Deep Expectation-Consistent Approximation for Fourier Phase Retrieval

Saurav K. Shastri and Philip Schniter

THE OHIO STATE UNIVERSITY

Supported by NSF grant CCF-1955587

Asilomar Conference on Signal, Systems, and Computers

October 2024

Generalized-linear inverse problems

- We consider generalized-linear inverse problems:
  - $\boldsymbol{x}$: image to recover
  - $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x}$: hidden transform outputs, known forward operator $\boldsymbol{A}$
  - $\boldsymbol{y}$: measurements modeled by $p_{\mathsf{y|z}}(\boldsymbol{y}|\boldsymbol{z}) = \prod_{i=1}^{m} p_{\mathsf{y|z}}(y_i|z_i)$ with known $p_{\mathsf{y|z}}$

- Application examples:

$p_{\mathsf{y\|z}}(y\|z) \propto \exp(-\frac{1}{2v}(y-z)^2)$	additive white Gaussian noise
$p_{\mathsf{y\|z}}(y\|z) \propto \exp(-\frac{1}{\lambda}\|y-z\|)$	additive white Laplacian noise
$p_{\mathsf{y\|z}}(y\|z) = \frac{1}{1+\exp(-yz)}$	logistic regression, $y \in \pm 1$
$p_{\mathsf{y\|z}}(y\|z) = \frac{e^{zy}}{y!} e^{-e^{z}}$	Poisson regression, $y \in \mathbb{Z}_+$
$p_{\mathsf{y\|z}}(y\|z) = \int_{t_y}^{t_{y+1}} \mathcal{N}(t; z, \sigma^2)\,\mathrm{d}t$	noisy quantization, thresholds $\{t_y\}_{y=1}^{Q}$
$p_{\mathsf{y\|z}}(y\|z) = \exp(-\frac{1}{2v}(y-\|z\|)^2)$	phase retrieval with $y \in \mathbb{R}$ and $z \in \mathbb{C}$

- Our approach is based on expectation-consistent (EC) approximation[1]

---

[1] Opper & Winther'04

Expectation-consistent (EC) approximation to infer $\boldsymbol{z}$

- Can write the posterior distribution $p_{\mathsf{z}|\mathsf{y}}$ as a variational optimization problem involving KL divergence:

$$p_{\mathsf{z}|\mathsf{y}} = \arg\min_{q} D(q \,\|\, p_{\mathsf{z}|\mathsf{y}}) = \underset{q_1=q_2=q_3}{\arg\min}\, \underbrace{D(q_1\|p_{\mathsf{y}|\mathsf{z}}) + D(q_2\|p_{\mathsf{z}}) + H(q_3)}_{\triangleq\, J_{\mathsf{Gibbs}}(q_1,q_2,q_3)}$$

- EC[2] relaxes the equality constraints to moment-matching constraints:

$$p_{\mathsf{z}|\mathsf{y}} \approx \underset{q_1,q_2,q_3}{\arg\min}\, J_{\mathsf{Gibbs}}(q_1,q_2,q_3)$$

$$\text{s.t. } \begin{cases} \mathrm{E}\{\boldsymbol{z}|q_1\} = \mathrm{E}\{\boldsymbol{z}|q_2\} = \mathrm{E}\{\boldsymbol{z}|q_3\} \\ \mathrm{tr}(\mathrm{Cov}\{\boldsymbol{z}|q_1\}) = \mathrm{tr}(\mathrm{Cov}\{\boldsymbol{z}|q_2\}) = \mathrm{tr}(\mathrm{Cov}\{\boldsymbol{z}|q_3\}) \end{cases}$$

  and iteratively solves for the means and variances of densities q_1 and q_2

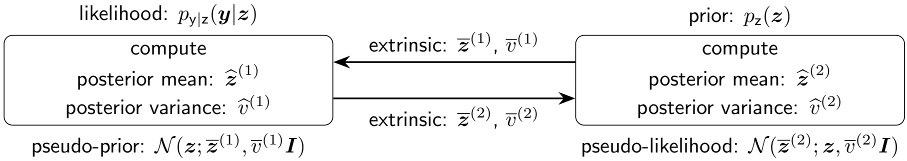
- Applied to generalized-linear inverse problems via extensions[3][4][5] of Vector AMP[6]

---

[2] Opper & Winther'04, [3] S et al'16, [4] He et al'17, [5] Meng et al'18, [6] Rangan et al'16

EC as message passing (or expectation propagation)

EC exchanges mean & variance messages between two estimation modules:



likelihood: $p_{\mathsf{y}|\mathsf{z}}(\boldsymbol{y}|\boldsymbol{z})$

compute
posterior mean: $\widehat{\boldsymbol{z}}^{(1)}$
posterior variance: $\widehat{v}^{(1)}$

pseudo-prior: $\mathcal{N}(\boldsymbol{z};\overline{\boldsymbol{z}}^{(1)},\overline{v}^{(1)}\boldsymbol{I})$

extrinsic: $\overline{\boldsymbol{z}}^{(1)}$, $\overline{v}^{(1)}$

extrinsic: $\overline{\boldsymbol{z}}^{(2)}$, $\overline{v}^{(2)}$

prior: $p_{\mathsf{z}}(\boldsymbol{z})$

compute
posterior mean: $\widehat{\boldsymbol{z}}^{(2)}$
posterior variance: $\widehat{v}^{(2)}$

pseudo-likelihood: $\mathcal{N}(\overline{\boldsymbol{z}}^{(2)};\boldsymbol{z},\overline{v}^{(2)}\boldsymbol{I})$

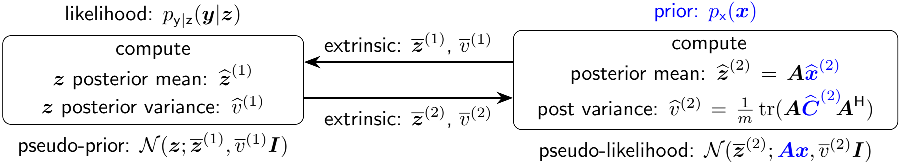
These "extrinsic" messages can be understood from Gaussian-pdf multiplication:

$$\mathcal{N}(z_i;\overline{z}_i^{(1)},\overline{v}^{(1)})\mathcal{N}(z_i;\overline{z}_i^{(2)},\overline{v}^{(2)}) = \mathcal{N}(z_i;\widehat{z}_i,\widehat{v}) \ \text{ for } \begin{cases} 1/\widehat{v} = 1/\overline{v}^{(1)} + 1/\overline{v}^{(2)} \\ \widehat{z}_i/\widehat{v} = \overline{z}_i^{(1)}/\overline{v}^{(1)} + \overline{z}_i^{(2)}/\overline{v}^{(2)} \end{cases}$$

- Given posterior $(\widehat{z}_i^{(2)},\widehat{v}^{(2)})$ and pseudo-likelihood $(\overline{z}_i^{(2)},\overline{v}^{(2)})$, solve for $(\overline{z}_i^{(1)},\overline{v}^{(1)})$
- Given posterior $(\widehat{z}_i^{(1)},\widehat{v}^{(1)})$ and pseudo-prior $(\overline{z}_i^{(1)},\overline{v}^{(1)})$, solve for $(\overline{z}_i^{(2)},\overline{v}^{(2)})$

Using EC for image recovery

Previously, we inferred $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x}$. We now modify EC to infer the image $\boldsymbol{x}$



likelihood: $p_{\mathsf{y|z}}(\boldsymbol{y}|\boldsymbol{z})$

compute

$\boldsymbol{z}$ posterior mean: $\widehat{\boldsymbol{z}}^{(1)}$

$\boldsymbol{z}$ posterior variance: $\widehat{v}^{(1)}$

pseudo-prior: $\mathcal{N}(\boldsymbol{z}; \overline{\boldsymbol{z}}^{(1)}, \overline{v}^{(1)}\boldsymbol{I})$

extrinsic: $\overline{\boldsymbol{z}}^{(1)}$, $\overline{v}^{(1)}$

extrinsic: $\overline{\boldsymbol{z}}^{(2)}$, $\overline{v}^{(2)}$

prior: $p_{\mathsf{x}}(\boldsymbol{x})$

compute

posterior mean: $\widehat{\boldsymbol{z}}^{(2)} = \boldsymbol{A}\widehat{\boldsymbol{x}}^{(2)}$

post variance: $\widehat{v}^{(2)} = \frac{1}{m}\operatorname{tr}(\boldsymbol{A}\widehat{\boldsymbol{C}}^{(2)}\boldsymbol{A}^{\mathsf{H}})$

pseudo-likelihood: $\mathcal{N}(\overline{\boldsymbol{z}}^{(2)}; \boldsymbol{A}\boldsymbol{x}, \overline{v}^{(2)}\boldsymbol{I})$

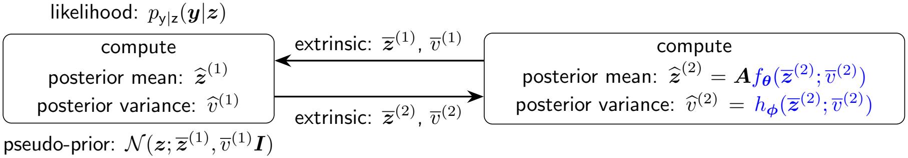
- Say we have a prior $p_{\mathsf{x}}(\boldsymbol{x})$ and we write the pseudo-likelihood in terms of $\boldsymbol{x}$
- In theory, could compute the posterior mean $\widehat{\boldsymbol{x}}^{(2)}$ and covariance $\widehat{\boldsymbol{C}}^{(2)}$ of $\boldsymbol{x}$
  - But both involve high-dimensional integrals
- When $\boldsymbol{A}$ is a large and rotationally-invariant random matrix and p_{x} is i.i.d., $\widehat{\boldsymbol{x}}^{(2)}$ and $\widehat{v}^{(2)}$ can be computed by Vector AMP[7]
  - But this does not work with deterministic imaging operators $\boldsymbol{A}$, such as Fourier

---

[7] Rangan,S,Fletcher'16

Proposed approach: deepEC

We propose to implement these challenging EC steps using deep networks:



likelihood: $p_{\mathsf{y}|\mathsf{z}}(\boldsymbol{y}|\boldsymbol{z})$

compute

posterior mean: $\widehat{\boldsymbol{z}}^{(1)}$

posterior variance: $\widehat{v}^{(1)}$

pseudo-prior: $\mathcal{N}(\boldsymbol{z}; \overline{\boldsymbol{z}}^{(1)}, \overline{v}^{(1)}\boldsymbol{I})$

extrinsic: $\overline{\boldsymbol{z}}^{(1)}$, $\overline{v}^{(1)}$

extrinsic: $\overline{\boldsymbol{z}}^{(2)}$, $\overline{v}^{(2)}$

compute

posterior mean: $\widehat{\boldsymbol{z}}^{(2)} = \boldsymbol{A} f_{\boldsymbol{\theta}}(\overline{\boldsymbol{z}}^{(2)}; \overline{v}^{(2)})$

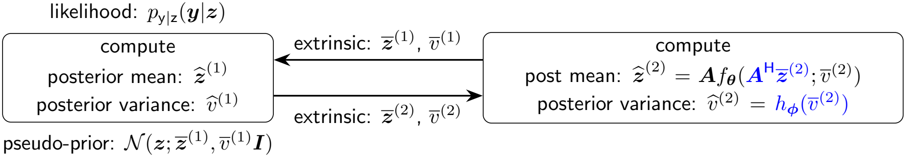
posterior variance: $\widehat{v}^{(2)} = h_{\boldsymbol{\phi}}(\overline{\boldsymbol{z}}^{(2)}; \overline{v}^{(2)})$

- Approximate posterior mean $\widehat{\boldsymbol{x}}^{(2)}$ using a neural network $f_{\boldsymbol{\theta}}(\overline{\boldsymbol{z}}^{(2)}; \overline{v}^{(2)})$ trained to recover $\boldsymbol{x}$ from $\overline{\boldsymbol{z}}^{(2)} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$ with $\boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \overline{v}^{(2)}\boldsymbol{I})$
- Approximate posterior variance $\overline{v}^{(2)}$ using a neural network $h_{\boldsymbol{\phi}}(\overline{\boldsymbol{z}}^{(2)}; \overline{v}^{(2)})$ trained to predict $\frac{1}{d}\| f_{\boldsymbol{\theta}}(\overline{\boldsymbol{z}}^{(2)}; \overline{v}^{(2)}) - \boldsymbol{x}\|^2$

A downside of this approach is that both $\boldsymbol{f}_{\boldsymbol{\theta}}$ and $h_{\boldsymbol{\phi}}$ depend on $\boldsymbol{A}$.

Simplifications for phase retrieval: deepECpr

Simplifications arise when $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A} = \boldsymbol{I}$, as in phase retrieval with Fourier or CDP $\boldsymbol{A}$:



likelihood: $p_{\mathsf{y}|\mathsf{z}}(\boldsymbol{y}|\boldsymbol{z})$

compute
posterior mean: $\widehat{\boldsymbol{z}}^{(1)}$
posterior variance: $\widehat{v}^{(1)}$

pseudo-prior: $\mathcal{N}(\boldsymbol{z}; \overline{\boldsymbol{z}}^{(1)}, \overline{v}^{(1)}\boldsymbol{I})$

extrinsic: $\overline{\boldsymbol{z}}^{(1)}$, $\overline{v}^{(1)}$

extrinsic: $\overline{\boldsymbol{z}}^{(2)}$, $\overline{v}^{(2)}$

compute
post mean: $\widehat{\boldsymbol{z}}^{(2)} = \boldsymbol{A} f_{\boldsymbol{\theta}}(\boldsymbol{A}^{\mathsf{H}}\overline{\boldsymbol{z}}^{(2)}; \overline{v}^{(2)})$
posterior variance: $\widehat{v}^{(2)} = h_{\boldsymbol{\phi}}(\overline{v}^{(2)})$

- Rather than recovering $\boldsymbol{x}$ from pseudo-measurements $\overline{\boldsymbol{z}}^{(2)} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$, recover it from the sufficient statistic $\overline{\boldsymbol{r}}^{(2)} \triangleq \boldsymbol{A}^{\mathsf{H}}\overline{\boldsymbol{z}}^{(2)} = \boldsymbol{x} + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \overline{v}^{(2)}\boldsymbol{I})$
  - In this case, $f_{\boldsymbol{\theta}}(\cdot; \overline{v}^{(2)})$ becomes a deep denoiser
- With the same sufficient statistic, $h_{\boldsymbol{\phi}}(\overline{\boldsymbol{r}}^{(2)}; \overline{v}^{(2)})$ reduces to $h_{\boldsymbol{\phi}}(\overline{v}^{(2)})$
  - $h_{\boldsymbol{\phi}}(\cdot)$ is simply the denoiser's input-output variance curve
- The resulting deepECpr is essentially a variance-tracking plug-and-play alg!
  - The full paper[8] describes a few more tricks, such as stochastic damping

---

[8] https://arxiv.org/pdf/2407.09687

Numerical experiments on phase retrieval

Measurements:

- forward operator $\boldsymbol{A}$: both 4x oversampled Fourier and CDP
- shot noise @ level α

Competitors:

- HIO[9]: classical method
- prDeep[10], Deep-ITA-F & Deep-ITA-S[11]: PnP methods
- DPS[12], DOLPH[13]: recent diffusion methods

All except HIO used the likelihood $p_{\mathsf{y}|\mathsf{z}}(y|z) = \exp(-\frac{1}{2v}(y - |z|)^2)$

Test images:

- 256x256 color FFHQ[12] face images . . . FFHQ diffusion denoiser
- 128x128 grayscale natural[11] and unnatural[10] . . . BSD400 DnCNN denoisers

---

[9] Fienup'82, [10] Metzler et al'18, [11] Wang et al'20, [12] Chung et al'23, [13] Shoushtari et al'22

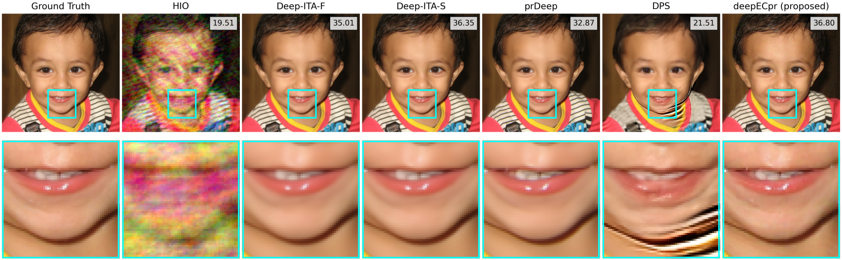
Phase retrieval of FFHQ faces at noise level α

method	OSF						CDP					
	$\alpha = 4$		$\alpha = 6$		$\alpha = 8$		$\alpha = 5$		$\alpha = 15$		$\alpha = 45$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
HIO	27.37	0.6759	26.08	0.6163	25.03	0.5664	36.48	0.9140	26.79	0.6064	17.58	0.2330
Deep-ITA-F	35.05	0.9420	34.94	0.9374	34.50	0.9321	41.69	0.9786	37.20	0.9556	29.93	0.8402
Deep-ITA-S	34.01	0.9365	34.54	0.9367	**35.15**	**0.9412**	41.87	0.9795	37.15	0.9531	27.28	0.7817
prDeep	37.69	0.9654	35.28	0.9523	33.68	0.9410	42.43	0.9816	37.48	0.9584	23.14	0.4477
DPS	27.22	0.7674	25.84	0.7530	24.57	0.7408	41.52	0.9786	35.68	0.9381	30.12	0.8428
DOLPH	14.65	0.3426	14.65	0.3420	14.64	0.3393	40.74	0.9755	33.84	0.8806	20.98	0.3541
deepECpr (proposed)	**39.75**	**0.9720**	**37.01**	**0.9567**	34.86	0.9404	**43.12**	**0.9846**	**37.55**	**0.9589**	**32.15**	**0.8941**

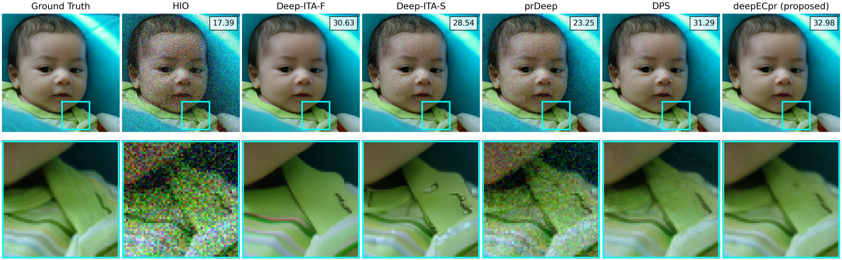
- The proposed deepECpr wins in all but OSF at $\alpha = 8$
- The well-known prDeep comes in second, except at the highest noise levels
- The DPS diffusion method works well with CDP but poorly with OSF

FFHQ examples

OSF recoveries at $\alpha = 8$:



CDP recoveries at $\alpha = 45$:



Phase retrieval of grayscale images

OSF measurements:

method	natural						unnatural					
	$\alpha = 4$		$\alpha = 6$		$\alpha = 8$		$\alpha = 4$		$\alpha = 6$		$\alpha = 8$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
HIO	24.32	0.6932	22.75	0.6267	21.63	0.5702	26.66	0.6221	25.10	0.5653	24.04	0.5231
Deep-ITA-F	31.19	0.8996	29.64	0.8642	28.35	0.8278	28.30	0.6904	26.76	0.6466	26.58	0.6450
Deep-ITA-S	30.98	0.8933	29.71	0.8603	28.78	0.8343	28.20	0.6894	26.96	0.6472	26.75	0.6390
prDeep	35.71	0.9632	32.56	0.9242	29.57	0.8670	30.22	0.7533	26.66	0.6630	25.96	0.6297
deepECpr (proposed)	**37.30**	**0.9766**	**34.06**	**0.9546**	**31.85**	**0.9302**	**30.98**	**0.7675**	**27.45**	**0.6897**	**27.09**	**0.6780**

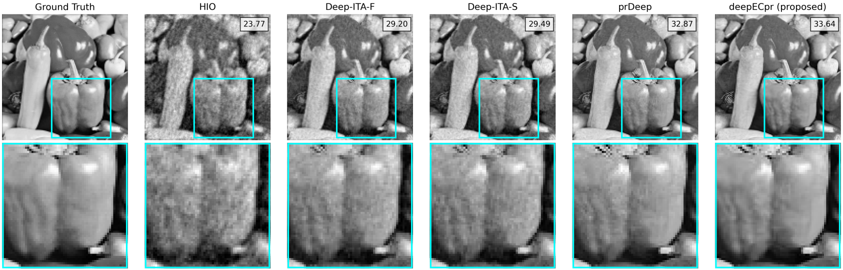
CDP measurements:

method	natural						unnatural					
	$\alpha = 5$		$\alpha = 15$		$\alpha = 45$		$\alpha = 5$		$\alpha = 15$		$\alpha = 45$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
HIO	36.39	0.9541	26.65	0.7663	17.13	0.4155	36.45	0.8891	26.94	0.6133	18.55	0.3175
Deep-ITA-F	38.67	0.9783	29.57	0.8595	22.77	0.6301	39.42	0.9607	29.80	0.7526	21.40	0.4352
Deep-ITA-S	38.80	0.9797	28.56	0.8357	17.25	0.4260	39.80	0.9679	28.56	0.7077	18.74	0.3211
prDeep	38.73	0.9785	32.49	0.9388	26.38	0.8167	39.63	0.9660	33.84	0.9267	27.57	0.8220
DOLPH	39.19	0.9738	28.37	0.7435	17.63	0.3336	40.40	0.9708	29.11	0.6571	19.28	0.2797
DPS	37.66	0.9617	32.07	0.9038	26.76	0.7634	39.05	0.9603	33.15	0.8978	27.22	0.7423
deepECpr (proposed)	**39.46**	**0.9845**	**32.75**	**0.9439**	**27.06**	**0.8428**	**40.92**	**0.9808**	**34.17**	**0.9356**	**28.14**	**0.8292**

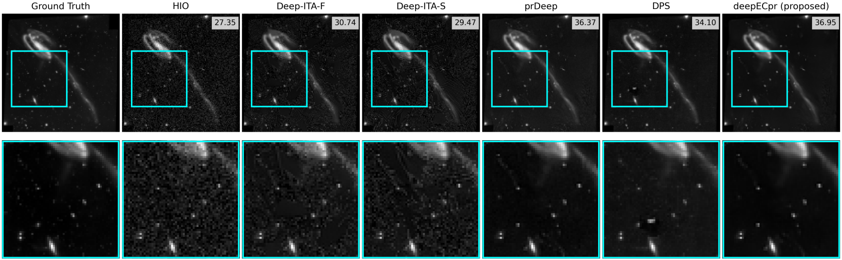
- The proposed deepECpr wins in all cases

Grayscale examples

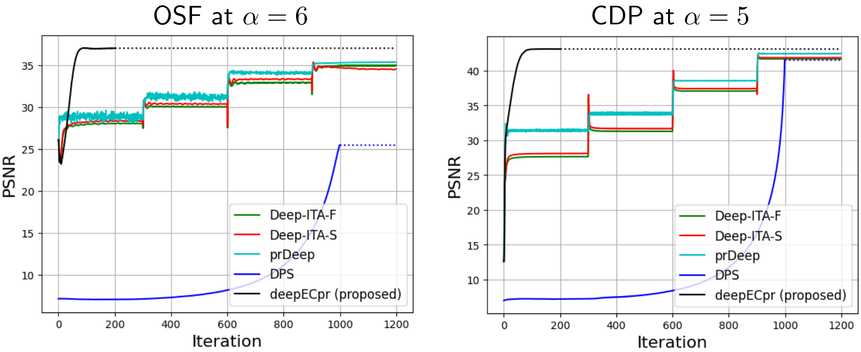
OSF recoveries at $\alpha = 8$:



CDP recoveries at $\alpha = 15$:



Convergence speed



- The proposed **deepECpr** (black curve) converges in $<$100 iterations, which is much faster than the competitors

Summary

- For linear and generalized-linear inverse problems, we proposed a novel EC-based approach called deepEC that exploits deep neural networks
- Unlike existing EC-methods for inverse problems (e.g., VAMP), deepEC does not require a rotationally invariant random $\boldsymbol{A}$
- For phase retrieval, we proposed a simplified variant called deepECpr that is essentially a variance-tracking plug-and-play algorithm
- Compared to the phase-retrieval methods prDeep, Deep-ITA, and DPS, deepECpr is more accurate and converges much faster