Thanks to the reviewers for their thoughtful feedback. R1 criticized that our NAC is basically an extended DIP, which we respectively cannot agree. Please see below our detailed responses to the major concerns of all reviewers.

"Differences from DIP" (R1). Our NAC is essentially different from DIP: **1**. NAC is a novel strategy for unsupervised learning of **adaptive network parameters** for the degraded image, while DIP aims to investigate **adaptive network structure** before any of its parameters are learned; **2**. NAC learns a mapping from $\boldsymbol{z}$ to $\boldsymbol{y}$, which approximates the mapping from $\boldsymbol{y}$ to $\boldsymbol{x}$. DIP maps a random noise map to $\boldsymbol{y}$; **3**. Due to **1** and **2**, DIP needs early stopping for different images, while NAC achieves more robust (and better) denoising performance than DIP on diverse images.

"NAC on DnCNN" (R1&R3). Thanks for the suggestion. We trained DnCNN with our proposed NAC strategy (DnCNN-NAC), and the comparison results with DnCNN are listed in Tab. 1. One can see that DnCNN-NAC achieves better PSNR results than the original DnCNN when $\sigma = 5, 10, 15$ (but worse when $\sigma = 20, 25$).

Others comments by R1. **1**. **noise assumption**: as mentioned in Lines (**L**) **105-107**: "*Different with the "zero-mean" assumption ...*" of the main paper, we assume that the noise is weak (Eqn. (4)), but not necessarily zero-mean. **2**. **imprecise arguments**: to make better use of the degraded image, we did data augmentation and trained NAC with *8 pairs* of noisy/noisy images, so we think the learned NAC are approximating a supervised network learned with *pairs* of noisy/clean images. **3**. **Eqn. (13)** treats the synthetic noise *dependent* on observation, please refer to Eqn. (12).

σ	5	10	15	20	25
DnCNN	38.76	34.78	32.86	31.45	30.43
DnCNN-NAC	43.18	37.16	33.65	31.16	29.23

Table 1: Average PSNR (dB) of DnCNN and DnCNN-NAC on *Set12* on AWGN noise with different levels σ.

σ	5	10	15	20	25
NAC	41.43	37.31	34.67	32.92	31.40
NAC-Blind	40.44	37.50	35.01	33.12	31.70

Table 2: Average PSNR (dB) results of NAC trained with known noise level and without known noise level (NAC-Blind) on *Set12* with AWGN noise (at level σ).

"training curves" (R2). Thanks for the suggestion. In Figs. 1 and 2, we plotted the curves of training loss and testing PSNR of NAC and DIP on two images, DIP needs early stopping to select the best results, while our NAC can stably achieve satisfactory results with 1000 epochs.

"blind denoising" (R2&R3). Thanks for the suggestion. We trained the blind NAC networks (NAC-blind) using the strategies as in N2N. The results on *Set12* are listed in Tab. 2. We observe that blind NAC achieves even better results.

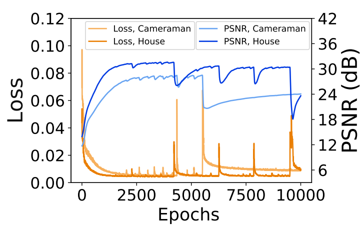

Figure 1: Curves of DIP.

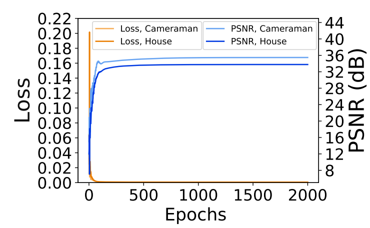

Figure 2: Curves of NAC.

"number of blocks" (R2). The PSNR(dB)/SSIM results of NAC with different number of Residual Blocks (ResBlocks) are provided in Table 3. The NAC network with only 1 or 2 Res-Block cannot achieve promising results on PSNR, while the NAC with more ResBlocks can achieve satisfactory results on both PSNR and SSIM. This indicates that when the training data is only the degraded image, a relatively "shallow" network is enough (and with faster speed).

# of ResBlocks	1	2	3	10
PSNR	33.55	33.94	34.00	34.67
SSIM	0.9252	0.9263	0.9267	0.9206

Table 3: Average PSNR (dB)/SSIM of NAC on *Set12* with different number of blocks on AWGN noise with $\sigma = 15$.

# of Epochs	100	200	500	1000
PSNR	32.71	33.70	34.18	34.67
Time (s)	67.4	132.5	302.0	583.2

Table 4: Average PSNR (dB) and time (s) of NAC on *Set12* with different number of epochs on AWGN noise $\sigma = 15$.

Others comments by R2. Thank you very much for your suggestions on revising our paper. **1**. **Runtime of all methods** are 8.41s (BM3D), 1.20s (NI), 0.05s (DnCNN), 0.40s (CBDNet), 0.19s (N2N). **2**. **PSNR/SSIM Results of NAC on different noise levels** σ are 29.86/0.8361 (30), 28.05/0.7674 (40), 25.59/0.6786 (50), 24.89/0.6224 (60), 23.96/0.5517 (75). **3**. **"training details"**. The batch size (BS) is 1, so iterations is $8000 = 1000 \times 8$, where 1000 is the epoch number and 8 is the number of augmented images. We will add this in revision. We also run experiments with BS=8, the iteration number is 1000. It suffers a little performance drop, but with faster training speed. **4**. We will add Figure 1 of the *Supp. File* to main paper. **5**. "Val" will be replaced with "Var". **6**. The suggested paper by Jain *et al.* will be cited.

"gap between DnCNN and N2N" (R3). We trained DnCNN on the 50K images from ImageNet-val dataset used in N2N. The PSNR (dB) results on *Set12* with AWGN noise are 33.21 ($\sigma = 15$) and 30.84 ($\sigma = 25$), still inferior to N2N.

"training DnCNN or N2N with mixed Poisson and AWGN noise" (R3). We agree that comparison is not exactly fair. The used DnCNN+ is trained with additional noise maps estimated by FFDNet, which is not used in our NAC. The iterations for training N2N is 300,000, while that for NAC is only 8000. We retrained DnCNN and N2N with the mixed noise, and achieve 35.73 and 35.16 on PSNR (dB) on the Cross-Channel dataset, respectively. The results are still inferior to NAC. Note that due to time limitation, we only trained the N2N with mixed noise for 180,000 iterations. The CBDNet is also trained also with the mixed noise, with an additional noise map estimation step, but achieves lower PSNR and SSIM results than our NAC, as shown in the Tab. 6 of the main paper.

"training time" (R3). As show in Fig. 2, our NAC trained in 500 epochs is enough to achieve state-of-the-art performance. The averaged running time on a 256×256 image w.r.t. the number of blocks are listed in Tab. 4.