A. Latent code and kernel shape

In Section 3.3 we describe the role of the latent code z and how it encodes local image information to guide the kernel network. In particular, noise and image features determine the support and shape of the kernel (Figure 9).

The burst sampling pattern also affects the learned kernels and final reconstruction. We create bursts from the same ground truth image at a fixed noise level, $\sigma_s = 0.01$, but with different translation offsets. The output images are similar in quality (PSNR average of 33.18 and variance 0.173 over 100 bursts) though not identical, as shown in Figure 10, first row. More evenly spaced samples result in better reconstructions (Figs. 10a, 10c). When the sampling misses large bands of the underlying scene, image degradations still arise in the output, such as less sharp edges and patchy smooth areas (Fig. 10d).

Though not explicitly provided, the encoder network learns the sampling pattern from the input frame stack and encodes this information in the latent z. For identical image content and noise levels, the learned kernels vary with burst sampling patterns (Fig. 10 top). Fixing the latent code image to be that of a particular set of burst offsets instead of the one corresponding to the current burst induces more severe color artifacts (Fig. 10, bottom). Bursts with more similar sampling patterns to the original burst (c vs d,e) are less affected as their latent codes, and kernels, are more similar.

B. Synthetic Data Burst Denoising and Demosaicking Additional Results

Figure 11 shows additional results on our synthetic test set of $N = 5$ frames for our method and the baselines detailed in Section 4.1.

C. Real Burst Denoising and Demosaicking Additional Results

Figure 12 shows a full resolution output from a real burst, using $N = 5$ frames of size 4032×3024. We show additional results on $N = 5, 512 \times 512$ real bursts in Figures 13, 14, and 15. The HDR+ result using 10 frames is shown for reference. The post-processing pipeline after demosaicking and denoising differs from ours so tone and geometry are expected to be different from the inputs and our outputs. All other baselines follow the same post-processing as in our method.

D. Chromatic Aberration Correction Additional Results

We show additional results on real bursts for our method and the baselines detailed in Section 5 in Figure 16.

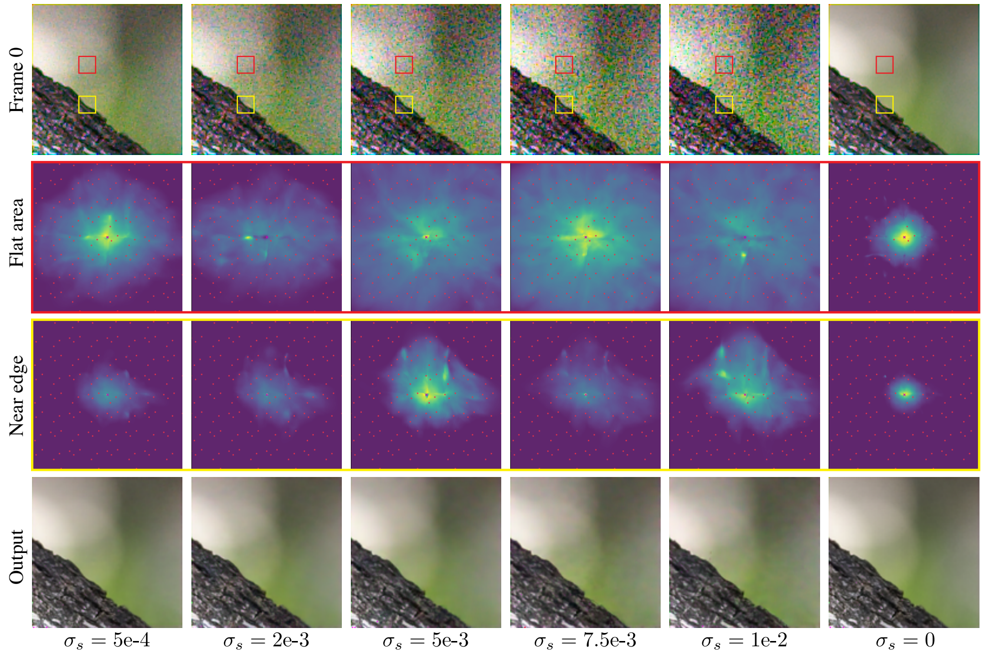

Figure 9. **Continuous kernels adapt to noise and local image information.** Our method lets the the kernels adapt their shape to the noise level in the image, resulting in output reconstructions with smooth flat regions and sharp detailed areas at all noise levels.

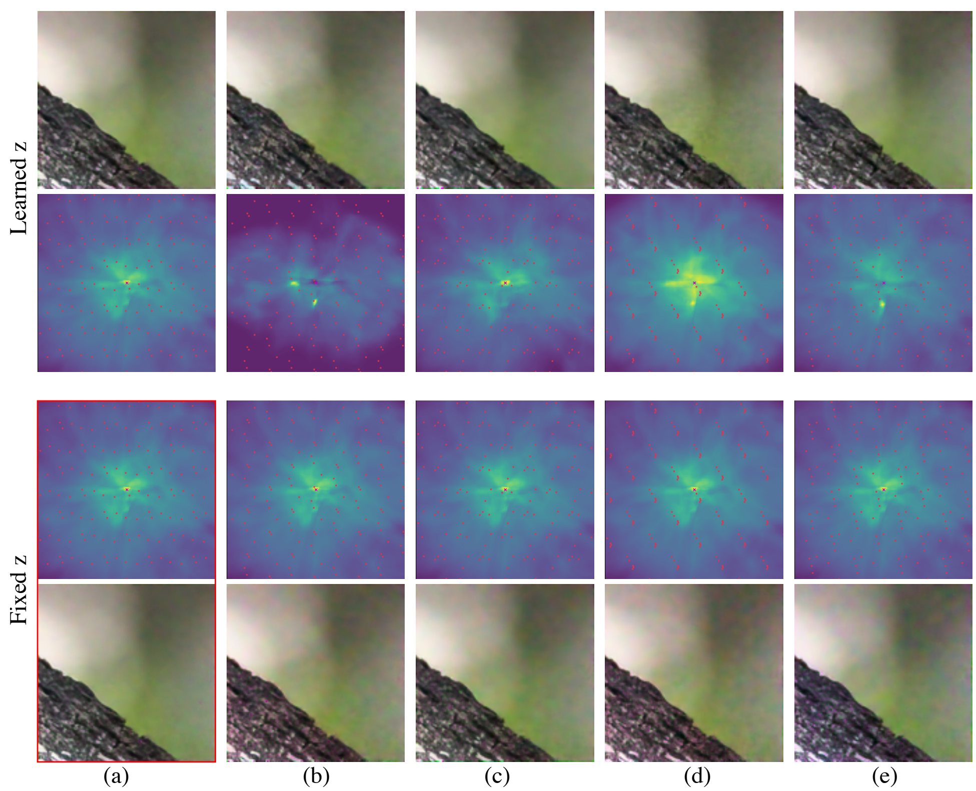

Figure 10. **Continuous kernels adapt to the sampling pattern.** Bursts with different sampling patterns result in reconstructions of similar quality but with very different kernels. Top: reconstruction output and example kernel when the latent code z is jointly learned with the kernel network. Bottom: reconstruction output and example kernel when using the latent code image of burst (a) for bursts with different sampling patterns (b)-(e). Red dots show the local sampling pattern.

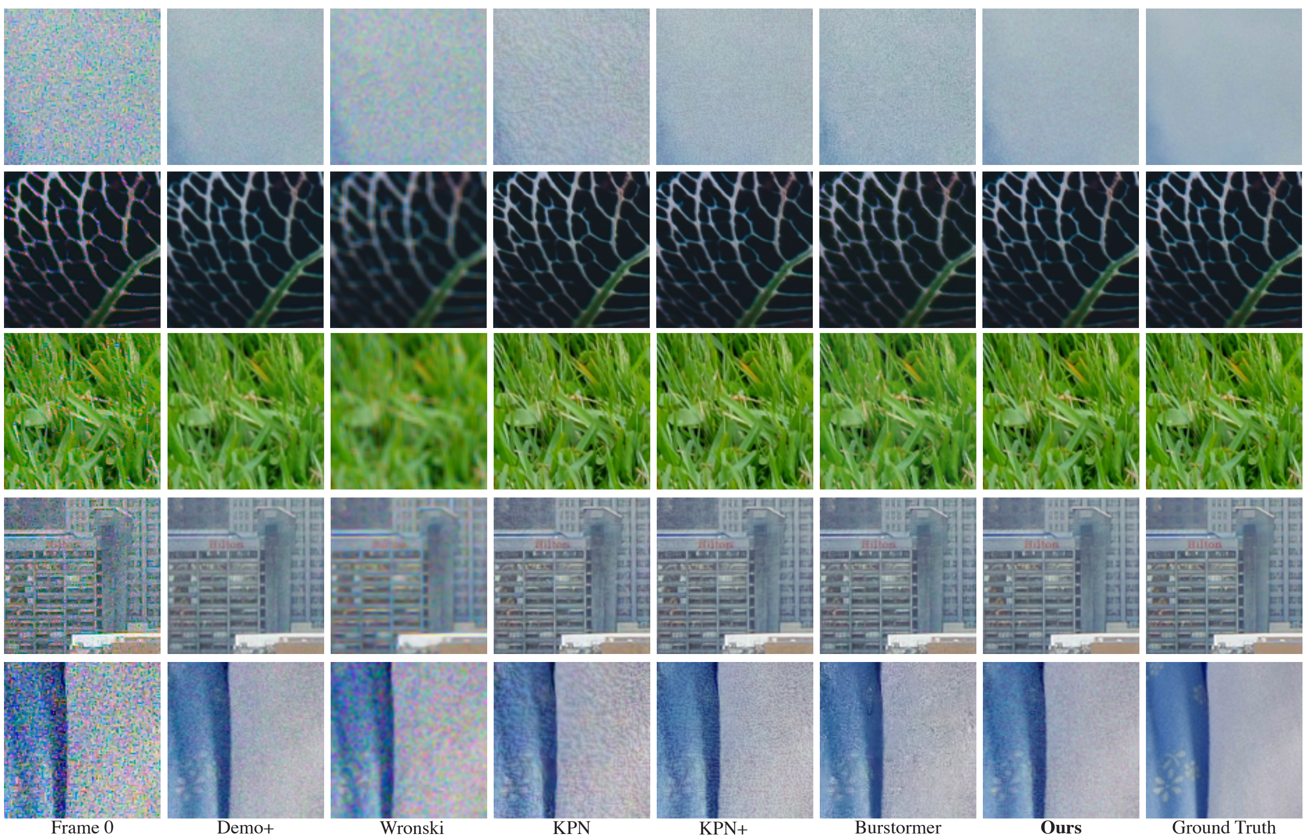

Figure 11. **Burst demosaicking and denoising on synthetic data.** The input is demosaicked for visualization

Figure 12. **Burst demosaicking and denoising on a real, full resolution image.** Our method can reconstruct bursts of any resolution, like this 12MP image. The first frame is demosaicked for visualisation.

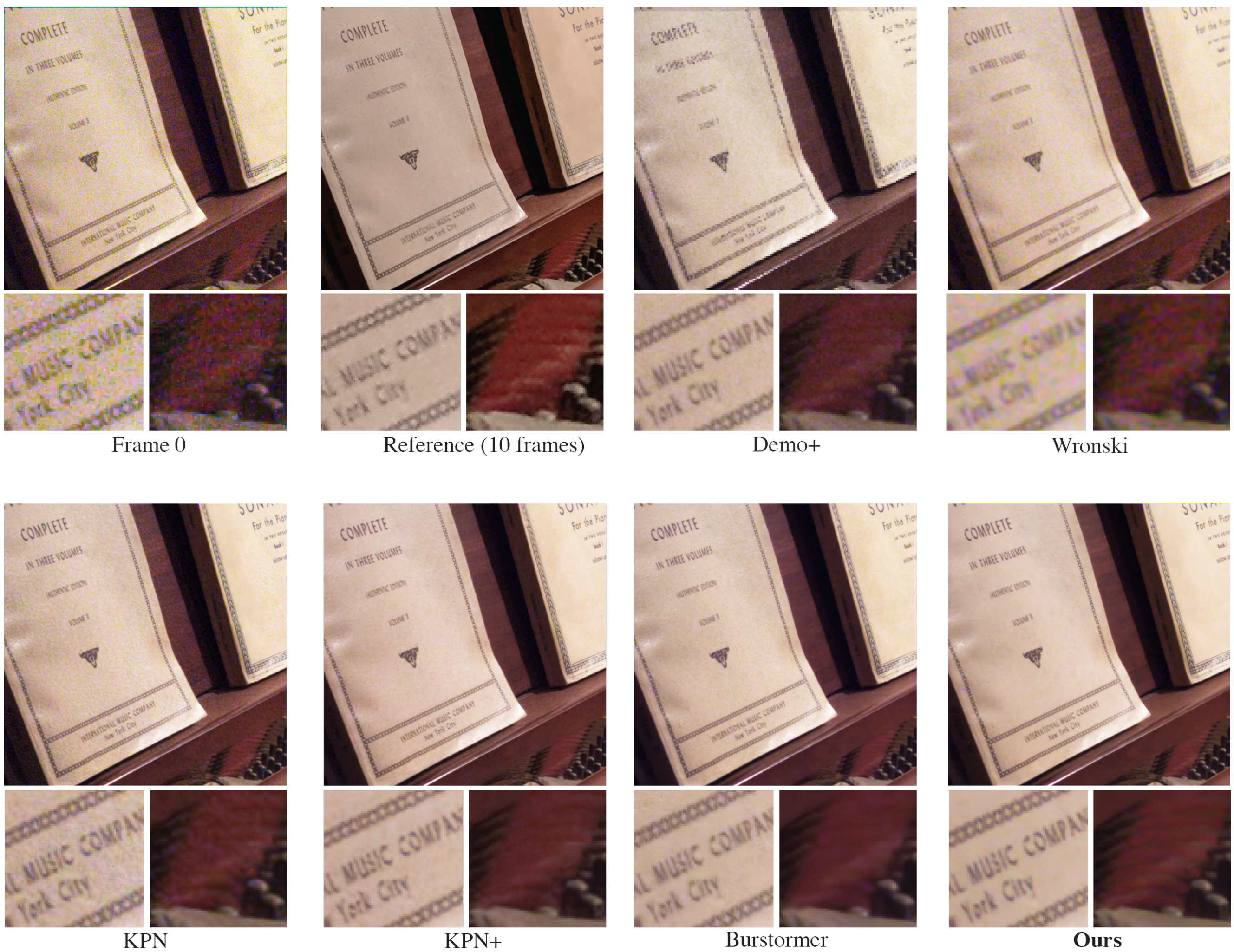

Figure 13. **Burst demosaicking and denoising on real bursts.** Our method better retains the detail in the piano strings and text on the sheet music. The first frame is demosaicked for visualization.

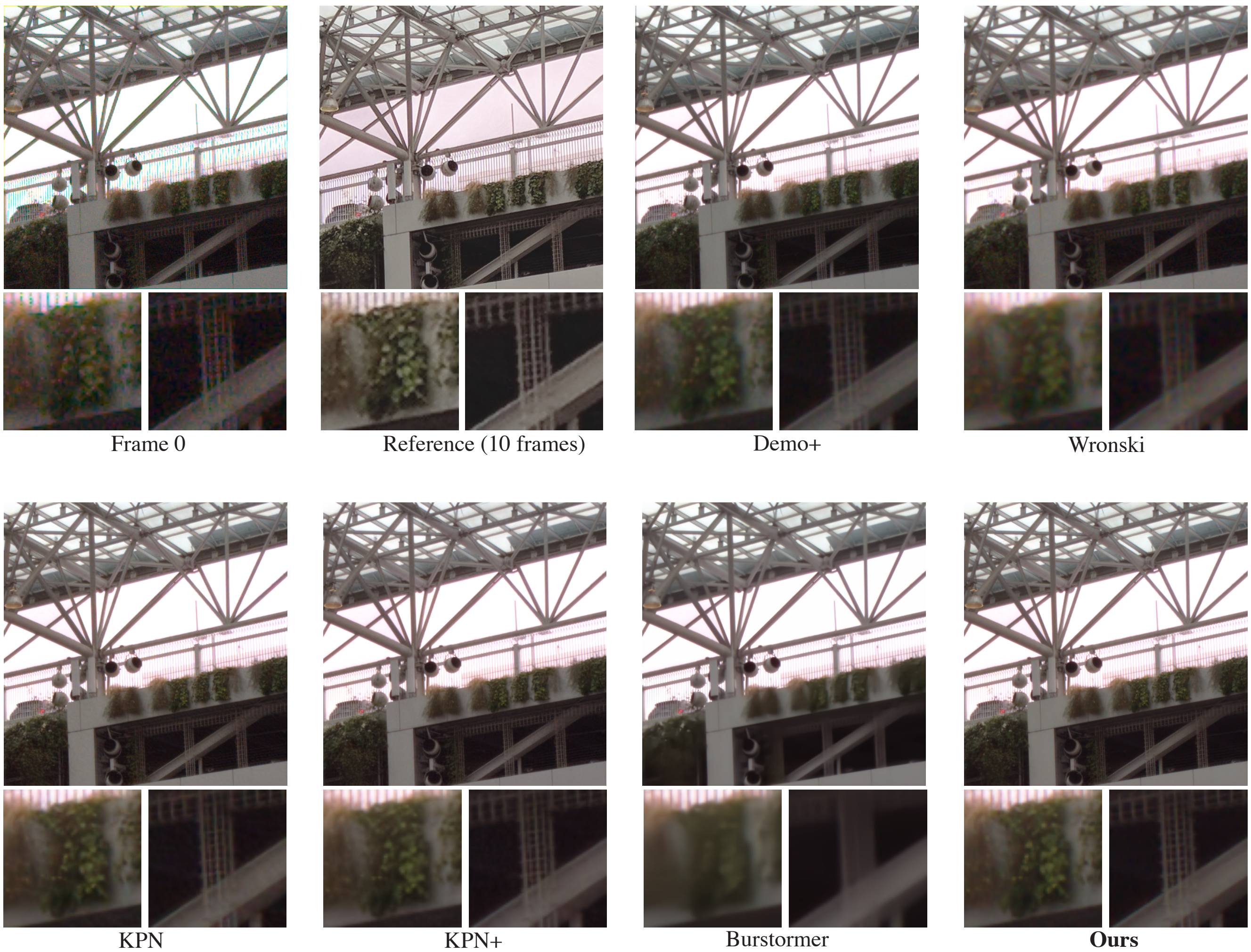

Figure 14. **Burst demosaicking and denoising on real bursts.** The first frame is demosaicked for visualization.

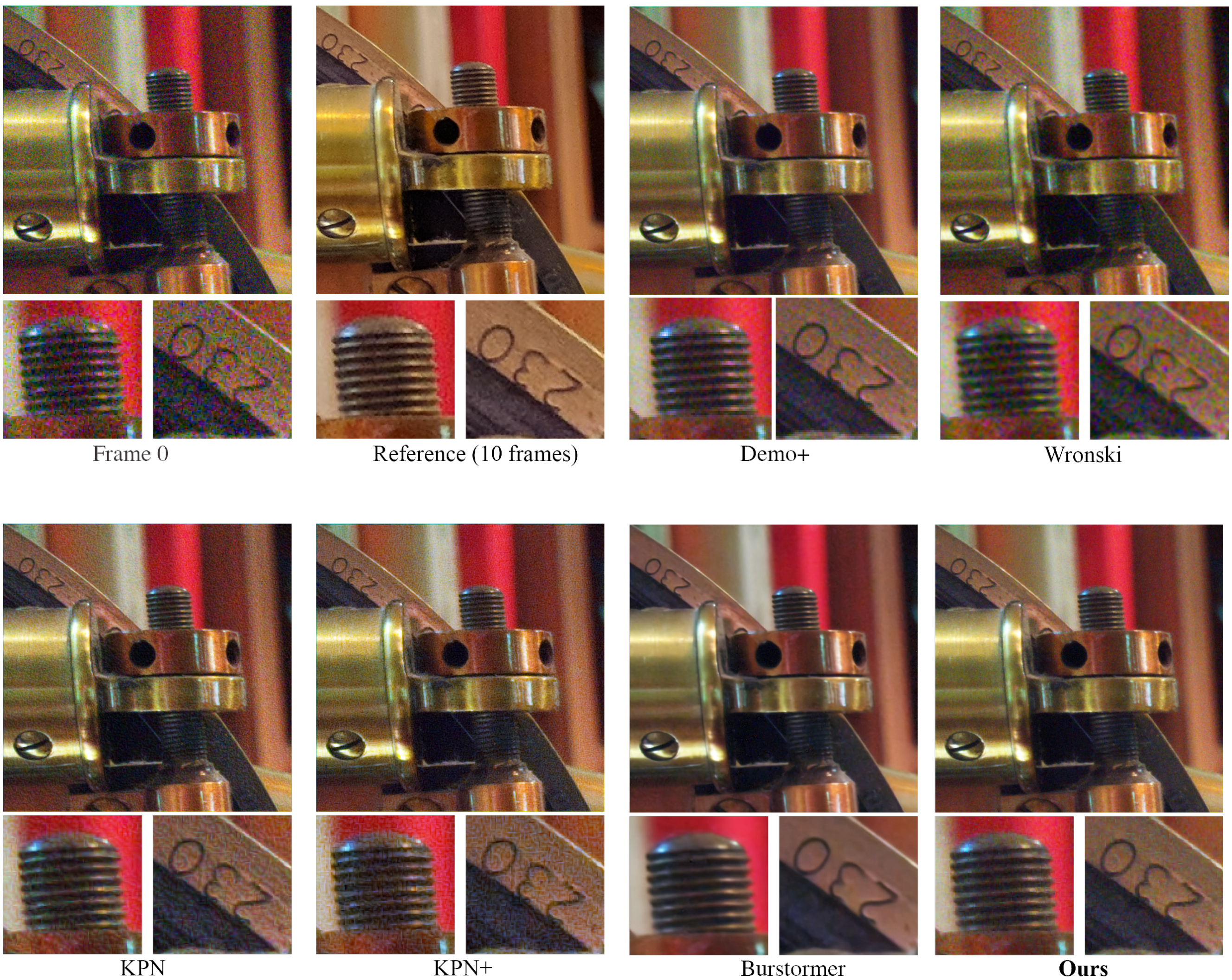

Figure 15. **Burst demosaicking and denoising on real bursts.** The first frame is demosaicked for visualization.

Figure 16. **Joint demosaicking, denoising and chromatic aberration correction on real images.** We show the reconstructed output from our model (right) when using real images as input (left), as well as detailed crops of the the input after the alignment and denoising steps of the sequential approach (center). The input is shown demosaicked for visualization.