Figure 2. Schematic illustration of the dimensionality stretching strategy. For an LR image of size $W \times H$, the vectorized blur kernel is first projected onto a space of dimension t and then stretched into a tensor $\mathcal{M}$ of size $W \times H \times (t+1)$ with the noise level.

Figure 3. The architecture of the proposed convolutional super-resolution network. In contrast to other CNN-based SISR methods which only take the LR image as input and lack scalability to handle other degradations, the proposed network takes the concatenated LR image and degradation maps as input, thus allowing a single model to manipulate multiple and even spatially variant degradations.

Figure 2. The network architecture of the proposed UDVD framework.

Figure 1. Schematic illustration of the proposed degradation model for scale factor 2. For an HR image, the randomly shuffled degradation sequences $\{B_{1\cdots7}, B_{shuffle}, D^{2}, N_{G}, N_{JPEG}, N_{S}\}$ are first performed, then a JPEG compression degradation N_{JPEG} is applied to save the LR image into JPEG format. The downscaling operation with scale factor 2, *i.e.*, D^{2}, is uniformly chosen from $\{D^{2}_{nearest}, D^{2}_{linear}, D^{2}_{bicubic}, D^{2}_{area\text{-}up}\}$.

Figure 4. The architecture of the proposed SFTMD network. The design of the proposed SFT layer is shown in pink box.

Algorithm 1 Iterative Kernel Correction

Require: the LR image I^{LR}
Require: the max iteration number t

1: $h_0 \leftarrow \mathcal{P}(I^{LR})$ (Initialize the kernel estimation)
2: $I_0^{SR} \leftarrow \mathcal{F}(I^{LR}, h_0)$ (The initial SR result)
3: $i \leftarrow 0$ (Initialize counter)
4: **while** $i < t$ **do**
5: $\quad i \leftarrow i + 1$
6: $\quad \Delta h_i \leftarrow \mathcal{C}(I_{i-1}^{SR}, h_{i-1})$ (Estimate the kernel error using the intermediate SR results)
7: $\quad h_i \leftarrow h_{i-1} + \Delta h_i$ (Update kernel estimation)
8: $\quad I_i^{SR} \leftarrow \mathcal{F}(I^{LR}, h_i)$ (Update the SR result)
9: **return** I_t^{SR} (Output the final SR result)

Figure 2: Overview of the pure synthetic data generation adopted in Real-ESRGAN. It utilizes a second-order degradation process to model more practical degradations, where each degradation process adopts the classical degradation model. The detailed choices for blur , resize, noise and JPEG compression are listed. We also employ *sinc* filter to synthesize common ringing and overshoot artifacts.

Figure 1: The overview structure of the deep alternating network (DAN).