Phase Retrieval via Deep Expectation-Consistent Approximation

Saurav K. Shastri
Dept. ECE
The Ohio State University
Columbus, OH 43210
shastri.19@osu.edu

Philip Schniter
Dept. ECE
The Ohio State University
Columbus, OH 43201
schniter.1@osu.edu

Abstract

The expectation consistent (EC) approximation framework is a state-of-the-art approach for solving (generalized) linear inverse problems with random forward operators and i.i.d. signal priors. In image inverse problems, however, both the forward operator and image pixels are structured, which plagues traditional EC implementations. In this work, we propose a novel incarnation of EC that exploits deep neural networks to handle structured operators and signals. For phase-retrieval, we propose a simplified variant called "deepECpr" that reduces to iterative denoising. In experiments recovering natural images from phaseless, shot-noise corrupted, coded-diffraction-pattern outputs, we observe accuracy surpassing the state-of-the-art prDeep (Metzler et al., 2018) and Diffusion Posterior Sampling (Chung et al., 2023) approaches with two-orders-of-magnitude complexity reduction.

1 Introduction

In nonlinear inverse problems, we observe corrupted measurements $\boldsymbol{y} \in \mathcal{Y}^m$ of a signal/image $\boldsymbol{x} \in \mathbb{R}^d$ or $\mathbb{C}^d$ and we would like to recover $\boldsymbol{x}$ from $\boldsymbol{y}$. We consider problems for which the relationship between $\boldsymbol{y}$ and $\boldsymbol{x}$ can be described using a likelihood model of the form

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{m} p_{\mathsf{y}|\mathsf{z}}(y_i|z_i) \text{ for } \boldsymbol{z} \triangleq \boldsymbol{A}\boldsymbol{x}, \tag{1}$$

where the forward operator $\boldsymbol{A} \in \mathbb{C}^{m\times d}$ and scalar measurement channel $p_{\mathsf{y}|\mathsf{z}}$ are both known. In the statistics literature, (1) is known as the generalized linear model (GLM). Versions of $p_{\mathsf{y}|\mathsf{z}}$ exist for, e.g., additive noise of an arbitrary distribution, logistic regression [1], Poisson regression [2], noisy quantization [3], and phase retrieval [4, 5]. In this work, we focus on phase retrieval, although many of the ideas that we describe can be applied more generally.

For phase retrieval, although many forward operators $\boldsymbol{A}$ can be considered, the most common are the (possibly oversampled) 2D Fourier transform and the coded diffraction pattern (CDP) [6] in (2), where $\boldsymbol{F}$ is the unitary 2D Fourier transform and $\{\boldsymbol{D}_k\}_{k=1}^K$ are diagonal matrices with entries drawn independently and uniformly from the unit circle in the complex plane. Although several choices of $p_{\mathsf{y}|\mathsf{z}}$ have been applied to phase retrieval, we will focus on the choice in (2) since it is tractable and has been shown to work well in a variety of real-world settings [7].

$$\mathbf{A} = \frac{1}{\sqrt{K}} \begin{bmatrix} \mathbf{F}\mathbf{D}_1 \\ \vdots \\ \mathbf{F}\mathbf{D}_K \end{bmatrix}, \qquad p_{\mathsf{y}|\mathsf{z}}(y|z) = \exp\big(-\tfrac{1}{2v}(y-|z|)^2\big). \tag{2}$$

Various computational approaches have been proposed for image phase retrieval. Classical methods, like the Gerchberg-Saxton [8] and Hybrid Input-Output [9] algorithms are based on iterative projection. A more modern approach is to minimize the negative log-likelihood, i.e.,

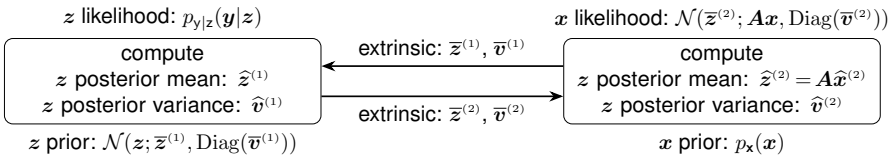

Figure 1: Summary of EC applied to the GLM in (1). The algorithm iteratively estimates the transform outputs $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x}$ by alternating between two estimation modules that exchange extrinsic messages.

$\arg\min_{\boldsymbol{x}}\{-\ln p_{\mathsf{y}|\mathsf{z}}(\boldsymbol{y}|\boldsymbol{A}\boldsymbol{x})\}$, using gradient-based iterative methods with a spectral initialization [10–12]. Although convex-relaxation-based methods like PhaseLift [13] have also been proposed (see the overview in [14]), they tend to be computationally impractical at typical image sizes. Approximate message-passing (AMP) algorithms have been proposed for phase retrieval [15, 16] that are near-optimal for high-dimensional i.i.d. or rotationally invariant random $\boldsymbol{A}$ [17, 18], but they tend to diverge with Fourier or CDP $\boldsymbol{A}$.

The aforementioned phase-retrieval methods do not exploit sophisticated structure among the pixels in $\boldsymbol{x}$, as occurs in imaging applications. To exploit this prior knowledge, several approaches have been proposed that involve deep neural networks. For example, using the plug-and-play (PnP) [19] or RED [20] frameworks, one can iteratively alternate between negative-log-likelihood reduction and neural-network based image denoising [21, 22]. Or, inspired by CSGM [23], when given an image-generator network $\boldsymbol{x} = g_{\boldsymbol{\theta}}(\boldsymbol{z})$, one could search for the code vector $\boldsymbol{z}$ such that the magnitude of $\boldsymbol{A}g_{\boldsymbol{\theta}}(\boldsymbol{z})$ matches the phaseless measurements $\boldsymbol{y}$ [24]. A variation on this idea, inspired by DIP [25], is to optimize the generator parameters $\boldsymbol{\theta}$ instead of the code vector $\boldsymbol{z}$ [26, 27]. The most recent trend is to use a diffusion method, like DPS [28], for phase retrieval.

In this work, we propose a novel approach to phase retrieval that builds on the expectation consistent (EC) approximation algorithm from [29]. Although there are connections to AMP algorithms like [16], our approach does not require random $\boldsymbol{A}$. And although our approach involves iterative denoising (like with plug-and-play, RED, and diffusion methods) it converges two orders-of-magnitude faster.

2 Proposed Approach

The expectation-consistent (EC) approximation algorithm [29] is a message-passing algorithm for iterative inference of vector-valued variables. Unlike the sum-product algorithm, it passes mean and variance messages rather than full probability distributions. More precisely, EC is a parallel version of the expectation propagation (EP) algorithm [30] that (locally) minimizes a known cost function. Over the last decade, EC has become famous as a method that admits rigorous analysis, and in some cases optimal performance, in high-dimensional random settings. Perhaps the best known setting is the standard linear model (i.e., the GLM (1) with additive Gaussian $p_{\mathsf{y}|\mathsf{z}}$) with asymptotically high-dimensional rotationally invariant $\boldsymbol{A}$ and i.i.d. signal $\boldsymbol{x}$. This application of EC is known as Vector AMP (VAMP), which obeys a rigorous state-evolution whose fixed points are minimum mean-squared error (MMSE)-optimal whenever they are unique [31]. EC has also been proposed for the GLM [16] (see also [32] and [33]) and rigorously analyzed in [34] under the same random-$\boldsymbol{A}$ and i.i.d.-$\boldsymbol{x}$ assumptions. Although the i.i.d.-$\boldsymbol{x}$ assumption was circumvented in [35], random-$\boldsymbol{A}$ remains an important ingredient in existing applications/analyses of EC.

In image inverse problems, applying EC is challenging because $\boldsymbol{A}$ is not random. Consequently, EC behaves unpredictably and may not even converge [36]. Although some heuristic work-arounds have been proposed for the case of magnetic resonance imaging [37, 38], successful applications of EC to imaging inverse problems have remained elusive.

Figure 1 shows how EC iteratively estimates the transform outputs $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x}$ by alternating between two estimation modules. The left module computes the posterior mean and (pixel-wise) variance of $\{z_i\}_{i=1}^m$ using the likelihoods $p_{\mathsf{y}|\mathsf{z}}(y_i|\cdot)$ and a Gaussian prior informed by the extrinsic message $\{(\overline{z}_i^{(1)}, \overline{v}_i^{(1)})\}_{i=1}^m$ received from the right module:

$$\forall i: \begin{array}{l} \widehat{z}_i^{(1)} = \mathrm{E}\{z_i|y_i; \overline{z}_i^{(1)}, \overline{v}_i^{(1)}\} \\ \widehat{v}_i^{(1)} = \mathrm{var}\{z_i|y_i; \overline{z}_i^{(1)}, \overline{v}_i^{(1)}\} \end{array} \text{ via } p(z_i|y_i; \overline{z}_i^{(1)}, \overline{v}_i^{(1)}) = \frac{p_{\mathsf{y}|\mathsf{z}}(y_i|z_i)\mathcal{N}(z_i; \overline{z}_i^{(1)}, \overline{v}_i^{(1)})}{\int p_{\mathsf{y}|\mathsf{z}}(y_i|z_i)\mathcal{N}(z_i; \overline{z}_i^{(1)}, \overline{v}_i^{(1)})\,\mathrm{d}z_i}. \quad (3)$$

Meanwhile, the right module computes the posterior mean and covariance of $\boldsymbol{x}$ using the prior p_{x} and a linear-Gaussian likelihood informed by the extrinsic message from the left module:

$$\begin{array}{l}\widehat{\boldsymbol{x}}^{(2)} = \mathrm{E}\{\boldsymbol{x}|\boldsymbol{y};\overline{\boldsymbol{z}}^{(2)},\overline{\boldsymbol{v}}^{(2)}\} \\ \widehat{\boldsymbol{C}}^{(2)} = \mathrm{Cov}\{\boldsymbol{x}|\boldsymbol{y};\overline{\boldsymbol{z}}^{(2)},\overline{\boldsymbol{v}}^{(2)}\}\end{array} \text{ via } p(\boldsymbol{x}|\boldsymbol{y};\overline{\boldsymbol{z}}^{(2)},\overline{\boldsymbol{v}}^{(2)}) = \frac{\mathcal{N}(\overline{\boldsymbol{z}}^{(2)};\boldsymbol{A}\boldsymbol{x},\mathrm{Diag}(\overline{\boldsymbol{v}}^{(2)}))\,p_{\mathsf{x}}(\boldsymbol{x})}{\int\mathcal{N}(\overline{\boldsymbol{z}}^{(2)};\boldsymbol{A}\boldsymbol{x},\mathrm{Diag}(\overline{\boldsymbol{v}}^{(2)}))\,p_{\mathsf{x}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}, \quad (4)$$

and then uses those quantities to computes the posterior mean and (pixel-wise) variance of $\boldsymbol{z}$:

$$\widehat{\boldsymbol{z}}^{(2)} = \boldsymbol{A}\widehat{\boldsymbol{x}}^{(2)} \quad \text{and} \quad \widehat{\boldsymbol{v}}^{(2)} = \mathrm{diag}(\boldsymbol{A}\widehat{\boldsymbol{C}}^{(2)}\boldsymbol{A}^{\mathsf{H}}). \quad (5)$$

Finally, the messages passed between the two modules take the form

$$\forall i: \begin{array}{l}\overline{v}_i^{(2)} = \big(1/\widehat{v}_i^{(1)} - \gamma^{(1)}/\overline{v}_i^{(1)}\big)^{-1} \\ \overline{z}_i^{(2)} = \big(\widehat{z}_i^{(1)}/\widehat{v}_i^{(1)} - \gamma^{(1)}\overline{z}_i^{(1)}/\overline{v}_i^{(1)}\big)\overline{v}_i^{(2)}\end{array} \text{ and } \begin{array}{l}\overline{v}_i^{(1)} = \big(1/\widehat{v}_i^{(2)} - \gamma^{(2)}/\overline{v}_i^{(2)}\big)^{-1} \\ \overline{z}_i^{(1)} = \big(\widehat{z}_i^{(2)}/\widehat{v}_i^{(2)} - \gamma^{(2)}\overline{z}_i^{(2)}/\overline{v}_i^{(2)}\big)\overline{v}_i^{(1)}.\end{array} \quad (6)$$

which are "extrinsic" when $\gamma^{(1)} = 1 = \gamma^{(2)}$. When $\boldsymbol{x}$ is i.i.d. and $\boldsymbol{A}$ is a large, rotationally invariant random matrix, it is possible to avoid the high-dimensional integral and posterior covariance matrix $\widehat{\boldsymbol{C}}^{(2)}$ in (4), as detailed in [16, 32–34]. But what can be done when $\boldsymbol{x}$ and $\boldsymbol{A}$ are structured and non-random, as in most imaging applications?

For GLM image recovery, we propose to use EC as above, but with (4)-(5) approximated as follows:

1. Compute the posterior mean $\widehat{\boldsymbol{x}}^{(2)}$ using a neural network $f_{\boldsymbol{\theta}}(\overline{\boldsymbol{z}}^{(2)};\overline{\boldsymbol{v}}^{(2)})$ trained to minimize $J_f(\boldsymbol{\theta}) = \sum_{t=1}^{T}\mathrm{E}\,\|\boldsymbol{x}_t - \widehat{\boldsymbol{x}}_t^{(2)}\|^2$, with training $\{\boldsymbol{x}_t\}_{t=1}^{T}$, output $\widehat{\boldsymbol{x}}_t^{(2)} = f_{\boldsymbol{\theta}}(\boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{e}_t;\overline{\boldsymbol{v}}^{(2)})$, noise $\boldsymbol{e}_t \sim \mathcal{N}(\boldsymbol{0},\mathrm{Diag}(\overline{\boldsymbol{v}}^{(2)}))$, and variances $\overline{v}_i^{(2)} \sim$ i.i.d. $\mathrm{Unif}[0, v_{\max}]$ for some $v_{\max}$.
2. Compute the posterior variances $\widehat{\boldsymbol{v}}^{(2)}$ using a neural network $h_{\boldsymbol{\phi}}(\overline{\boldsymbol{z}}^{(2)};\overline{\boldsymbol{v}}^{(2)})$ trained to minimize $J_h(\boldsymbol{\phi}) = \sum_{t=1}^{T}\mathrm{E}\,\||\boldsymbol{z}_t - \widehat{\boldsymbol{z}}_t^{(2)}|^{\odot 2} - \widehat{\boldsymbol{v}}_t^{(2)}\|^2$, with true $\boldsymbol{z}_t = \boldsymbol{A}\boldsymbol{x}_t$, estimated $\widehat{\boldsymbol{z}}_t^{(2)} = \boldsymbol{A}\widehat{\boldsymbol{x}}_t^{(2)}$, elementwise square $(\cdot)^{\odot 2}$, and $\widehat{\boldsymbol{v}}_t^{(2)} = h_{\boldsymbol{\phi}}(\overline{\boldsymbol{z}}_t^{(2)};\overline{\boldsymbol{v}}^{(2)})$. This avoids the computation of $\widehat{\boldsymbol{C}}^{(2)}$.

For phase retrieval, we make additional simplifications, some of which exploit $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A} = \boldsymbol{I}$, which holds for both the (possibly oversampled) Fourier and CDP (2) incarnations of $\boldsymbol{A}$.

1. Use the Laplace approximation [39] of $(\widehat{z}_i^{(1)}, \widehat{v}_i^{(1)})$ in (3), computable in closed-form.
2. Approximate the variance vector $\overline{\boldsymbol{v}}^{(2)}$ by $\boldsymbol{1}$ times its average value $\overline{v}^{(2)} \triangleq \frac{1}{m}\boldsymbol{1}^{\top}\overline{\boldsymbol{v}}^{(2)}$.
3. Replace the linearized measurement $\overline{\boldsymbol{z}}_t^{(2)} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{e}_t$, where $\boldsymbol{e}_t \sim \mathcal{N}(\boldsymbol{0},\overline{v}^{(2)}\boldsymbol{I})$, with the sufficient statistic $\overline{\boldsymbol{r}}_t^{(2)} \triangleq \boldsymbol{A}^{\mathsf{H}}\overline{\boldsymbol{z}}_t^{(2)} = \boldsymbol{x}_t + \boldsymbol{\epsilon}_t$, where again $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0},\overline{v}^{(2)}\boldsymbol{I})$. Consequently, estimating $\boldsymbol{x}_t$ from $\overline{\boldsymbol{r}}_t^{(2)}$ becomes a *denoising* task, for which we use $\widehat{\boldsymbol{x}}^{(2)} = f_{\boldsymbol{\theta}}(\overline{\boldsymbol{r}}^{(2)},\overline{v}^{(2)}\boldsymbol{1})$.
4. Approximate $\widehat{\boldsymbol{v}}^{(2)} = h_{\boldsymbol{\phi}}(\overline{\boldsymbol{z}}^{(2)};\overline{v}^{(2)}\boldsymbol{1})$ by $\boldsymbol{1}$ times $\widehat{v}^{(2)} \triangleq \frac{1}{m}\|\widehat{\boldsymbol{z}}^{(2)} - \overline{\boldsymbol{z}}^{(2)}\|$. This is admittedly heuristic, but it works well empirically and avoids the need to use (and hence train) $h_{\boldsymbol{\phi}}$.

The resulting phase-retrieval algorithm, which we'll refer to as "deepECpr," performs iterative denoising, similar to PnP or RED. However, it requires many fewer iterations, as we show next.

3 Numerical Experiments

To compare to prDeep [21], we repeat one of the experiments in [21], where 128×128 grayscale images from Set12 [40] were recovered from phaseless CDP measurements $\boldsymbol{y}$ under the noise model

$$y_i^2 = |z_i|^2 + w_i \text{ with } w_i \sim \mathcal{N}(0, \alpha^2|z_i|^2). \quad (7)$$

This approximates shot-noise corruption at noise level α since y_i^2/α^2 is approximately $\mathrm{Poisson}(|z_i|^2/\alpha^2)$ [21]. For prDeep, we use the authors' implementation [41] under default settings, which uses $p_{\mathsf{y}|\mathsf{z}}$ from (2) and a bank of four Matlab DnCNN denoisers [40] trained on the BSD400 dataset [42]. For deepECpr, we use the same $p_{\mathsf{y}|\mathsf{z}}$ with two Python DnCNN denoisers trained on BSD400 using the bias-free approach from [43]. For (6), deepECpr uses $\gamma^{(1)} = 1$ and $\gamma^{(2)} = 0$. We also compare to the classical HIO algorithm [9], as implemented in [41], using 1000 iterations. Both prDeep and deepECpr were initialized using 50 iterations of HIO, and they set v in (2) at the variance of $y_i - |z_i|$ averaged over the training data. See the Appendix for further details.

Table 1: Average PSNR and SSIM at various shot-noise levels α for Set12 test images

	$\alpha = 9$		$\alpha = 18$		$\alpha = 27$		
method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	denoiser calls
HIO	30.92	0.9557	25.79	0.8568	22.44	0.7413	-
prDeep	38.68	0.9895	34.34	0.9760	31.91	0.9622	800
deepECpr	**39.02**	**0.9909**	**34.5**	**0.9777**	**32.04**	**0.9635**	**25**

Table 2: Improvement of deepECpr over prDeep versus α

α	ΔPSNR $\pm$ SE	ΔSSIM $\pm$ SE
9	0.34 ± 0.09	0.0013 ± 0.0003
18	0.16 ± 0.06	0.0017 ± 0.0004
27	0.13 ± 0.06	0.0012 ± 0.0004

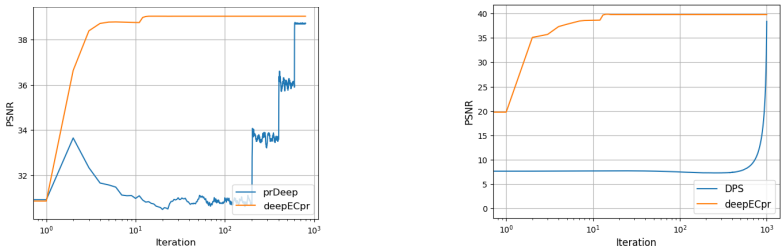

Figure 2: Average PSNR versus iteration for Set12 test data with shot-noise level $\alpha = 9$ (left) and FFHQ test data with shot-noise level $\alpha = 18$ (right).

Table 1 shows PSNR and SSIM averaged over the Set12 test images, where the proposed deepECpr outperformed prDeep and HIO at all three noise levels α. Table 2 shows that the PSNR and SSIM gains are statistically significant. Furthermore, Fig. 2 shows that deepECpr converges more smoothly, and two orders-of-magnitude faster, than prDeep. Example reconstruction plots in the Appendix show that deepECpr obtained superior visual quality relative to HIO and prDeep.

In a second experiment, we compare to the recent "DPS" conditional diffusion method from [28], again using phaseless CDP measurements at $K = 4$ with shot noise from (7). As in [28], we use 256×256 RGB images from the FFHQ dataset [44] and the DPS codebase from [45], which includes a pre-trained unconditional FFHQ diffusion model. For deepECpr, this unconditional diffusion model is rescaled to act as a denoiser, as in [28, eq. (9)]. Both DPS and deepECpr are configured to use $p_{\mathsf{y}|\mathsf{z}}$ from (2). As before, 1000-iteration HIO is used as a baseline, 50-iteration HIO is used to initialize deepECpr, v in (2) is set at the variance of $y_i - |z_i|$ averaged over the training data, and deepECpr used $\gamma^{(1)} = 1$ and $\gamma^{(2)} = 0$.

Table 3: Average PSNR and SSIM at various shot-noise levels α for FFHQ test images

	$\alpha = 9$		$\alpha = 18$		$\alpha = 27$		
method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	denoiser calls
HIO	20.84	0.7821	20.02	0.5519	18.89	0.3959	-
DPS	41.98	0.9803	38.4	0.9619	36.14	0.9432	1000
deepECpr	**43.52**	**0.9862**	**39.79**	**0.9724**	**37.47**	**0.9570**	**15**

Table 4: Improvement of deepECpr over DPS versus α

α	ΔPSNR $\pm$ SE	ΔSSIM $\pm$ SE
9	1.54 ± 0.03	0.0059 ± 0.0007
18	1.39 ± 0.02	0.0105 ± 0.0009
27	1.34 ± 0.03	0.0138 ± 0.0006

Tables 3 and 4 show PSNR and SSIM averaged over the FFHQ test images, where the proposed deepECpr outperformed DPS and HIO at all three noise levels α. Furthermore, Fig. 2 shows that deepECpr converges two orders-of-magnitude faster than prDeep. Example reconstruction plots in the Appendix show that deepECpr obtained superior visual quality over HIO and DPS.

In a third experiment, we compare deepECpr to DPS and HIO using phaseless CDP measurements at $K = 4$ with additive Gaussian noise, and we observe similar results. Please see the Appendix.

4 Conclusion

For generalized linear models, we proposed a novel variant of expectation consistent (EC) approximation [29] that exploits deep neural networks. Unlike the traditional EC implementations, the proposed "deepEC" framework does not require random forward operators nor an i.i.d. signal prior. For phase retrieval, we proposed a simplified variant called "deepECpr." In experiments recovering natural images from phaseless, shot-noise corrupted, coded-diffraction-pattern outputs, we observed

deepECpr outperforming the state-of-the-art prDeep [21] and the DPS [28] methods in reconstruction accuracy, while reducing complexity by two orders of magnitude.

Acknowledgments and Disclosure of Funding

The authors are funded in part by the National Science Foundation under grant CCF-1955587.

References

[1] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd ed., 2009. 1

[2] R. M. Willett, R. F. Marcia, and J. M. Nichols, "Compressed sensing for practical optical imaging systems: A tutorial," *Optical Eng.*, vol. 50, July 2011. 1

[3] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Prentice-Hall, 1984. 1

[4] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, "Phase retrieval with application to optical imaging: A contemporary overview," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 87–109, 2015. 1

[5] J. Dong, L. Valzania, A. Maillard, T.-a. Pham, S. Gigan, and M. Unser, "Phase retrieval: From computational imaging to machine learning: A tutorial," *IEEE Signal Process. Mag.*, vol. 40, no. 1, pp. 45–57, 2023. 1

[6] E. J. Candès, X. Li, and M. Soltanolkotabi, "Phase retrieval from coded diffraction patterns," *Appl. Comput. Harmonic Anal.*, vol. 39, no. 2, pp. 277–299, 2015. 1

[7] L.-H. Yeh, J. Dong, J. Zhong, L. Tian, M. Chen, G. Tang, M. Soltanolkotabi, and L. Waller, "Experimental robustness of Fourier ptychography phase retrieval algorithms," *Optical Eng.*, vol. 23, no. 26, pp. 33214–33240, 2015. 1

[8] R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of the phase from image and diffraction plane pictures," *Optika*, vol. 35, no. 2, pp. 237–246, 1972. 1

[9] J. R. Fienup, "Phase retrieval algorithms: A comparison," *Appl. Optics*, vol. 21, pp. 2758–2769, Aug. 1982. 1, 3

[10] E. J. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval via Wirtinger flow: Theory and algorithms," *IEEE Trans. Inf. Theory*, vol. 61, no. 4, pp. 1985–2007, 2015. 2

[11] M. Mondelli and A. Montanari, "Fundamental limits of weak recovery with applications to phase retrieval," in *Proc. Conf. Learn. Theory*, pp. 1445–1450, 2018. 2

[12] W. Luo, W. Alghamdi, and Y. M. Lu, "Optimal spectral initialization for signal recovery with applications to phase retrieval," *IEEE Trans. Signal Process.*, vol. 67, no. 9, pp. 2347–2356, 2019. 2

[13] E. J. Candès, T. Strohmer, and V. Voroninski, "PhaseLift: Exact and stable signal recovery from magnitude measurements via convex programming," *Commun. Pure & Appl. Math.*, vol. 66, no. 8, pp. 1241–1274, 2013. 2

[14] A. Fannjiang and T. Strohmer, "The numerics of phase retrieval," *Acta Numerica*, vol. 29, pp. 125–228, 2020. 2

[15] P. Schniter and S. Rangan, "Compressive phase retrieval via generalized approximate message passing," *IEEE Trans. Signal Process.*, vol. 63, pp. 1043–1055, Feb. 2015. 2

[16] P. Schniter, S. Rangan, and A. K. Fletcher, "Vector approximate message passing for the generalized linear model," in *Proc. Asilomar Conf. Signals Syst. Comput.*, pp. 1525–1529, 2016. 2, 3

[17] A. Maillard, B. Loureiro, F. Krzakala, and L. Zdeborová, "Phase retrieval in high dimensions: Statistical and computational phase transitions," in *Proc. Neural Inf. Process. Syst. Conf.*, vol. 33, pp. 11071–11082, 2020. 2

[18] M. Mondelli and R. Venkataramanan, "Approximate message passing with spectral initialization for generalized linear models," in *Proc. Int. Conf. Artificial Intell. Statist.*, pp. 397–405, 2021. 2

[19] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proc. IEEE Global Conf. Signal Inf. Process.*, pp. 945–948, 2013. 2

[20] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *SIAM J. Imag. Sci.*, vol. 10, no. 4, pp. 1804–1844, 2017. 2

[21] C. A. Metzler, P. Schniter, A. Veeraraghavan, and R. G. Baraniuk, "prDeep: Robust phase retrieval with flexible deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, pp. 3501–3510, 2018. 2, 3, 5, 7

[22] X. Chang, L. Bian, and J. Zhang, "Large-scale phase retrieval," *eLight*, vol. 1, no. 1, pp. 1–12, 2021. 2

[23] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed sensing using generative models," in *Proc. Int. Conf. Mach. Learn.*, pp. 537–546, 2017. 2

[24] P. Hand, O. Leong, and V. Voroninski, "Phase retrieval under a generative prior," in *Proc. Neural Inf. Process. Syst. Conf.*, vol. 31, 2018. 2

[25] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proc. IEEE Conf. Comp. Vision Pattern Recog.*, 2018. 2

[26] F. Wang, Y. Bian, H. Wang, M. Lyu, G. Pedrini, W. Osten, G. Barbastathis, and G. Situ, "Phase imaging with an untrained neural network," *Light: Science & Applications*, vol. 9, no. 1, p. 77, 2020. 2

[27] E. Bostan, R. Heckel, M. Chen, M. Kellman, and L. Waller, "Deep phase decoder: Self-calibrating phase microscopy with an untrained deep neural network," *Optica*, vol. 7, no. 6, pp. 559–562, 2020. 2

[28] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye, "Diffusion posterior sampling for general noisy inverse problems," in *Proc. Int. Conf. on Learn. Rep.*, 2023. 2, 4, 5, 7

[29] M. Opper and O. Winther, "Expectation consistent approximate inference," *J. Mach. Learn. Res.*, vol. 1, pp. 2177–2204, 2005. 2, 4

[30] T. Minka, *A Family of Approximate Algorithms for Bayesian Inference*. PhD thesis, Dept. Comp. Sci. Eng., MIT, Cambridge, MA, USA, 2001. 2

[31] S. Rangan, P. Schniter, and A. K. Fletcher, "Vector approximate message passing," *IEEE Trans. Inf. Theory*, vol. 65, pp. 6664–6684, Oct. 2019. 2

[32] H. He, C.-K. Wen, and S. Jin, "Generalized expectation consistent signal recovery for nonlinear measurements," in *Proc. IEEE Int. Symp. Inf. Theory*, pp. 2333–2337, 2017. 2, 3

[33] X. Meng, S. Wu, and J. Zhu, "A unified Bayesian inference framework for generalized linear models," *IEEE Signal Process. Lett.*, vol. 25, no. 3, pp. 398–402, 2018. 2, 3

[34] P. Pandit, M. Sahraee-Ardakan, S. Rangan, P. Schniter, and A. K. Fletcher, "Inference with deep generative priors in high dimensions," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 336–347, 2020. 2, 3

[35] A. K. Fletcher, P. Pandit, S. Rangan, S. Sarkar, and P. Schniter, "Plug-in estimation in high-dimensional linear inverse problems: A rigorous analysis," in *Proc. Neural Inf. Process. Syst. Conf.*, pp. 7440–7449, 2018. 2

[36] A. K. Fletcher, M. Sahraee-Ardakan, S. Rangan, and P. Schniter, "Expectation consistent approximate inference: Generalizations and convergence," in *Proc. IEEE Int. Symp. Inf. Theory*, pp. 190–194, 2016. 2

[37] C. Millard, A. T. Hess, B. Mailhé, and J. Tanner, "Approximate message passing with a colored aliasing model for variable density Fourier sampled images," *IEEE Open J. Signal Process.*, vol. 1, pp. 146–158, 2020. 2

[38] S. K. Shastri, R. Ahmad, C. A. Metzler, and P. Schniter, "Denoising generalized expectation-consistent approximation for mr image recovery," *IEEE J. Sel. Areas Inf. Theory*, vol. 3, no. 3, pp. 528–542, 2022. 2

[39] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2007. 3

[40] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, pp. 3142–3155, July 2017. 3, 7

[41] C. A. Metzler, "prdeep." `https://github.com/ricedsp/prDeep/tree/master`, 2018. 3, 7

[42] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, pp. 416–423, July 2001. 3

[43] S. Mohan, Z. Kadkhodaie, E. P. Simoncelli, and C. Fernandez-Granda, "Robust and interpretable blind image denoising via bias-free convolutional neural networks," in *Proc. Int. Conf. on Learn. Rep.*, 2020. 3, 7

[44] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE Conf. Comp. Vision Pattern Recog.*, pp. 4396–4405, 2019. 4, 7

[45] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye, "diffusion-posterior-sampling." `https://github.com/ricedsp/prDeep/tree/master`, Mar. 2023. 4, 7

Appendix

Test images: For our grayscale-image experiments, we test performance using all 12 images from the Set12 dataset [40]. For our RGB-image experiments, we test performance using the first 30 images from the FFHQ dataset [44].

Algorithm parameters: HIO was run for 1000 iterations using the code from [41]. prDeep was run using the authors' code from [41] under default settings. DPS was run using the authors' code from [45], modified to accommodate our $p_{\mathsf{y}|\mathsf{z}}$ from (2). In DPS, tuning the hyperparameter v in $p_{\mathsf{y}|\mathsf{z}}$ is equivalent to tuning the step-size on the gradient of the log likelihood, which we did by maximizing the PSNR on validation data that comprised FFHQ images 31-60.

Denoisers: As described in [21], prDeep uses four Matlab DnCNN denoisers trained on BSD400 at noise standard deviation $\sigma = 60$, 40, 20, and 10, respectively, where 255 is the maximum pixel intensity. First prDeep is run for 200 iterations using the $\sigma = 60$ DnCNN, then the result is improved by running another 200 iterations using the $\sigma = 40$ DnCNN, and so on, for a total of 800 iterations. For our grayscale-image experiment, deepECpr uses two bias-free [43] Python DnCNN denoisers trained on BSD400 with σ drawn uniformly over the intervals $[0, 10]$ and $[0, 5]$, respectively. First deepECpr is run for 15 iterations using the first denoiser, and then the result is improved by running for 10 iterations using the second denoiser. For the RGB-image experiments, deepECpr used the pre-trained FFHQ unconditional diffusion model from [28] as a denoiser. Since this denoiser accepts a noise variance, we feed it $\overline{v}^{(2)}$ for the first 10 iterations, but use $\overline{v}^{(2)} = 0$ for the remaining 5 iterations.

Additional experiments: We also considered the Gaussian noise corruption model:

$$y_i = |z_i| + w_i \text{ with } w_i \sim \mathcal{N}(0, v), \tag{8}$$

where v controls the variance of the additive white Gaussian noise. Table 5 demonstrate that deepECpr performs better than both HIO and DPS in this scenario, and Table 6 suggests that the performance gap between deepECpr and DPS is statistically significant.

Table 5: Average PSNR and SSIM at various Gaussian noise levels $\sqrt{v}$ for FFHQ test images

	$\sqrt{v} = 0.04$		$\sqrt{v} = 0.06$		$\sqrt{v} = 0.08$		
method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	denoiser calls
HIO	18.98	0.5154	18.05	0.3692	16.87	0.2697	-
DPS	37.95	0.9584	35.91	0.9392	34.47	0.9206	1000
deepECpr	**39.33**	**0.9699**	**37.22**	**0.9542**	**35.20**	**0.9325**	**15**

Table 6: Improvement of deepECpr over DPS versus $\sqrt{v}$

$\sqrt{v}$	Δ PSNR $\pm$ SE	Δ SSIM $\pm$ SE
0.04	1.38 ± 0.02	0.0115 ± 0.0008
0.06	1.31 ± 0.04	0.0150 ± 0.0006
0.08	0.73 ± 0.13	0.0119 ± 0.0026

Reconstruction examples under shot noise: Reconstruction examples for phaseless CDP reconstruction under shot noise are shown below, for both the Set12 and FFHQ experiments.

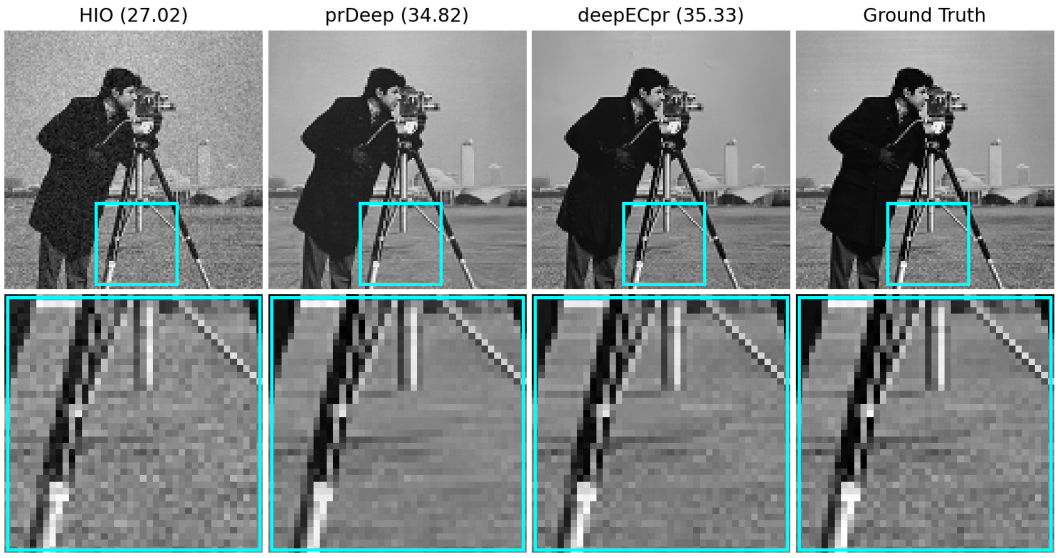

Figure 3: Top: Reconstructions of a 128×128 Set12 image from phaseless CDP measurements at $K = 4$ in shot noise with $\alpha = 18$. PSNR is shown in parentheses. Bottom: Zoomed versions of the cyan squares in the top row. Note how deepECpr reconstruction has less over-smoothing than prDeep and less noise than HIO.

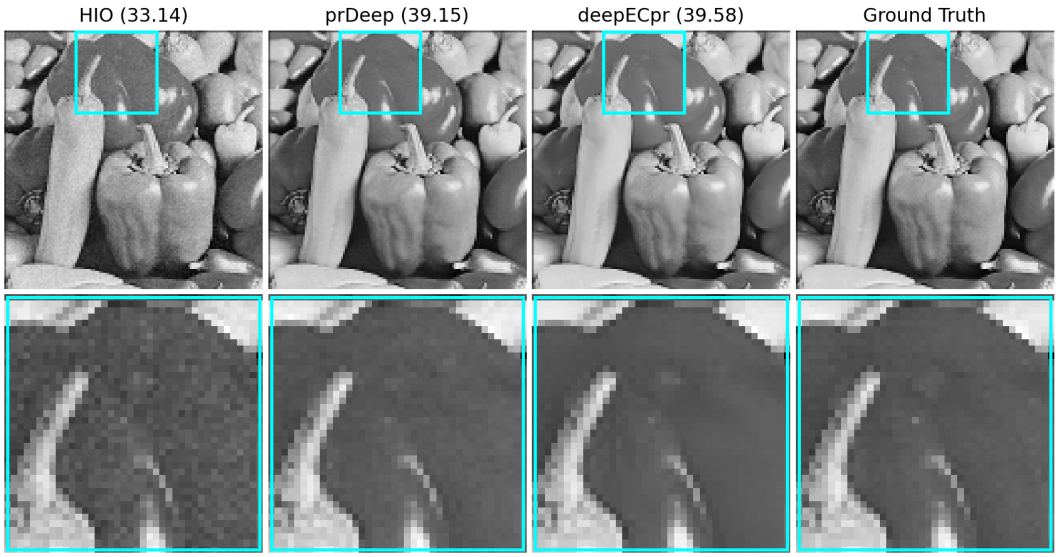

Figure 4: Top: Reconstructions of a 128×128 Set12 image from phaseless CDP measurements at $K = 4$ in shot noise with $\alpha = 9$. PSNR is shown in parentheses. Bottom: Zoomed versions of the cyan squares in the top row. Note how deepECpr removes more noise than prDeep and HIO while still preserving fine details.

Figure 5: Top: Reconstructions of a 256×256 FFHQ image from phaseless CDP measurements at $K = 4$ in shot noise with $\alpha = 9$. PSNR is shown in parentheses. Bottom: Zoomed versions of the cyan squares in the top row. Note how DPS failed to reconstruct the skin blemish near the center of the zoomed plot.

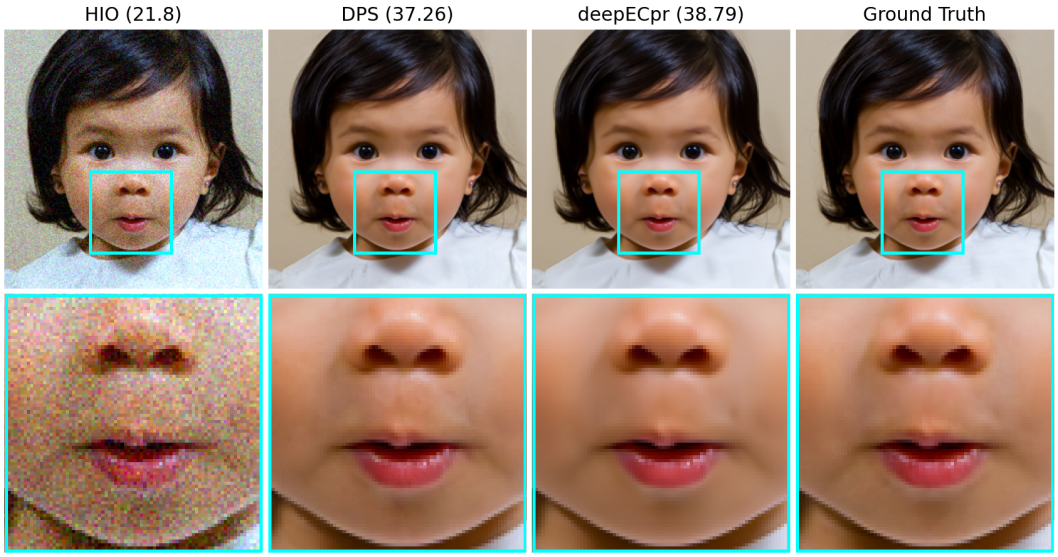

Figure 6: Top: Reconstructions of a 256×256 FFHQ image from phaseless CDP measurements at $K = 4$ in shot noise with $\alpha = 27$. PSNR is shown in parentheses. Bottom: Zoomed versions of the cyan squares in the top row. Note how DPS created visual artifacts near the center of the zoomed image.

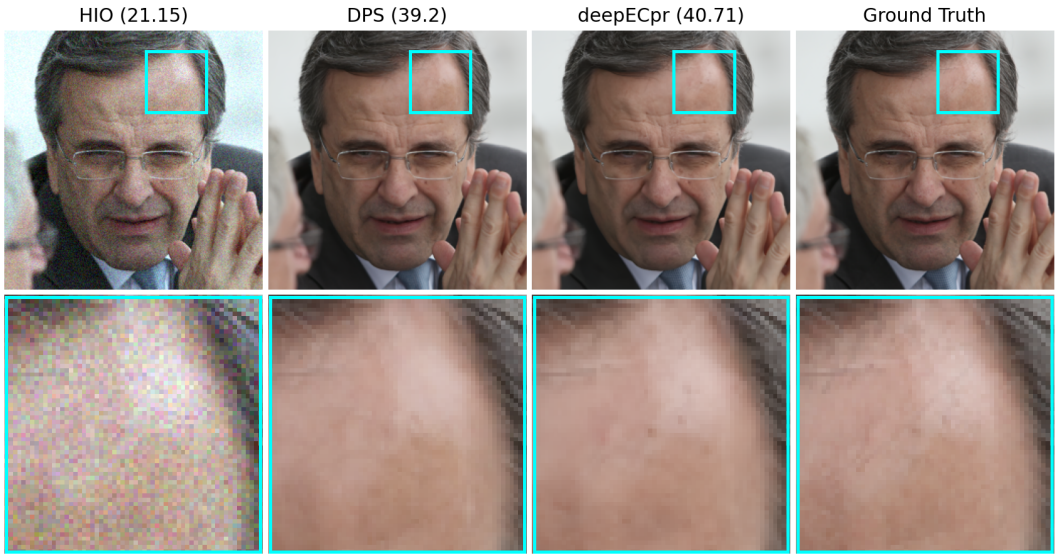

Figure 7: Top: Reconstructions of a 256×256 FFHQ image from phaseless CDP measurements at $K = 4$ in shot noise with $\alpha = 18$. PSNR is shown in parentheses. Bottom: Zoomed versions of the cyan squares in the top row. Note how DPS failed to reconstruct the skin blemish near the center of the zoomed plot.

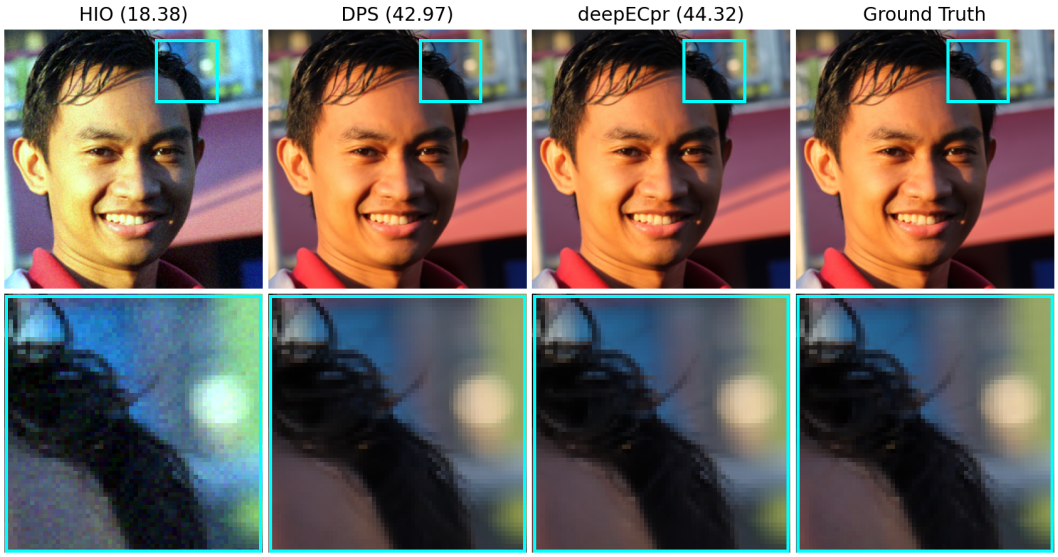

Figure 8: Top: Reconstructions of a 256×256 FFHQ image from phaseless CDP measurements at $K = 4$ in shot noise with $\alpha = 9$. PSNR is shown in parentheses. Bottom: Zoomed versions of the cyan squares in the top row. Note how DPS failed to reconstruct the fine hair stands near the center of the zoomed plot.

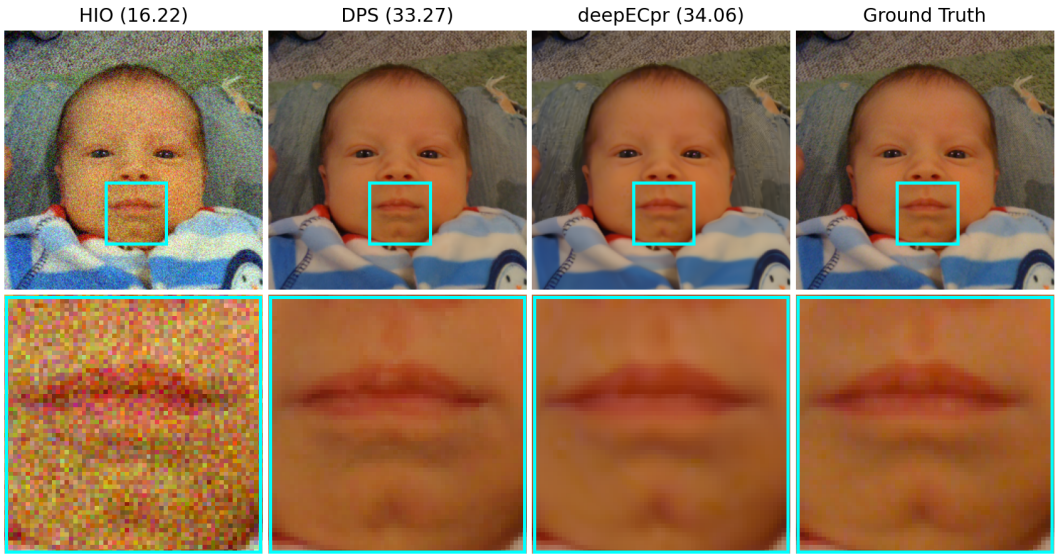

Figure 9: Top: Reconstructions of a 256×256 FFHQ image from phaseless CDP measurements at $K = 4$ in shot noise with $\alpha = 27$. PSNR is shown in parentheses. Bottom: Zoomed versions of the cyan squares in the top row. Note how DPS generated artifacts in and around the lips.