Expansive Supervision for Neural Radiance Fields

Weixiang Zhang[1], Wei Yao[1], Shuzhao Xie[1], Shijia Ge[1], Chen Tang[2], Zhi Wang[1*]
[1]Tsinghua University [2]Chinese University of Hong Kong

***Abstract*—Neural Radiance Field (NeRF) has achieved remarkable success in creating immersive media representations through its exceptional reconstruction capabilities. However, the computational demands of dense forward passes and volume rendering during training continue to challenge its real-world applications. In this paper, we introduce Expansive Supervision to reduce time and memory costs during NeRF training from the perspective of partial ray selection for supervision. Specifically, we observe that training errors exhibit a long-tail distribution correlated with image content. Based on this observation, our method selectively renders a small but crucial subset of pixels and expands their values to estimate errors across the entire area for each iteration. Compared to conventional supervision, our approach effectively bypasses redundant rendering processes, resulting in substantial reductions in both time and memory consumption. Experimental results demonstrate that integrating Expansive Supervision within existing state-of-the-art acceleration frameworks achieves 52% memory savings and 16% time savings while maintaining comparable visual quality.**

***Index Terms*—neural radiance fields, novel view synthesis, sparse supervision, acceleration**

I. Introduction

Neural Radiance Field (NeRF) [1] has emerged as a promising approach for representing 3D media content in the field of photorealistic novel view synthesis. NeRF employs a neural network $F(\Theta)$ to implicitly encode scenes by mapping position $\mathbf{x} = (x, y, z)$ and direction $\mathbf{d} = (\theta, \varphi)$ to view-dependent color $\mathbf{c} = (r, g, b)$ and view-independent volumetric density τ, i.e., $F(\Theta) : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$, where Θ is parameters of the neural network. Through its powerful implicit neural scene representation, NeRF synthesizes and infers target pixels via volume rendering using sampled query pairs (color $\mathbf{c}$ and density σ) along rays, significantly surpassing traditional multi-view stereo methods in view synthesis and 3D construction.

Despite the impressive performance of NeRF, its training speed remains a significant limitation. In the original NeRF architecture, rendering each pixel requires sampling N points to compute the color $\mathbf{c}$ and density σ. For a scene with dimensions (h, w), this process necessitates $h \cdot w \cdot N$ neural network forward passes, potentially exceeding 10^8 computations for rendering a single 1080p resolution view. This computational burden substantially prolongs the training duration and impedes the practical generation of neural 3D media content. To address these computational challenges, various acceleration methods have been proposed, including decomposition schemes [2], distillation to light fields [3], and incorporation of explicit representations [4], [5]. Among these, the most effective approaches leverage explicit representations to cache

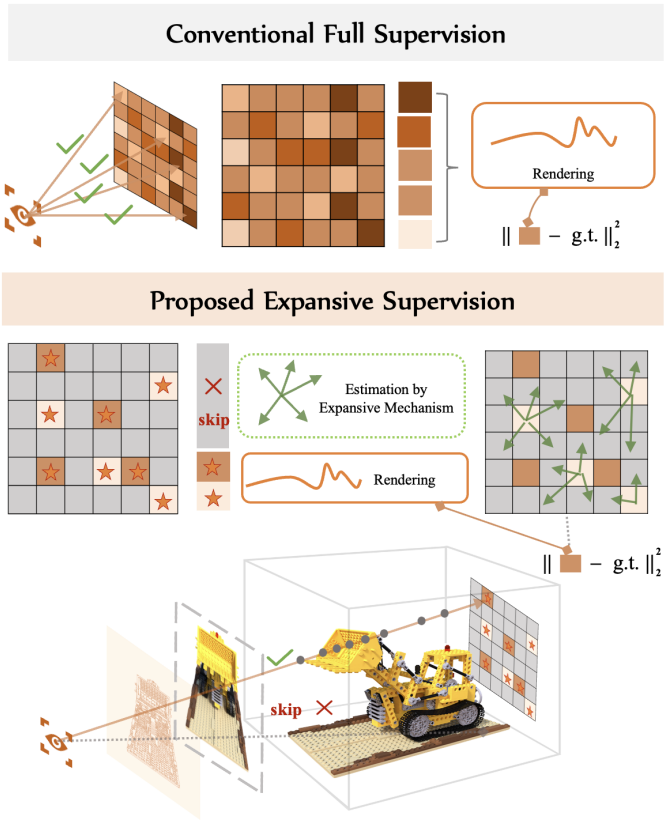

Fig. 1. **Overview of proposed method.** Our approach adopts an expansive supervision technique to selectively render a subset of crucial pixels to estimate the error by expansive mechanism. Unlike conventional full supervision, which blindly renders all pixels, our method intelligently avoids redundant rendering processes, leading to significant reductions in training time and memory consumption.

view-independent features, trading memory consumption for reduced training time. These efficient explicit representations include voxel grids [6], [7], point cloud [8], hash tables [5], and low-rank tensors [4]. By incorporating such explicit structures, training time for a scene can be reduced significantly, from approximately 10 hours to 30 minutes.

Differing from previous methods, we propose Expansive Supervision, an acceleration method for NeRF training based on ray selection and partial supervision, which can be integrated with existing acceleration frameworks to further reduce time and memory consumption. Our approach is motivated by the observation that training errors exhibit a long-tail distribution strongly correlated with image content: regions with higher frequencies display larger errors, while smoother areas show smaller errors. This observation leads us to selectively render

*Corresponding author.

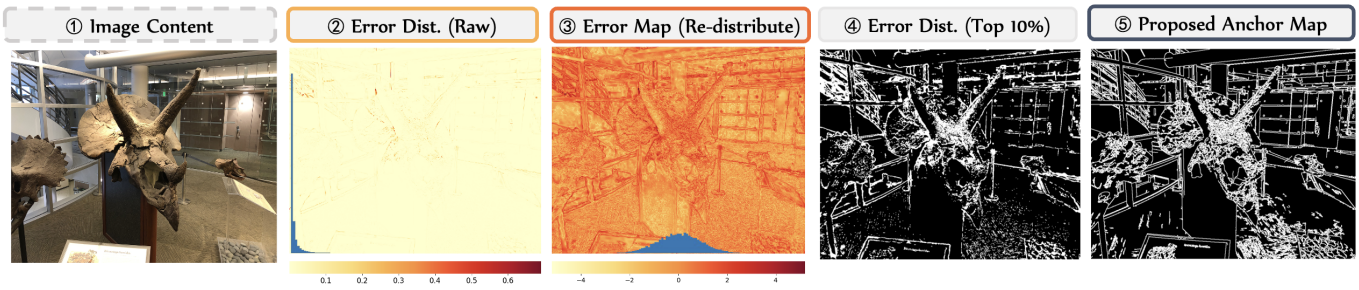

Fig. 2. **Motivational observation.** (#2) The blue histogram shows the error distribution after 1000 iterations, revealing a pronounced long-tail characteristic. (#3) To enhance the visibility of error map, we transformed the data into a normal distribution, revealing correlations between redistributed errors and image content. (#4) The top 10% of errors identified during training are visualized, corresponding to regions with high-frequency details in the image content. (#5) The top 10% error map generated by our expansive supervision exhibits a high correlation with the actual error distribution.

a small but crucial subset of pixels $R' \subset R$ guided by image content prior during training, and expand the error of these precisely rendered pixels to estimate the loss for the entire area in each iteration. Specifically, the selected pixel set R' comprises two areas: the anchor area A and the source area S. The anchor area is determined by a meticulously designed frequency prior extractor to capture prominent error patterns, while the source area is sampled to expand its values to the remaining regions to maintain reconstruction quality. The final error estimate $\hat{L}$ is synthesized from the selected pixels $R' = \{A \cup S\}$ through the expansive mechanism, and the model is updated via $\Theta := \Theta - \eta \nabla \hat{L}(R')$, where η denotes the learning rate. By avoiding costly yet marginal renderings, our method theoretically achieves computational savings proportion to $(1 - \frac{|R'|}{|R|})$. Experimental results demonstrate that by rendering only 50% of pixels to supervise the entire model, we achieve 0.52× memory and 0.16× time savings.

Our method shares similar mechanisms with recent sampling-based neural field optimization approaches, such as EGRA [9] and Soft Mining [10], which resample rays in each batch based on image edge areas and Langevin Monte Carlo (LMC) sampling, respectively. While these methods achieve better reconstruction quality than uniform sampling with the same number of iterations, our method distinctively reduces both memory consumption and training time simultaneously. Furthermore, under identical conditions, our method demonstrates superior reconstruction quality compared to these sampling-based methods.

Extensive experiments validate the effectiveness of our proposed method. Compared to conventional full supervision, our approach achieves substantial reductions in both time and memory usage while maintaining comparable rendering quality. Notably, our method seamlessly integrates with existing explicit caching acceleration frameworks without requiring custom modifications. Our main contributions are summarized as follows: **(1)** We observe a strong correlation between error distribution and image content, motivating the development of partial supervision methods to accelerate NeRF training within existing acceleration frameworks. **(2)** We propose Expansive Supervision, which selectively renders a small yet crucial subset of pixels for acceleration. This method significantly reduces training time and memory consumption by eliminating redundant rendering operations. **(3)** We present comprehensive experimental results validating the effectiveness of our method and detailed analysis of quality-cost trade-offs.

II. Methods

A. Preliminary and Formulation

Neural Radiance Field learns a function $F(\Theta) : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$ using a multilayer perceptron (MLP), where $\mathbf{x} \in \mathbb{R}^3, \mathbf{d} \in \mathbb{R}^2$ represent the position and view direction of a point, while $\mathbf{c} \in \mathbb{R}^3$ and $\sigma \in \mathbb{R}$ represent the emitted color and density, respectively. Volume rendering allows computing the expected color $\mathbf{C}(\mathbf{r})$ in a novel view as:

$$\mathbf{C}(\mathbf{r}) = \int_0^t \mathcal{T}(t; \mathbf{r}) \cdot \tau(\mathbf{r}(t)) \cdot \mathbf{c}(\mathbf{r}(t), \mathbf{d}) dt, \tag{1}$$

where $\mathcal{T}(t; \mathbf{r}) = \exp\left(-\int_0^t \tau(\mathbf{r}(s)) ds\right)$ represents the accumulated transmittance along the ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$.

In practice, numerical estimation of the rendering integral involves sampling N points from partitioned bins along the ray, allowing the estimation of $\mathbf{C}(\mathbf{r})$ as:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^{N} \mathcal{T}_i \left(1 - \exp\left(-\tau_i \delta_i\right)\right) \mathbf{c}_i, \tag{2}$$

where $\mathcal{T}_i = \exp\left(-\sum_{j=1}^{i-1} \tau_j \delta_j\right)$, and δ_i represents the distance between adjacent sampled points.

The training of radiance fields is computationally intensive due to the large number of required neural network forward passes, which amounts to $N \times b$ for each iteration, where N and b denotes the number of sampled points along the ray and batch size. Our method aims to reduce the computational resources by selecting a subset of pixels $R' \in R$ to supervise the model, bypassing redundant renderings. We utilize image context $\mathcal{I}$ to estimate the total error $L(R)$ with the expansive mechanism based on estimated error $\hat{L}(R')$ from partial significant pixels R'. This mechanism allows for the conservation of both time and memory, resulting in a savings of $(1 - \frac{|R'|}{|R|})v\times$ computational resources, where $v \in (0, 1)$ represents the ratio

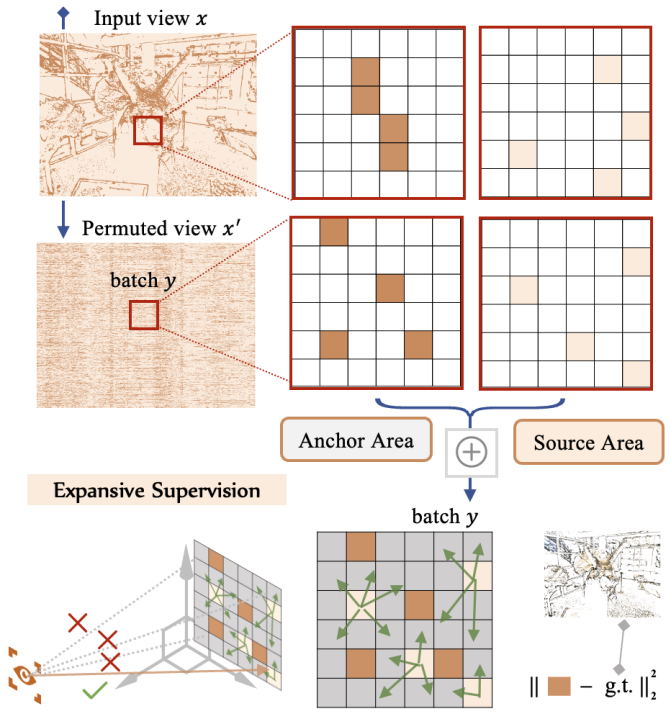

Fig. 3. **Pipeline of expansive supervision.** The mechanism of expansive supervision is to exclusively render the crucial pixels, which consist of the pre-computed anchor area and sampled source areas, to estimate the loss. This estimation is accomplished through the expansive strategy described in Section II-B. Note that the expansive strategy is implemented on permuted pixels, i.e., random batch training in NeRF. Our expansive supervision method results in significant time and memory savings while maintaining negligible compromise in visual quality.

of resources used by volume rendering in the total training process.

B. Expansive Supervision

The objective of *Expansive Supervision* is to utilize an important subset of supervision signals to guide radiance field training while maintaining rendering quality and achieving substantial time and memory savings. Our approach is motivated by the observation that error distribution strongly correlates with image content: regions with high-frequency details exhibit larger training errors and require more attention.

This observation is validated through a preliminary study, as illustrated in Fig. 2. During standard training, the error distribution exhibits a clear long-tail phenomenon, where approximately 99.4% of the data points with the smallest errors contribute only 10% of the overall loss. Moreover, higher error values predominantly concentrate in high-frequency image regions, particularly edges and texture-rich areas, as demonstrated in column 3 of Fig. 2. These findings suggest that the global error distribution can be effectively estimated from partial renderings, enabling expansive loss calculation.

The pipeline of Expansive Supervision is demonstrated in Fig. 3. The selected pixel set R' comprises two distinct areas within the input view I:

1) Anchor area $A \subset I$. The anchor area A represents regions within I that exhibit larger error patterns. It is computed using the anchor extractor function $\mathcal{F}A(\cdot)$, which is detailed in Sec. II-C. Formally, $A = \mathcal{F}A(I, \xi_a)$, where ξ_a controls the anchor area size such that $|A| = \xi_a|I|$.

2) Source area $S \subset I \backslash A$. The source area S consists of sampled points from the remaining regions after excluding the anchor set. It is formulated as $S \sim \mathcal{U}_{\xi_s|I|}(I \backslash A)$, where $\mathcal{U}(\cdot)$ denotes uniform sampling and ξ_s controls the source area size such that $|S| = \xi_s|I|$. The source area is regenerated for each iteration, and its errors are expanded to the remaining regions to achieve comprehensive loss estimation.

We define B_t^* as the batch data after random permutation at iteration t, with A_t^* and S_t^* representing its corresponding anchor and source areas. The global estimated error $\hat{L}_t$ at iteration t can be represented as:

$$\hat{L}_t = \underbrace{\sum_{r_a \in A_i^*} ||\hat{C}(r_a) - C(r_a)||_2^2}_{\text{anchor supervision}} + \underbrace{[\gamma_{s/a} + \frac{t}{T}(1 - \gamma_{s/a})]}_{\text{expansive mechanism}} \underbrace{\sum_{r_s \in S_i^*} ||\hat{C}(r_s) - C(r_s)||_2^2}_{\text{source supervision}}, \tag{3}$$

where $C(\cdot)$ and $\hat{C}(\cdot)$ denote ground truth and predicted RGB colors from given rays. The expansive factor $\gamma s/a$ is computed as $(1 - \xi_a)/\xi_a$, and T denotes the total number of iterations. At each iteration's end, the parameter Θ is updated via $\Theta := \Theta - \eta \nabla \hat{L}_t$. Compared to standard full supervision, Expansive Supervision renders only a subset of rays $R' = A \cup S \subset R$ to guide the model learning process. This selective rendering theoretically reduces computational resources (time and memory) proportional to ξ, where $\xi = \xi_a + \xi_s = |R'|/|R|$.

C. Anchor Area Extractor

Here we detail the design of the anchor area extractor $\mathcal{F}_A(\cdot)$. Given that Expansive Supervision aims to accelerate training, we employ the lightweight edge detector [11] as our basic extractor for frequency prior. To maintain consistent anchor area intensity of $\xi_a|I|$, we implement a progressive threshold adjustment mechanism for the edge detector. At iteration i, the threshold T_i is updated according to:

$$T_i = 1 + \mu(\mathcal{E}(T_{i-1}) - \xi_a|I|) \tag{4}$$

where $\mathcal{E}(T_{i-1})$ represents the sum of the detector output map with threshold T_{i-1}, and μ denotes the step rate. The iteration continues until the condition $0.8 \leqslant \frac{\mathcal{E}(T_i)}{\xi_a|I|} \leqslant 1.2$ is satisfied.

III. Experiments

A. Implementation Details

Backbones & Hyper-parameters. To demonstrate the compatibility of our method with state-of-the-art NeRF acceleration frameworks, we employ TensoRF [4] as the backbone model for all experiments. The key hyper-parameters are set as follows: $\xi_a = \xi_s = 0.25$, $T = 30,000$, and $\mu = 15$.

TABLE I
QUANTITATIVE COMPARISON OF DIFFERENT STRATEGIES ON SYNTHETIC-NERF DATASETS.

Strategies	Iteration: 5k PSNR (dB) ↑	SSIM ↑	L(V) ↓	Iteration: 10k PSNR (dB) ↑	SSIM ↑	Iteration: 15k PSNR (dB) ↑	SSIM ↑	Iteration: 30k PSNR (dB) ↑	SSIM ↑	L(V) ↓	Consumption Time (sec) ↓	Memo. (GB) ↓
Standard	29.30	0.933	9.52e-2	30.78	0.946	31.78	0.954	32.91	0.962	4.97e-2	587.25	21.47
EGRA [9]	29.33	0.933	9.48e-2	30.81	0.947	31.74	0.954	32.85	0.961	5.10e-2	1771.85	17.10
Soft [10]	29.44	0.938	9.35e-2	30.88	0.950	31.61	0.955	32.34	0.959	6.05e-2	585.32	16.29
E.S. (ours)	30.36	0.939	9.20e-3	31.77	0.949	32.42	0.954	33.02	0.961	6.12e-2	578.54	18.33
E.S. ($\beta = 0.9$)	30.21	0.938	9.37e-2	31.65	0.948	32.36	0.953	32.96	0.955	6.04e-2	576.17	16.78
E.S. ($\beta = 0.7$)	29.92	0.934	9.85e-2	31.42	0.946	32.16	0.951	32.87	0.954	6.25e-2	529.47	12.57
E.S. ($\beta = 0.5$)†	29.41	0.929	1.06e-1	30.95	0.941	31.77	0.948	32.64	0.952	6.56e-2	491.32	10.37
E.S. ($\beta = 0.3$)†	28.73	0.922	1.17e-1	30.30	0.935	31.24	0.943	32.31	0.950	6.93e-2	438.91	6.29
E.S. ($\beta = 0.1$)	26.74	0.898	1.52e-2	28.29	0.914	29.37	0.925	30.86	0.939	8.51e-2	388.46	2.56

† denotes the settings strike a balance between quality and efficiency. The best result ; The second best result . The following tables use the same notation.

TABLE II
COMPARISON OF DIFFERENT STRATEGIES ON LLFF DATASETS.

	5k PSNR↑	10k PSNR↑	20k PSNR↑	SSIM↑	L(V)↓
Standard	23.86	25.43	26.10	0.813	0.241
EGRA [9]	23.83	25.30	26.08	0.811	0.237
Soft [10]	23.94	25.47	25.96	0.819	0.227
E.S. ($\beta = 1.0$)	24.11	25.60	26.13	0.813	0.236
E.S. ($\beta = 0.7$)	23.83	25.41	26.11	0.812	0.243
E.S. ($\beta = 0.5$)	23.57	25.21	26.01	0.811	0.250

The ray filtering operation is disabled to ensure compatibility with various strategies. We introduce a quality-computation trade-off parameter $\beta \in [0, 1]$ that modulates the supervision intensity through $\xi_a := \beta \times \xi_a$ and $\xi_s := \beta \times \xi_s$, where higher β values yield better reconstruction quality at the cost of increased computational resources. All other parameters follow the default TensoRF configuration.

Datasets & Metrics. We evaluate our method using both the object-centric Synthetic-NeRF [1] and real-world forward-facing LLFF [12] datasets. For quantitative assessment, we employ PSNR, SSIM [13], and LPIPS [14] metrics, where L(V) denotes the VGG [15] versions of LPIPS.

Experimental Settings. All experiments were conducted using PyTorch [16] on a system equipped with four NVIDIA RTX 3090 GPUs (24.58GB VRAM).

B. Comparison with State-of-the-arts Strategies

Comparing Baselines. We compare our method with standard training (random batch sampling) and recent sampling-based NeRF acceleration methods: EGRA [9] and Soft Mining [10]. For EGRA, we re-implement the algorithm following the original paper's specifications, as their official implementation is not available. For Soft Mining, we directly adapt their official implementation without modifications, maintaining their recommended soft coefficients ($\alpha = 0.6$ for Synthetic-NeRF and $\alpha = 0.8$ for LLFF datasets). Time measurements exclude I/O operations and preprocessing costs. Memory consumption during training is recorded and reported as Memo.

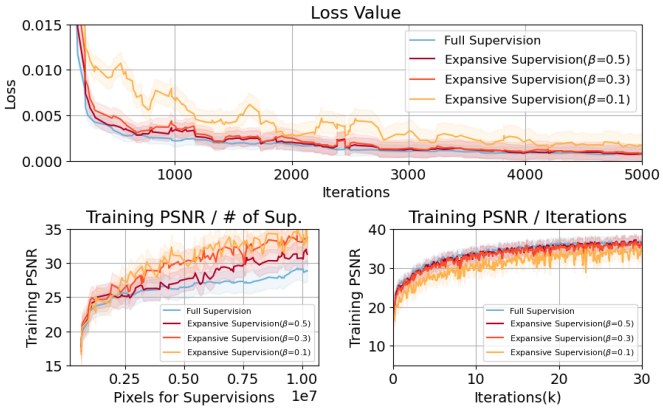

Fig. 4. **Convergence performance of expansive supervision.** Our method achieves precise error estimation comparable to full supervision and exhibits faster convergence as the number of supervised pixels increases.

Quantitative Results. The quantitative comparison results on Synthetic-NeRF and LLFF datasets are presented in Tab.I and Tab.II, respectively. Compared to existing sampling-based acceleration methods, Expansive Supervision (E.S.) achieves superior reconstruction quality under comparable time and memory constraints, with particularly notable improvements during early training stages. This performance advantage is consistently demonstrated across both synthetic and real scenes. Furthermore, our ablation study with varying β values shows that with $\beta = 0.5$, our method achieves comparable reconstruction quality while reducing training time by 84% and memory consumption by 48% relative to standard training.

Convergence & Visualization Results. The convergence performance of our method is visualized in Fig. 4, demonstrating the effectiveness of Expansive Supervision. Our approach with $\beta = 0.3$ and $\beta = 0.5$ achieves error estimation accuracy comparable to full supervision, as evidenced in the upper and lower right sub-figures. The convergence rate accelerates as the number of supervised pixels increases. The visual quality comparison is presented in Fig. 5. We evaluate our method against full supervision under varying computational con-

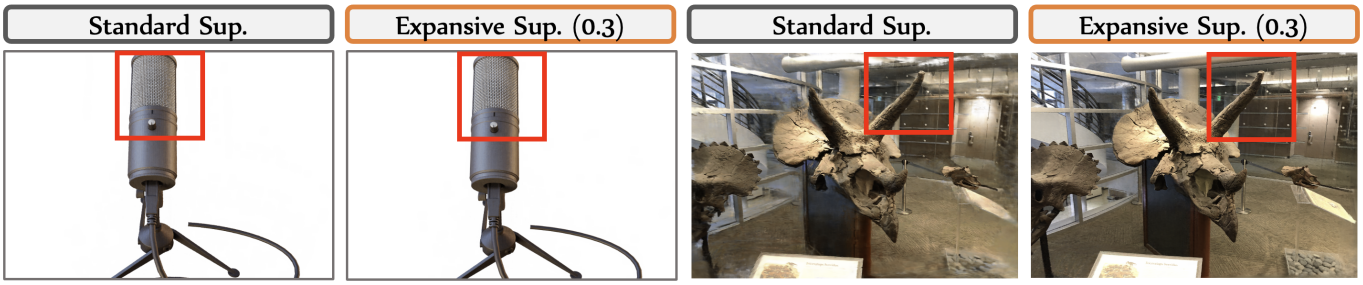

Fig. 5. **Visual quality comparison with standard supervision**. Under the same constrained computational resources, expansive supervision demonstrates higher quality reconstruction compared to standard supervision.

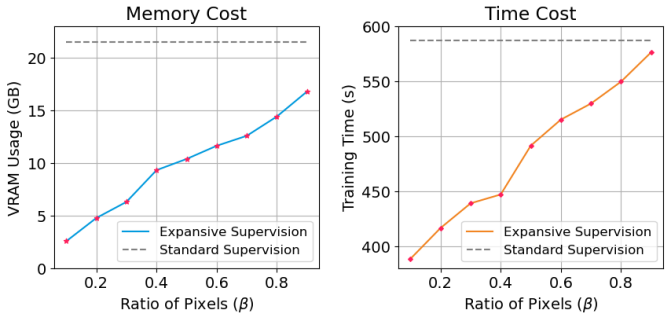

Fig. 6. **Memory and training time cost of expansive supervision.** Both memory and time consumption demonstrate linear scaling with β

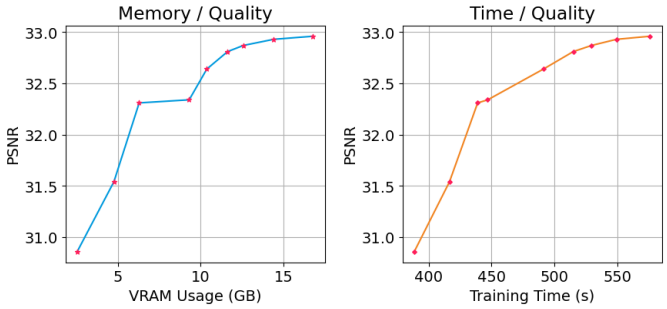

Fig. 7. **Memory and training time cost of expansive supervision.** As the supervised pixel ratio increases, the margin of resource savings decreases.

straints, simulated by adjusting batch sizes while maintaining consistent iteration counts. Our method, with its emphasis on high-frequency regions, achieves superior detail preservation compared to standard supervision under similar computational budgets.

C. Analysis of Resource Savings

Measurements. To validate the theoretical resource savings outlined in Sec. II-A, we conducted systematic measurements of practical memory and time consumption under Expansive Supervision. The experiments were performed in isolation on a dedicated server, focusing exclusively on training operations (forward passes, loss computation, backward passes, and volume rendering), excluding I/O and pre-processing overhead. To ensure measurement precision, all extraneous processes were terminated, and experiments were conducted sequentially. We evaluated computational costs across ten β values ranging from 0.1 to 1.0, with results visualized in Fig. 6. Both memory and time consumption demonstrate linear scaling with β, confirming its effectiveness as a control parameter for balancing reconstruction quality against computational resources.

Time-Quality Trade-off. We analyze the relationship between resource utilization and model performance across various configurations. The resource-quality curves illustrated in Fig. 7 demonstrate how β modulates both time and memory consumption. Our analysis reveals an optimal reduction in supervised pixels at approximately $\beta = 0.3$, where $\frac{\mathrm{d}C(\text{Time})}{\mathrm{d}\beta}$ and $\frac{\mathrm{d}C(\text{Memo.})}{\mathrm{d}\beta}$ reach their maximal values. Beyond this point, the marginal resource savings diminish significantly. While lower supervision ratios ($\beta = 0.1$) lead to substantial resource savings, they introduce noticeable artifacts incompatible with our mechanism. Thus, $\beta = 0.3$ emerges as the optimal trade-off between rendering quality and computational efficiency.

D. Extension to Implicit Neural Representations for Images

Settings. To further validate the effectiveness and generalizability of our method, we extend it to Implicit Neural Representations (INR) for images. Image fitting can be formulated as $F(\Theta) : (x, y) \mapsto (r, g, b)$. We employ SIREN [17] with a 3×256 network architecture as the backbone, evaluated on the processed DIV2K dataset [18], [19]. Training proceeds for 5,000 iterations with a batch size of $0.5 \times h \times w$, where h and w denote the image height and width, respectively.

Results. The comparative results, averaged across the dataset, are presented in Tab. III. Our method outperforms standard training (uniform sampling), EGRA, and Soft Mining, with particularly notable improvements in later training stages. These results demonstrate that Expansive Supervision generalizes beyond NeRF applications to other neural representation modalities.

E. Ablation Studies

Settings. To validate the effectiveness of each component within our proposed Expansive Supervision mechanism, we conducted comprehensive ablation experiments. Using Expansive Supervision ($\beta = 1.0$) as the baseline, we first examined

TABLE III
COMPARISON ON 2D IMAGE FITTING WITH INRS.

	1k PSNR↑	2k PSNR↑	5k PSNR↑	5k SSIM↑	5k L(V)↓
Standard	30.35	33.33	36.11	0.956	0.026
EGRA [9]	30.37	33.40	36.22	0.957	0.025
Soft [10]	31.49	33.52	35.31	0.948	0.045
E.S. ($\beta = 1.0$)	30.95	33.70	36.28	0.955	0.023

TABLE IV
ABLATION STUDY.

Settings	5k PSNR↑	30k PSNR↑	30k SSIM↑	30k L(A)↓	30k L(V)↓
E.S. ($\beta = 1.0$)	30.36	33.02	0.961	3.69e-2	6.12e-2
w/o anchor area	27.29	29.68	0.946	4.53e-2	6.41e-2
w/o source area	17.36	18.34	0.451	6.14e-1	5.24e-1
w/o anchor sup.	26.59	29.10	0.939	5.45e-2	7.38e-2
w/o source sup.	19.34	20.21	0.586	5.28e-1	4.59e-1
w/o expansive	29.98	32.69	0.949	4.40e-2	8.56e-2

the impact of different sampling areas: *w/o anchor area* samples only from non-important regions, while *w/o source area* samples exclusively from fixed anchor areas. Furthermore, we evaluated each component in Eq. 3: anchor supervision, source supervision, and the expansive mechanism.

Results. The results presented in Tab. IV demonstrate that each component contributes distinctly to the overall performance. Removing either anchor or source areas effectively reduces our method to uniform sampling in different spaces. The *w/o source area*, which samples only from anchor areas, shows severe performance degradation due to its limited sampling space. In the analysis of Eq. 3 components, removing source supervision (*w/o source sup.*) introduces significant training bias by eliminating uncertainty supervision. The *w/o expansive* configuration, which merely supervises two areas independently, reduces Eq. 3 to approximate squared error, resulting in suboptimal reconstruction quality.

IV. CONCLUSION

In this paper, we introduce an expansive supervision mechanism that enhances the efficiency of neural radiance field training. Our approach leverages the observation that training error distributions exhibit strong correlations with image content, following a long-tail distribution pattern. By strategically selecting ray subsets for rendering and utilizing image context for expansive error estimation, our method achieves significant computational efficiency while maintaining reconstruction quality. Experimental results demonstrate that our method outperforms existing sampling-based acceleration approaches under equivalent computational constraints. Moreover, when integrated with current acceleration frameworks, our approach achieves substantial memory savings with minimal implementation effort.

REFERENCES

[1] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV (1)*. 2020, vol. 12346 of *Lecture Notes in Computer Science*, pp. 405–421, Springer. 1, 4

[2] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger, "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps," in *ICCV*. 2021, pp. 14315–14325, IEEE. 1

[3] Junli Cao, Huan Wang, Pavlo Chemerys, Vladislav Shakhrai, Ju Hu, Yun Fu, Denys Makoviichuk, Sergey Tulyakov, and Jian Ren, "Real-time neural light field on mobile devices," in *CVPR*. 2023, pp. 8328–8337, IEEE. 1

[4] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su, "Tensorf: Tensorial radiance fields," in *ECCV (32)*. 2022, vol. 13692 of *Lecture Notes in Computer Science*, pp. 333–350, Springer. 1, 3

[5] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, 2022. 1

[6] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa, "Plenoxels: Radiance fields without neural networks," in *CVPR*. 2022, pp. 5491–5500, IEEE. 1

[7] Cheng Sun, Min Sun, and Hwann-Tzong Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *CVPR*. 2022, pp. 5449–5459, IEEE. 1

[8] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann, "Point-nerf: Point-based neural radiance fields," in *CVPR*. 2022, pp. 5428–5438, IEEE. 1

[9] Zhenbiao Gai, Zhenyang Liu, Min Tan, Jiajun Ding, Jun Yu, Mingzhao Tong, and Junqing Yuan, "Egra-nerf: Edge-guided ray allocation for neural radiance fields," *Image and Vision Computing*, vol. 134, pp. 104670, 2023. 2, 4, 6

[10] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi, "Accelerating neural field training via soft mining," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20071–20080. 2, 4, 6

[11] John F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, 1986. 3

[12] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar, "Local light field fusion: practical view synthesis with prescriptive sampling guidelines," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 29:1–29:14, 2019. 4

[13] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004. 4

[14] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*. 2018, pp. 586–595, Computer Vision Foundation / IEEE Computer Society. 4

[15] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015. 4

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019, pp. 8024–8035. 4

[17] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in neural information processing systems*, vol. 33, pp. 7462–7473, 2020. 5

[18] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," in *NeurIPS*, 2020. 5

[19] Eirikur Agustsson and Radu Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 126–135. 5