Self-Supervised Burst Super-Resolution
Supplementary Material

In this supplementary material, we provide additional details and analysis. First we provide more details about our method in Section 1. Section 2 provides details about the SynBSR benchmark used for our comparison with alternate supervised training strategies in Section 4.1 of the main paper. Section 3 provides details about the comparison on SynBSR benchmark. Section 4 provides details about the comparison on real-world BurstSR dataset in Section 4.1 of the main paper. More details about our experiment on HDR+ dataset in Section 4.2 of the main paper are provided in Section 5. Next, in Section 6, we provide details about our analysis of the self-supervised training on synthetic data in Section 4.3 of the main paper. An analysis of training convergence is provided in Section 7. Section 8 shows the impact of error in motion estimation in our self-supervised training framework. Further qualitative results on the BurstSR dataset are provided in Section 9. Results in the linear camera color space is provided in Section 10, while Section 11 provides qualitative results on our synthetic setup. Finally, we discuss the limitations of our approach in Section 12.

1. Additional method details

Here, we provide some additional implementation details of our method.

Motion Estimation: We use a pre-trained optical flow network PWC-Net [7] to compute the motion between the first image in the burst b_1, and the images b_i from the unseen set B_{unseen}. Since the images from the burst are in the Raw domain, we first perform a nearest-neighbor demosaicking to obtain an RGB image, which is then passed to the optical flow network.

Image Formation Model: We use the bilinear interpolation when warping the images using the warp operator Φ. In all our experiments, the Raw images are obtained using the Bayer filter mosaic, although our method is applicable to any color filter array.

2. Details about the SynBSR benchmark in Sec. 4.1

Quantitatively evaluating burst super-resolution models on real data is hard due to challenges associated with collecting accurately aligned ground truths. Thus we develop a synthetic benchmark **SynBSR** which aims to simulate the real world training challenges, while allowing accurate quantitative evaluation. The benchmark is designed for 4x burst super-resolution task. We generate the training and test bursts for our synthetic setup using the pipeline introduced in [1]. We use the Isotropic Gaussian kernel as the blur kernel, instead of Bilinear kernel employed in [1]. We use the Isotropic Gaussian kernel with a standard deviation of 1.35 pixels as our blur kernel. Crucially, the methods are not allowed to use the knowledge of the blur kernel during training. The bursts are corrupted using the mixed read and shot noise distribution, with the same parameters as in [1]. For each training burst, we also generate a weakly paired high resolution groundtruth. In practise, the groundtruth is captured using optical zoom, often using a different camera as the input [1]. Hence, there exists spatial and color misalignments between the input burst and the groundtruth. We simulate these misalignments synthetically. In order to model the spatial shift between cameras, we first apply random translation (uniformly sampled between -24 to 24 pixels) and rotation (uniformly sampled between -1 to 1 degree) to the original high-resolution ground truth. Next, we apply a linear color transformation to the spatially shifted reference image to model the color space difference between the cameras used to capture the burst, and the HR reference. The color transformation is applied before clipping the image intensity values to be between 0 to 1, which models saturation in cameras. Note that these are simple approximations of the real-world misalignment between cameras. In practise, the spatial shift between images cannot be modelled using a simple rigid transformation due to perspective changes caused by differences in focal lengths [12]. Furthermore, the color space mapping between two cameras can be non-linear, and depend on the scene illumination.

We generate the training bursts using the sRGB images from the training split of Zurich Raw to RGB dataset [6].

The test set, containing 200 bursts, is generated using the sRGB images from the test split of Zurich Raw to RGB dataset. Note that for we provide accurately aligned groundtruth for the test bursts for quantitative evaluation. Both the training and test bursts contain 16 96×96 Raw images.

3. Details about comparison on SynBSR in Sec. 4.1

Training details: For each network architecture (DBSR, DeepRep, BIPNet, and Burstormer), we use the official model released by authors, trained on synthetic data as our **Synthetic data** baseline model. Note that these models are trained on synthetic bursts generated using the Bilinear blur kernel. We then finetune the synthetic data model on weakly-paired bursts from SynBSR for 60k iterations, using the aligned L1 loss introduced in [1] to obtain the **weakly-paired** baseline. For the models trained using our self-supervised approach, we also start with the official models trained on synthetic data. These models are finetuned for 60k iterations using our self-supervision loss. We start with a base learning rate of 10^{-4} for the burst SR model, which is reduced by a factor of 5 after 20k iterations. We initialize the blur kernel with an Isotropic Gaussian with standard deviation of 1.8 pixels. The blur kernel is trained with an initial learning rate of 10^{-3}, which is reduced by a factor of 5 after 20k and 40k iterations.

Evaluation details: Since a perfectly aligned ground truth is available in our synthetic setup, we use the fidelity based metrics PSNR and SSIM [8] and the perceptual metric LPIPS [11] for evaluating the methods. Due to the training on misaligned data using explicit alignment, we found that the network trained using weakly-paired method can introduce global spatial shifts in the predictions. If not accounted for, these spatial shifts can lead to very low PSNR. In order to not penalize the method for these global shifts, we first compute a per-image global translation between the network prediction and the ground truth for the weakly-paired trained network. The performance metrics are computed after aligning the prediction to the ground truth using the estimated global translation. The boundary regions (40 pixels on each side) are omitted when computing the performance metrics for all the methods.

4. Details about the comparison on BurstSR in Sec. 4.1

We train our models on the Raw bursts from the BurstSR [1] dataset. The dataset contains bursts captured from a smarthphone camera, along with a corresponding high-resolution reference captured using a DSLR camera with optical zoom. We use the pre-processed version of the dataset which contains 160×160 crops extracted from the central region of the images which cover the same field of view as the DSLR high-resolution reference. This ensures that we use the same inputs bursts for training as the weakly-paired training approach, which utilizes the DSLR reference. However, unlike the weakly-paired approach, our method is not restricted to only using the central crops since we do not use any high-resolution reference images for training.

For each model architecture, we use the official model released by authors, trained on synthetic data and BurstSR dataset as our **Synthetic data** and **Weakly-paired** baseline models, respectively. For the models trained using our self-supervised approach, we also start with the official models trained on synthetic data. These models are finetuned using our self-supervision loss. We initialize the blur kernel with an Isotropic Gaussian with standard deviation of 2.25 pixels.

5. Details about generalization to other sensor (Sec. 4.2)

We train a DeepRep [2] using the 77 bursts captured using the Nexus 6 camera from the HDR+ dataset [5]. The bursts from the HDR+ dataset contain varying number of images (2-10). We only utilize bursts containing at least 9 images for training. We use $K = 6$ of the images as input to the burst SR model, while 3 images are used to compute the self-supervision loss. We use the network pre-trained on the BurstSR dataset using our self-supervised loss as our initial model. This model is then finetuned for 40k iterations on the HDR+ bursts, using a batch size of 5. We use an initial learning rate of 10^{-4} for the burst SR model, and 10^{-3} for the blur kernel parameters. The learning rate is reduced by a factor of 5 after 20k iterations. We use crops of resolution 112×112 for training. During training, we discard textureless crops, i.e. crops with very low variance, since these groups do not sufficient provide supervision for super-resolution.

6. Details about synthetic experiments in Sec. 4.3

Here, we provide more details about our experiments on synthetic data, presented in Section 4.3 in the main paper.

6.1. Data Generation

We use the pipeline introduced in [1] to generate synthetic bursts for our experiments. We use the default parameters utilized in [1] for our inverse camera pipeline, and to simulate camera motion. The different blur kernels utilized in our data distributions are described next.

Unit Impulse: We use a 9×9 kernel with a weight of 1.0 in the central location, and 0.0 everywhere else.

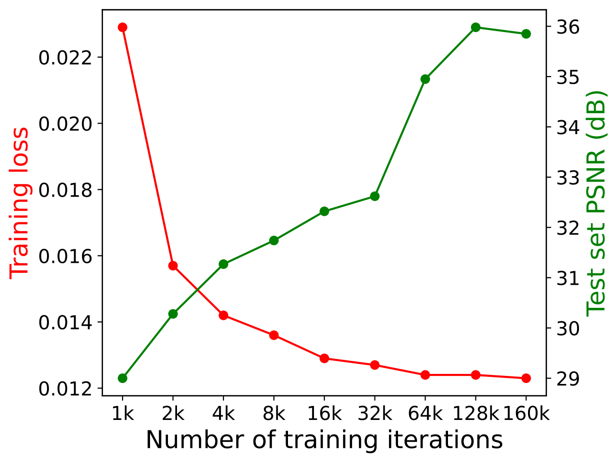

Figure 1: Evolution of training loss and performance on test set (in terms of PSNR) with number of training iterations.

Isotropic Gaussian: We use a 9×9 kernel representing a discritized isotropic 2D Gaussian with a standard deviation of 0.9 pixels.

Anisotropic Gaussian: We use a 9×9 kernel representing a discritized anisotropic 2D Gaussian with a standard deviation of 0.45 and 1.35 pixels along the two-axes, oriented at 45 degrees w.r.t. the X-axis.

We employ the commonly used camera noise model consisting of a read noise and the shot noise [3]. This can be modelled as a heteroscedastic Gaussian noise with a variance $\sigma_{\text{read}}^2 + \sigma_{\text{shot}}x$, where σ_{read}^2 and σ_{shot} are the read and shot noise parameters, respectively. x denotes the clean pixel value. Next, we describe the read and shot noise parameter utilized in the different data distributions in our experiments.

Read & Shot, Mix: We use the similar noise distribution as in [3]. The shot noise level σ_{shot} is sampled uniformly in the log-domain from the range $[0.0001, 0.01]$, while the read noise level σ_{read}^2 is sampled in the log-domain from a normal distribution $\log(\sigma_{\text{read}}^2) \sim \mathcal{N}(2.18\log(\sigma_{\text{shot}}) + 1.2, 0.26)$.

Read & Shot, Fixed: We use fixed noise levels with the shot noise parameter set to $\sigma_{\text{shot}} = 0.001$, and the read noise parameter set to $\sigma_{\text{read}}^2 = 9.57 \cdot 10^{-7}$.

Only Read, Fixed: We use fixed noise levels with the shot noise parameter set to $\sigma_{\text{shot}} = 0.0$, and the read noise parameter set to $\sigma_{\text{read}}^2 = 0.0002$.

6.2. Network Architecture

We utilize the DeepRep architecture introduced in [2] for our experiments. We use a smaller variant of the model used in [2] for the $2\times$ super-resolution task. In particular, we an encoder network consisting of 5 residual blocks with 32 channels each. The encoder output is mapped to a 128 dimensional embedding space. We use a single iteration in our optimizer module to minimize the reconstruction error, instead of 3 employed in [2]. The output of the optimizer is a 64 dimensional feature map. This is processed by 3 residual blocks, upsampled by a factor of 2 using bilinear interpolation followed by a convolution layer. The resulting 32 channel feature map is further processed by 3 residual blocks to generate the output RGB image.

For our oracle single-image super-resolution (Fully Sup. SISR) network, we use a residual network with bilinear upsampling, with a similar depth as the burst super-resolution network.

6.3. Training details

We train our networks using the synthetic bursts generated using sRGB images from the training split of Zurich Raw to RGB dataset [6]. We use bursts containing 14 128×128 Raw images for training. During our training, we use $K = 8$ input images as the model input $\mathcal{B}_{\text{model}}$, while the remaining 6 images are used for computing the self-supervision loss. We train our networks with a batch size of 12, for 160k iterations. We start with a base learning rate of 10^{-4}, which is reduced by a factor of 5 after 100k and 140k iterations.

When learning the blur kernel k along with the burst SR network, we empirically found it beneficial to first train only the burst SR model using a fixed Unit Impulse blur kernel, and then jointly learn both the kernel and the SR model. Thus, we train our model first for 60k iterations with a fixed Unit Impulse blur kernel. Next, we initialize the blur kernel to a wide isotropic Gaussian, and jointly train the blur kernel, as well as the model parameters. We use a 10 times higher learning rate (i.e. base learning rate of 10^{-3}) for the blur kernel parameters, as compared to the burst SR model parameters. For the data distributions generated using the Unit Impulse, and the Anisotropic Gaussian, we initialize the blur kernel with a standard deviation of 0.9 pixels. In order to ensure that we do not initialize the blur kernel to the ground truth, we use a standard deviation of 1.35 pixels when initializing the kernel for the Isotropic Gaussian data distribution.

6.4. Evaluation details

For each of the 5 data distributions, we evaluate the trained models on corresponding test datasets. The test bursts are synthetically generated in the same manner as the training bursts. We use the sRGB images from the test split of the Zurich Raw to RGB dataset [6] to generate the test bursts. Our test datasets contain 200 bursts containing 14 200×200 Raw images each.

7. Analysis of training convergence

Here, we analyse how the network performance evolves during training. Our analysis is performed on the synthetic

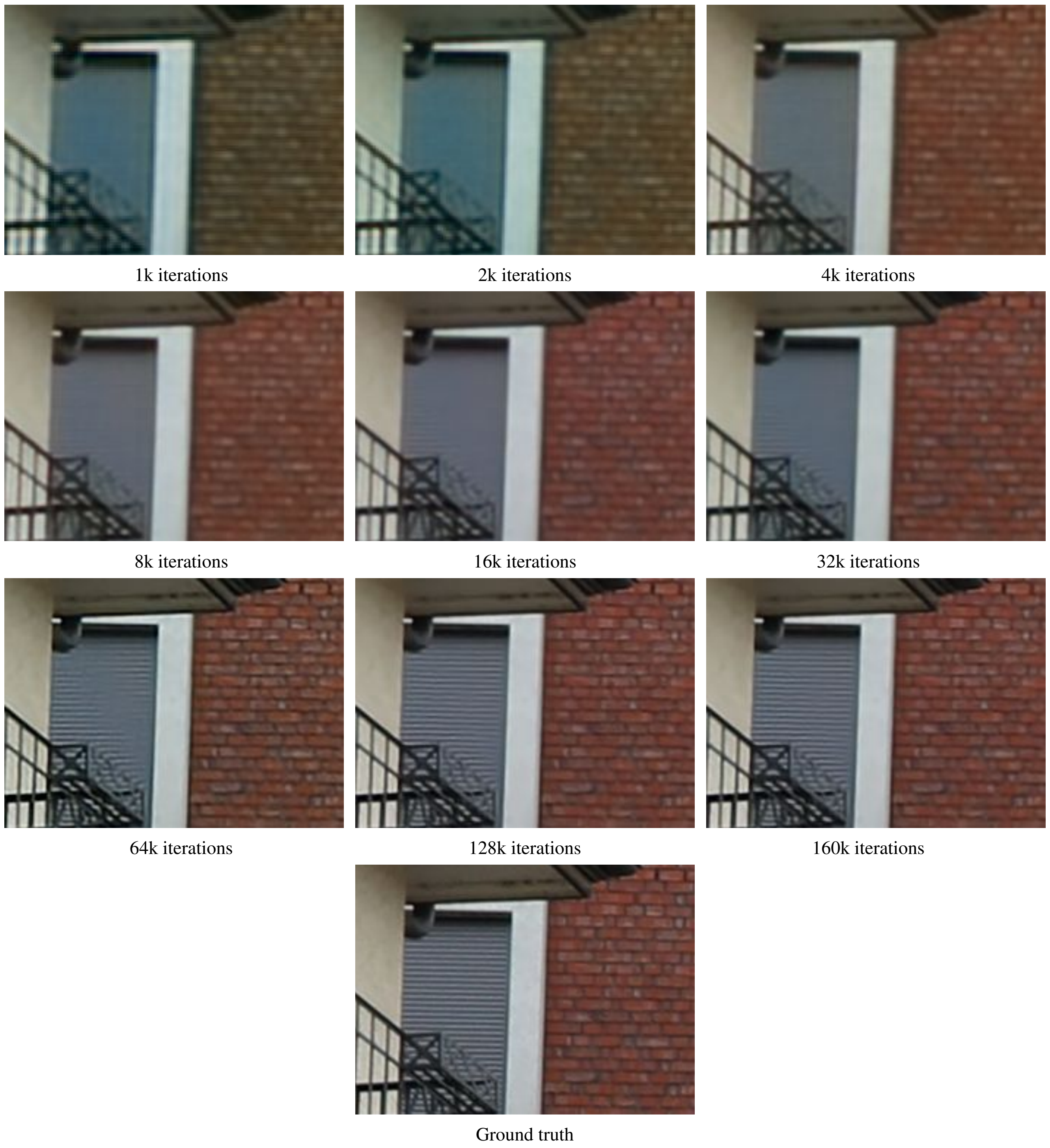

Figure 2: Evolution of network prediction with respect to the number of training iterations. The network predictions have large color shifts in the initial training iterations. As the training progresses, the network results become sharper and it can recover more details.

bursts generated using the Isotropic Gaussian kernel and mixed read and shot noise (see Section 6.1). In Figure 1, we plot the training loss, as well as the performance on the test set (in terms of PSNR) for different number of training iterations. Note that we start with randomly initialized network. We keep the blur kernel k fixed during the first 60k training iterations, and finetune it with the rest of the model parameters thereafter. This leads to a substantial PSNR improvement between the models trained for 32k and 64k iterations. Both the training loss, as well as the test set performance, converge in 160k iterations.

A visual comparison between networks trained for different number of iterations if provided in Figure 2. We observe that the network predictions have large color shifts in the initial training iterations. These color shifts are largely fixed by 4k training iterations. The early predictions are also more blurry due to the use of incorrect blur kernel k. As the training progresses, the network results become sharper and it can recover more details, *e.g*. the window blinds.

8. Additional analysis of motion estimation

In Section 3.3 in the main paper, we detail how the motion between two frames is estimated when computing our self-supervision loss. Here, we analyse the impact of error in the motion estimation. Our experiments are performed on the synthetic bursts generated using the Isotropic Gaussian kernel and mixed read and shot noise. We use the same model as employed in Section 4.1 in the main paper. We also assume that the blur kernel k is known for this analysis.

In order to analyse the impact of motion estimation error, we start with the ground truth motion between frames and add independent Gaussian noise to the per-pixel flow vectors. The results obtained using varying amounts of noise are shown in Table 1. In particular, we add Gaussian noise with standard deviations of 0.1, 0.2, 0.5, 1.0, 2.0, and 5.0 pixels. We also include the results obtained using the ground truth motion, as well as estimated motion, for comparison. The results show the our approach can handle small amount of error in the motion estimation ($<$ 0.2 pixels) without any degradation in performance. However, larger errors can lead to substantial decrease in performance.

9. Additional qualitative results on BurstSR dataset

In Figure 4 and 5 in the main paper, we provide a qualitative comparison of our self-supervised training approach with alternative methods. In Figure 3, we provide more qualitative examples on the BurstSR dataset [1], using the DeepRep architecture [2]. Our approach obtained comparable results to the weakly-paired alternative, despite using only low-resolution noisy bursts for training. A similar comparison using the BIPNet architecture [4] is provided in Figure 4.

10. Results in linear camera color space

All our networks are trained to generate predictions in the same color space as the camera. In all our qualitative results on the real-world bursts, we use Adobe Camera Raw to postprocess these predictions in order to generate sRGB images for visual comparison. This entails applying camera white balance and converting to linear RGB color space, followed by tone mapping, gamma correction, and brightness adjustment. We additionally perform some sharpening to generate visually pleasing results. For each method, we use the same set of manually tuned "slider" parameters for all the images.

Here, we present qualitative results in the linear camera color space, after performing white balance. We compare our self-supervised training approach with the supervised synthetic and real data training alternatives on the BurstSR dataset. The results are shown in Figure 5. Compared to the weakly-paired training approach, the results of our self-supervised training are softer, with less contrast. As shown in the main paper (Figure 5), this can be addressed with simple post-processing to generate visually pleasing results. Crucially, we observe that our self-supervised approach can recover most of the image details, similar to the weakly-paired training approach. Furthermore, we notice that the network trained using weakly-paired real data introduces noticeable color shifts, compared to the results of both the synthetic data training, as well as the self-supervised training approaches. We believe that this is due to the use of high-resolution DSLR images as ground truth. While the DSLR images captured using a zoom lens are sharper, they lie in different color space compared to the burst images, due to the differences in the imaging sensors.

11. Qualitative results for synthetic experiments

Here, we provide a qualitative comparison between the different models analysed on the synthetic data in Section 4.3 of the main paper. The results are shown in Figure 6 for the data distribution generated using the isotropic Gaussian kernel, with mixed read and shot noise. Our approach obtains results comparable to the fully-supervised training approach, even when the blur kernel k and motion m are unknown and estimated. Note that our approach obtains significantly better results compared to the single image SR network trained in a fully-supervised manner. This demonstrates that we can learn to effectively fuse information from the input burst to recover the high resolution details, despite using only low-resolution bursts for training.

12. Limitation and discussion

Our image formation model jointly learns a lens blur kernel that is spatially-invariant and fixed for images captured with the same device. This is a common assumption in super-resolution literature [9, 10]. One could extend our method to learn a spatially-varying blur kernel by changing

Ground truth m	Noisy m, std. 0.1	Noisy m, std. 0.2	Noisy m, std. 0.5	Noisy m, std. 1.0	Noisy m, std, 2.0	Noisy m, std. 5.0	Estimated m
36.765	36.854	36.625	36.106	34.800	31.612	27.100	36.347

Table 1: Impact of error in motion estimation. The experiments are performed on synthetic bursts. In order to obtain noisy motion estimates, we add Gaussian noise with varying noise levels to the ground truth motion vectors. The results show that our approach can provide good results when the motion estimation error is small (< 0.2 pixels).

our parameterization of the blur kernel. Furthermore, we do not model any motion blur, which may limit the performance of the model. Extending our approach to incorporate spatially varying blur is an interesting future work.

Another future direction is to integrate additional supervision in the form of unpaired high-resolution images, using *e.g.* adversarial training. This could help the network to learn to generate visually pleasing sharp images. Another issue that we observe in our results is that our method can introduce some artifacts near over-exposed regions. Note that our image formation model described in Section 3.1 in the main paper is not valid for over-exposed regions due to clipping in the sensor. Thus, it is hard to get reliable supervision around over-exposed regions in our training. This could also be addressed using an adversarial loss on the high-resolution outputs.

References

[1] Goutam Bhat, Martin Danelljan, L. Gool, and R. Timofte. Deep burst super-resolution. In *CVPR*, 2021. 1, 2, 5, 7, 8, 9

[2] Goutam Bhat, Martin Danelljan, Fisher Yu, Luc Van Gool, and Radu Timofte. Deep reparametrization of multi-frame super-resolution and denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2460–2470, 2021. 2, 3, 5, 7, 9

[3] T. Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and J. Barron. Unprocessing images for learned raw denoising. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11028–11037, 2019. 3

[4] Akshay Dudhane, Syed Waqas Zamir, Salman Khan, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Burst image restoration and enhancement. In *CVPR*, 2022. 5, 8

[5] Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 35(6), 2016. 2

[6] Andrey Ignatov, Luc Van Gool, and Radu Timofte. Replacing mobile camera isp with a single deep learning model. *arXiv preprint arXiv:2002.05509*, 2020. 1, 3

[7] Deqing Sun, X. Yang, Ming-Yu Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 1

[8] Zhou Wang, A. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004. 2

[9] Zhihao Xia and Ayan Chakrabarti. Training image estimators without image ground-truth. In *NeurIPS*, 2019. 5

[10] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *IEEE International Conference on Computer Vision*, pages 4791–4800, 2021. 5

[11] Richard Zhang, Phillip Isola, Alexei A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 2

[12] X. Zhang, Qi feng Chen, R. Ng, and V. Koltun. Zoom to learn, learn to zoom. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3757–3765, 2019. 1

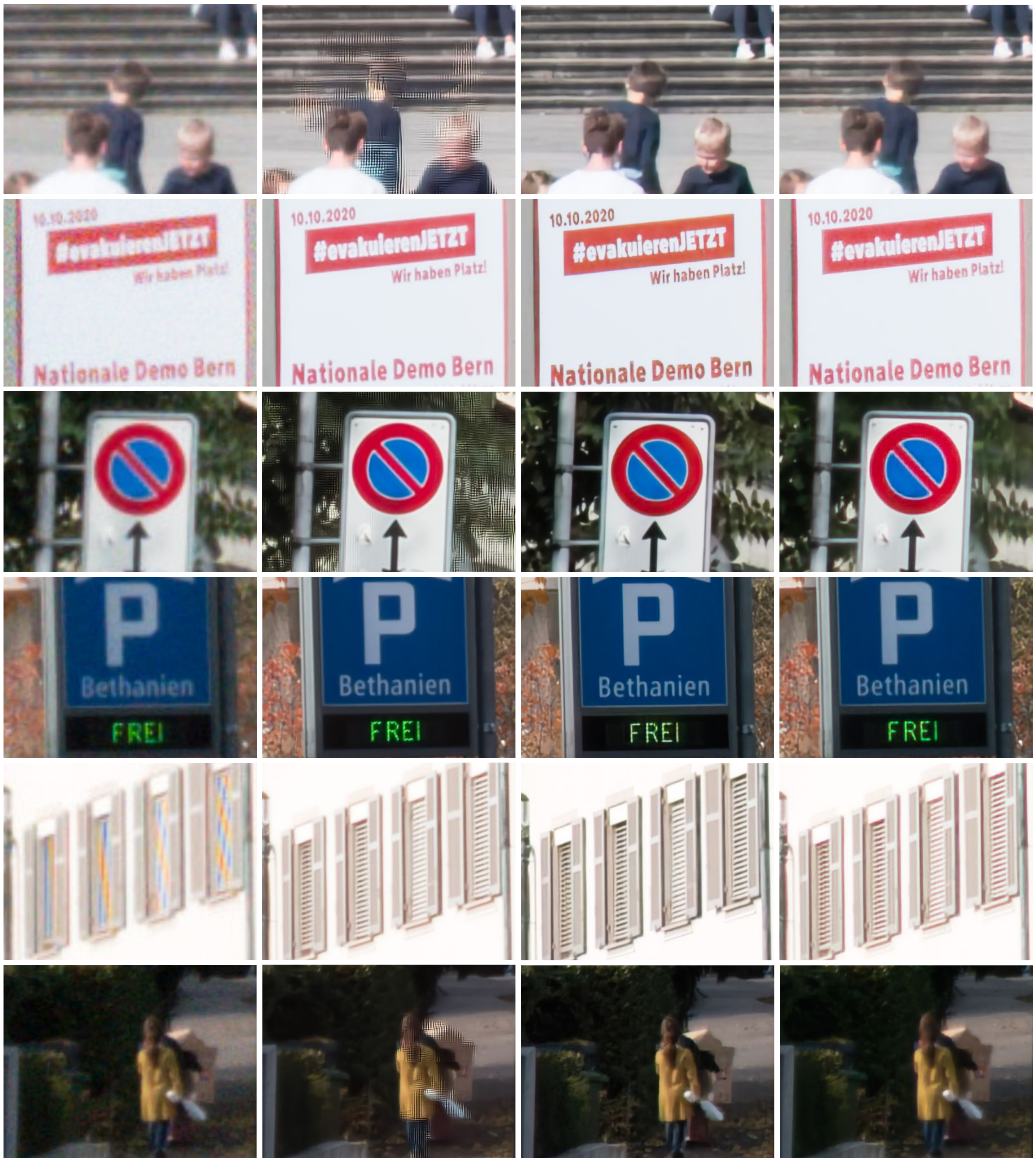

Figure 3: Comparison of our self-supervised training approach with supervised training alternatives using synthetic and weakly-paired real data on the BurstSR dataset [1] using the DeepRep architecture [2]. The network trained using synthetic data cannot handle dynamic objects, introducing severe artifacts. Our self-supervised training approach using only the noisy bursts for training obtains promising results, comparable to the weakly-paired alternative which relies on the availability of high-resolution reference images for each burst.

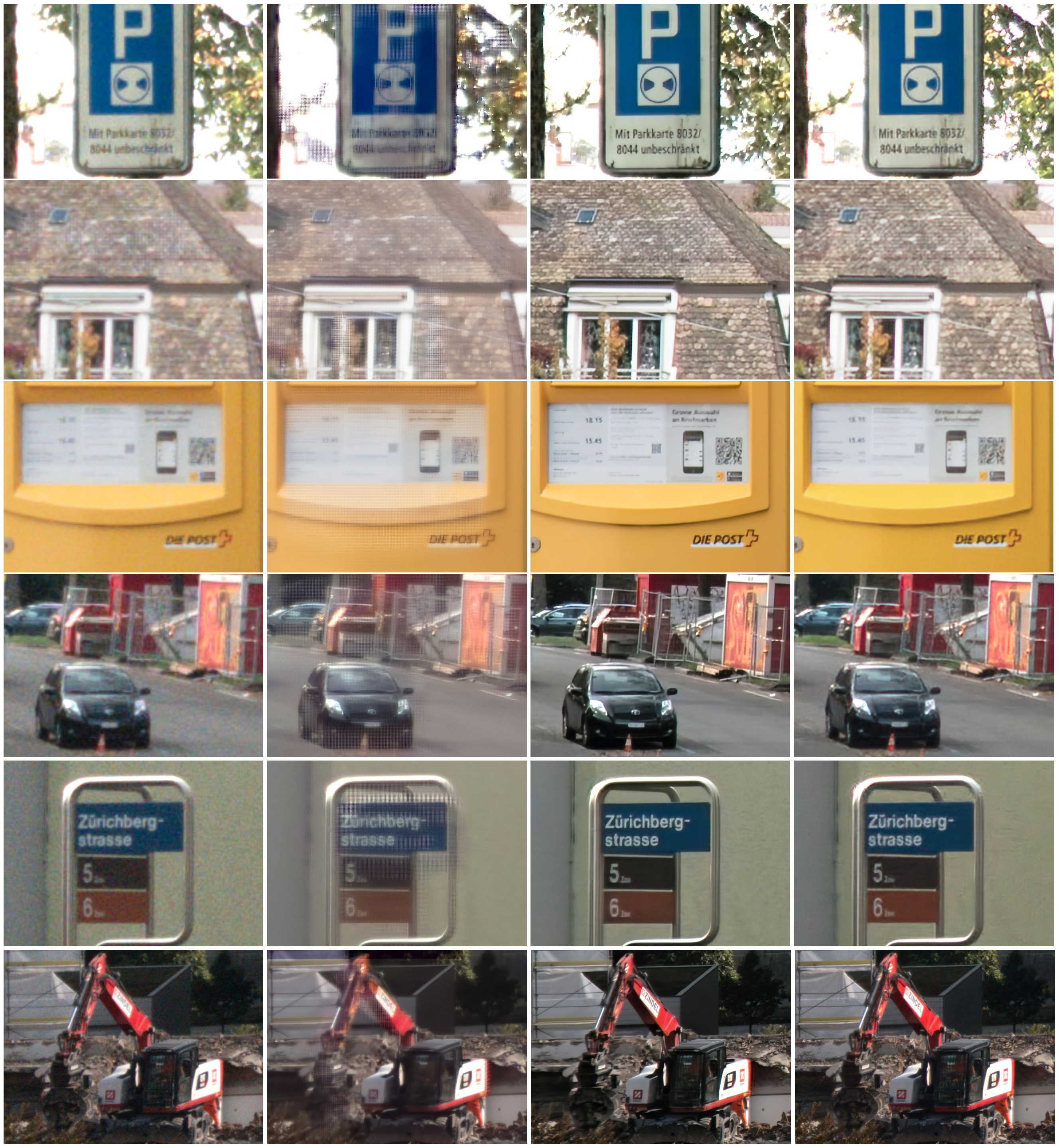

Figure 4: Comparison of our self-supervised training approach with supervised training alternatives using synthetic and weakly-paired real data on the BurstSR dataset [1] using the BIPNet architecture [4]. The network trained using synthetic data introduces severe artifacts due to domain shift. The network trained using the weakly-paired approach on real data provides good results. However such a training strategy requires aligned high-resolution references for each training burst, which are cubmersome to collect. Furthermore, it can introduce some color shifts since it is trained using the high-resolution image captured using a different camera, with a different color response. Our self-supervised training approach provides promising results, despite being trained using only low-resolution noisy bursts.

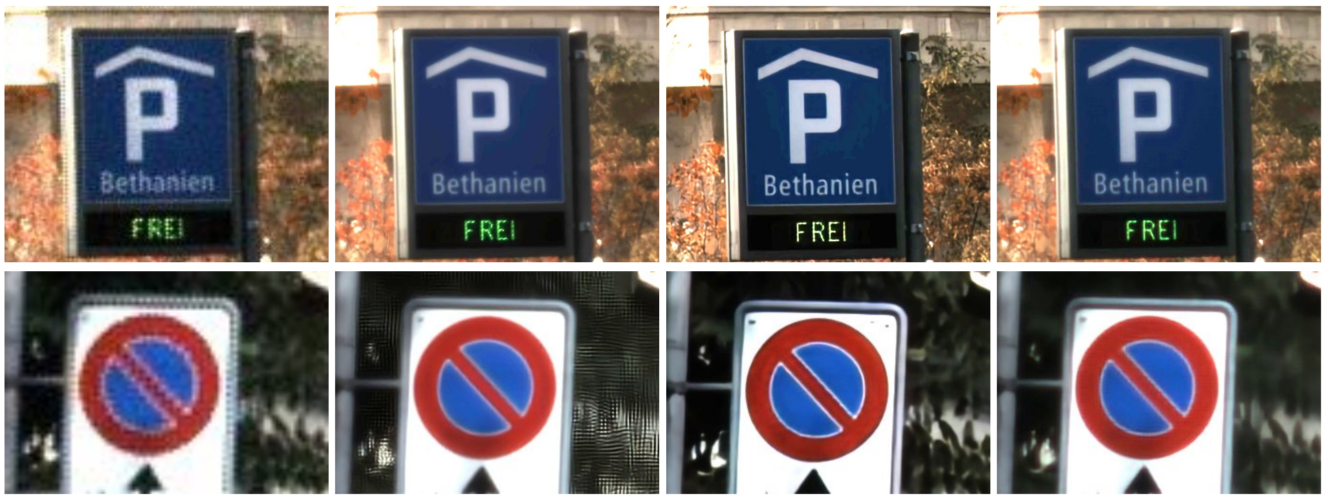

Figure 5: Comparison of our self-supervised training approach with supervised training alternatives using synthetic and weakly-paired real data on the BurstSR dataset [1] using the DeepRep architecture [2]. The results are shown in the linear camera color space, before any postprocessing. While being a bit softer, the results of our approach contain most of the image details compared to the weakly-paired training approach. As shown in the results in the main paper (Figure 5), the softness of our results can be easily addressed using standard post-processing techniques.

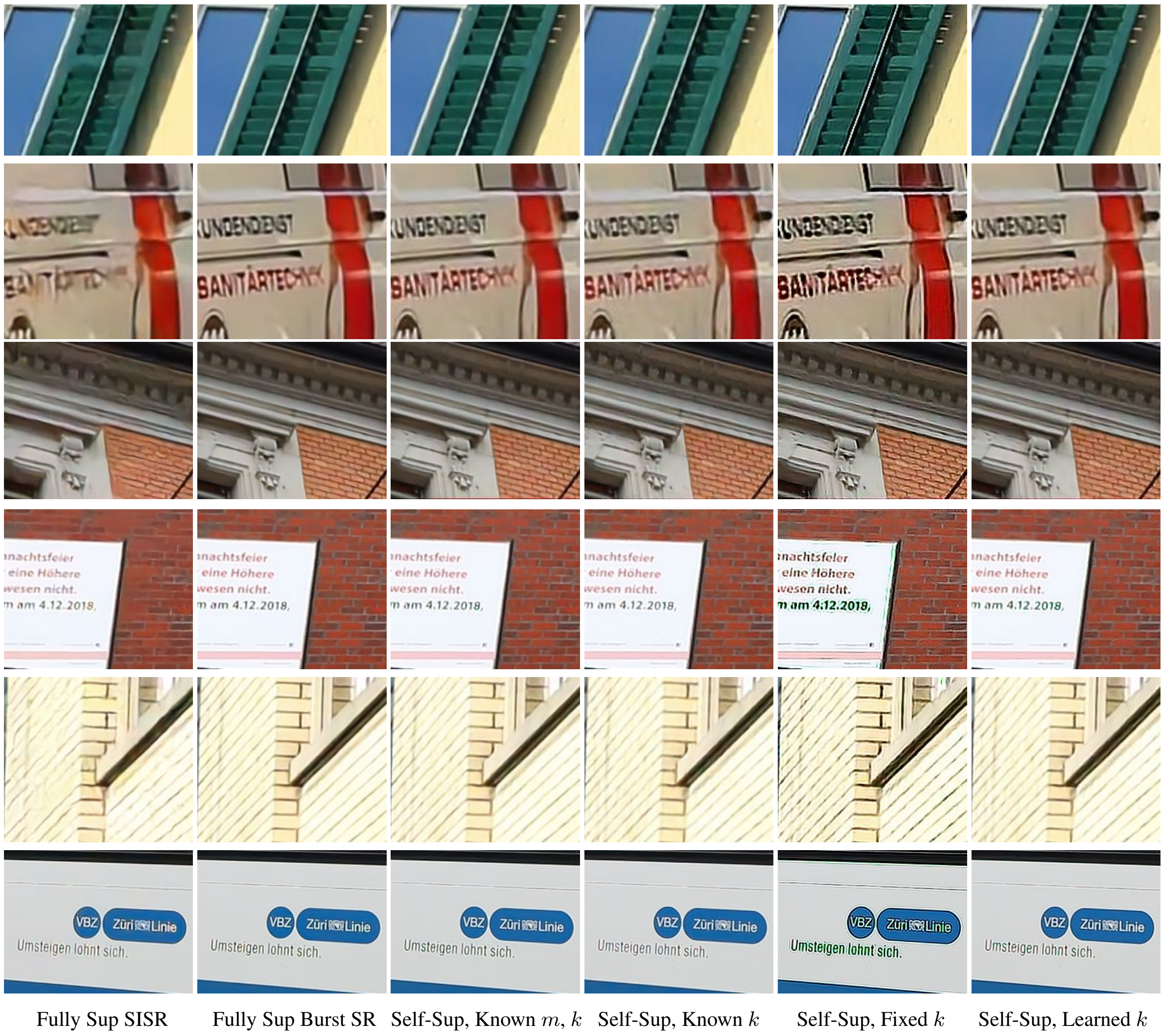

Figure 6: Qualitative analysis of our self-supervised training approach. The results are shown for the data distribution generated using the isotropic Gaussian kernel, with mixed read and shot noise. The model trained using our self-supervised approach with unknown blur kernel k and motion m (last column) obtains promising results, comparable to the fully-supervised alternative (second column). Compared to the single image SR model trained in a fully-supervised manner (first column), our self-supervised burst SR model can recover more details. We observe that using a fixed, incorrect kernel for self-supervised training can lead to poor results (fifth column).