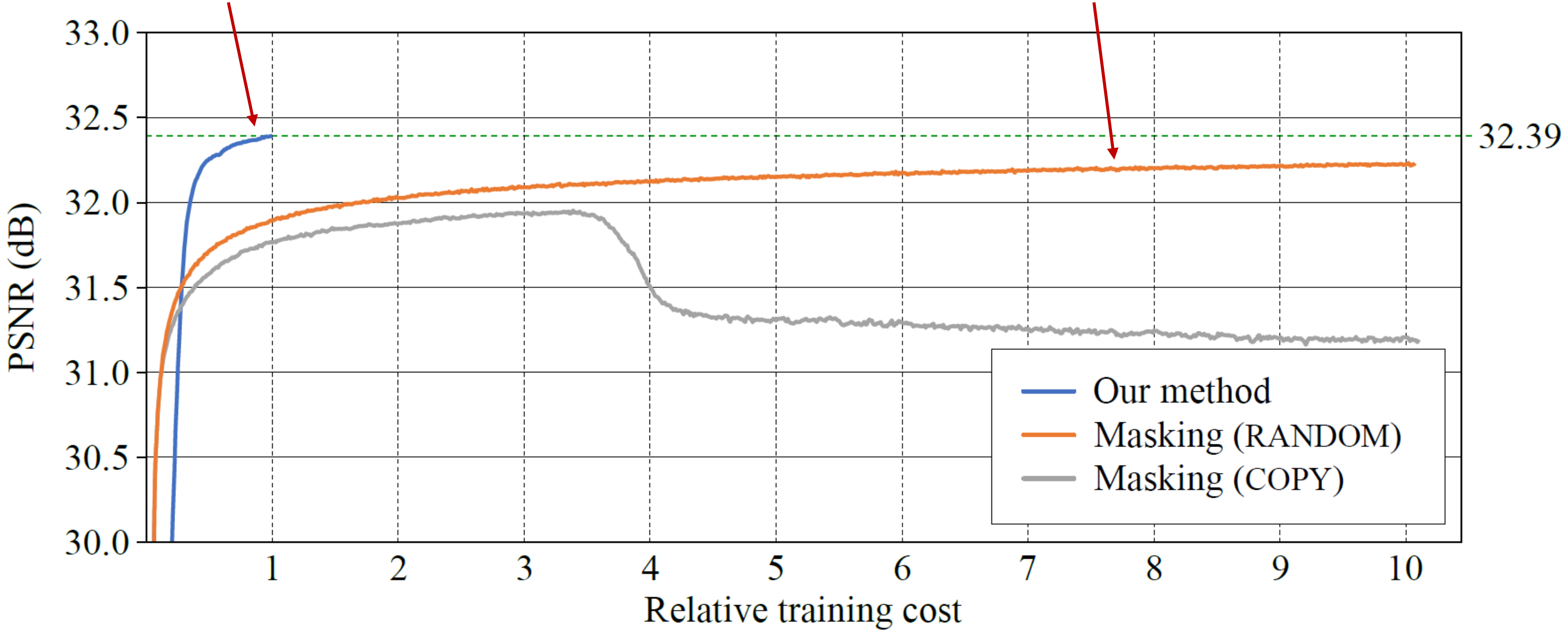
Results: Impulse noise ($\alpha = 0.5$)

	Noisy input	Supervised training, clean training data	Our method, noisy data only
			
KODAK-20	9.30 dB	34.90 dB	34.55 dB
			
KODAK-19	12.09 dB	33.62 dB	33.35 dB

Evaluation of network architecture

Our blind-spot network architecture converges quickly

Standard network architecture with masking-based training [Krull et al., 2018]



Conclusions

- Training high-quality denoisers is possible **with noisy data only**, when we have **just one** noisy realization of each training image
  - Can train a denoiser from a corpus of noisy data — no separate training set is required
- Result quality is comparable to traditionally trained networks
- Future work: Extend to more general corruptions?
  - Can we relax the assumption that noise is independent between pixels?

Thank you

Paper: https://arxiv.org/abs/1901.10277

Code: https://github.com/NVlabs/selfsupervised-denoising

Feel free to contact with any questions: slaine@nvidia.com

References

J. Batson and L. Royer. Noise2Self: Blind denoising by self-supervision. In *Proc. International Conference on Machine Learning (ICML)*, pages 524–533, 2019.

A. Krull, T.-O. Buchholz, and F. Jug. Noise2Void – Learning denoising from single noisy images. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2129–2137, 2019.

A. Krull, T. Vicar, and F. Jug. Probabilistic Noise2Void: Unsupervised content-aware denoising. *CoRR*, abs/1906.00651, 2019.

J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila. Noise2Noise: Learning image restoration without clean data. In *Proc. International Conference on Machine Learning (ICML)*, 2018.

D. A. Nix and A. S. Weigend. Estimating the mean and variance of the target probability distribution. *Proc. IEEE International Conference on Neural Networks (ICNN)*, pages 55–60, 1994.