Stochastic Convolutional Sparse Coding
Supplementary Material

Anonymous CVPR submission

Paper ID 3722

1. Solvers for LASSO and QCQP

Algorithm 1 is used to optimize a LASSO problem in the following form:

$$\underset{\mathbf{z}}{\operatorname{argmin}} \ \frac{1}{2}\|\mathbf{x}-\mathbf{D}\mathbf{z}\|_2^2+\lambda\|\mathbf{z}\|_1 \tag{1}$$

Algorithm 1 ADMM framework for solving LASSO

> **for** $s = 1$ to S **do**
>   // z-update step (quadratic programming)
> $\mathbf{z}^{s+1} \leftarrow (\mathbf{D}^\top\mathbf{D}+\rho\mathbb{I})^{-1}(\mathbf{D}^\top\mathbf{x}+\rho(\mathbf{y}^s-\mathbf{q}^s))$
>   // y-update step (soft thresholding)
> $\mathbf{y}^{s+1} \leftarrow (\mathbf{z}^{s+1}+\mathbf{q}^s-\frac{\lambda}{\rho})_+ - (-\mathbf{z}^{s+1}-\mathbf{q}^s-\frac{\lambda}{\rho})_+$
>   // scaled dual variables update
> $\mathbf{q}^{s+1} \leftarrow \mathbf{q}^s+\mathbf{z}^{s+1}-\mathbf{y}^{s+1}$
> **end for**

S is the total number of ADMM iteration. $\mathbf{y}$ is the introduced slack variable, $\mathbf{q}$ is the scaled dual variable, and ρ is the augmented Lagrangian penalty.

Algorithm 2 Projected Block Coordinate Descent for solving QCQP

> $\mathbf{r} \leftarrow \mathbf{x}-\mathbf{Z}\mathbf{d}$
> **while** not converge **do**
>   **for** $k = 1$ to K **do**
> $\mathbf{d}_\mathrm{k}^* \leftarrow \mathbf{d}_\mathrm{k}+\mathbf{Z}_\mathrm{k}^\top\mathbf{r} \,/\, L_\mathrm{k}$
> $\mathbf{d}_\mathrm{k}^* \leftarrow \mathbf{d}_\mathrm{k}^* \,/\, \max(\|\mathbf{d}_\mathrm{k}^*\|, 1)$
> $\mathbf{r} \ \leftarrow \mathbf{r}+\mathbf{Z}_\mathrm{k}(\mathbf{d}_\mathrm{k}-\mathbf{d}_\mathrm{k}^*)$
>   **end for**
> $\mathbf{d} \leftarrow \mathbf{d}^*$
> **end while**

Algorithm 2 is used to optimize a QCQP problem in the following form:

$$\begin{aligned} &\underset{\mathbf{d}}{\operatorname{argmin}} \ \frac{1}{2}\|\mathbf{x}-\mathbf{Z}\mathbf{d}\|_2^2 \\ &\text{subject to } \|\mathbf{d}_\mathrm{k}\|_2^2 \le 1 \ \ \forall k \in \{1,\dots,K\}, \end{aligned} \tag{2}$$

L_k is the Lipschitz constant of $\mathbf{Z}_\mathrm{k}^\top\mathbf{Z}_\mathrm{k}$.

2. Zooming in on the Filters

Here we zoom in on the filters from Fig. 1 in the main manuscript. Notice that our filters look more smooth for those Gabor-like filters and also contains less number of noise-like filters.

3. Additional Experiments

In order to show the robustness of the proposed algorithms, We also conduct experiments on city dataset. We first compare SBCSC with the state-of-the-art batch-mode algorithm, and the results are shown in Fig. 2. Similar to the results shown in Fig. 1 in the main manuscript, SBCSC, with $p = 0.1$, outperforms the compared batch-mode algorithm with better outcomes and runtime performance. Since the learning is performed on handful datasets, both of them learn quite a few data-specific image features. These data-specific image features have the shortage of the generalization ability, which can be revealed in the image inpainting application. For example, the eighth image, which contains a large portion of the texture information existed in the filters learned from city dataset, can be significantly better reconstructed by such filters compared to using those learned from fruit dataset. However, it may exhibit poorer performance on other types of images.

We then compare SOCSC with the state-of-the-art online algorithm on city dataset, the results of which are shown in Fig. 3. Similar to the results presented in Fig. 3 of the main manuscript, our method obtains comparable outcomes, meanwhile, achieves roughly $6\times$ speedup.

4. Over-complete Dictionary and Large Datasets

In this section, we first show the improved sparse representation of the natural images with over-complete dictionaries ($K = 400$) over under-complete ones ($K = 100$). The numerical comparisons of number of non-zero ele-

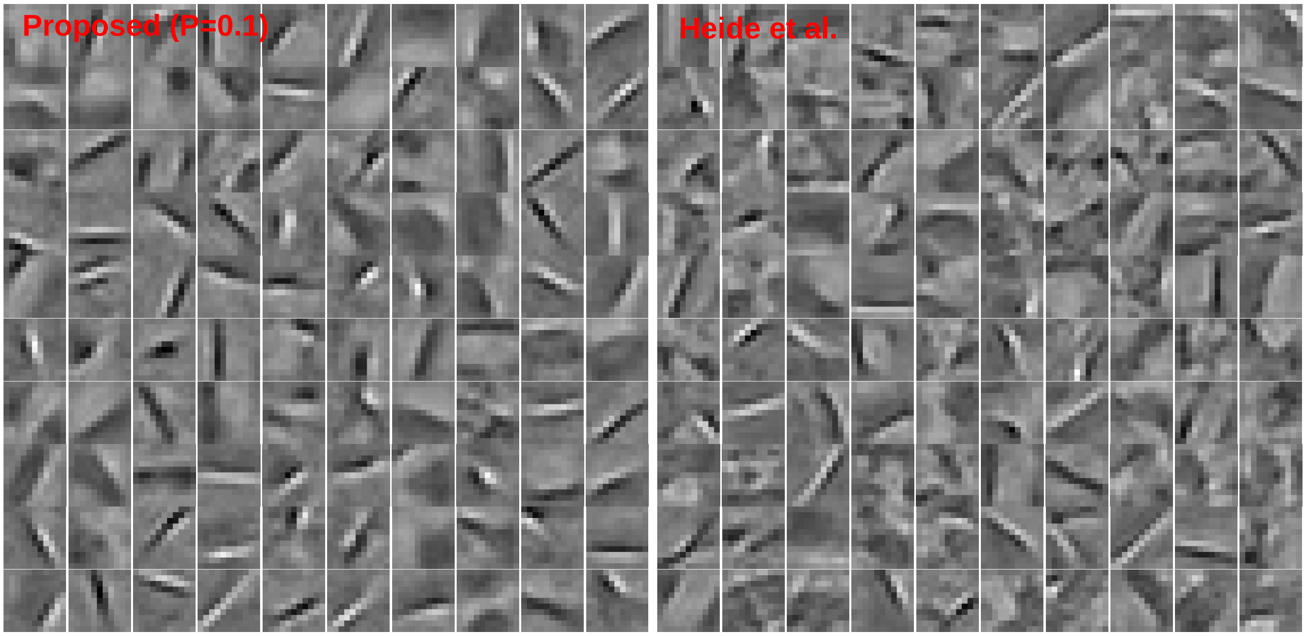

Figure 1: Zoom-in view of the filters from Fig. 1 of the main manuscript.

ments and its corresponding reconstruction PSNR for various images are shown in Fig. 4. Here, we define the non-zero elements as the codes whose coefficient is no less than 0.1. We could observe that at all times, using over-complete dictionary leads to a sparser representation of the images, roughly $8\% - 10\%$ reduction on the non-zero elements. Meanwhile, it achieves dramatically improved reconstruction quality, over 1 dB on average.

We then verify the importance of a large dataset when learning the over-complete dictionary. in Fig. 5, we show a visual and quantitative comparisons between the over-complete dictionaries respectively learned by batch-mode algorithm on small dataset (the fruit dataset) and online-mode algorithm on large dataset (1000 images). Most of the filters learned from small dataset have poor structures and reveal very limited representative ability. The numerical results also demonstrate that it even shows a degraded reconstruction performance compared to the under-complete dictionary. Owing to plentiful training samples, our over-complete dictionary not only shows visually decent structures and more representative image features, but also leads to a significant improvement on image reconstruction. Based on all experimental results, it implies that the number of filters and number of training samples are both essential in the CSC model, therefore, the proposed algorithm has prominent advantages over the existing approaches.

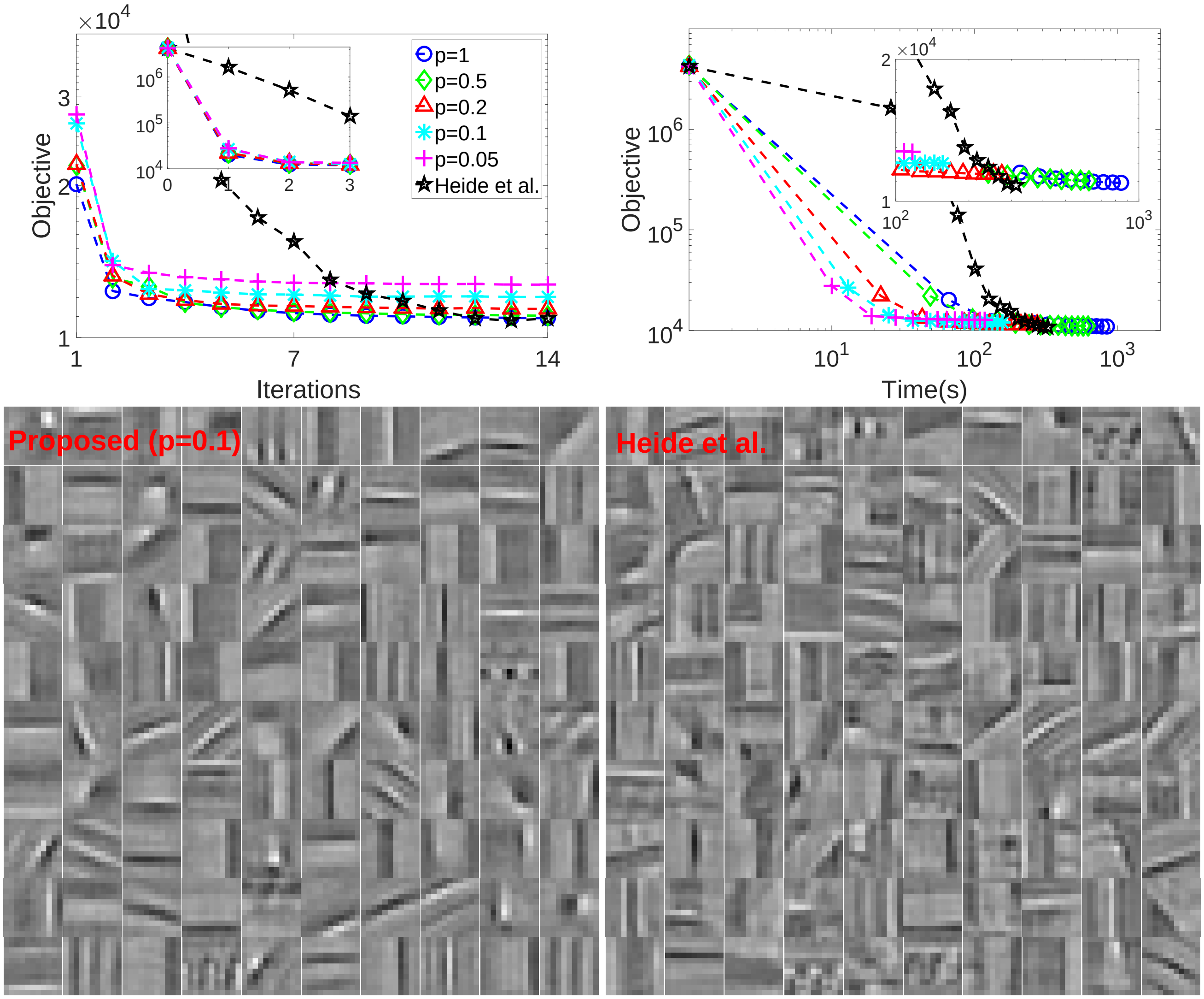

Image	1	2	3	4	5	6	7	8	9	10
PSNR [10]	29.67	**28.14**	29.58	29.69	28.74	29.43	27.89	30.13	27.03	30.61
PSNR ours	**29.78**	**28.14**	**29.67**	**29.85**	**28.88**	**29.95**	**27.97**	**30.37**	**27.14**	**30.89**

Figure 2: Experimental results obtained on the city dataset. Top: Convergence comparison between the-state-of-art batch method and SBCSC with different subsampling probability. Middle: Learned filters by SBCSC with $p = 0.1$ and the comparable method. Even though these dictionaries look similar, our method learns more smooth filters. Compared to the learned filters in Fig. 1 of the main manuscript, they have different representations of the dictionaries, where data-specific features are learned from handful training images. Bottom: Numerical comparisons of the reconstruction quality obtained by the presented filters in the image inpainting application.

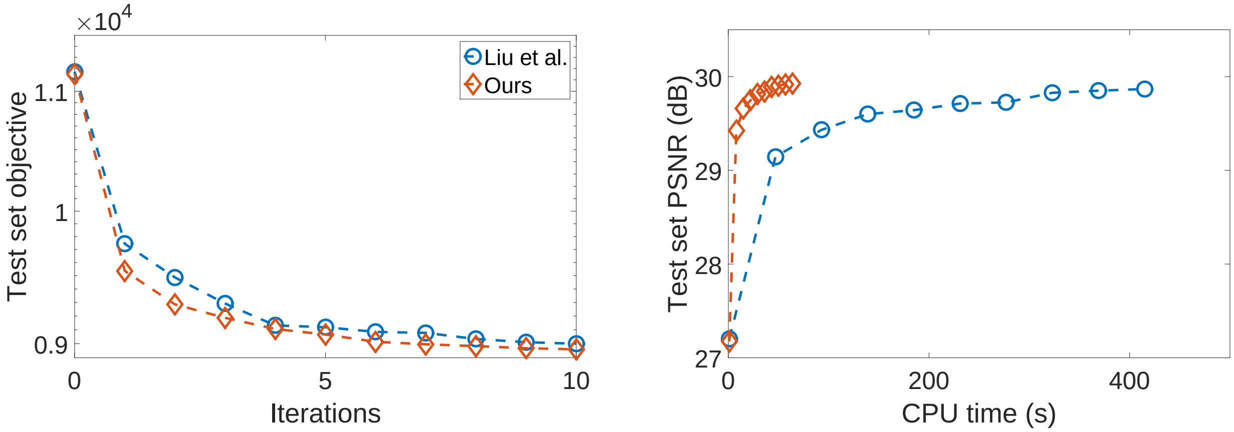

Figure 3: Experimental results obtained on city dataset. Left: Convergence of the test set objectives for our method (SOCSC) and the state-of-the-art online approach. Right: Testing PSNR with respect to execution time.

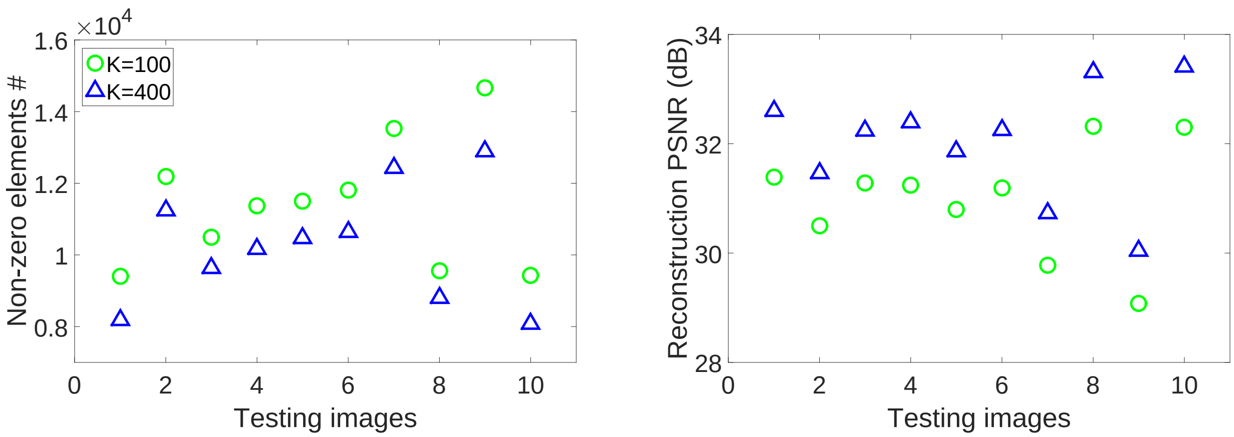

Figure 4: Left: number of non-zero elements in the codes for different images (10 256×256 images). Right: PSNR between the reconstructed images and the original ones.

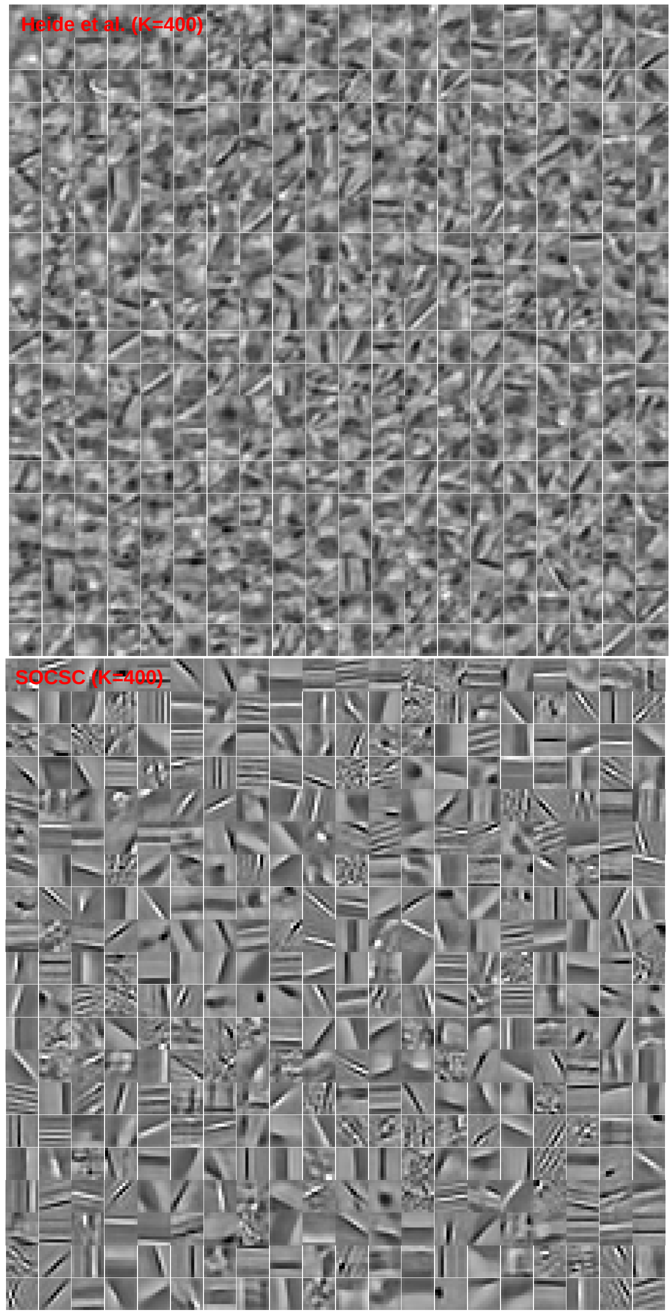

Image	1	2	3	4	5	6	7	8	9	10
PSNR [10]	29.60	28.11	29.47	28.98	28.79	29.21	28.03	29.31	27.42	30.52
PSNR ours	**30.24**	**28.34**	**29.95**	**30.30**	**29.43**	**29.96**	**28.24**	**30.57**	**27.72**	**31.67**

Figure 5: Visual and numerical comparisons between the learned over-complete dictionaries by batch-mode algorithm and by online-mode algorithm. Top: Over-complete dictionary learned by batch CSC model on small dataset, and proposed online CSC model (SOCSC) on large dataset. Bottom: Respective reconstruction quality for these two over-complete dictionaries applied for image inpainting.