Eco-NeRF: Energy-Aware Cloud-to-Edge Training and Deployment of Neural Radiance Fields for Satellite IoT

Kriti Ghosh[1], Braveenan Sritharan[1], Lakshmish Ramaswamy[1], In Kee Kim[1],
Suchendra M. Bhandarkar[1], Deepak Mishra[2], Aditya Joshi[1], Nancy O'Hare[2],
[1]School of Computing, [2]Center for Geospatial Research and Department of Geography
The University of Georgia, Athens, Georgia 30602, USA
Email: {kriti.ghosh, sritharan.braveenan, laksmr, inkee.kim, suchi, dmishra, aditya.joshi, nkohare}@uga.edu

***Abstract*—Neural Radiance Fields (NeRFs) are widely adopted for novel view synthesis and 3D scene reconstruction. However, their training is computationally demanding and energy-intensive, particularly for satellite imagery variants that have high model complexity. While prior work has largely focused on improving runtime or memory usage, the actual energy consumption of NeRF training remains largely unexplored.**

This paper presents Eco-NeRF, a cloud-to-edge energy-aware pipeline for NeRF-based satellite reconstruction. We first characterize energy consumption across multiple NeRF variants on satellite imagery, jointly analyzing training energy, runtime, and reconstruction accuracy. We observe that backward propagation dominates training cost, yet many gradient updates are redundant. Building on this insight, we introduce a selective backpropagation strategy that reduces training energy by up to 30% with negligible impact on reconstruction quality. We further apply knowledge distillation to compress models to 20% of their original size and evaluate edge-side inference on the Jetson Orin platform, achieving a 70% inference energy reduction and a 3.2x latency improvement with only modest quality loss. Our results demonstrate that energy-aware optimization is effective throughout the NeRF lifecycle, from cloud training to edge deployment.

***Index Terms*—Neural Radiance Fields, energy-aware training, edge deployment, knowledge distillation, satellite imagery**

I. Introduction

Neural Radiance Fields (NeRFs) have emerged as a powerful paradigm for realistic novel view synthesis and 3D scene reconstruction across diverse settings including synthetic objects, real-world indoor-outdoor scenes, and large-scale aerial imagery [1], [2]. More recently, NeRFs have been adapted for remote sensing and satellite imagery, with variants incorporating sensor geometry, uncertainty estimation, and shadow modeling to enable 3D reconstruction from sparse-view satellite images [3]–[5]. However, these satellite-specific models introduce high computational demand, making energy consumption a critical concern. This is especially important for edge deployments for satellite imagery processing, which often operate under strict power constraints and in remote locations with limited power availability.

Despite growing interest in efficient NeRF training, prior work has largely focused on reducing wall-clock time or memory usage through architectural innovations such as hash-encodings, tensor factorization, and voxel-based representations [6]–[10]. While these approaches accelerate training, they rarely measure or report actual energy consumption. A recent study analyzes NeRF's energy footprint [11], but remains diagnostic in nature without proposing optimization strategies. As a result, the true energy consumption of NeRF training remains largely underexplored.

To address this gap, we begin by characterizing energy consumption during NeRF training. A key observation is that backward propagation dominates training, accounting for about 47% of iteration time, which is more than twice that of forward passes (Table I in Section II). Prior work has shown that many of these gradient updates are redundant in deep learning [12] and NeRF training [13], but without quantifying energy savings. This suggests that selectively computing gradients only for informative rays can substantially reduce energy consumption without sacrificing reconstruction quality.

Building on this observation, we present **Eco-NeRF**, a cloud-to-edge energy-aware pipeline for NeRF-based satellite reconstruction. Our approach operates in three stages: (1) during training, we apply selective backpropagation (SBP) that identifies high-impact rays based on reconstruction loss and geometric cues; (2) after training, we compress the model via knowledge distillation; and (3) the compressed model is deployed on edge devices for energy-efficient inference.

Our evaluation on four satellite scenes shows that Eco-NeRF achieves 29% – 31% training energy reduction by computing gradients for only 50% of rays, with negligible quality loss (ΔPSNR $\leq$ 0.45 dB, ΔSSIM $\leq$ 0.003). For edge deployment, model compression reduces model size to 20% of original parameters and yields 70% inference energy reduction and 3.2$\times$ latency improvement with only modest quality loss. These results highlight the importance of energy-aware design for NeRF deployment.

Specifically, we make the following scientific contributions:

- **An energy-aware benchmark of NeRF-based models for satellite imagery**, jointly analyzing training energy, runtime, and reconstruction accuracy across multiple scenes and model variants.
- **A selective backpropagation strategy for NeRF training** that prioritizes informative rays based on reconstruction loss and geometric cues, reducing training energy by up to 30% without observable quality degradation.
- **A knowledge distillation-based compression pipeline** that reduces model size to $\sim$20% of original parameters while preserving practical reconstruction quality.

- **A cloud-to-edge deployment validation on the Jetson platform**, characterizing inference latency and energy consumption under realistic edge IoT constraints.

The remainder of this paper is organized as follows. Section II provides background on NeRFs and identifies energy bottlenecks in training. Section III presents the Eco-NeRF pipeline, including SBP and model compression. Section IV evaluates our approach on satellite imagery. Section V discusses our findings, and Section VI concludes.

II. Background and Motivation

A. Neural Radiance Fields (NeRFs)

NeRFs represent a scene as a continuous volumetric function and map a given spatial location and viewing direction to color and volume density enabling realistic novel view synthesis. Given a camera ray, $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, the rendered color can be obtained from the following volume rendering equation:

$$\mathbf{C}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\,\sigma(\mathbf{r}(t))\,\mathbf{c}(\mathbf{r}(t), \mathbf{d})\,dt,$$

$$T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\,ds\right)$$

where σ denotes volume density and $\mathbf{c}$ the view-dependent radiance [1]. While the original NeRF formulation assumes static scenes with consistent illumination, several variants, such as NeRF-W [14], S-NeRF [4], Sat-NeRF [3], introduce modifications to address illumination variation, transient factors, uncertainty estimation and shadow-awareness. They extend the application scope from a wide range of on-the-go photography to large-scale satellite images. Even though these variants improve reconstruction quality, they also increase model complexity and training time, which leads to further energy costs.

B. Identifying Energy Bottlenecks in NeRF Training

Despite recent advances in accelerating NeRF training, the energy implications of training remain underexplored. To identify dominant factors contributing to the runtime and energy impact, we profile the NeRF training pipeline on NVIDIA RTX-A6000 using PyTorch Lightning's built-in profiler. The results reveal that backward propagation consistently dominates execution time, accounting for $\sim$45% of the total iteration cost (Table I), exceeding the costs of the model forward pass and other training components. The remaining runtime is dominated by optimizer bookkeeping, validation passes, and framework overhead, which lie outside gradient computation.

Since GPU power draw closely aligns with active compute time, this observation directly translates into backpropagation being the primary energy bottleneck during NeRF training. Existing NeRF optimization efforts largely target architectural choices, sampling strategies, or faster convergence [6]–[8], [15]. However, energy consumption is considered an implicit by-product of reduced runtime and is rarely quantified directly in existing NeRF literature for optimization strategies. Some studies that highlight the energy impact are diagnostic in nature, without coupling it with an effective optimization strategy. In contrast, our profiling analysis motivates a targeted energy-aware optimization of backward computation without modifying model architecture. To the best of our knowledge, few studies jointly analyze energy profiles and optimization strategies for NeRFs, particularly in cloud-to-edge settings relevant to satellite IoT deployments.

TABLE I
Runtime Breakdown of Sat-NeRF Training (JAX-068)

Training Phase	Runtime Share (%)
Forward pass (`model_forward`)	19.93
Backward propagation (`backward`)	**46.60**

C. Previous Advancements and Limitations

A large body of work accelerates NeRF training and inference through architectural, representation-level, or sampling-based optimizations [6]–[10], [15]. However, they primarily focus on forward computation and do not explicitly analyze or target training-time energy consumption, particularly the cost of backward propagation. Several existing works have emphasized the impact of backward propagation on the computational overhead [12]. However, such studies based in the context of NeRF are very limited. Studies which suggest alternative backpropagation strategies for NeRF [13], do not always quantify the energy impact. A few existing works have explored the energy impact of Neural Radiance Fields. While some of them [16], [17] explore the energy impact of a substantial redesign of NeRF architecture using a spiking neural network, others are purely diagnostic [11] in nature and do not couple an optimization strategy for NeRF training with energy impact analysis. Prior work has also explored edge-based NeRF acceleration focused on inference efficiency [18]. However, our work targets the NeRF training itself to reduce energy consumption without changing model architecture. Moreover, we also demonstrate the feasibility of NeRF inference on edge with energy impact analysis to provide an optimized energy-aware life-cycle.

III. Eco-NeRF: Energy-Aware Cloud-to-Edge Pipeline

A. Problem Formulation

While backward propagation remains a significant contributor to the energy impact, NeRF training involves other factors, including scene characteristics, architectural choices, etc. A key challenge is to determine how these factors interact with training dynamics and contribute to the total energy footprint. Even though our profiling results indicate that optimizing backward propagation should be effective, naive sub-sampling of rays can degrade reconstruction quality. Hence, an optimal ray selection strategy is required to prioritize informative rays.

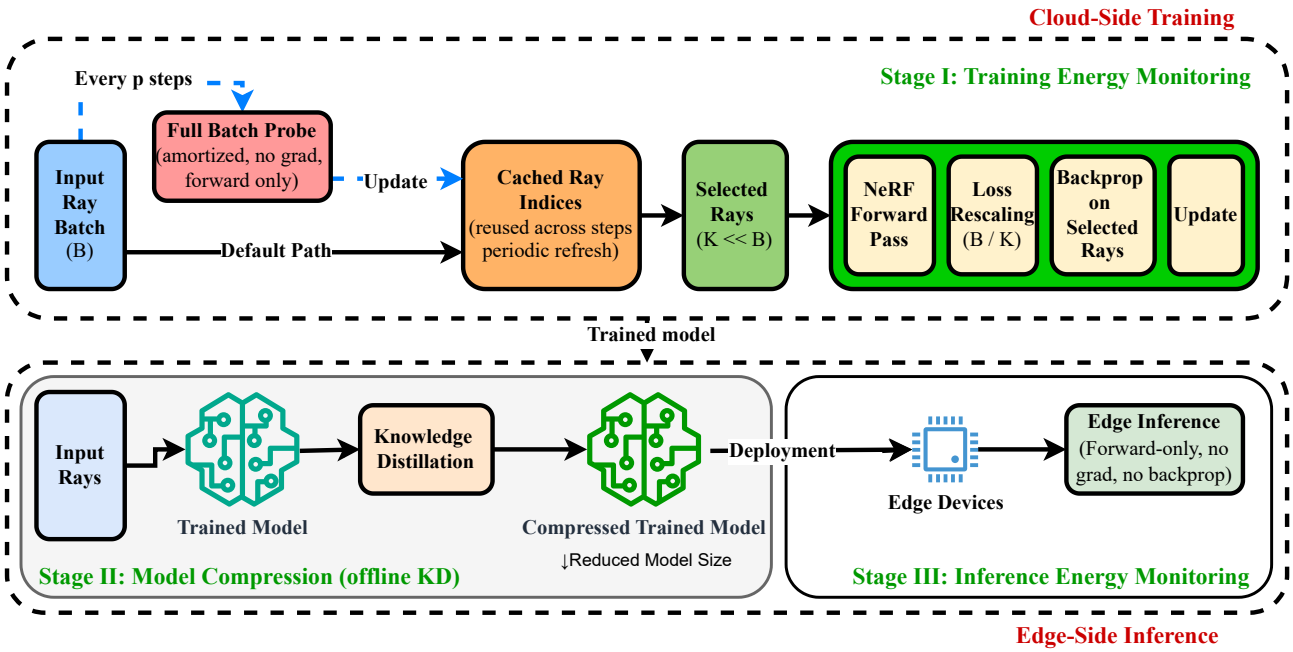

Fig. 1. **Eco-NeRF Pipeline with selective backpropagation (SBP)**: A full-batch forward-only probe is performed periodically to update cached ray importance, while gradients are computed only for a selected subset of rays during training.

Furthermore, efficient deployment requires reducing inference-time cost on resource-constrained edge devices without sacrificing performance. Based on these challenges, we study and address the following research questions:

RQ1: Beyond backward propagation, which factors most strongly influence NeRF energy consumption?

RQ2: How can selective backpropagation be designed to reduce training-time energy and carbon footprint while maintaining reconstruction accuracy?

RQ3: How can model compression via knowledge distillation preserve these energy savings during edge inference?

To address the above, first, we do an energy analysis on our baseline models without any optimization to identify contributing factors. Second, we discuss our ray selection and SBP strategy with explanations for our design choices. And finally, we show that knowledge distillation on the original model compresses and preserves most of the original performance while reducing model size and inference energy.

B. Energy-aware Training

Eco-NeRF is an end-to-end, energy-aware cloud-to-edge pipeline for training and deploying NeRFs. Figure 1 provides a clear overview of the system. The pipeline has three stages.

1) Stage I : Training with SBP: Eco-NeRF alters the standard NeRF training with an SBP strategy which reduces backward-pass computation without changing the NeRF architecture. This is possible with the help of the following two components.

a) Ray Selection Algorithm: Algorithm 1 selects a subset of informative rays based on their estimated contribution to the training loss. Each ray is assigned a difficulty score based on its reconstruction error, modulated by an opacity-based validity proxy and a lightweight geometric cue. These terms down-weight rays traversing empty space while prioritizing rays with informative geometry. The highest-scoring rays are selected and a small fraction of rays is drawn uniformly at random and merged with the difficult set to preserve stability and prevent sampling bias.

Algorithm 1 Loss-driven Ray Selection (SBP RaySelector)

Require: Per-ray predictions $\hat{\mathbf{c}}_i$, GT colors $\mathbf{c}_i$, (optional) weights $\mathbf{w}_i$; keep ratio ρ

$K \leftarrow \max(1, \lfloor \rho B \rfloor)$

$e_i \leftarrow \text{mean}\big((\hat{\mathbf{c}}_i - \mathbf{c}_i)^2\big)$ ▷ per-ray RGB MSE (hardness)

$\alpha_i \leftarrow \textsc{OpacityProxy}(\mathbf{w}_i)$ ▷ if unavailable, set $\alpha_i = 1$

$s_i \leftarrow \textsc{GeomCue}(\mathbf{w}_i)$ ▷ surface/ambiguity cue (optional)

$\textsc{score}_i \leftarrow e_i \cdot g(\alpha_i) + \lambda s_i$ ▷ loss + validity/geometry

$\mathcal{I}_{\text{hard}} \leftarrow \textsc{TopK}(\textsc{score}, K(1-\gamma))$

$\mathcal{I}_{\text{rand}} \leftarrow \textsc{UniformSample}(\{1..B\}, K\gamma)$ ▷ anchors for stability

$\mathcal{I} \leftarrow \textsc{Unique}(\mathcal{I}_{\text{hard}} \cup \mathcal{I}_{\text{rand}})$

return $\mathcal{I}$ ▷ $|\mathcal{I}| = K$

b) Selective Backpropagation (SBP): Algorithm 2 implements SBP by limiting gradient computation to a subset of informative rays. The indices produced by the ray selection algorithm are cached and reused across multiple training iterations. Periodically, a full forward-only probing pass is performed to refresh the cache without gradient tracking.

Algorithm 2 Selective Backpropagation (SBP)

Require: Ray batch (r, c) of size B, keep ratio ρ, warmup steps T_w
Ensure: Training loss $\mathcal{L}$

$t \leftarrow t + 1$ ▷ global step
$K \leftarrow \max(1, \lfloor \rho B \rfloor)$
if $t < T_w$ **or** $\rho = 1$ **then**
    $\hat{y} \leftarrow f_\theta(r)$
    $\mathcal{L} \leftarrow \text{Loss}(\hat{y}, c)$
    Backpropagate through $\mathcal{L}$
else
    Periodically run a forward-only probe to score rays
    Select top-K informative rays (with random anchors)
    $\hat{y}_K \leftarrow f_\theta(r_K)$
    $\mathcal{L} \leftarrow \text{Loss}(\hat{y}_K, c_K)$
    $\mathcal{L} \leftarrow \mathcal{L} \cdot \frac{B}{K}$ ▷ rescale
    Backpropagate only through selected rays
end if
return $\mathcal{L}$

During regular training steps, gradients are computed only for the selected rays with the loss rescaled to match full-batch magnitude.

Training-time energy consumption is monitored throughout Stage I at the cloud side, capturing the impact of SBP on GPU utilization, runtime, total energy draw, and carbon footprint.

2) Stage II: Model Compression: In this stage, knowledge distillation is used to transfer scene-specific representations from the full-capacity teacher (8 layers, 512 hidden units) to a smaller student. Intuitively, instead of learning only from ground-truth images, the student is guided by the teacher's predictions, allowing it to approximate the behavior of a larger network with fewer parameters. The smaller student network has reduced depth and width (6 layers, 256 hidden units) but follows the same architectural structure as the teacher. It is trained using the standard NeRF color reconstruction loss, augmented with an auxiliary knowledge distillation term that encourages the student's RGB predictions to match those of the frozen teacher using mean squared error (MSE).

3) Energy-aware Edge Inference: The compressed model is then used for forward-only inference on resource-constrained edge devices. Since inference requires only forward passes (without gradient computation), this stage evaluates how model compression translates into reduced runtime and energy consumption under practical deployment conditions. During edge inference, we measure reconstruction quality, latency, and energy usage.

IV. Evaluation

A. Experimental setup

1) Dataset: We evaluate Eco-NeRF using four scenes from the 2019 IEEE GRSS Data Fusion Contest dataset [19], which consists of Maxar WorldView-3 satellite imagery collected between 2014 and 2016 over Jacksonville, Florida, USA. We use scenes JAX-004, JAX-068, JAX-214, and JAX-260 for our results. For each scene, RGB image crops of approximately 800×800 pixels at a spatial resolution of 0.3 m/pixel at nadir are used. This corresponds to a ground coverage of 256×256 meters.

2) Model configurations: We evaluate the Eco-NeRF pipeline using three NeRF-based models with increasing complexity: NeRF, S-NeRF, and Sat-NeRF [1], [3], [4].

NeRF [1] is the baseline model that represents a scene using an MLP mapping 3D coordinates and viewing directions to color and density, trained using a photometric reconstruction loss. S-NeRF [4] extends NeRF with shadow-aware illumination modeling. Sat-NeRF [3] further incorporates satellite-specific camera models based on rational polynomial coefficients (RPCs) and uncertainty-aware loss weighting, introducing the highest model complexity. IC9600 [20] complexity metric is used to calculate scene complexity.

3) Training setup: All models are trained with the same Adam optimizer for fair comparison, learning rate 0.0005, batch size 1024, chunk size 5120, and a fixed budget of 300000 training steps. Eco-NeRF enables SBP after 200 warm-up steps, keeping only 50% of the rays for gradient computation. The forward-only probe interval is scheduled adaptively during training, using frequent probing in early stages (every 1–10 steps) and gradually increasing to a fixed interval of 50 steps in later training to amortize the cost of full-batch evaluation. The loss is scaled by B/K to match full-batch magnitude.

4) Hardware and System configurations: All training experiments are conducted on a single NVIDIA RTX A6000 GPU (48 GB memory) per run, using CUDA 12.2 and driver version 535. Training jobs are restricted to a single GPU to ensure consistent and fair energy measurements across all configurations.

For edge inference evaluation, we use NVIDIA's Jetson Orin Nano platform, representing a resource-constrained edge platform. Jetson Orin Nano has the Ampere architecture with six Cortex-A78AE cores, 1024 CUDA cores, and 8GB LPDDR5 memory. Models trained in the cloud are transferred to the Orin Nano to measure inference-time performance and energy consumption under realistic deployment conditions.

5) Energy Measurement: Energy consumption and carbon footprint are measured using CodeCarbon [21] and verified with CarbonTracker [22]. Carbon emissions are measured using CodeCarbon, which reports total CO_2-equivalent emissions ($kgCO_2e$) based on measured energy consumption and carbon intensity of regional electricity. On the Jetson Orin Nano, inference energy is measured using `tegrastats`.

B. Baseline Results

In this section, we characterize the energy cost of baseline training, while the resulting profile helps both quantify the impact of the model and identify major energy contributors. Figure 2 visualizes the energy-accuracy trade-off. Sat-NeRF consistently achieves the best reconstruction quality, improving PSNR by up to +9.33 dB over NeRF (JAX-004), but at a significantly higher energy cost. Across all scenes, SatNeRF consumes approximately 41% – 43% more training energy

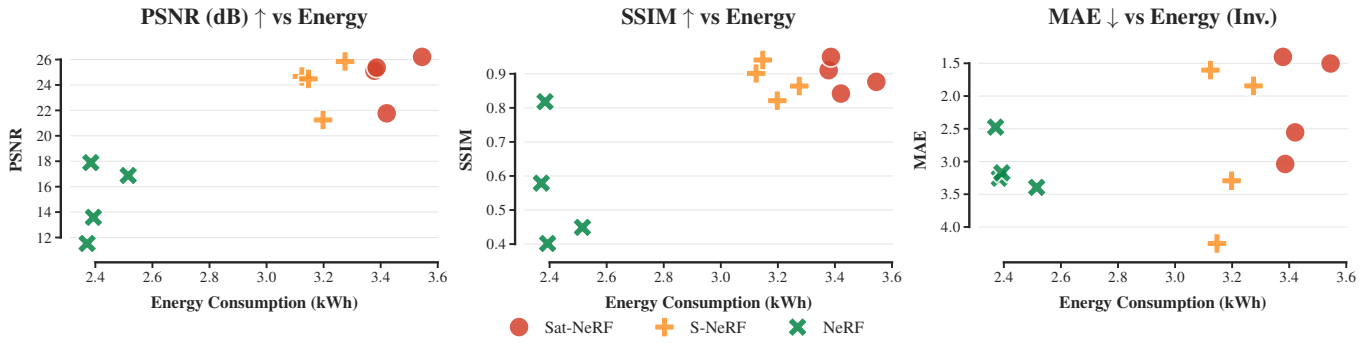

Fig. 2. **Baseline diagnosis: Energy vs accuracy** (PSNR, SSIM, MAE) across all scenes (JAX004-JAX260) and all models namely, NeRF, S-NeRF and Sat-NeRF.

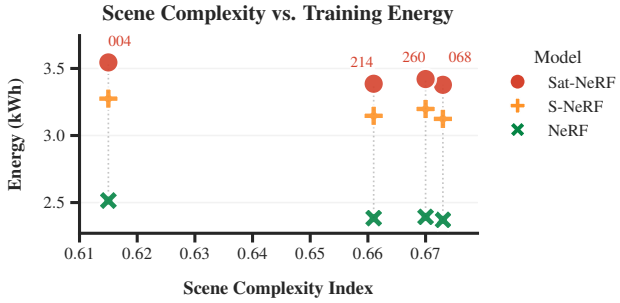

Fig. 3. Baseline diagnosis: Energy vs scene complexity.

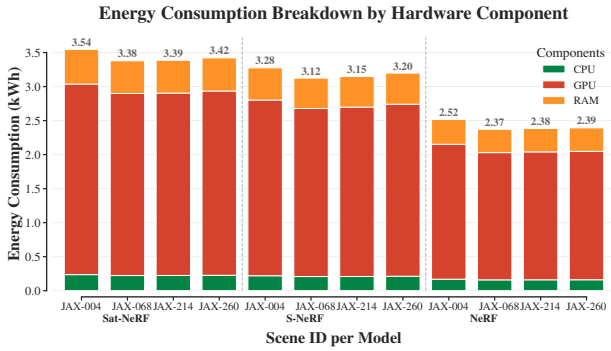

Fig. 4. Baseline diagnosis: Energy components: CPU, GPU and RAM

than NeRF. S-NeRF lies between these extremes, offering moderate quality gains with intermediate energy consumption. These show that higher fidelity requires a higher energy trade-off (Figure 2).

Energy consumption is also strongly architecture dependent and has a weak relationship with scene complexity as displayed in Figure 3. Finally, Figure 4 shows that irrespective of architecture and data, GPU computation dominates overall energy usage. Combined with our profiler results (Table I), these observations identify backward propagation as the dominant bottleneck during training, and GPU-intensive forward computation in complex architectures as the primary bottleneck during inference, both in terms of runtime and energy. Baseline Sat-NeRF training emits 1.53–1.61 kg CO_2 per scene. Carbon emissions mirror training energy across models, with Sat-NeRF highest and NeRF lowest.

These results motivate the need for intentional optimization that targets dominant costs without performance degradation.

C. Eco-NeRF Results

In this section, we evaluate the Eco-NeRF pipeline, showing the effectiveness of SBP across four satellite scenes (JAX-004, 068, 214, 260), focusing on training energy, runtime, and reconstruction quality of the most complex model, Sat-NeRF. Figure 5 decomposes the training into forward and backward phases, revealing that compared to baseline, SBP substantially reduces backward computation, which dominates the baseline training. Figure 6 presents the main result where

TABLE II
EFFECT OF SBP ON TRAINING EFFICIENCY AND RECONSTRUCTION QUALITY FOR SAT-NERF. VALUES REPORT PERCENTAGE REDUCTIONS IN ENERGY, CARBON AND TIME, AND ABSOLUTE CHANGES IN RECONSTRUCTION METRICS (SBP MINUS BASELINE).

Scene	Energy ↓ (%)	CO2 ↓ (%)	Time ↓ (%)	ΔPSNR	ΔSSIM	ΔMAE
004	29.13	29.1	26.76	−0.25	−0.0024	+0.042
068	30.50	30.5	27.58	−0.09	−0.0022	−0.054
214	30.60	30.7	27.49	−0.45	−0.0031	−0.016
260	30.91	30.9	28.01	−0.02	−0.0001	−0.081

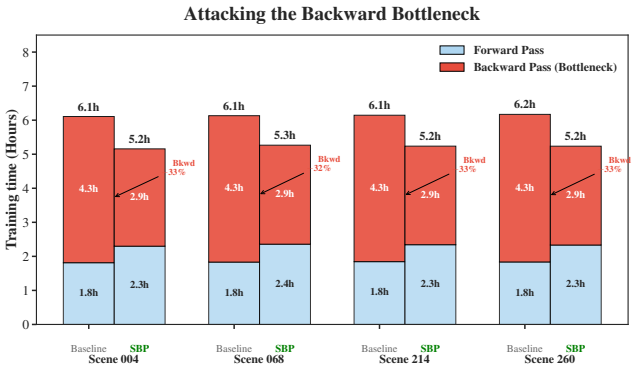

Fig. 5. **Effect of EcoNeRF on forward and backward training phases**: While training is dominated by the backward propagation time, SBP substantially reduces the backpropagation duration and thereby reduces total training time.

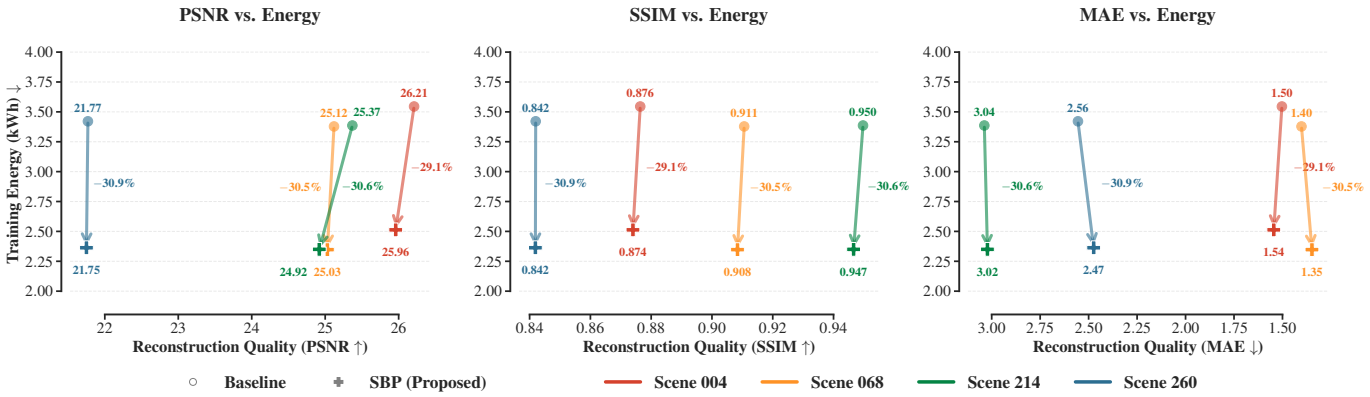

Fig. 6. **Eco-NeRF Training: Energy–Accuracy Efficiency of SBP vs. Baseline (Sat-NeRF).** Each arrow represents the transition from the baseline to the proposed SBP strategy for a specific scene. The vertical descent signifies a reduction in training energy (kWh), while the horizontal displacement represents the change in reconstruction quality. The near-vertical inclination across all scenes (004–260) and metrics (PSNR, SSIM, MAE) demonstrates a Pareto-efficient trade-off, achieving up to ∼30% training energy savings with negligible degradation in performance.

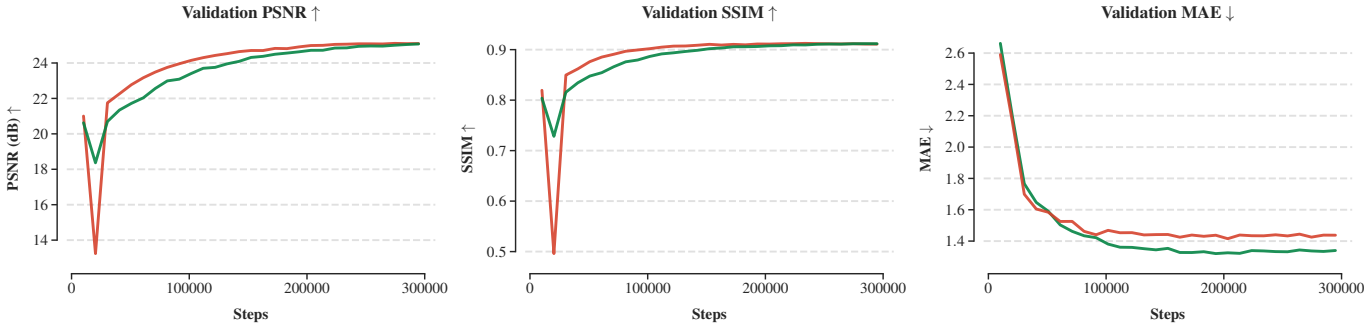

Fig. 7. **Evidence of Training Stability**: Validation curves on example scene JAX-068 using Sat-NeRF. SBP (green) achieves comparable performance with the baseline (red) on appearance metrics. SBP achieves better MAE compared to baseline.

Fig. 8. **Qualitative comparison: Baseline vs SBP**: Both SBP RGB and DSM predictions for JAX-068 using Sat-NeRF, match baseline predictions.

SBP consistently shifts each operating point from baseline to Eco-NeRF downward with minimal horizontal shift, indicating minimal accuracy drop with steady energy savings. Quantitatively, SBP achieves 29% – 31% energy savings and 26% – 28% runtime reduction across all scenes (Table II), while incurring only negligible changes in PSNR ($\leq$ 0.45 dB), SSIM ($\leq$ 0.0031), and MAE ($\leq$ 0.081). Additionally, SBP consistently reduces estimated carbon emissions by up to 31% across Sat-NeRF scenes. This shows SBP as a Pareto-efficient mechanism leading to Eco-NeRF's effectiveness. Table III shows a generalization check across less complex models (NeRF and S-NeRF) using scene 260. This shows that the savings, though present, are less pronounced in these models. Figure 7 confirms that SBP preserves training stability and tracks baseline performance closely. In fact, the depth-aware ray selection provides up to 3.85% better MAE compared to the baseline. Finally, qualitative results show visually indistinguishable RGB and DSM predictions despite large energy savings (Figure 8).

D. Edge Inference Results

The inference is evaluated using compressed models on resource constrained jetson edge devices, comparing them

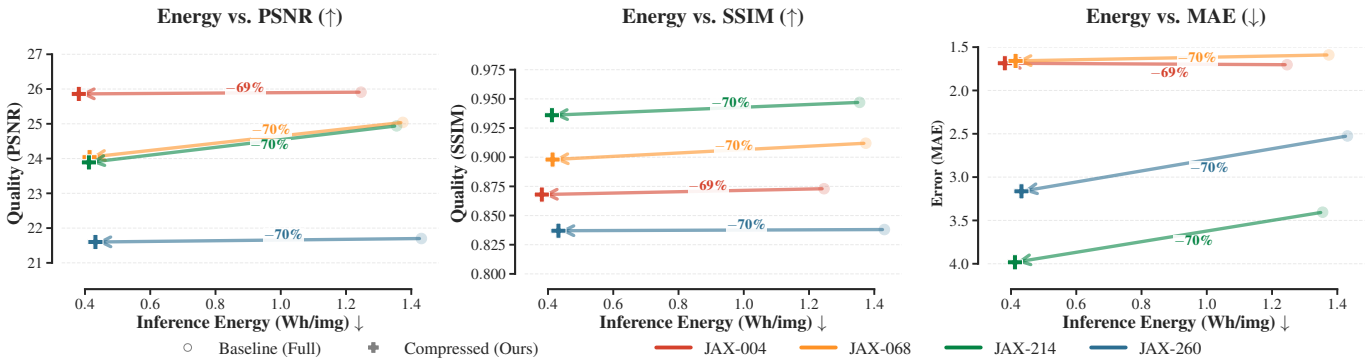

Fig. 9. **Eco-NeRF Inference: Energy–Accuracy Efficiency of Model Compression vs. Full Sat-NeRF.** Each arrow represents the transition from the high-capacity baseline to the proposed compressed model for a specific scene (JAX 004–260). The horizontal displacement signifies a drastic reduction in inference energy (Wh/img), while the vertical inclination represents the stability of reconstruction quality. Across all metrics—PSNR, SSIM, and MAE—the trajectories demonstrate a dominant Pareto improvement, achieving a significant ∼70% reduction in inference energy consumption with minimal impact on structural and spectral fidelity.

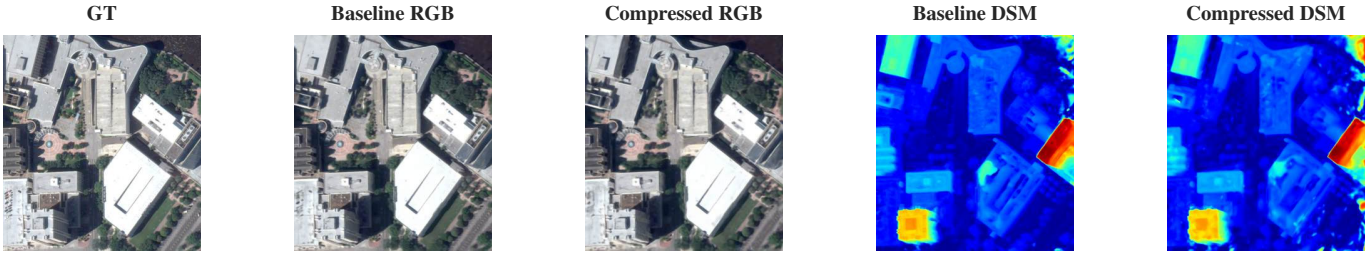

Fig. 10. **Qualitative comparison of edge outputs: Baseline Full vs Compressed**: Compressed model has preserved the majority of the reconstruction quality of the full model.

TABLE III
GENERALIZATION OF SBP TO LIGHTER NERF VARIANTS. PERCENTAGE REDUCTIONS IN TRAINING ENERGY AND TIME, AND ABSOLUTE CHANGES IN RECONSTRUCTION METRICS (SBP MINUS BASELINE).

Model	Scene	Energy ↓ (%)	Time ↓ (%)	ΔPSNR	ΔSSIM	ΔMAE
S-NeRF	260	28.01	25.29	−0.19	+0.0030	−0.0667
NeRF	260	19.82	11.92	+0.52	+0.0208	+0.154

against the full Sat-NeRF models in terms of per-view inference energy, latency, and reconstruction quality. Figure 9 visualizes the main energy-accuracy trade-off for the compressed models. Across all scenes, model compression yields a 70% reduction in inference energy with only a marginal degradation in PSNR ($\leq$ 1.05dB), SSIM ($\leq$ 0.014), and MAE ($\leq$ 0.64). This confirms that most of the representational capacity of the full model is retained despite a 5$\times$ parameter reduction (2.6M $\rightarrow$ 0.5M) achieved by reducing network depth and width in stage-II. Qualitative comparisons in Figure 10 confirm that both RGB appearance and the DSM structure remain visually consistent with the baseline. Figure 11 shows the reduction in inference energy due to the compression. Figure 12 further shows that compression reduces latency by up to 3.2$\times$, enabling fast, energy-aware real-time edge deployment.

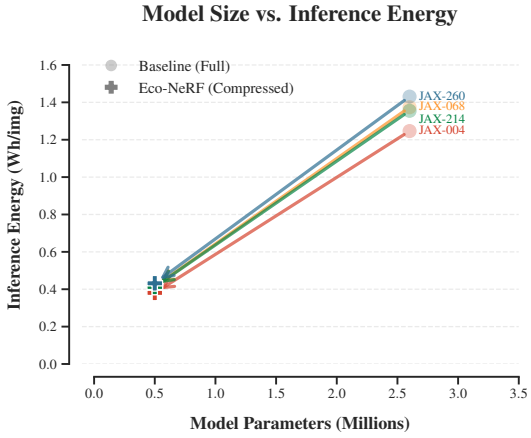

Fig. 11. Sat-NeRF: Model Size vs Inference Energy

V. DISCUSSION

Our results indicate backward propagation, GPU computation, and architectural complexity to be the sources of the primary runtime and energy bottleneck during NeRF training. Our SBP addresses this issue by selectively reducing backward computation while preserving informative gradients, producing comparable reconstruction with improved geometry at up to 30% less energy.

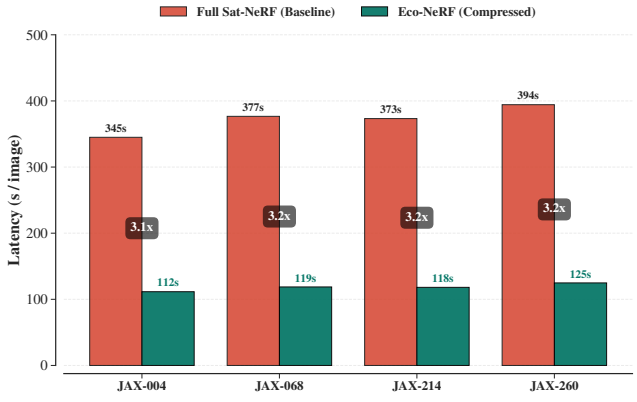

Fig. 12. **Latency Comparison, Baseline vs Compressed**: Compressed model reduces latency up to 3.2x.

The effectiveness of SBP is highly influenced by model complexity. Although backward propagation accounts for a substantial portion of training cost across all NeRF variants, Sat-NeRF amplifies this effect by introducing additional prediction heads (sun visibility, sky color, and uncertainty), increasing both gradient diversity and redundancy; SBP therefore targets a larger portion of low-impact backward computation in Sat-NeRF, resulting in greater relative energy savings compared to lighter variants.

At inference time, backward computation is absent, and efficiency is instead governed by forward-pass complexity. Here, model compression, being the dominant optimization strategy, directly reduces latency and energy on edge devices. Together, SBP for training and compression for inference form a unified, phase-and-energy-aware optimization pipeline spanning cloud training to edge deployment.

VI. Conclusion

In this paper, we present Eco-NeRF, an energy-aware NeRF training and deployment pipeline that targets the dominant sources of computational and energy cost in NeRFs. By using SBP, Eco-NeRF reduces training-time energy, while model compression leads to reduced energy for edge inference. Across multiple scenes and model variants, our results demonstrate consistent energy savings, highlighting the need for an energy-aware lifecycle for NeRF models.

Future work will extend our evaluation to diverse geographic regions, additional GPU architectures, and alternative compression strategies (e.g., pruning and quantization), as well as explore adaptive feedback-driven SBP policies that dynamically balance energy and accuracy.

References

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, 2021.

[2] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[3] R. Marí, G. Facciolo, and T. Ehret, "Sat-NeRF: Learning Multi-View Satellite Photogrammetry With Transient Objects and Shadow Modeling Using RPC Cameras," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022.

[4] D. Derksen and D. Izzo, "Shadow Neural Radiance Fields for Multi-view Satellite Photogrammetry," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021.

[5] D. Chakraborty, K. Ghosh, Z. Sukma, I. K. Kim, L. Ramaswamy, S. M. Bhandarkar, and D. R. Mishra, "An Empirical Evaluation of the Impact of Solar Correction in NeRFs for Satellite Imagery," in *27th International Conference on Pattern Recognition (ICPR)*, 2024.

[6] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding," *ACM Transactions on Graphics*, vol. 41, no. 4, pp. 1–15, 2022.

[7] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "TensoRF: Tensorial Radiance Fields," in *European Conference on Computer Vision (ECCV)*, 2022.

[8] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance Fields without Neural Networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[9] T. Hu, S. Liu, Y. Chen, T. Shen, and J. Jia, "EfficientNeRF: Efficient Neural Radiance Fields," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[10] W. Zhang, R. Xing, Y. Zeng, Y.-S. Liu, K. Shi, and Z. Han, "Fast learning radiance fields by shooting much fewer rays," *IEEE Transactions on Image Processing*, vol. 32, pp. 2703–2718, 2023.

[11] Z. R. Lim Sy, S. Luo, J. Flaherty, and T. C. Thang, "Towards Green NeRF: an Exploration of Energy Influence Factors in NeRF Models," in *2024 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)*, 2024.

[12] A. H. Jiang, D. L. Wong, G. Zhou, D. G. Andersen, J. Dean, G. R. Ganger, G. Joshi, M. Kaminsky, M. Kozuch, Z. C. Lipton, and P. Pillai, "Accelerating Deep Learning by Focusing on the Biggest Losers," *CoRR*, vol. abs/1910.00762, 2019.

[13] J. Korhonen, G. Rangu, H. R. Tavakoli, and J. Kannala, "Efficient NeRF Optimization – Not All Samples Remain Equally Hard," in *European Conference on Computer Vision (ECCV)*, 2024.

[14] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[15] C. Sun, M. Sun, and H.-T. Chen, "Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[16] Z. Li, Y. Ma, J. Zhou, and P. Zhou, "Spiking-NeRF: Spiking Neural Network for Energy-Efficient Neural Rendering," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 20, no. 3, pp. 10:1–10:23, 2024.

[17] X. Yao, Q. Hu, F. Zhou, T. Liu, Z. Mo, Z. Zhu, Z. Zhuge, and J. Cheng, "SpiNeRF: direct-trained spiking neural networks for efficient neural radiance field rendering," *Frontiers in Neuroscience*, vol. 19, 2025.

[18] C. Li, S. Li, Y. Zhao, W. Zhu, and Y. Lin, "RT-NeRF: Real-Time On-Device Neural Radiance Fields Towards Immersive AR/VR Rendering," in *The 41st IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2022.

[19] B. L. Saux, N. Yokoya, R. Hänsch, and M. Brown, "Data fusion contest 2019 (dfc2019)," 2019. [Online]. Available: https://dx.doi.org/10.21227/c6tm-vw12

[20] T. Feng, Y. Zhai, J. Yang, J. Liang, D.-P. Fan, J. Zhang, L. Shao, and D. Tao, "IC9600: A Benchmark Dataset for Automatic Image Complexity Assessment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 01, pp. 1–17, 2023.

[21] B. Courty, V. Schmidt, S. Luccioni, Goyal-Kamal, M. Coutarel, B. Feld, J. Lecourt, L. Connell, A. Saboni, Inimaz, supatomic, M. Leval, L. Blanche, A. Cruveiller, ouminasara, F. Zhao, A. Joshi, A. Bogroff, H. de Lavoreille, N. Laskaris, E. Abati, D. Blank, Z. Wang, A. Catovic, M. Alencon, M. Stechly, C. Bauer, L. O. N. de Araujo, JPW, and MinervaBooks, "mlco2/codecarbon: v2.4.1," May 2024. [Online]. Available: https://doi.org/10.5281/zenodo.11171501

[22] L. F. W. Anthony, B. Kanding, and R. Selvan, "Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models," *CoRR*, vol. abs/2007.03051, 2020.