Local Positional Encoding for Multi-Layer Perceptrons: Supplemental Document

Shin Fujieda
Advanced Micro Devices, Inc.

Atsushi Yoshimura
Advanced Micro Devices, Inc.

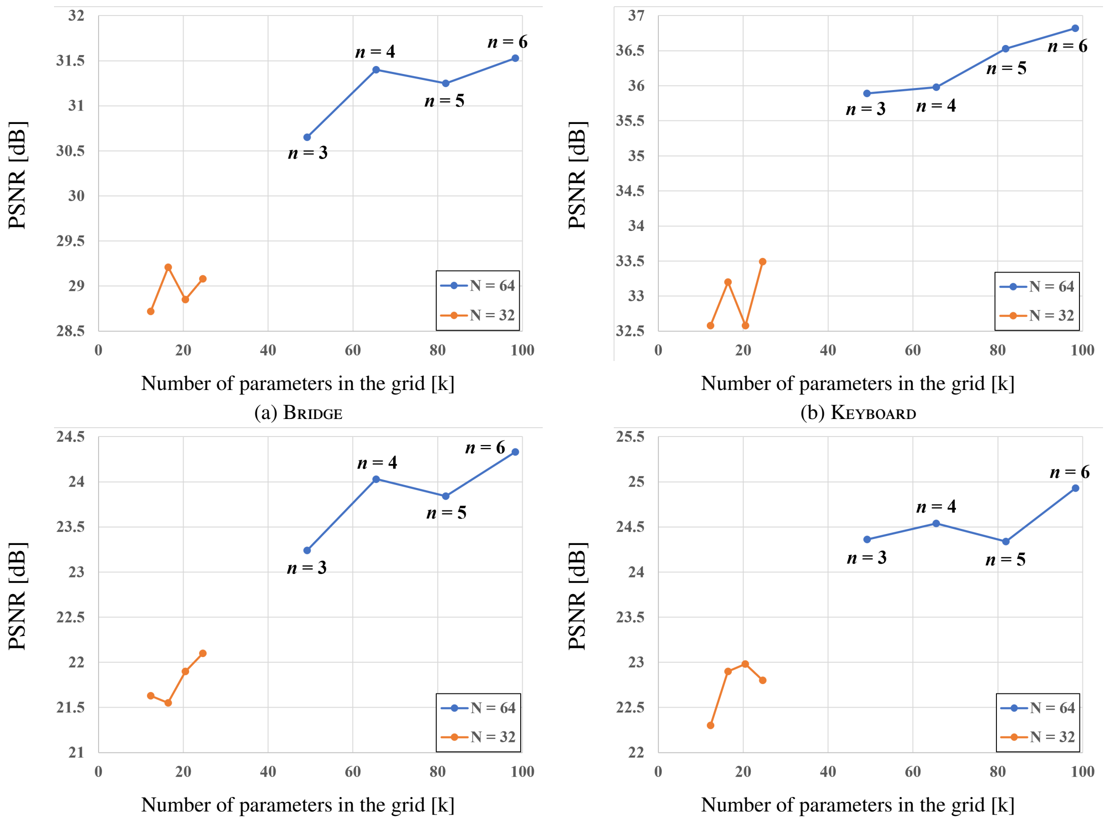
Takahiro Harada
Advanced Micro Devices, Inc.

1 Various Conditions for Local Positional Encoding

Local positional encoding has two hyperparameters, the grid resolution N and the number of frequencies n. We analyze how they impact the performance of our proposed method in the image reconstruction task and the task of representing signed distance functions (SDFs). In both tasks, we use a small MLP with three hidden layers with 64 neurons, each followed by a leaky ReLU activation function with $\alpha = 0.01$. Also, we apply a sigmoid activation function to the output layer for the image reconstruction task.

1.1 Image Reconstruction

We evaluate the performance of the 2D image reconstruction under various conditions of N and n for local positional encoding in Fig. 1. The memory footprint is linear to N and n, and the dimension of the input layer of the MLP is also linear to n. In this analysis, we found $(N = 64, n = 4)$ often shows better quality results. As illustrated in Fig. 2, we can see the rendered image of Fish Market with $n = 4$ captures high-frequency details well, especially compared with those with $n = 3$ and $n = 5$. The image with $n = 6$ also shows a better result, but it consumes 1.5× the amount of memory than that with $n = 4$. As a result, $(N = 64, n = 4)$ achieves well-balanced results between the quality and the memory consumption, so we use these values as the default configuration in the experiments in the main document.



(a) Bridge (b) Keyboard (c) Fish Market (d) Trees

Figure 1: *Performance comparisons with 4 images for varying the grid resolution N and the number of frequencies n for local positional encoding, which determine the number of trainable parameters in the grid. The MLP has three hidden layers with 64 neurons each, which is trained for 1k iterations.*

PSNR: 23.24 dB $n = 3$ | PSNR: 24.03 dB $n = 4$ | PSNR: 23.84 dB $n = 5$ | PSNR: 24.33 dB $n = 6$

Figure 2: *Qualitative comparison of various n-values with $N = 64$ for the* FISH MARKET *image.*

2 dimensions | 4 dimensions | 8 dimensions | 16 dimensions

Figure 3: *Comparison of closed-up* FISH MARKET *images reconstructed using grid encoding with different feature vector sizes.*

Local positional encoding with $n = 4$ results in a 16-dimenstional input vector to the MLP. However, some other approaches [KMX*21, MESK22, WZK*23] use smaller dimensions of latent vectors stored in the grid (e.g. 2, 4, 8-dimensional vectors). Thus, we also analyze the impact of dimensions of latent vectors in grid encoding. As shown in Fig. 3, smaller dimensions of latent vectors fail to represent the results of a sufficiently expressive quality. In our paper, we evaluate our proposed method against a superior configuration of grid encoding (i.e. 16-dimensional vectors) for a fair comparison.

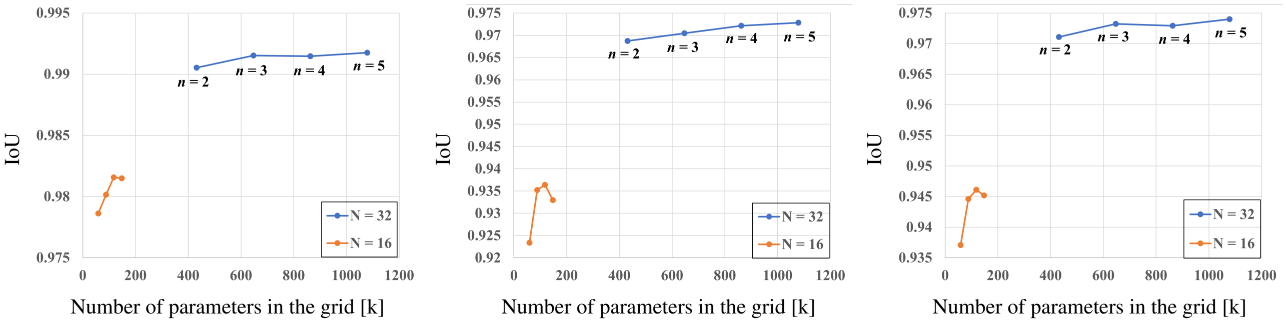
1.2 Signed Distance Functions

We evaluate the performance of local positional encoding for the 3D SDFs under various conditions of N and n in Fig. 4. In this analysis, we found $(N = 32, n = 3)$ often shows better quality results. As illustrated in Fig. 5, we can see the rendered image of the THAI STATUE model with $n = 3$ captures enough high-frequency details with the smaller size of memory, even compared with those with $n = 4$ and $n = 5$. The result with $n = 5$ achieves a higher IoU, but it also uses about 1.7× larger size of memory than that with $n = 3$. Therefore, $(N = 32, n = 3)$ is a well-balanced parameter, so we use it as the default configuration for our encoding in the experiments in the main document.

2 Additional Results for Signed Distance Functions

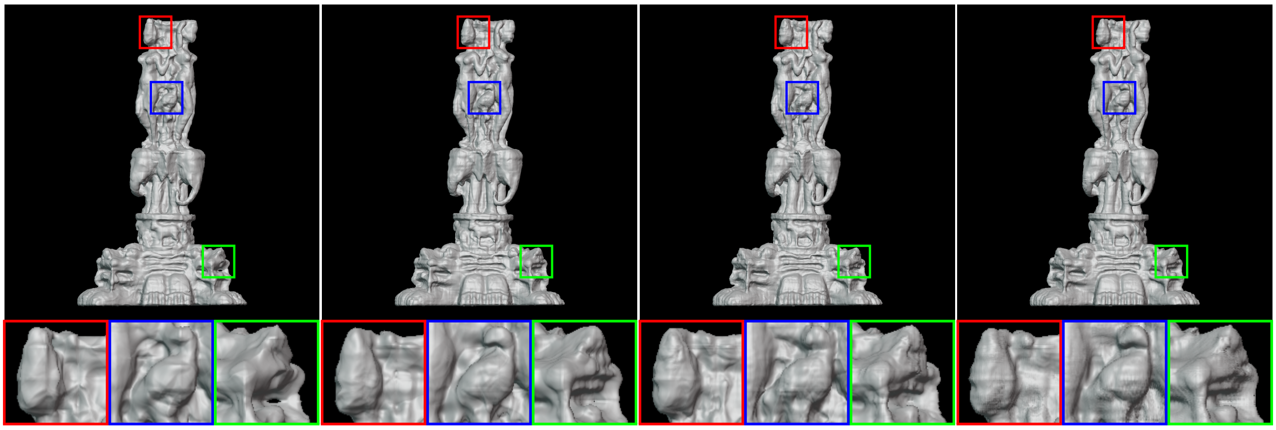
In the main document, we illustrate the reconstructed SDFs geometries only for the THAI STATUE model. We show the additional results for ARMADILLO, LUCY models in this section.

We first show the additional results for the comparison of local positional encoding with positional and grid encodings. As described in Sec. 1.2, we use the parameters of $(N = 32, n = 3)$ for local positional encoding, $n = 3$ for positional encoding, and $N = 32$ with an 18-dimensional latent vector for grid encoding for a fair comparison concerning the number of trainable parameters in the grid. Every encoding results in an 18-dimensional vector as an input for the MLP. Fig. 6 shows rendered images with different encodings for three models along with the intersection-over-union (IoU) metric. Similar to the THAI STATUE model which we mention in the main document, close-up views of our method for ARMADILLO and LUCY models (Fig. 6c) visually show higher-frequency details than those



(a) Armadillo (b) Lucy (c) Thai Statue

Figure 4: *Performance comparisons for varying the grid resolution N and the number of frequencies n for local positional encoding, which determine the number of trainable parameters in the grid. The MLP has three hidden layers with* 64 *neurons each, which is trained for* 16*k iterations.*



IoU: 0.9711	IoU: 0.9732	IoU: 0.9729	IoU: 0.9740
$n = 2$	$n = 3$	$n = 4$	$n = 5$

Figure 5: *Qualitative comparison of various n-values with* $N = 32$ *for the* Thai Statue *model.*

of grid encoding (Fig. 6b). However, though our encoding produced a slightly higher value for the Armadillo model, grid encoding achieved better results for the other models using the IoU metric.

We also show the additional rendered images for the comparison of our encoding with multi-resolution grids in Fig. 7. For local positional encoding, we use the same configuration as the default configuration as described in Sec. 1.2 ($N = 32, n = 3$). And to make memory consumption equivalent, we choose 3-level grids ($N = 16, 32, 64$) with 2-dimensional latent vectors as multi-resolution grid encoding, and 4-level grids ($N = 16, 32, 64, 128$) with 2-dimensional latent vectors using 2^{17} hash table size for multi-resolution hash encoding [MESK22]. These settings result in roughly 590k parameters for all the encodings in the grids. As shown in Fig. 7, though local positional encoding uses only a single-level grid, it achieved higher IoUs and reconstructed higher-frequency details than multi-resolution grid encoding for all three models. On the other hand, multi-resolution hash encoding showed the highest IoUs for all the models while our encoding achieved visually comparable results even with a small single-level grid. Also, taking a closer look at close-up views in Fig. 7b, it produces micro-structured artifacts on the smooth surface due to hash collisions, which cannot be seen with our encoding. Therefore, higher-resolution grids accepting hash collisions can capture high-frequency details well with artifacts due to hash collisions, which is the limitation of multi-resolution hash encoding. Adopting multi-resolution grids in local positional encoding which we expect improves the performance further is our interesting future work. In summary, local positional encoding can achieve better results than multi-resolution grid encoding and comparable results to multi-resolution hash encoding only using a small single-level grid with equivalent memory consumption.

Acknowledgments

We thank colleagues in Advanced Rendering Research (ARR) Group for discussion, neural network implementation, proof reading. We would also like to thank Sofía Rabassa for the Fish Market image, and the Stanford Computer Graphics Laboratory for Armadillo, Lucy, and Thai Statue models.

(a) Positional Encoding (b) Grid Encoding (c) Local Positional Encoding (d) Reference

Figure 6: *Qualitative and quantitative comparison with positional encoding, grid encoding, and local positional encoding with SDFs geometry rendering. They all use a* $32\times32\times32$ *grid with* 18*-dimensional latent vectors and* 3 *frequencies in positional encoding. Each close-up view enclosed by red and blue squares represents how the fine details are captured by the encodings. IoU metrics are shown for each geometry and encoding as well. The bold numbers represent the best in the encodings for each geometry.*

References

[KMX*21] Kuznetsov A., Mullia K., Xu Z., Hašan M., Ramamoorthi R.: Neumip: Multi-resolution neural materials. *ACM Trans. Graph. 40*, 4 (jul 2021). URL: https://doi.org/10.1145/3450626.3459795, doi:10.1145/3450626.3459795.

[MESK22] Müller T., Evans A., Schied C., Keller A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph. 41*, 4 (jul 2022). URL: https://doi.org/10.1145/3528223.3530127, doi:10.1145/3528223.3530127.

[WZK*23] Weier P., Zirr T., Kaplanyan A., Yan L.-Q., Slusallek P.: Neural prefiltering for correlation-aware levels of detail. In *Proceedings of SIGGRAPH 2023* (2023).

IoU = 0.9904 | **IoU = 0.9941** | IoU = 0.9920

IoU = 0.9702 | **IoU = 0.9856** | IoU = 0.9723

IoU = 0.9712 | **IoU = 0.9848** | IoU = 0.9736

(a) Multi-resolution Grid Encoding (b) Multi-resolution Hash Encoding (c) Local Positional Encoding (d) Reference

Figure 7: *Qualitative and quantitative comparison with multi-resolution grid encoding, multi-resolution hash encoding [MESK22], and local positional encoding with SDFs geometry rendering. All the encodings are configured for approximately the same number of latent vectors in grids for a fair comparison. The bold number represents the best in the encoding for the geometry.*