Coordinate-based Feature Compensation Network for Screen Content Image Super-Resolution (Supplementary Material)

Anonymous submission

We offer additional discussions and experiments in this supplementary material. First, we describe the Broader Impacts of the paper, and then supplement the relevant experimental results on the SCI1K dataset, including quantitative and qualitative results. Then we give the performance results of our method on natural images, illustrating the versatility of our method. Finally, we summarize and discuss the entire paper.

We have also provided the core code to enhance understanding and evaluation by the reviewers in an accompanying package. The complete and comprehensive codebase will be made available upon acceptance of the paper.

Broader Impacts

The broader impacts of our work on the Coordinate-based continuous implicit Feature Compensation Network (CFCNet) for screen content image super-resolution extend across various domains in both academia and industry. By addressing the unique challenges of screen content images (SCIs), our method not only improves the visual quality of SCIs but also enhances user experiences across a wide range of digital platforms.

In education and remote work, where high-quality screen content is essential, our approach can significantly improve the clarity and legibility of digital content, aiding in better communication and understanding. This is particularly important in scenarios where screen sharing or online collaboration tools are used, as it ensures that all participants can view content with minimal loss of detail, regardless of the resolution of their devices.

In the gaming and entertainment industry, where screen content often includes detailed graphics and text, our method can enhance the visual experience by providing higher resolution outputs, thereby preserving the integrity of original designs and improving the overall user experience.

Furthermore, we proposes a novel **C**oordinate-based continuous implicit **F**eature **C**ompensation Network (**CFCNet**) for arbitrary scale SCI SR. This can inspire future research in the domain of image super-resolution, particularly for specialized content types.

By making our code and models available to the research community, we encourage further exploration and development in this area, potentially leading to new applications and improvements in various digital imaging technologies. Our approach can also contribute to reducing bandwidth and storage requirements, as it allows for lower-resolution images to be transmitted or stored while still enabling high-quality reconstruction on the user side. This has the potential to optimize resource usage in network and cloud-based services, aligning with broader efforts toward sustainable computing.

One limitation of our method is the introduction of additional learnable parameters, which increases the overall parameter count compared to the baseline version. Despite this increase, we carefully balanced the model's complexity with performance gains. Specifically, our approach results in a 7% increase in the number of parameters, but this modest growth allows the model to achieve optimal results, demonstrating the effectiveness of the additional features in enhancing performance.

Implementation Details

We implement our CFCNet in the PyTorch framework and train all parameters on NVIDIA GeForce RTX 3080Ti GPU. Following (Chen, Liu, and Wang 2021), we train the models for 1,000 epochs with a batch size of 8. The learning rate starts at 1×10^{-4} for all modules and is halved every 200 epochs. We use the L1 loss as the loss function and the Adam (Kingma and Ba 2014) optimizer is used with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

In the training phase, to simulate continuous magnification, the downsampling scale is sampled from a uniform distribution $U(1, 4)$. We then randomly crop 48×48 patches from the LR images and augment them via flipping and rotation. To supervise the training process, we randomly sample 48×48 pixels from the corresponding HR patch and coordinates.

During the test phase, the entire LR image is directly fed into our network (as long as GPU memory permits) to generate the super-resolution result.

Algorithm 1 Coordinate-based continuous implicit Feature Compensation Network (CFCNet) for SCI SR

Input:
I^{LR}: Low-resolution input image
x_q: Query coordinates for target high-resolution (HR) image
L: Number of Coordinate-based Feature Compensation Layers
Encoder: The encoder network to extract initial features
Decoder: The decoder f_θ implemented with a trainable MLP, maps the final unified feature $\mathbf{y}_{fuse}^{[L+1]}$ into RGB values.
Output:
I^{HR}: High-resolution output image

1: *Encode low-resolution image to latent code z*
   $z \leftarrow \text{Encoder}(I^{LR})$
2: **for** each coordinate x_q in the HR grid **do**
3: *Coordinate-based Feature Compensation Layer*
4: **for** i =1 to L do **do**
5: *Apply Coordinate-Aware Feature Compensation Module*
   $y_i^{[l+1]} \leftarrow \text{CAFCM}(z, x_q)$
6: *Apply Dynamic Multi-Coordinate Feature Integration Module*
   $\mathbf{y}_{fuse}^{[l+1]} \leftarrow \text{DMFIM}(y_i^{[l+1]})$
7: **end for**
8: *Map feature to RGB value at coordinate x_q*
   $I^{HR}(x_q) \leftarrow f_\theta(y_i^{[L+1]}, x_q)$
9: **end for**
10: *Return the reconstructed high-resolution image*
11: **return** I^{HR}

Follow (Chen, Liu, and Wang 2021), our method is compatible with various encoders. In our experiments, we utilize RDN (Zhang et al. 2018) as an encoder, removing the last upsampling layer to produce a feature map with the same resolution as the input image. The decoding function f_θ is implemented as a 5-layer MLP with ReLU activation and hidden dimensions of 256. To balance computational efficiency and performance, we set $L = 2$.

Algorithm & Pseudocode

The pseudocode presented in Algorithm 1 outlines the core process of our proposed Continuous Implicit Feature Compensation Network (CFCNet) for screen content image super-resolution. The algorithm takes a low-resolution image I^{LR} as inputs and outputs a high-resolution image I^{HR}. Below is a detailed explanation of each step:

1. **Encoding the Low-Resolution Image (Line 1):** The algorithm begins by encoding the low-resolution image I^{LR} into a latent code z. This step involves extracting relevant features from the input image using an encoder, which could be any standard feature extractor used in single image super-resolution (SISR) methods.
2. **Coordinate-based Feature Compensation Layer (Lines 5-6):** The algorithm iterates over each coordinate x_q in the target high-resolution grid. For each coordinate:
   - **Coordinate-Aware Feature Compensation Module(Line 5):** The Coordinate-Aware Feature Compensation Module (CAFCM) is applied to the latent code z and the current coordinate x_q. This module aims to compensate for feature discrepancies between the query coordinate and its neighboring coordinates, ensuring that the feature representation at x_q is accurate and informative.
   - **Dynamic Multi-Coordinate Feature Integration Module(Line 6):** The resulting compensated feature z_q is then passed through the Dynamic Multi-Coordinate Feature Integration Module (DMFIM). This module adaptively integrates features across different coordinates, allowing the network to capture the dynamics of interactions between neighboring points.
3. **Predicting the RGB Value (Line 8):** After obtaining the integrated feature f_q, the algorithm uses a decoding function f_θ, which is implemented as a multi-layer perceptron (MLP), to map the feature f_q at coordinate x_q to its corresponding RGB value. This step reconstructs the pixel value at the given coordinate in the high-resolution image.
4. **Output the High-Resolution Image (Line 11):** Finally, the algorithm assembles the predicted RGB values for all coordinates in the high-resolution grid, producing the high-resolution image I^{HR}.

Train Set: SCI1K(n = 800)			In-training-scale			Out-of-training-scale					
Test set	**Method**	**# Params.**	×2	×3	×4	×5	×6	×7	×8	×9	×10
SCI1K (n = 200)	Bicubic	-	28.81	25.15	23.18	22.02	21.23	20.72	20.26	19.96	19.67
	RDN (Zhang et al. 2018)	21.97M	38.45	33.59	29.81	-	-	-	-	-	-
	MetaSR (Hu et al. 2019)	22.42M	38.57	33.67	30.12	27.52	26.13	23.91	23.19	22.02	21.73
	LIIF (Chen, Liu, and Wang 2021)	22.32M	38.65	33.97	30.55	27.77	26.07	23.99	23.24	22.18	21.81
	ITSRN (Yang et al. 2021)	22.62M	38.74	34.32	30.82	28.15	26.07	24.36	23.12	22.36	21.77
	CiaoSR (Cao et al. 2023)	32.50M	39.05	34.35	30.80	28.19	26.11	24.39	23.15	22.38	21.81
	LMI (Fu et al. 2024)	22.10M	38.77	34.12	30.67	28.02	26.03	24.23	23.13	22.32	21.74
	LTE (Lee and Jin 2022)	22.53M	39.14	34.50	30.93	28.22	26.19	24.28	23.17	22.39	21.85
	BTC (Pak, Lee, and Jin 2023)	22.40M	39.17	34.58	31.10	28.33	26.31	24.47	23.38	22.48	21.89
	CFCNet (Ours)	24.00M	**39.19**	**34.63**	**31.12**	**28.46**	**26.37**	**24.52**	**23.42**	**22.51**	**21.92**
SCID (n = 40)	Bicubic	-	25.22	22.78	21.60	20.9	20.42	20.04	19.77	19.51	19.29
	RDN (Zhang et al. 2018)	21.97M	34.00	28.34	25.74	-	-	-	-	-	-
	MetaSR (Hu et al. 2019)	22.42M	33.84	29.08	25.76	23.62	22.38	21.59	21.07	20.71	20.41
	LIIF (Chen, Liu, and Wang 2021)	22.32M	34.24	29.10	25.89	23.77	22.53	21.73	21.21	20.84	20.54
	ITSRN (Yang et al. 2021)	22.62M	34.19	29.46	26.22	23.96	22.64	21.80	21.26	20.87	20.56
	CiaoSR (Cao et al. 2023)	32.50M	34.45	29.56	26.30	24.23	22.65	21.79	21.27	20.88	20.58
	LMI (Fu et al. 2024)	22.10M	34.20	29.41	26.15	23.89	22.58	21.76	21.24	20.85	20.55
	LTE (Lee and Jin 2022)	22.53M	34.49	29.60	26.34	24.06	22.67	21.81	21.28	20.90	20.59
	BTC (Pak, Lee, and Jin 2023)	22.40M	34.48	29.56	26.30	24.09	22.69	21.84	21.29	20.90	20.61
	CFCNet (Ours)	24.00M	**34.52**	**29.65**	**26.39**	**24.13**	**22.72**	**21.87**	**21.32**	**20.92**	**20.63**
SIQAD (n = 22)	Bicubic	-	22.89	20.66	19.70	19.18	18.79	18.46	18.20	17.94	17.68
	RDN (Zhang et al. 2018)	21.97M	33.53	26.89	23.38	-	-	-	-	-	-
	MetaSR (Hu et al. 2019)	22.42M	34.12	28.40	23.55	21.18	20.18	19.63	19.25	18.94	18.65
	LIIF (Chen, Liu, and Wang 2021)	22.32M	34.31	28.27	23.44	21.16	20.25	19.70	19.36	19.02	18.70
	ITSRN (Yang et al. 2021)	22.62M	34.68	29.07	24.03	21.44	20.38	19.77	19.40	19.09	18.79
	CiaoSR (Cao et al. 2023)	32.50M	35.02	29.28	24.16	21.50	20.38	19.76	19.41	19.19	18.80
	LMI (Fu et al. 2024)	22.10M	34.76	29.11	24.05	21.46	20.31	19.75	19.38	19.11	18.73
	LTE (Lee and Jin 2022)	22.53M	35.07	29.33	24.21	21.52	20.39	19.78	19.43	19.11	18.81
	BTC (Pak, Lee, and Jin 2023)	22.40M	34.91	29.36	24.25	21.57	20.43	19.82	19.45	19.11	18.84
	CFCNet (Ours)	24.00M	**35.11**	**29.38**	**24.27**	**21.59**	**20.45**	**19.84**	**19.46**	**19.13**	**18.88**

Table 1: Quantitative comparison on SCI1K test set, SCID, and SIQAD (PSNR (dB)) for **integer scales**. The best and second results are in **Bold** and underlined, respectively. RDN trains different models for each scale. MetaSR, LIIF, ITSRN, CiaoSR, LMI, LTE, BTC and CFCNet use one model for all scales, and the six models utilize RDN as an encoder.

Additional experimentss

Quantitative Results In our quantitative analysis, we've included the results of CiaoSR and LMI. The results show that our method consistently maintains optimal performance across all evaluated methods. While CiaoSR uses an attention-based approach to weight features, similar to our concept, our method employs feature similarity for weighting, which provides a more rational and effective approach. This distinction underscores the superiority of our method in accurately capturing and enhancing feature representation.

Visual Comparisons of Model Figures 1 and 2 present a qualitative comparison of our method against other state-of-the-art ASSR methods. We evaluate performance across both integer scales and non-integer scales, highlighting the robustness of our method under varying conditions. As demonstrated in Figure 1, our approach excels in reconstructing the intricate details of screen content images, surpassing other methods in preserving the sharpness and clarity of text and graphical elements. This superior performance is especially evident at higher scales. In Figure 2, the efficacy of our method at non-integer scales is showcased, where it consistently outperforms competitors in restoring fine details. Even when subjected to unconventional scaling factors, our method preserves the integrity of both text and graphics with remarkable precision.

Experiments on natural image

Datasets & Implementation Details

Datasets We utilize the DIV2K dataset (Agustsson and Timofte 2017) for training our network. This dataset contains 1,000 images at 2K resolution, along with their low-resolution counterparts generated using bicubic interpolation at downsampling scales of ×2, ×3, and ×4. For evaluation, we assess performance on the DIV2K validation set (Agustsson and Timofte 2017), as well as on standard benchmark datasets such as Set14 (Zeyde, Elad, and Protter 2012), B100 (Martin et al. 2001), and Urban100 (Huang, Singh, and Ahuja 2015), using peak signal-to-noise ratio (PSNR) as the evaluation metric.

Implementation Details. During training, we follow the experimental setup from previous works (Chen, Liu, and Wang 2021; Lee and Jin 2022). Initially, 48×48 patches are cropped from high-resolution (HR) images as inputs. For each batch, a

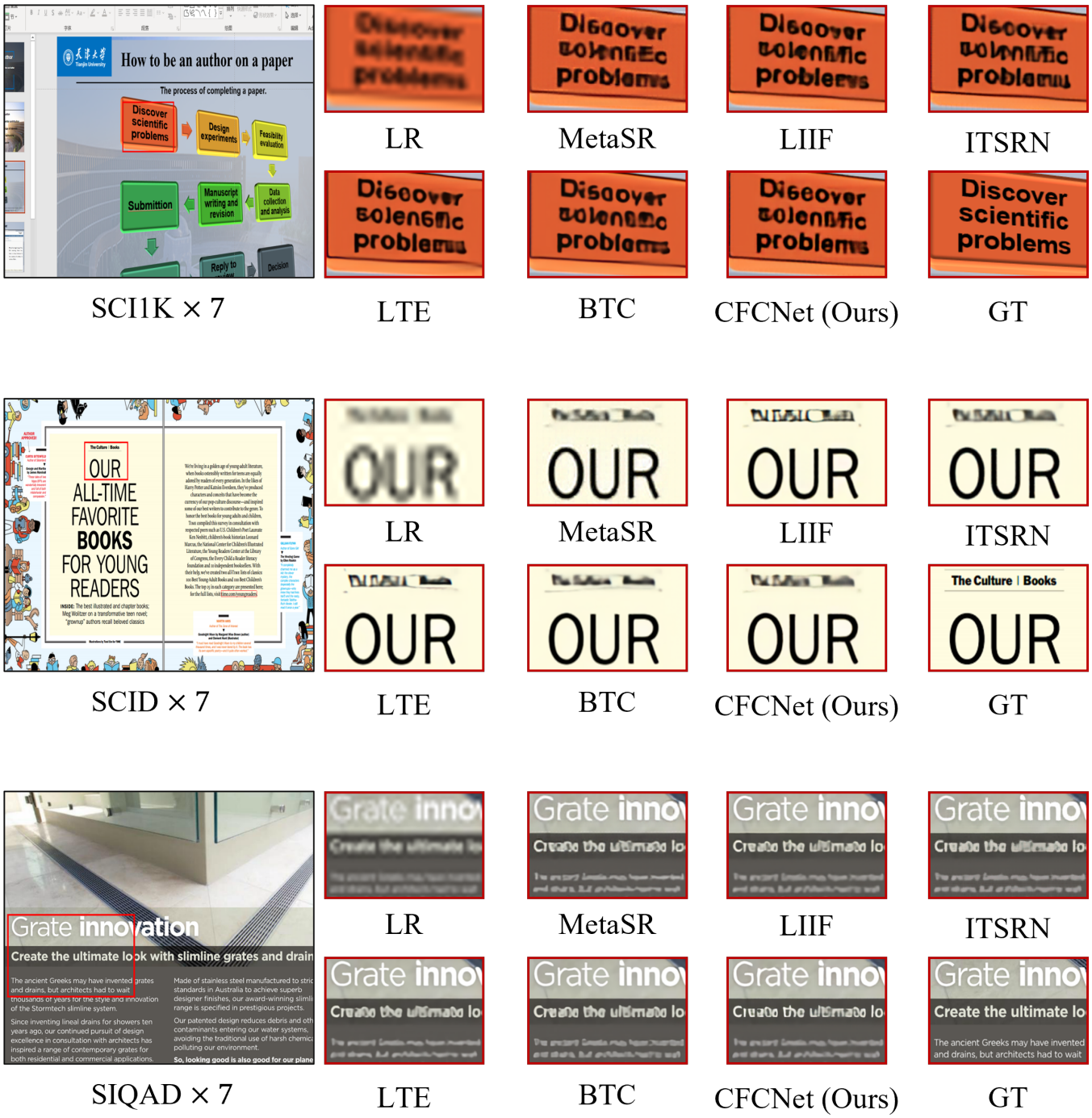

Figure 1: Visual comparison of different methods on SCI1K test set, SCID test set and SIQAD test set for **integer scales**. All of the ASSR methods use RDN as the encoder.

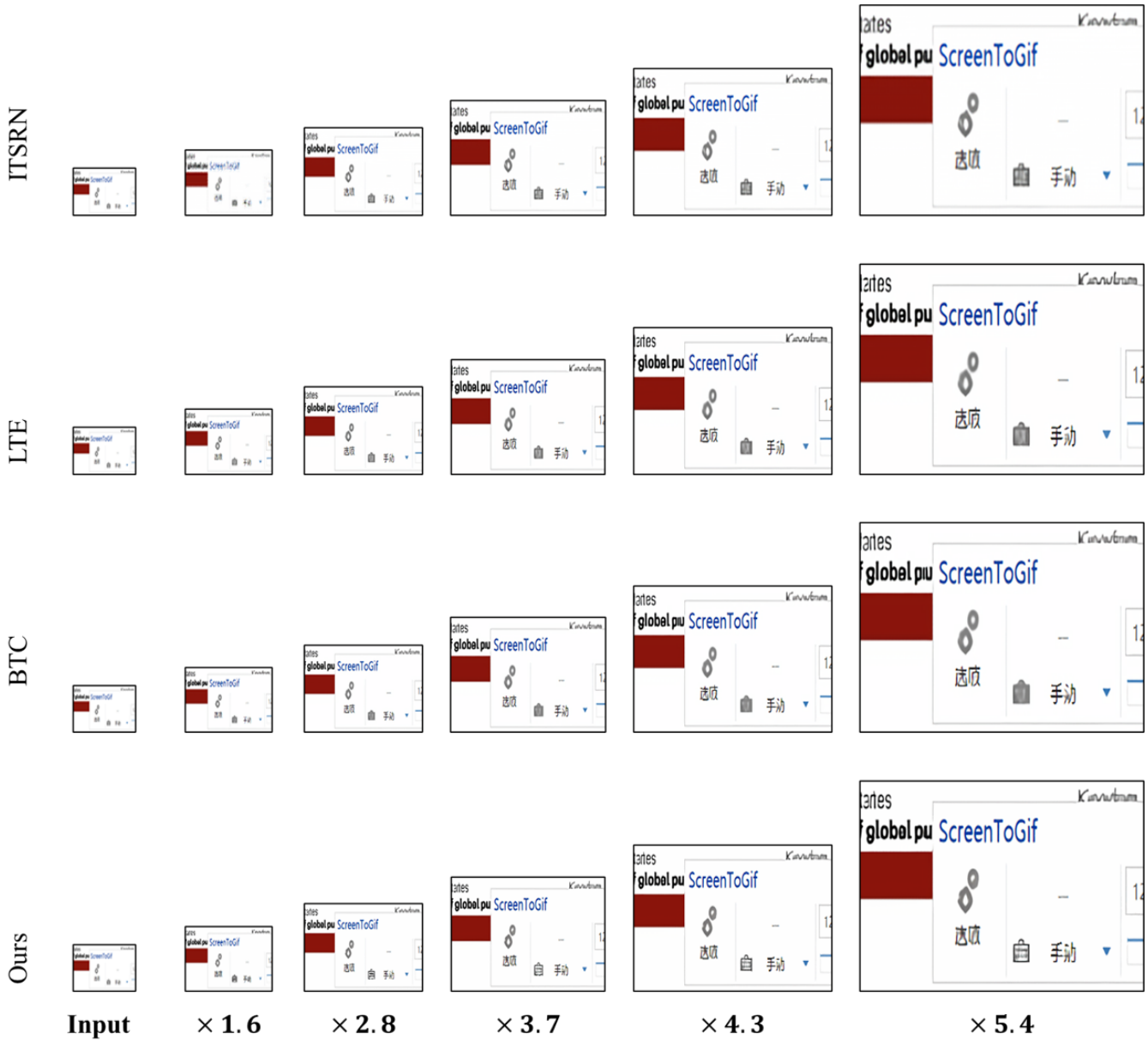

Figure 2: The qualitative results of ITSRN (Yang et al. 2021), LTE (Lee and Jin 2022), BTC (Pak, Lee, and Jin 2023) and Our proposed CFCNet with RDN as the decoder on SCI1K test set for non-integer scales.

Methods	×2	×3	×4	×6	×12	×18	×24	×30
Bicubic	31.01	28.22	26.66	24.82	22.27	21.00	20.19	19.59
EDSR-baseline	34.55	30.90	28.94	-	-	-	-	-
EDSR-baseline-MetaSR	34.64	30.93	28.92	26.61	23.55	22.03	21.06	20.37
EDSR-baseline-LIIF	34.67	30.96	29.00	26.75	23.71	22.17	21.18	20.48
EDSR-baseline-ITSRN	34.71	30.95	29.03	26.77	23.71	22.17	21.18	20.49
EDSR-baseline-LTE	34.72	31.02	29.04	26.81	23.78	22.23	21.24	20.53
EDSR-baseline-LIT	34.81	31.12	29.15	26.92	23.83	22.29	21.26	20.53
EDSR-baseline-Ours	**34.95**	**31.16**	**29.19**	**26.97**	**23.91**	**22.36**	**21.35**	**20.62**
RDN-baseline	34.94	31.22	29.19	-	-	-	-	-
RDN-MetaSR	35.00	31.27	29.25	26.88	23.73	22.18	21.17	20.47
RDN-LIIF	34.99	31.26	29.27	26.99	23.89	22.34	21.31	20.59
RDN-ITSRN	35.09	31.36	29.38	27.06	23.93	22.36	21.32	20.61
RDN-LTE	35.04	31.32	29.33	27.04	23.95	22.40	21.36	20.64
RDN-LIT	35.10	31.39	29.39	27.12	24.01	22.45	21.38	20.64
RDN-Ours	**35.18**	**31.46**	**29.46**	**27.20**	**24.05**	**22.51**	**21.46**	**20.73**
SwinIR-baseline	34.94	31.22	29.19	-	-	-	-	
SwinIR-MetaSR	35.15	31.40	29.33	26.94	23.80	22.26	21.26	20.54
SwinIR-LIIF	35.17	31.46	29.46	27.15	24.02	22.43	21.40	20.67
SwinIR-ITSRN	35.19	31.42	29.48	27.13	23.83	22.31	21.31	20.55
SwinIR-LTE	35.24	31.50	29.51	27.20	24.09	22.50	21.47	20.73
SwinIR-LIT	35.29	31.55	29.55	27.26	24.11	22.51	21.45	20.70
SwinIR-Ours	**35.38**	**31.60**	**29.59**	**27.32**	**24.19**	**22.57**	**21.58**	**20.78**

Table 2: Quantitative evaluation against state-of-the-art methods for **arbitrary-scale SR** on the DIV2K validation dataset (PSNR in dB). The **Bold** highlights the best value, while the underlined indicates the second-best result.

scaling factor r is randomly sampled from a uniform distribution $r \sim U(1, 4)$. The HR patches are then resized to $48r \times 48r$, while the corresponding low-resolution patches remain at 48×48 pixels. Data augmentation is applied to HR patches, including random horizontal flips, vertical flips, and 90° rotations. We then randomly sample coordinate-RGB pairs from each HR patch for training. The network is trained using the L1 loss function. We apply Adam (Kingma and Ba 2014) as the optimizer. For the encoder, we employ the existing SR models (e.g., EDSR (Lim et al. 2017), RDN (Zhang et al. 2018) and SwinIR (Liang et al. 2021)) as encoders with their upsampling modules removed. The encoder are trained for 1000 epochs with a batch size of 16, starting with a learning rate of 1e-4, which is halved every 200 epochs. Finally, we evaluate the network's generalization ability across multiple scales.

Results Figures 3 presents a qualitative comparison of our method against other state-of-the-art ASSR methods on natural images. We evaluate performance on B100 dataset, Urban100 dataset and Set14 dataset, highlighting the robustness of our method under varying conditions.

Figure 3: Qualitative comparison to other **arbitrary-scale SR** on natural images. RDN is used as an encoder for all methods.

Methods	Set5					Set14					B100					Urban100				
	×2	×3	×4	×6	×8	×2	×3	×4	×6	×8	×2	×3	×4	×6	×8	×2	×3	×4	×6	×8
RDN	38.24	34.71	32.47	-	-	34.01	30.57	28.81	-	-	32.34	29.26	27.72	-	-	32.89	28.80	26.61	-	-
RDN-MetaSR	38.22	34.63	32.38	29.04	26.96	33.98	30.54	28.78	26.51	24.97	32.33	29.26	27.71	25.90	24.83	32.92	28.82	26.55	23.99	22.59
RDN-LIIF	38.17	34.68	32.50	29.15	27.14	33.97	30.53	28.80	26.64	25.15	32.32	29.26	27.74	25.98	24.91	32.87	28.82	26.68	24.20	22.79
RDN-ITSRN	38.23	34.76	32.55	29.32	27.25	34.19	30.59	28.88	26.68	25.17	32.38	29.32	27.79	26.01	24.93	33.07	28.96	26.77	24.23	22.81
RDN-LTE	38.23	34.72	32.61	29.32	27.26	34.09	30.58	28.88	26.71	25.16	32.36	29.30	27.77	26.01	24.95	33.04	28.97	26.81	24.28	22.88
RDN-LIT	38.26	34.79	32.69	29.54	27.34	34.09	30.69	28.93	26.83	25.36	32.39	29.33	27.80	26.07	25.00	33.14	29.05	26.93	24.44	23.04
RDN-Ours	**38.30**	**34.83**	**32.72**	**29.53**	**27.36**	**34.15**	**30.74**	**28.97**	**26.85**	**25.40**	**32.39**	**29.34**	**27.81**	**26.08**	**25.02**	**33.21**	**29.14**	**27.01**	**24.49**	**23.07**
SwinIR	38.35	34.89	32.72	-	-	34.14	30.77	28.94	-	-	32.44	29.37	27.83	-	-	33.40	29.29	27.07	-	-
SwinIR-MetaSR	38.26	34.77	32.47	29.09	27.02	34.14	30.66	28.85	26.58	25.09	32.39	29.31	27.75	25.94	24.86	33.29	29.12	26.76	24.16	22.75
SwinIR-LIIF	38.28	34.87	32.73	29.46	27.36	34.14	30.75	28.98	26.82	25.34	32.39	29.34	27.84	26.07	25.01	33.36	29.33	27.15	24.59	23.14
SwinIR-ITSRN	38.22	34.75	32.63	29.31	27.24	34.26	30.75	28.97	26.71	25.32	32.42	29.38	27.85	26.05	24.96	33.46	29.34	27.12	24.50	23.06
SwinIR-LTE	38.33	34.89	32.81	29.50	27.35	34.25	30.80	29.06	26.86	25.42	32.44	29.39	27.86	26.09	25.03	33.50	29.41	27.24	24.62	23.17
SwinIR-LIT	38.41	34.97	32.86	29.69	27.62	34.27	30.85	29.08	26.94	25.55	32.46	29.42	27.91	26.15	25.09	33.56	29.43	27.25	24.77	23.33
SwinIR-Ours	**38.48**	**35.01**	**32.92**	**29.71**	**27.65**	**34.33**	**30.92**	**29.13**	**26.98**	**25.62**	**32.49**	**29.48**	**27.95**	**26.18**	**25.13**	**33.65**	**29.52**	**27.42**	**24.84**	**23.43**

Table 3: Quantitative comparison with state-of-the-art methods for **arbitrary-scale SR** on various benchmark datasets (PSNR in dB). The **Bold** markers denote the best values, while the underlined markers indicate the second-highest values.

Discussion

In this paper, we presented a novel Coordinate-based continuous implicit Feature Compensation Network (CFCNet) designed specifically for the super-resolution of screen content images (SCIs). Our approach addresses the limitations of existing methods that fail to fully capture the dynamic interactions between neighboring coordinates, which are critical in accurately reconstructing SCIs. By introducing the Coordinate-Aware Feature Compensation Module (CAFCM) and the Dynamic Multi-Coordinate Feature Integration Module (DMFIM), our model effectively bridges feature gaps and adaptively learns cross-coordinate feature similarities, leading to superior performance in SCI super-resolution.

Our extensive experiments on the SCI1K, SCID, and SIQAD datasets demonstrate that CFCNet consistently outperforms existing methods, including CiaoSR and LMI. Notably, while CiaoSR employs an attention-based mechanism to weight point features, our approach uses feature similarity for weighting, which we found to be more effective in capturing the intricate details of SCIs. This is particularly evident in our results, where CFCNet excels at reconstructing fine details such as text and graphics, even at challenging fractional scales.

Furthermore, the ablation studies conducted on the SCI1K dataset confirm the significance of both CAFCM and DMFIM in enhancing the network's performance. Our findings show that the integration of these modules not only improves the quality of the super-resolved images but also provides a balanced trade-off between computational efficiency and accuracy. The experiments reveal that while increasing the number of Coordinate-based Feature Compensation Layers gradually enhances performance, it also adds computational overhead. We determined that using L=2 offers an optimal balance.

However, we acknowledge that our method introduces additional learnable parameters, resulting in a 7% increase in the overall parameter count compared to the baseline version. Despite this increase, our approach achieves optimal performance, demonstrating that the additional complexity is justified by the gains in image quality.

Overall, our proposed method advances the state of the art in SCI super-resolution by offering a more principled approach to feature weighting and integration. The consistent superiority of CFCNet across various datasets and magnification scales underscores its robustness and generalizability. Future work may explore further optimization of the network architecture to reduce parameter overhead while maintaining or even enhancing performance. Additionally, extending this framework to other domains, such as natural images, could provide valuable insights into the broader applicability of our method.

Code for Reproducibility

In order to illustrate the reproducibility, we provide the implementation details of CFCNet in https://anonymous.4open.science/r/CFCNet.[1]

Code — https://anonymous.4open.science/r/CFCNet

As stated in the manuscript, CFCNet consists of three main components: *an encoder*, L *coordinate-based feature compensation layers*, and *a decoder* (f_θ).

First, the encoder processes the low-resolution screen image I^{LR} into a latent code z. This latent code z is then passed through L coordinate-based feature compensation layers, which along with the 2D coordinates, generates an optimized feature map for any arbitrary query coordinate x_q. Finally, the RGB values are predicted from the generated feature map, resulting in the final generated HR image I^{HR}.

```
        z_direction = [(inpx[:,:,:-4]).cuda() for inpx in inp_list]
        dire_feats = []

        bs, q = coord.shape[:2]

        fusion_feature = torch.stack([self.weights[i] * f[:, :, :-4] for i, f in
            enumerate(inp_list)], dim=0).sum(dim=0) # + self.bias

        z_f = fusion_feature.contiguous().view(bs * q, -1) # torch.stack([f[:, :, :-4] for i,
            f in enumerate(inp_list)], dim=0).mean(dim=0).view(bs * q, -1)
        for i in range(self.num_branches):

            for j in range(len(inp_list)):
                z_direction[j] = self.block(z_direction[j].contiguous().view(bs * q, -1),
                    inp_list[j].contiguous().view(bs * q, -1))
                dire_feats.append(z_direction[j])

            # pdb.set_trace()
            ext_f = []
            for i, dire_feat in enumerate(dire_feats):
                alp_f = torch.mul(z_f, self.gate(dire_feat + z_f)) # bs * q, 576
                ext_f.append(alp_f)

            branch_out = self.fuse(torch.stack(ext_f, dim=0).sum(dim=0)) +
                fusion_feature.contiguous().view(bs * q, -1)

            z_f = self.gn3(F.relu(branch_out))

        ret = self.imnet(z_f).view(bs, q, -1)
```

[1] Not all codes are provided here, only some related codes are shown.

References

Agustsson, E.; and Timofte, R. 2017. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1122–1131. IEEE Computer Society.

Cao, J.; Wang, Q.; Xian, Y.; Li, Y.; Ni, B.; Pi, Z.; Zhang, K.; Zhang, Y.; Timofte, R.; and Van Gool, L. 2023. Ciaosr: Continuous implicit attention-in-attention network for arbitrary-scale image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1796–1807.

Chen, Y.; Liu, S.; and Wang, X. 2021. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8628–8638.

Fu, H.; Peng, F.; Li, X.; Li, Y.; Wang, X.; and Ma, H. 2024. Continuous Optical Zooming: A Benchmark for Arbitrary-Scale Image Super-Resolution in Real World. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3035–3044.

Hu, X.; Mu, H.; Zhang, X.; Wang, Z.; Tan, T.; and Sun, J. 2019. Meta-SR: A magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1575–1584.

Huang, J.-B.; Singh, A.; and Ahuja, N. 2015. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5197–5206.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lee, J.; and Jin, K. H. 2022. Local texture estimator for implicit representation function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1929–1938.

Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; and Timofte, R. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1833–1844.

Lim, B.; Son, S.; Kim, H.; Nah, S.; and Mu Lee, K. 2017. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 136–144.

Martin, D.; Fowlkes, C.; Tal, D.; and Malik, J. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, 416–423. IEEE.

Pak, B.; Lee, J.; and Jin, K. H. 2023. B-spline texture coefficients estimator for screen content image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10062–10071.

Yang, J.; Shen, S.; Yue, H.; and Li, K. 2021. Implicit transformer network for screen content image continuous super-resolution. *Advances in Neural Information Processing Systems*, 34: 13304–13315.

Zeyde, R.; Elad, M.; and Protter, M. 2012. On single image scale-up using sparse-representations. In *Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7*, 711–730. Springer.

Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; and Fu, Y. 2018. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2472–2481.