Fusion-3D: Integrated Acceleration for Instant 3D Reconstruction and Real-Time Rendering

Sixu Li, Yang Zhao, Chaojian Li, Bowei Guo, Jingqun Zhang, Wenbo Zhu,
Zhifan Ye, Cheng Wan, Yingyan (Celine) Lin
School of Computer Science
Georgia Institute of Technology
Atlanta, GA, USA
{sli941, celine.lin}@gatech.edu

2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO) | 979-8-3503-5057-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/MICRO61859.2024.00016

***Abstract*—Recent breakthroughs in Neural Radiance Field (NeRF) based 3D reconstruction and rendering have spurred the possibility of immersive experiences in augmented and virtual reality (AR/VR). However, current NeRF acceleration techniques are still inadequate for real-world AR/VR applications due to: 1) the lack of end-to-end pipeline acceleration support, which causes impractical off-chip bandwidth demands for edge devices, and 2) limited scalability in handling large-scale scenes. To tackle these limitations, we have developed an end-to-end, scalable 3D acceleration framework called Fusion-3D, capable of instant scene reconstruction and real-time rendering. Fusion-3D achieves these goals through two key innovations: 1) an optimized end-to-end processor for all three stages of the NeRF pipeline, featuring dynamic scheduling and hardware-aware sampling in the first stage, and a shared, reconfigurable pipeline with mixed-precision arithmetic in the second and third stages; 2) a multi-chip architecture for handling large-scale scenes, integrating a three-level hierarchical tiling scheme that minimizes inter-chip communication and balances workloads across chips. Extensive experiments validate the effectiveness of Fusion-3D in facilitating real-time, energy-efficient 3D reconstruction and rendering. Specifically, we tape out a prototype chip in 28nm CMOS to evaluate the effectiveness of the proposed end-to-end processor. Extensive simulation based on the on-silicon measurements demonstrates a 2.5× and 6× throughput improvement in training and inference, respectively, compared to state-of-the-art accelerators. Furthermore, to assess the multi-chip architecture, we integrate four chips into a single PCB as a prototype. Further simulation results show that the multi-chip system achieves a 7.3× and 6.5× throughput improvement in training and inference, respectively, over the Nvidia 2080Ti GPU. To the best of our knowledge, Fusion-3D is the first to achieve both instant (≤ 2 seconds) 3D reconstruction and real-time (≥ 30 FPS) rendering, while only requiring the bandwidth of the most commonly used USB port (0.625 GB/s, 5 Gbps) in edge devices for off-chip communication.**

***Index Terms*—Neural Rendering, VLSI, Accelerator.**

I. Introduction

3D reconstruction from sparsely sampled 2D images of a scene is a foundational task in numerous augmented and virtual reality (AR/VR) applications, as shown in Fig. 1 [48]. Neural Radiance Fields (NeRFs) have emerged as the state-of-the-art (SOTA) method for 3D reconstruction, thanks to their photorealistic rendering quality [28], [45]. Another advantage of NeRFs is their relatively small storage footprint, approximately 10 MB of parameters [13], [28], which is notably smaller than traditional methods such as point cloud

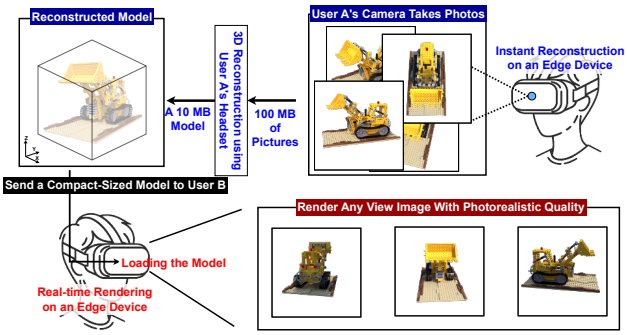


Fig. 1. Illustrating NeRF-based 3D reconstruction and an application scenario.

reconstructions. This efficiency not only reduces communication bandwidth in latency-sensitive applications but is also advantageous in scenarios with unstable network connections. Moreover, real-world 3D reconstruction applications often demand high performance, including instant training (reconstruction) within 2 seconds [22], [29], [32] and real-time inference (rendering) at a minimum of 30 FPS [47], [51], which are essential for immersive experiences like virtual telepresence [2]. Given the low storage needs of NeRFs [28] and stringent performance requirements, there is a trend towards conducting NeRF training and inference at the edge [6], [22] to achieve lower latency and conserve network bandwidth.

Despite the growing demand for 3D reconstruction at the edge, existing commercial edge devices, e.g., the NVIDIA Xavier NX embedded system [33], still struggle to achieve the aforementioned requirements of instant reconstruction and real-time rendering. Currently, these capabilities are primarily confined to high-end GPUs, like the NVIDIA RTX 3090 [5], [31]. To bridge this gap, recent studies [10], [13], [16], [18], [22], [30] have developed dedicated accelerators, aiming to enable instant reconstruction and real-time rendering on edge devices. These works propose tailored acceleration methods to overcome the execution bottlenecks, advancing the potential of edge 3D reconstruction solutions for real-world applications.

Despite existing NeRF accelerators' promises, they still fall short in meeting two critical requirements imposed by real-world deployments: 1) the practical off-chip bandwidth demand and 2) the efficient scaling-up strategy to support

979-8-3503-5057-9/24/$31.00 ©2024 IEEE
DOI 10.1109/MICRO61859.2024.00016

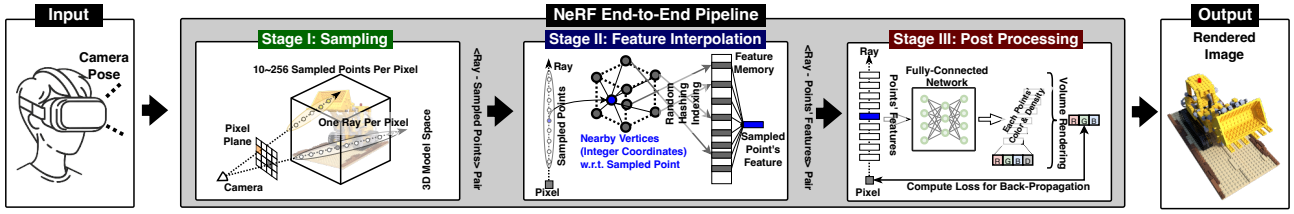


Fig. 2. A visualization of the NeRF pipeline [31], including three stages: Stage I - sampling, Stage II - feature interpolation, and Stage III - post processing.

large-scale scenes [38], [52]. To understand the first requirement, let's examine the NeRF pipeline, which typically involves three stages: Sampling, Feature Interpolation, and Post-Processing (see Sec. II-A). Existing NeRF accelerators focus on accelerating the dominant operations within one or two stages, relying on the host processor for the remaining stages. This approach results in substantial inter-stage communication, often leading to full utilization or even exceeding the practical off-chip bandwidth limitations. For example, [22] and [30] assume off-chip DRAM bandwidths of 59 GB/s and 231 GB/s, respectively. These configurations exceed the bandwidth capacities typically found in the latest edge devices, where the most commonly utilized interface is the 5 Gbps (0.625 GB/s) USB port [27], [33]. Additionally, real-world applications often involve large-scale scenes, e.g., outdoor environments, rather than single objects [1]. Previous solutions for enhancing performance in larger scenes typically involve scaling up the accelerator by adding more cores [16], [18]. However, this incurs higher chip fabrication costs [15] and poses risks of reduced yield due to larger chip areas [9]. Furthermore, increasing the size of the accelerator exacerbates the off-chip memory bandwidth issues and thus hampers its ability to achieve both low latency and high throughput, which is essential for real-time applications.

To overcome these challenges, we developed Fusion-3D, which involves an end-to-end NeRF accelerator designed to minimize off-chip communication and eliminate the external memory access during inter-stage data transfers in the NeRF pipeline and a multi-chip architecture as a more efficient alternative to larger single accelerators for handling 3D reconstruction and rendering of large-scale scenes. It is worth noting that the aforementioned challenges cannot be effectively addressed by simply adopting the end-to-end pipeline paired with the multi-chip architecture. The main reasons for this include inefficient sampling (C1), high precision requirements (C2), heavy chip-to-chip communication (C3), and workload imbalance between chips (C4), as detailed in Sec. II (here we use C1/2/3/4 for challenges and T1/2/3/4 for corresponding techniques). To address these challenges within the designs of the end-to-end pipeline and multi-chip architecture, we propose dedicated techniques in Sec. IV and Sec. V.

In summary, our contributions are as follows:

- We present Fusion-3D, which, to the best of our knowledge, is the first accelerator that simultaneously supports instant reconstruction (i.e., training) and real-time rendering (i.e., inference). It operates within the practical off-chip bandwidth limits of the most commonly used 5 Gbps USB port in edge devices and incorporates an efficient strategy to scale up for handling large-scale scenes.
- We have developed a single-chip NeRF accelerator, the first designed for end-to-end acceleration across all stages of the NeRF pipeline. This accelerator integrates two key techniques: 1) model normalization and partitioning combined with dynamic workload scheduling, which streamline complex computations and balance workloads in Stage I (i.e., Sampling of NeRF processing); and 2) a shared, mixed-precision pipeline that enhances computing unit utilization and reduces numeric conversion overheads. To validate the effectiveness of these proposed techniques, a prototype chip has been fabricated in 28nm CMOS.
- Leveraging the aforementioned accelerator, we have co-designed a multi-chip system to handle large-scale scenes, offering a more efficient alternative to increasing chip size. This system features a three-level hierarchical tiling: 1) mixture-of-expert-based multi-chip tiling for spatial scaling with minimized chip-to-chip communication; and 2) two-level hash tiling to eliminate irregular memory accesses and conflicts, ensuring balanced chip-to-chip workloads. We have implemented a proof-of-concept by integrating four chips onto a PCB board.
- Experiments and ablation studies validate the effectiveness of our Fusion-3D. Specifically, the single-chip accelerator achieves a 3.3× energy saving and a 2.5× speedup in training, along with an 18.6× energy saving and a 6× speedup in inference, compared to the latest NeRF solutions. The multi-chip system demonstrates a 304× energy saving and a 7.3× speedup in training, as well as a 270× energy saving and a 6.5× speedup in inference when compared against the Nvidia RTX 2080 Ti GPU.

II. Background, Motivations, and Challenges

A. Background of NeRFs

The SOTA NeRF pipeline, shown in Fig. 2 [31], comprises three stages: Stage I: the NeRF algorithm generates rays for each pixel on the image to be rendered, and then samples 3D points in the 3D model space along these pixel rays. The number of sampled points can either be small (e.g., 4, 5) or large (e.g., 128, 144, 255). Subsequent to sampling, an occupancy grid is employed to filter out points in empty spaces, reducing the computational demand for the other two stages. In other words, only the points within non-zero occupancy grids are stored and processed. Thus, the occupancy grid naturally serves as a "gating function" for our multi-chip architecture, a discovery first made by this work (see

TABLE I
OFF-CHIP BANDWIDTH REQUIREMENTS OF EXISTING NERF ACCELERATORS VERSUS OFF-CHIP BANDWIDTH AVAILABLE ON THREE COMMERCIAL EDGE PLATFORMS

Platform	Support Training	Off-Chip Connection Type	Bandwidth
Prior Accelerators			
RT-NeRF (Edge) [18]	No	LPDDR4-1600	17 GB/s
Gen-NeRF [10]	No	LPDDR4-2400	17.8 GB/s
NeuRex (Edge) [16]	No	LPDDR4-3200	25.6 GB/s[1]
Instant-3D [22]	Yes (Instant)	LPDDR4-1866	59.7 GB/s
NGPC [30]	No	GDDR6X	231 GB/s
RT-NeRF (Server) [18]	No	HBM2	510 GB/s
NeuRex (Server) [16]	No	HBM2	256 GB/s[1]
SOTA Edge Platforms			
Nvidia XNX [33]		USB 3.2 Gen 1	0.625 GB/s[2]
Meta Quest 2/3/Pro [27]		USB 3.2 Gen 1	0.625 GB/s[2]
Samsung S24 Ultra [41]		USB 3.2 Gen 1	0.625 GB/s[2]
This Work			
This Work	**Yes (Instant)**	**USB 3.2 Gen 1**	**0.6 GB/s**

[1] : Detailed numbers not reported in the paper, estimated based on the used memory type (DDR4-3200, HBM2).
[2] : Available bandwidth for connecting to dedicated accelerators.

Sec. V-A). Stage II: The feature interpolation stage extracts features for each sampled 3D point, where each point will be assigned features across different levels with distinct grid resolutions. During this process, the target point's features are interpolated from the eight nearby vertices. Stage III: After extracting features for each 3D point, an MLP is used to determine the density and color for all points. Subsequently, the renderer performs volumetric rendering to integrate these attributes along each ray, resulting in a pixel.

B. Motivation 1: Bandwidth Limitation of Edge AR/VR Calls for an End-to-End Accelerator

As shown in Fig. 3, training a NeRF model to an acceptable quality (e.g., 25 PSNR [24], [49]) involves a total of 155 GB intermediate data volume. This indicates a substantial memory bandwidth requirement for a 2-second instant training process: around 12.5 GB/s for data exchange among the three stages and 77.5 GB/s for off-chip data transfers within the three stages. However, the off-chip bandwidths of typical edge devices do not meet these requirements. Tab. I lists the bandwidth requirements reported in previous accelerators alongside the available bandwidth for dedicated accelerators when integrated into SOTA edge platforms [27], [33], [41]. This bandwidth limitation prevents the seamless integration of NeRF accelerators into existing systems without costly

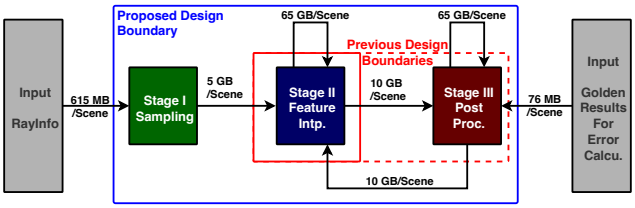


Fig. 3. Illustrating the data volume for the three stages in the NeRF pipeline during training and the design boundaries covering different stages. The data was obtained using Instant-NGP [31] on the NeRF-Synthetic Dataset [28]. Unlike prior works that cover varying numbers of stages, our work encompasses all stages to minimize inter-stage communication.

TABLE II
RENDERING QUALITY WITH QUANTIZED TRAINING MODELS

Quantization Frequency[1]	Never	1000 Iter.	200 Iter	Every Iter.
PSNR (INT8)[2]	31.7	30.1 (↓1.6)	26.0 (↓5.7)	Not Convergent

[1] : Quantize all the weights after every N iteration.
[2] : The exact PSNR number may change on different termination conditions. The reported number is measured with a total iteration of 5000 on the NeRF-Synthetic Dataset [28], averaged over all eight scenes of the dataset.

hardware upgrades (e.g., adding additional high-bandwidth memories). Moreover, allocating less memory bandwidth to these accelerators would compromise their ability to achieve instant training or real-time rendering, thus failing to deliver the immersive experience they are designed for.

However, as indicated in Fig. 3, only 700 MB of data transfer is required for the input and output of the entire pipeline. Thus, to address the above issue of impractical bandwidth requirement, we propose an end-to-end NeRF accelerator with heterogeneous modules for each stage, allowing all necessary computations to be completed on-chip. This minimizes the need for large volumes of off-chip data exchanges for partial sums, as outlined in the blue box of Fig. 3. Additionally, by organizing the three stages in a pipeline, our accelerator further enhances its performance.

C. Challenges in Designing an End-to-End Accelerator

Challenge C1: Inefficient Sampling. As discussed in Sec. II-B, the impractical off-chip bandwidth requirements necessitate accelerator solutions to support the end-to-end NeRF pipeline. Although Stages II and III in Fig. 3 dominate the overall latency of the NeRF pipeline [16], [18], [22], [30], once they are properly accelerated, e.g., by over 10× with dedicated optimization [16], [22], the workload associated with Stage I will become the new bottleneck. This necessitates dedicated hardware optimizations for Stage I to achieve efficient sampling and thus keep pace with the other two stages, as corroborated by [21].

Challenge C2: Distinct Workload Patterns and Data Precisions in Training and Inference. Previous research on NeRF accelerators for inference [16], [18], [30] and training [22] has indicated the necessity for both real-time (>30 FPS) inference and instant (<2 s) training in edge AR/VR systems. However, the workload patterns during inference and training can differ significantly. For instance, during inference, Stage II primarily aggregates features, while in training, it focuses on the distribution of gradients. Moreover, NeRF training demands higher precision and relies on floating-point operations. Tab. II illustrates that quantized training can nontrivially hurt model quality, based on experiments with Instant-NGP [31]. The requirement to support both training and inference, along with their unique workload patterns and data precisions, poses a significant challenge: how to efficiently accelerate both processes within a single system.

D. Motivation 2: Scalability Limitation of Prior NeRF Accelerators Calls for a Multi-Chip System

To accelerate 3D reconstruction and rendering for large real-world scenes, or to meet higher performance requirements,

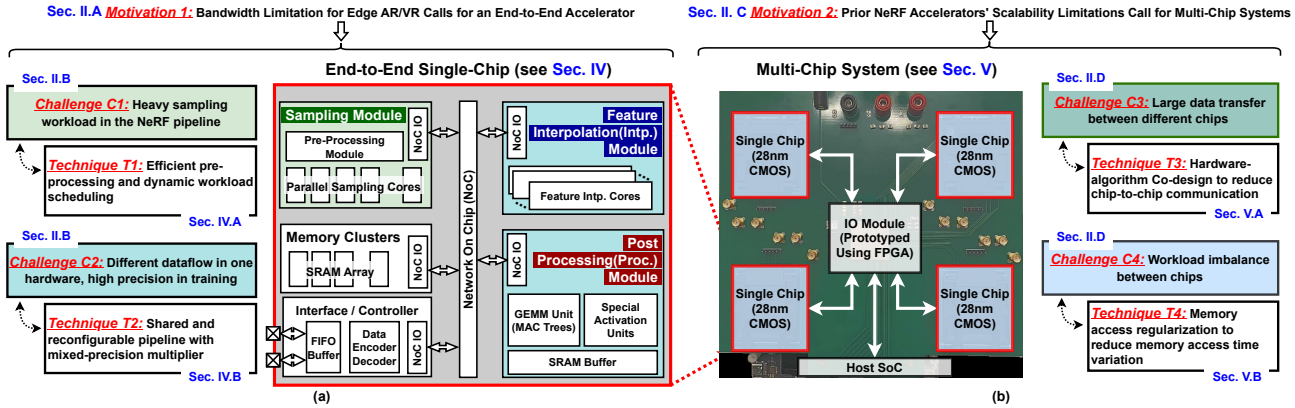


Fig. 4. Overview of the proposed Fusion-3D system: (a) the block diagram of our single-chip accelerator and (b) our multi-chip system.

spatially scaling up an accelerator by adding more cores is commonly adopted, as observed in [16], [18], [22]. However, the increased area reduces manufacturing yield and raises tapeout costs. For instance, the yield decreases from 99% to 72% for RT-NeRF [18] based on the yield model in [9], which results in a doubling of the cost per unit area. Additionally, scaling up the chip area limits flexibility in adapting to the diverse core number requirements of different devices, such as high-end and middle/low-end AR/VR devices. These challenges are further exacerbated by off-chip bandwidth limitations, as discussed in Sec. II-B. For example, there is an increase of 20-fold in memory bandwidth requirements when scaling up RT-NeRF [18] and NeuRex [16]. For example, RT-NeRF (Server) needs 510 GB/s of bandwidth but RT-NeRF (Edge) only requires 17 GB/s, as detailed in Tab. I.

In this work, we explore an orthogonal approach to scaling up by using multiple chips. Such a design can reduce chip fabrication costs, but it introduces challenges associated with managing higher-cost chip-to-chip communication [25], [37], [43], [53]. To address this, we propose algorithm-hardware co-design techniques to minimize chip-to-chip communication volume and prevent workload imbalance across chips.

E. Challanges in Designing the Multi-chip System

Challenge C3: Heavy Chip-to-Chip Communication. Given the trend toward larger neural networks/models, multi-chip systems have become a cost-effective alternative to scaling up chip area, as discussed in [9], [42]. Despite the advantages, these systems may suffer from performance degradation due to the latency in chip-to-chip communication and the limited bandwidth available for such interactions [12]. To this end, we introduce an algorithm-hardware co-designed approach (see Sec. V-A). Specifically, we are the first to exploit the built-in gating function of NeRF (i.e., the occupancy grid as mentioned in Sec. II-A) to build a Mixture-of-Experts (MoE) NeRF model that reduces the need for intermediate data communication among chips, thus largely reducing the chip-to-chip communication volume.

Challenge C4: Workload Imbalance among Chips. Another challenge in a multi-chip system is workload balancing. All chips must complete their jobs before the aggregated result can be derived. Consequently, if one chip's runtime is significantly longer than others, it creates a bottleneck for the entire system. We propose a method to regularize memory accesses, as detailed in Sec. V-B. This method specifically targets the mitigation of memory access conflicts, which often lead to varied runtimes due to the differences in access times. By diminishing these conflicts, we effectively ensure a more balanced runtime across all chips.

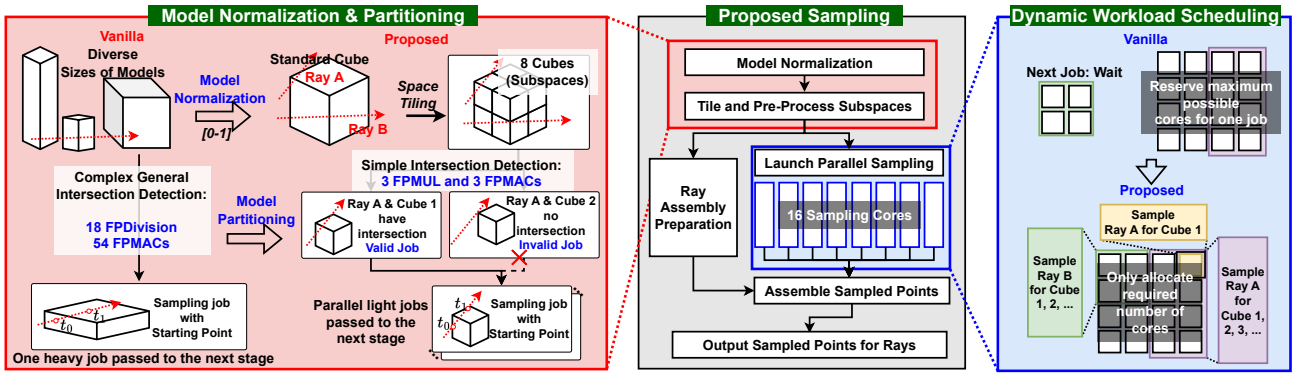
III. System Overview

Fig. 4 shows an overview of the proposed Fusion-3D system, comprising two levels of hierarchy: Single-Chip and Multi-Chip. This section briefly introduces these two levels and then provides their details in Sec. IV and Sec. V, respectively.

A. Overview of the Fusion-3D's Single-Chip Accelerator

Fusion-3D's single-chip accelerator is designed to address the impractical bandwidth requirement limitations of existing NeRF acceleration, as discussed in Sec. II-B. To achieve this, it offers end-to-end acceleration for all three stages of the NeRF pipeline and addresses the challenges associated with end-to-end acceleration. As shown in Fig. 4(a), the single-chip architecture comprises six major components: three computing modules and three supporting modules, introduced below:

- **1) The Sampling Module** is designed for processing Stage I of the NeRF pipeline. Here, we address **Challenge C1:** the heavy sampling workload in this stage. We develop **Technique T1:** efficient pre-processing and dynamic workload scheduling to mitigate this challenge, as detailed in Sec. IV-A. This module comprises a pre-processing unit and 16 parallel sampling cores.
- **2) The Feature Interpolation (Intp.) Module** and **3) the Post Processing (Proc.) Module** are designed to execute Stages II and III of the NeRF pipeline, respectively. In these stages, we identify **Challenge C2:** the need for efficiently supporting both training and inference within a single hardware system and the requirements for high precision during training. To address this, we propose **Technique T2:** a shared and reconfigurable pipeline with a mixed-precision multiplier, as detailed in Sec. IV-B.



(a) (b) (c)

Fig. 5. The proposed sampling optimizations include 1) Model Normalization & Partitioning, and 2) Dynamic Workload Scheduling. The following are the key aspects: (a) the Model Normalization & Partitioning pipeline and a visualization of the associated computation cost savings; (b) an overview of the proposed sampling pipeline; and (c) a comparison between the proposed Dynamic Workload Scheduling and the baseline, demonstrating that more cores can be utilized instead of remaining idle with our technique.

- **4) The Memory Clusters** serve as shared memory spaces for the three computing modules mentioned above. Equipped with multiple SRAM arrays, these clusters enable simultaneous access by all computing modules. Software-configurable connections between the arrays and computing cores facilitate a ping-pong buffer mechanism, further enhancing acceleration performance. **5) The Interface/Controller Module** is responsible for top-level control, managing data transmission and reception to and from external sources. **6) The Network-on-Chip (NoC) Module** acts as the connectivity layer, interlinking all the aforementioned modules for efficient communication.

B. Overview of Fusion-3D's Multi-Chip Accelerator

To enhance scalability as discussed in Sec. II-D, we propose a multi-chip system for handling larger scenes, as depicted in Fig. 4(b). In this multi-chip system, we identify **Challenge C3:** the latency and high energy consumption associated with data transfer between chips. To mitigate this, we introduce **Technique T3:** a hardware-algorithm co-designed multi-chip collaboration scheme, detailed in Sec. V-A, aimed at reducing inter-chip communication volume. Additionally, within the multi-chip system, we address **Challenge C4:** workload imbalance caused by irregular memory accesses, which can adversely affect system-level performance. Our response is **Technique T4:** a memory access regularization strategy, described in Sec. V-B, aiming to even out memory access variations and balance the workload across chips.

IV. Fusion-3D: Single-Chip End-to-End Acceleration

A. *Technique T1: Optimized Sampling: Model Normalization & Partitioning and Dynamic Workload Scheduling*

*1) **Technique T1-1:** Model Normalization & Partitioning:* In the sampling stage, the initial task is to identify intersection points between a ray and the 3D model bounding box, thereby establishing the sampling start (t_0) and end (t_1) points. The diverse and irregular shapes of objects/scenes result in bounding boxes of varying dimensions across 3D space. Consequently, detecting ray-model intersections is computationally demanding, involving solving six linear equations (requiring 18 divisions, 54 multiplications, and 54 additions [26]). This complexity, coupled with the limited computational capacity of AR/VR devices, hinders the acceleration of this and subsequent steps. To reduce the computational complexity in detecting start (t_0) and end (t_1) points, we propose the Model Normalization & Partitioning technique. This technique first normalizes the model region into a standard cube with coordinates ranging from [0,0,0] to [1,1,1] as depicted in the upper part of Fig. 5(a). As a result, the starting (t_0) and end (t_1) points can be efficiently calculated by solving a much-simplified equation, thanks to the fixed bounding box. Specifically, the model normalization helps to fix some of the parameters in the linear equations, allowing us to simplify the equations to solve t_0 and t_1. This simplified calculation only needs 3 multiplication and 3 Multiply-Accumulate (MAC) operations [36], significantly reducing the computational burden. Consequently, our NeRF accelerator can compute multiple intersections simultaneously without requiring more computation resources than the standard pipeline. To parallelize the sampling process between the ray and the normalized space, we divide the whole space into eight cubes and perform the aforementioned starting and end point computation on the ray and these eight cubes, as shown in the lower part of Fig. 5(a). Only the ray-cube pairs with valid intersections proceed to the subsequent sampling cores, while those without intersections are discarded. As the first end-to-end work investigating the sampling process in NeRF, this work, for the first time, finds and emphasizes that model normalization is vital for end-to-end customized hardware as it can greatly reduce the computation in the first stage of the pipeline (as validated in Sec. VI-C) and simplify the design of corresponding hardware.

*2) **Technique T1-2:** Dynamic Workload Scheduling:* The valid ray-cube pairs continue to the multi-core sampling processor to determine the sampled points. However, the number of valid ray-cube pairs for one ray (e.g., 1∼3) and the number of sampled points for each ray-cube pair

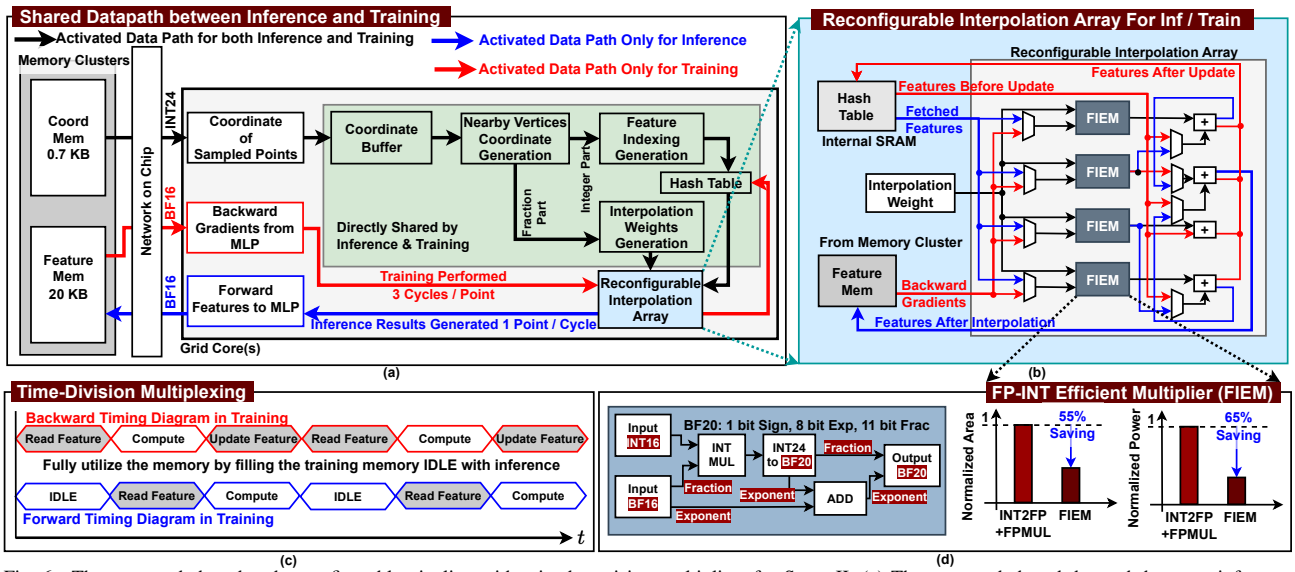


Fig. 6. The proposed shared and reconfigurable pipeline with mixed-precision multipliers for Stage II: (a) The proposed shared datapath between inference and training, which share most of the computation units, and a reconfigurable interpolation array is designed to switch between the two modes for computing unit sharing. (b) The schematic of the proposed reconfigurable interpolation array. (c) The time-division multiplexing between inference and training for the shared pipeline. (d) The schematic of the proposed FP-INT efficient multiplier, and its area & power saving.

(e.g., 3∼100) vary significantly. Employing a simple ray-by-ray or pair-by-pair workload scheduling approach can cause underutilization of the multiple sampling cores. Therefore, we introduce the Dynamic Workload Scheduling technique to enhance the utilization, as visualized in Fig. 5(c). This sampling controller continuously checks the availability of the sampling cores (see Fig. 4(a)). Once the number of available sampling cores exceeds the number of cores needed for computations corresponding to an entire ray, the controller dispatches all the ray-cube pairs in this ray for execution. The purpose of using this threshold, rather than launching execution as soon as a single core becomes available, is to strike a balance between the complexity of the control logic and the achievable performance. Additionally, this approach reduces the required intermediate buffer size by alleviating the need to store the partial sum associated with each ray. Hence, this dynamic scheduling technique ensures efficient utilization of sampling cores, enabling the processor with fewer stalls.

*B. **Technique T2:** Optimized Feature Interpolation and Post Processing - Shared and Reconfigurable Pipeline with Mixed-Precision Multiplier*

*1) **Technique T2-1:** Shared and Reconfigurable Datapath with Time-Division Multiplexing between Inference and Training:* As mentioned in **Challenge C2**, there is a need to support both inference and training. We address this need from three perspectives: 1) directly reusing hardware for both purposes, 2) reconfiguring hardware that cannot be directly reused, and 3) utilizing under-utilized hardware for inference tasks during training periods.

1) To achieve direct resource sharing between inference and training, we begin by examining their similarities. For Stage II, both inference and training require the computation of nearby vertices' coordinates, feature indices, and interpolation weights from input coordinates. Consequently, these hardware components—nearby vertices' coordinate generation, feature index computation, and interpolation weight determination—can be shared in both modes. This enhances area efficiency by eliminating redundant computing modules, as illustrated in Fig. 6(a). **2)** Next, let's examine the differing workloads between inference and training: In the forward pass, outputs of eight multiplications are accumulated, similar to an adder tree, whereas in the backward pass, results are added back to the original features, akin to an inverse adder tree, and stored in the feature SRAM. These differences are denoted by blue and red lines in Fig. 6(a), representing inference and training, respectively. In other words, inference and training share the same computation graph structure but with inverse edges. This symmetrical yet distinct workload leads to the proposed shared reconfigurable interpolation array, as shown in Fig. 6(b). This array enhances area efficiency by negating the need for separate hardware for inference and training. Functioning either as a MAC tree for the forward pass or a vector multiplication unit for the backward pass. **3)** After understanding the workloads between training and inference above, we investigate whether there is underutilized hardware during training that can be utilized to perform inference. Specifically, during training, the three-step feature updates—comprising one cycle each for reading, computing, and writing back to SRAM—create an IDLE slot in the feature memory during computation, which reduces system efficiency. To utilize this IDLE slot in memory effectively, we schedule an inference task with a training task, as depicted in Fig. 6(c).

*2) **Technique T2-2:** FP-INT Efficient Multiplier:* The experimental results in Tab. II confirm the necessity of floating-point representation for NeRF model training. However, certain operations, such as interpolation weight computation in

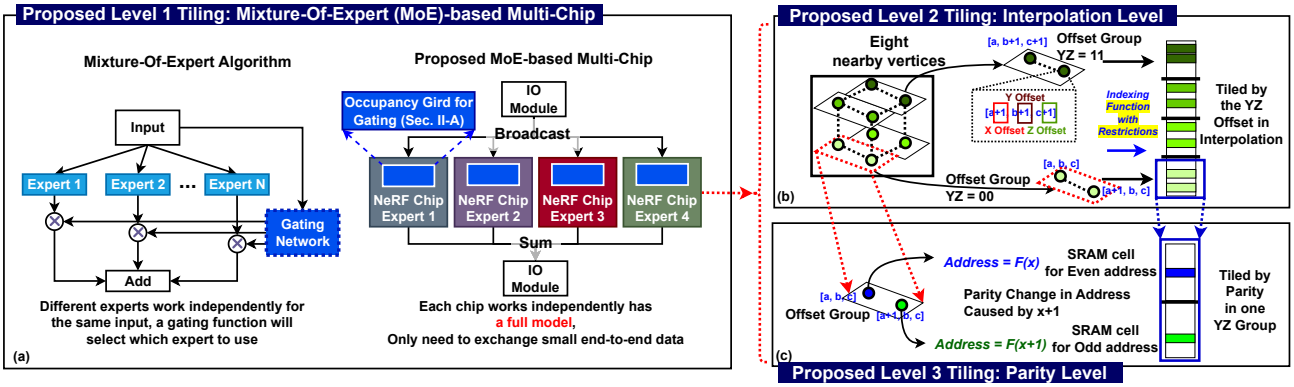


Fig. 7. Overview of the proposed hierarchical Multi-Chip design: (a) Level 1 of tiling: Input is broadcast to multiple chips, and a full inference/training process is performed in the spirit of MoE. Specifically, the pixel values from different chips are summed together to obtain the final pixel value. (b) Level 2 of tiling: the eight nearby vertices are grouped based on their YZ offset, and the feature table is tiled into four blocks. (c) Level 3 of tiling: the two vertices in each YZ group from the Level 2 tiling are separated based on the parity of the address. Data with addresses of different parity are stored in separate SRAM banks.

Stage II still utilize integers. The resulting mixed multiplications, combining integer and floating-point numbers, pose challenges. Traditionally, a design that includes an Integer-to-Floating Point (INT2FP) conversion unit followed by a complete Floating Point Multiplier (FPMUL) is used, which increases area/power overhead. To overcome this drawback, we develop an FP-INT Efficient Multiplier (FIEM). The FIEM module processes the fraction and exponent of the floating point input separately, first multiplying the fraction by the input integer and then combining this result with the exponent. This approach conserves area compared to using a full FPMUL. The module's schematic is shown in Fig. 6(d).

*3) **Ablation Study for Technique T2**:* Post-layout simulation results indicate that 87.4% of the area in Stage II is directly shared between inference and training, while 12.6% of the area is reused. Additionally, compared to the traditional approach of an INT2FP unit followed by a FPMUL, the FIEM achieves a 55% area reduction and a 65% power saving, as shown in Fig. 6(d).

V. Fusion-3D: Multi-Chip For Enhanced Scalability

A. *Technique T3: Multi-Chip Design with Mixture-Of-Expert*

To design a multi-chip system capable of handling large neural networks/models, a common approach involves mapping each layer to a specific chip or distributing several computationally intensive layers across multiple chips [12]. However, this strategy leads to significant data transfer demands for inter-layer and intra-layer communication, thereby increasing latency and energy costs associated with chip-to-chip communication, which can outweigh the benefits of parallelization across multiple chips. To reduce data transfer in multi-chip systems, we extend the Mixture-of-Experts (MoE) concept [7], [8], [14], [17], [39]. Typically, MoE utilizes small "experts" and a gating function for input-specific expert selection. Our method, referred to as "Level 1 Tiling" divides the entire model into complete, smaller models or "experts" and assigns each chip an expert, as illustrated in Fig. 7(a). Regarding the gating function in our proposed MoE method,

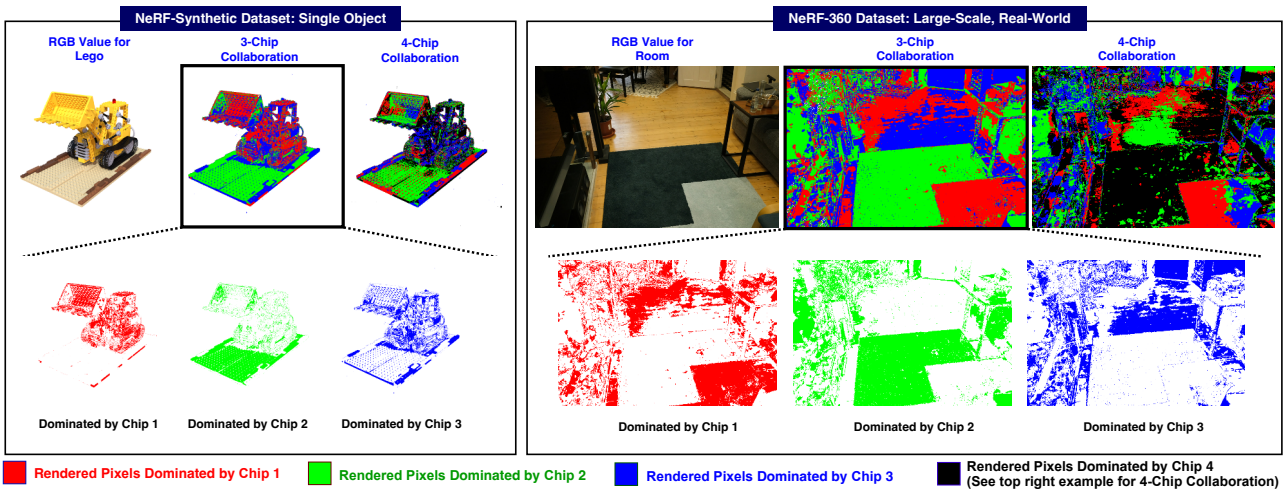


Fig. 8. Visualization of our MoE-based multi-chip design. Different region colors indicate the corresponding region features are mainly learned by different experts. Taking the NeRF-Synthetic tractor scene as an example, the left-bottom region is learned by Expert 2 (green), while the right-top region is learned by both Expert 1 (red) or Expert 3 (blue).

we use occupancy grids (as mentioned in Sec. II-A) as built-in gating functions within each expert (chip). Thus, we can simply use addition to fuse the outputs from all experts via an Input-Output (I/O) module. Through our MoE design, communication between chips during inference/training is greatly reduced (see Sec. VI-C). The workload distribution among chips is visually depicted in Fig. 8, highlighting how different experts tend to specialize, automatically dominating different pixels during the training process. Additionally, our MoE-based multi-chip system flexibly adapts to varying numbers of chips, as shown in the upper row of Fig. 8, where workload assignment automatically adjusts to accommodate the number of chips.

B. *Technique T4: Two-Level Hash Tiling*

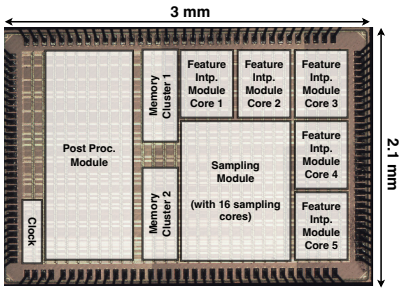
In the feature interpolation stage (i.e., Stage II), each sampled point requires eight memory accesses to retrieve features from its eight nearest vertices. This process can lead to bank conflicts because feature indexing functions, such as the random hash function in the SOTA design [31], may map nearby vertices to the same memory bank, causing conflicts during these accesses. These conflicts can cause memory access times for a single sampled point to vary from 1 (i.e., no conflict) to 8 cycles (i.e., all accesses target the same bank). This variability in runtime equivalently causes workload imbalances across chips, leading to the stalling of faster chips.

To tackle the memory access conflicts and, thus, variable runtimes, we propose a two-level hash tiling strategy to map features of vertices onto proper SRAM banks. The first level, termed "interpolation level tiling," is noted as "Level 2 Tiling" in conjunction with the previously mentioned "Level 1 Tiling" in Sec. V-A, as illustrated in Fig. 7(b). The eight nearest vertices of each sampled point are generated by offsetting the X, Y, and Z coordinates by one unit. Because the hash function applies larger factors to the Y and Z offsets [31], vertices with varying YZ offsets show a wider distribution in hash table addresses (average distance approximately 1/4 of the hash table size). Leveraging this, we partition the feature hash table into four groups, each characterized by a unique YZ offset and a dedicated SRAM group. To further mitigate memory conflicts, we utilize a property of the hash function: addresses that are offset by one unit along the X coordinate always exhibit opposite parities. To leverage this property, we allocate two SRAM banks for even and odd X address parities, respectively, thus resolving potential conflicts in vertex memory access within a YZ SRAM group. This configuration is illustrated in Fig. 7(c) and is referred to as "Level 3 Tiling," which is also noted as "parity level tiling."

VI. Experiments

A. *Evaluation of Fusion-3D's Single-Chip Accelerator*

Evaluation Methodology. To evaluate the performance and practical viability of Fusion-3D's end-to-end single-chip accelerator, we conduct two sets of evaluations. First, we tape out a silicon prototype chip and measure its performance to verify our design's functionality and to characterize its performance.



Process	28 nm HPC+ RF
Die Area	3 mm × 2.1 mm
Core Area	2.7 mm × 1.8 mm
Precision	BF16, FP20, INT16
Voltage	0.60 - 1.1 V
Frequency	200-750 MHz
On-Chip SRAM	659 KB
Power	170 - 1860 mW

(b)

	Sampling Module	Feature Intp. Core	Post Proc. Module	Memory Cluster
SRAM	243 KB	64 KB / Each	16 KB	40 KB / Each
Equivalent MACs	FP32-INT: 48 FP32: 12	BF16-INT: 16 INT16: 96	FP-INT: 48 BF16: 512	-

(c)

Fig. 9. The fabricated prototype for Fusion-3D's single-chip accelerator: (a) die photo, (b) measured results, and (c) resource breakdown for different modules.

Second, for a fair comparison with other NeRF accelerators, we design a scaled-up single-chip accelerator with resources similar to the baselines, for which we develop a cycle-accurate simulator based on the measured results of the prototype chip and validate through post-layout simulations. Finally, we evaluate this scaled-up accelerator against the baselines.

Details of Fusion-3D's Prototype Chip. Here are the implementation, chip configuration, and measurement of the single-chip accelerator prototype chip. **1) Chip implementation:** We implement the design in Verilog HDL, then adopt Cadence EDA tools [3], [4] to generate the layout, and finally tape out this silicon prototype chip in a commercial 28nm CMOS technology. Fig. 9(a) is the die photo. The spec table is shown in Fig. 9(b). **2) Chip configuration:** The prototype chip features a Sampling Module, a Feature Interpolation Module with five feature interpolation cores, one Post Processing Module, and two Memory Clusters. Their detailed configurations are summarized in Fig. 9(c). Additionally, their area and power breakdown are presented in Fig. 10(c). **3) Chip measurement set-up:** The chip measurement set-up is shown in Fig. 10(b). Specifically, the chip is mounted on a PCB board (see Fig. 10(a)); A power supplier provides the supply voltages for the chip and reports the current; Simultaneously, a clock generator supplies clock signals to the chip; Additionally, an FPGA board acts as an interface with the PC [54]. During measurement, we characterize the operating frequency under different chip supply voltage, the results are shown in Fig. 10(d). **4) Chip Measurement Results.** We evaluate the silicon-validated prototype chip across three metrics: **i)** We verify the chip's functionality by matching the chip results and the algorithm outputs, the results show a PSNR difference within 0.1. **ii)** We verify the chip's performance by measuring the start and end time for a specific task, results show that it can achieve 36 FPS in rendering or 1.8 s training at 600 MHz clock frequency separately. **iii)** We verify that the chip consumes 1.21 W of power when running at 600 MHz by multiplying the supply voltage and the average current read from the ammeter.

TABLE III
THE PROPOSED **SINGLE-CHIP** ACCELERATOR VS. SOTA NERF ACCELERATORS IN TERMS OF ACCELERATOR RESOURCES AND EFFICIENCY PERFORMANCE

	Nvidia Jetson Nano [35]	Nvidia Jetson XNX [33]	RT-NeRF ICCAD'22 [18]	Instant-3D ISCA'23 [22]	NeuRex ISCA'23 [16]	MetaVRain ISSCC'23 [13]	This Work
Silicon Prototype	No	No	No	No	No	Yes	**Yes**
Process	20 nm	12 nm	28 nm	28 nm	28 nm	28 nm	28 nm
Die Area	118 mm^2	350 mm^2	18.85 mm^2	6.8 mm^2	3.14 mm^2	20.25 mm^2	**8.7 mm^2**
Clock Frequency	900 MHz	1100 MHz	1000 MHz	800 MHz	1000 MHz	250 MHz	**600 MHz**
SRAM Size	2,500 KB	11,000 KB	3,500 KB	1,536 KB	884 KB	2,050 KB	**1,099 KB**
Core Voltage	N/A	N/A	1 V	1 V	N/A	0.95 V	**0.95 V**
NeRF Algorithm	Hash Grid	Hash Grid	Dense Grid	Hash Grid	Hash Grid	MLP	Hash Grid
Instant Training (<2s)	No	No	No	Yes	No	No	**Yes**
Real-time Inference (>30 FPS)	No	No	Yes	Yes	Yes	Yes[1]	**Yes**
End-to-End Train&Infer. Support	Yes	Yes	No	No	No	No	**Yes**
Inference Throughput[2]	2.5 M/s	12.5 M/s	288 M/s	N/R	112 M/s[4]	13.8 M/s	**591 M/s**
Training Throughput[2]	0.5 M/s	2.6 M/s	N/S	32 M/s	N/S	N/S	**199 M/s**
Inference Energy/Point[3]	192 nJ	486 nJ	27 nJ	N/R	41 nJ[4]	65 nJ	**2.5 nJ**
Training Energy/Point[3]	943 nJ	2357 nJ	N/S	59 nJ	N/S	N/S	**7.4 nJ**
Off-Chip Bandwidth	25.6 GB/s	59.7 GB/s	17 GB/s	59.7 GB/s	N/R	N/R	**0.6 GB/s**

[1] MetaVRain adopts image warping, so if there are more than 97% of the overlapped pixels between the current and last frames, it can achieve real-time inference.
[2] For throughput, we use *the no. of sampled points per second* following [13], 591 M/s means processing 591 million sampled points (from Stage I: sampling) per second.
[3] For energy, we use *energy consumption per sampled point* as per the standard in literature [13].
[4] Estimated based on the speedup and energy saving over [33] reported in [16], where only the result on one of the eight scenes in the NeRF-Synthetic dataset [28] is reported.
[5] N/A: Not Applicable, N/R: Not Reported and can not be computed based on the reported data, N/S: Not Supported

Details of Fusion-3D's Single-Chip Accelerator. Here, we provide details on the configuration, benchmark dataset, and evaluation standard of the single-chip accelerator: **1) Single-chip accelerator configuration:** To ensure a fair comparison with other NeRF accelerators (see Tab. III) that use larger on-chip resources than our prototype chip, we design a single-chip accelerator to includes five additional Feature Interpolation cores and three more Memory Clusters than the prototype chip. This scaled-up accelerator occupies an area of 8.7 mm^2, based on post-layout estimation. We then develop a cycle-accurate simulator to simulate and evaluate the performance of our single-chip accelerator. The simulator's technology-related parameters, such as memory access cost, are characterized by the prototype chip, and its functionality is validated through post-layout simulations. **2) Dataset and evaluation standard:** We use the NeRF-Synthetic dataset [28], commonly used as a normal-scale scene dataset (800×800 resolution), for evaluating the performance of both single-chip inference and training tasks. Because different NeRF accelerators may adopt different algorithms, we use inference/training quality on the same dataset (i.e., PSNR) as a unified standard for all the devices. Specifically, for NeRF training tasks, the runtime is measured when the training quality reaches 25 PSNR following prior works [22], which is recognized as "acceptable quality"; and for NeRF inference tasks, we use 30 PSNR as the standard following prior works [18], as this is recognized as good quality to human eyes [11], [19], [40].

Evaluation Results of Fusion-3D's Single-Chip Accelerator. We benchmark the single-chip accelerator against six baselines, including two SOTA edge GPUs [33], [35] and four SOTA NeRF accelerators [13], [16], [18], [22]. Tab. III summarizes their detailed performance comparison. First, the single-chip accelerator achieves 1.36× inference throughput speedup w.r.t. the best baseline [18] for inference and 4.15× training throughput speedup w.r.t. the best baseline [22] for training. In addition, compared with the best baselines in terms of inference [18] / training [22] throughput, our single-chip accelerator also has a 19×/25× energy efficiency improvement for inference/training, respectively. Finally, compared with the baseline accelerators [16], [22] adopting the same NeRF algorithm [31], our accelerator demonstrates 6× inference speedup w.r.t [16] and 2.5× training speedup w.r.t. [22].

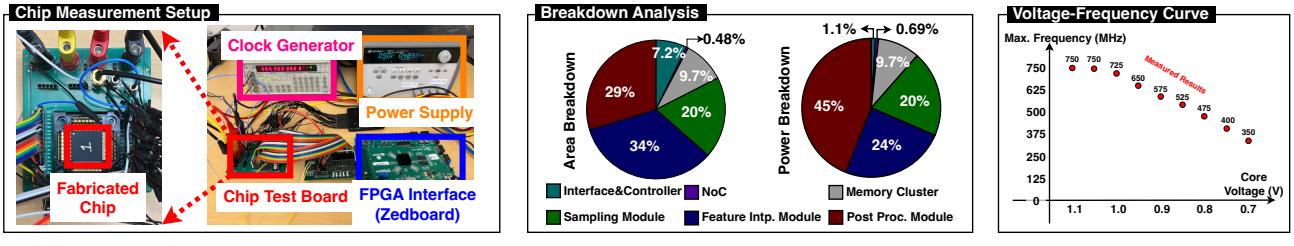


Fig. 10. Chip measurement setup and results: (a) The packaged chip is mounted onto the PCB board and uses a socket, (b) measurement set-up photo, (c) the area and power breakdown of the fabricated chip, and (d) the measured voltage-frequency curve of the fabricated chip.

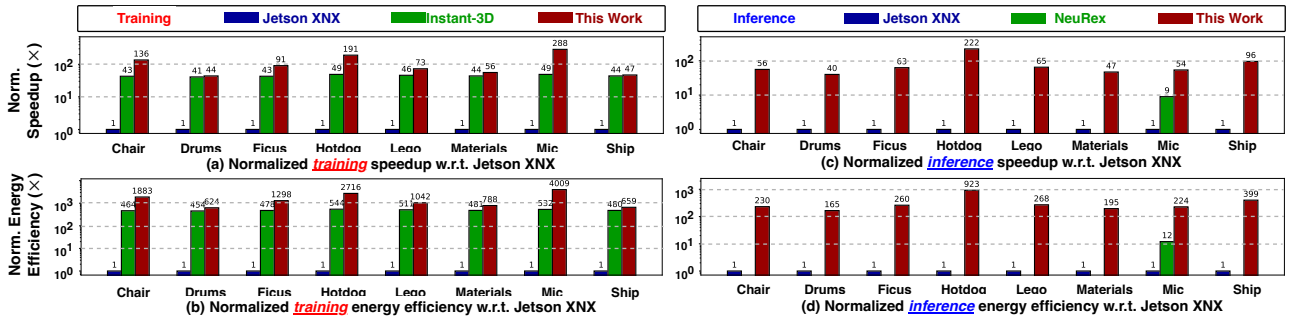


Fig. 11. The normalized speedup and energy efficiency of the single-chip accelerator and SOTA baseline devices on the eight scenes of the NeRF-synthetic dataset [28]. Note that Instant-3D [22] and NeuRex [16] are the two latest works, and [16] only provides the result on one of the eight scenes.

B. Evaluation of Fusion-3D's Multi-Chip System

Evaluation Methodology. To evaluate the multi-chip system, we first design a PCB board-based multi-chip prototype using an FPGA as the I/O module and integrating four prototype chips. This prototype is used to verify the functional correctness of the multi-chip system and to characterize the communication bandwidth supported by this low-end substrate, i.e., the PCB board. We then develop a cycle-accurate simulator to represent a multi-chip system, emulating the integration of four scaled-up single-chip accelerators on a PCB board. This setup enables us to make a fair comparison with the baselines.

Details of Fusion-3D's PCB-based Multi-Chip Prototype. We design the multi-chip prototype mentioned above, using an 8-layer PCB board (see Fig. 4(b)). This prototype incorporates an FPGA as the I/O module, which performs the gating function detailed in Sec. V-A; Additionally, four of the prototype chips (see Fig. 9(a)) are mounted on the PCB board and communicate with the I/O module. This multi-chip prototype's measurement setup is akin to that of the single-chip prototype shown in Fig. 10(b). We measure the PCB prototype using two metrics: **1)** We verify the functionality using the same protocol as the single-chip setting. **2)** We verify that the 0.6 GB/s bandwidth can be achieved by the low-cost PCB board.

Details of Fusion-3D's Multi-Chip System. 1) Multi-chip system configuration: For a fair comparison with the baseline cloud accelerators, we design a multi-chip system comprising four scaled-up single-chip accelerators as configured in Sec. VI-A, along with an I/O module. The I/O module runs the same functions as those on the FPGA I/O module in the multi-chip prototype. It is worth noting that the area and the SRAM size of the multi-chip system are not simply four times that of a single chip. These numbers account for both the four scaled-up single-chip accelerators and the I/O module, which introduces a 0.5% area overhead and a 2.3% SRAM size overhead in the I/O module. The I/O module is implemented using the same technology and follows the same synthesis, placement, and routing flow as the single-chip accelerators to ensure a fair comparison. We then design a cycle-accurate simulator to evaluate the multi-chip system. The simulator's I/O bandwidth is derived from the multi-chip PCB prototype, and its functionality is validated through post-layout simulations. **2) Dataset and evaluation standard:** We select the NeRF-360 dataset [1], widely recognized as a real-world large-scale scene dataset [52]. We set 25 PSNR as this unified standard for both large-scale training and inference tasks, referring to [38], [52].

TABLE IV
THE PROPOSED MULTI-CHIP SYSTEM VERSUS SOTA NERF ACCELERATORS IN TERMS OF ACCELERATOR RESOURCES AND ENERGY EFFICIENCY PERFORMANCE.

	Nvidia 2080Ti [34]	RT-NeRF-Cloud ICCAD'22 [18]	NeuRex-Server ISCA'23 [16]	This Work
Process	12 nm	28nm	28nm	28 nm
Die Area	754 mm^2	565 mm^2	21.37 mm^2	**35 mm^2**
Clock Frequency	1350 MHz	1000 MHz	1000 MHz	**600 MHz**
SRAM Size	27,394 KB	105,000 KB	4,644 KB	**4,500 KB**
Typical Power	250 W	240 W	6.1 W	**6.0 W**
Inference Throughput/Watt	0.4 M/s	34 M/s[1]	50 M/s[1]	**98.5 M/s**
Training Throughput/Watt	0.1 M/s	-	-	**33.2 M/s**
Off-Chip Bandwidth	616 GB/s	510 GB/s	512 GB/s[2]	**0.6 GB/s**

[1] Estimated based on the reported data on the reported dataset in the papers.
[2] Estimated based on the evaluation setup in [16].

TABLE V
SPEEDUP AND ENERGY SAVING ACHIEVED BY OUR PROPOSED MULTI-CHIP SYSTEM W.R.T. NVIDIA 2080TI [34] ON THE SEVEN SCENES OF NERF-360 [1] WITH THE SAME MODEL.

	bicycle	bonsai	counter	garden	kitchen	room	stump
Inference Speedup	9.2×	8.2×	6.1×	3.1	5.9×	7.3×	5.3×
Training Speedup	8.7×	8.8×	5.5×	6.7×	5.7×	7.1×	8.5×
Inference Energy Eff.	380×	342×	255×	128×	244×	302×	221×
Training Energy Eff.	359×	365×	229×	279×	236×	295×	351×

Evaluation Results of Fusion-3D's Multi-Chip System on Real-World Large Scale Scenes with Different Complexity. We benchmark the multi-chip system against a cloud GPU [34] and two SOTA cloud NeRF accelerators [16], [18], with a summary in Tab. IV. To align with the power constraints of AR/VR devices (around 8 W [27]), we compute throughput per watt for a fair comparison. Our system outperforms the best baseline [16] with a 1.97× improvement in inference throughput per watt and a 332× improvement in training throughput per watt compared to the cloud GPU [34]. Notably, we require only 0.6 GB/s off-chip bandwidth (plus 2.4 GB/s intra-system) to achieve this performance, two orders of magnitude less than the baselines. In additional comparisons with the cloud GPU baseline [34] on the NeRF-360 dataset [1] (see Tab. V), this work achieves up to 7.3× speedup and over 267.4× energy efficiency in training, demonstrating improvements in both speed and energy efficiency.

C. Ablation Study

Speedup Achieved by the Proposed Technique T1. To validate the effectiveness of the proposed technique T1 in Stage I, we conduct ablation studies to evaluate the speedup

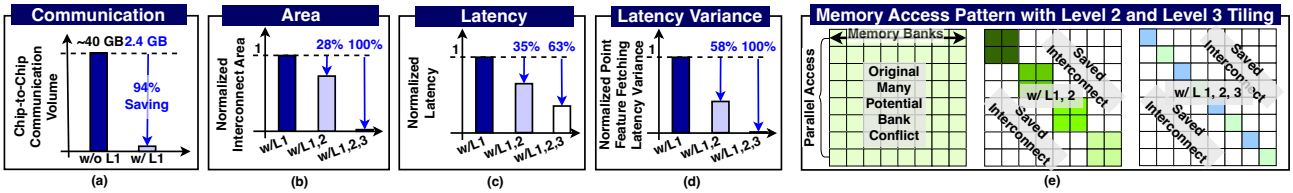


Fig. 12. Ablation study and visualization on techniques proposed in Sec. V: (a) Communication saving when adopting Level 1 Tiling, (b) Interconnect area saving when adopting Level 2 and Level 3 Tiling, (c) Latency saving for feature access when adopting Level 2 and Level 3 Tiling, (d) Feature fetching latency variance when adopting Level 2 and Level 3 Tiling, and (e) Memory access pattern when adopting Level 2 and Level 3 Tiling.

TABLE VI
ABLATION STUDY FOR THE SAMPLING MODULE ON THE EIGHT SCENES OF THE NERF-SYNTHERIC DATASET [28].

Dataset	Ship	Mic	Materials	Lego	Hotdog	Ficus	Drums	Chair
Speedup	5.4×	20.2×	10.6×	7.8×	7.3×	18.8×	14.4×	9.0×

attributable solely to technique T1. Following the single-chip evaluation methodology outlined in Section VI-A, technique T1 achieves a speedup ranging from 5.4× to 20.2× across various scenes, compared to naive sampling modules without the proposed technique, as detailed in Tab. VI.

Communication, Area, and Latency Savings Achieved by the Proposed Techniques T3 and T4. To validate the effectiveness of the proposed techniques T3 and T4, we conduct a series of ablation studies within the same settings described in Sec. VI-B. As depicted in Fig. 12(a), the communication volume across chips is reduced by 94% using the MoE-based multi-chip design. Owing to the regularized memory access, the need for a crossbar-like memory access unit is eliminated. Instead, a straightforward one-to-one connection is implemented, leading to significant area and latency savings as illustrated in Fig. 12(b) and (c). Furthermore, the reduction in randomness results in the variance of the feature fetching latency becoming zero, as indicated in Fig. 12(d), suggesting a more balanced latency. Lastly, the memory access pattern is depicted in Fig. 12(e), where, with the proposed level 2 and 3 tiling, each access is aligned with a single memory bank, effectively eliminating all access conflicts.

PSNR and Convergence Comparison Between the Proposed Technique T3 and the Original Single Large Model. To examine the impact on PSNR and convergence of the proposed technique T3, we conduct experiments comparing PSNR versus training time. These experiments involve various configurations of small models within the MoE model against the original single large model. As illustrated in Fig. 13(a), the MoE model, comprising four small models, is capable of achieving a PSNR comparable to that of the large model within the same training time, thereby matching the convergence speed. Notably, each small model in the MoE configuration has a hash table size of 2^{14}, whereas the large model has a hash table size of 2^{16}. Furthermore, it is observed that the convergent PSNR improves as the number of small models (i.e., the number of chips) increases.

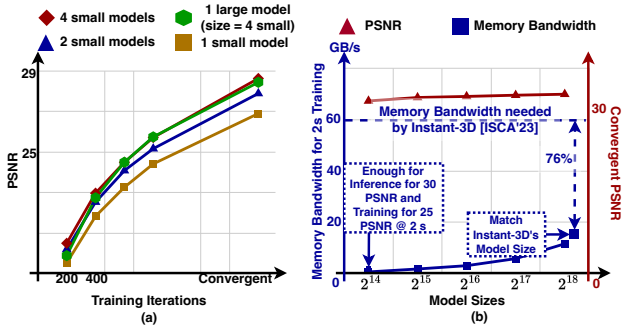


Fig. 13. Experiments on the scalability of this work: (a) The testing PSNR w.r.t. training iterations on the Room scene [1], which is commonly used in recent NeRF works for ablation study [20], [46]. (b) The testing PSNR and required memory bandwidth for 2-second training when adopting different model sizes on NeRF-Synthetic dataset [28].

Bandwidth Savings Across Different Model Sizes. We present a curve in Fig. 13(b) that illustrates the bandwidth requirements across various single-chip model sizes, which are sufficient to meet a 2-second training time constraint. Observations include: 1) Our proposed techniques consistently reduce the off-chip bandwidth requirements across different model sizes; 2) Using the same model size as the SOTA NeRF training accelerator (i.e., $2^{16}+2^{18}$ as reported in [22]), we achieve a 76% bandwidth reduction compared to this SOTA accelerator. This 76% reduction (44 GB/s) is solely attributable to our optimized end-to-end pipeline; 3) In our current configuration, all hash tables can be stored on-chip, utilizing only $2 \times 5 \times 64$ KB of SRAM, which results in a minimal bandwidth requirement of only 0.6 GB/s.

Effectiveness of the Proposed Techniques When Adapted to Other NeRF Pipelines. Our key components are adaptable to various NeRF pipelines as these pipelines share common components, such as the sampling and post-processing modules referenced in [18], [28], [31]. Consequently, our proposed designs for these modules can be employed in other NeRF implementations. Specifically, when integrating the proposed Sampling and Post-Processing modules into TensorRF [5] (while retaining its Feature Interpolation Module), we observe a 39% reduction in power consumption and an 11% reduction in area compared to RT-NeRF [5]. Furthermore, our MoE design integrates seamlessly with the final output stages of various NeRF pipelines, making it fundamentally compatible with different architectural designs. When the MoE configuration was applied to TensorRF [5], four smaller models (each with 128^3 parameters) achieve a PSNR difference of only -0.5 when compared to a single larger model (with 4×128^3 parameters) using the NeRF-Synthetic dataset [28].

Speedup Breakdown Analysis. The design methodology of this work is to first push the speed of Stage II to fit the need for

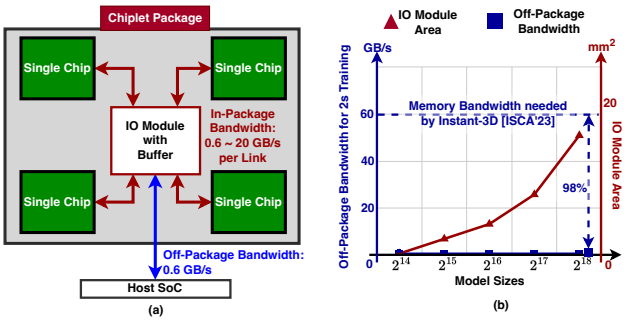


Fig. 14. (a) The block diagram of the chiplet-based multi-chip system. (b) The required I/O module area to keep off-package bandwidth to 0.6 GB/s with different model sizes. The area of the I/O module is the sum of the area of the current implemented I/O module without any buffer and the SRAM area computed using the datasheet.

real-time inference and instant training, then match the speed of Stages I and III by adjusting the number of computing cores. Thus, the speedups for different stages are the same, in other words, all the stages have a 47× and 76× speedup during inference and training w.r.t. Nvidia XNX [33], respectively.

D. Integration into Existing Systems.

This work integrates easily into existing AR/VR systems like a USB drive, as this dedicated accelerator requires only 0.6 GB/s of memory bandwidth, which is lower than the common USB connection bandwidth [50] of 0.625 GB/s in edge devices [27], [41]. We demonstrate an integration example in Fig. 4(b), using an FPGA for interfacing our prototype chip with the host SoC like Meta's avatar chip [44].

VII. Related Works

Several accelerators have been developed for NeRF. MetaVRain [13] utilizes MLP-based NeRF pipelines [28]. It employs image-warping to reuse the previous frame's image in the current frame to reduce computation. RT-NeRF [18], focusing on the TensorRF [5] algorithm, introduces a sparse encoding scheme for enhanced hardware efficiency. NGPC [30], Instant-3D [22], and NeuRex [16] target the Instant-NGP [31] algorithm. NGPC focuses on integrating NeRF units into existing GPUs. Instant-3D aims to shrink the feature table size by decoupling color and density branches and introduces a memory reordering & merging scheme to address memory access irregularities. NeuRex concentrates on space tiling and enhancing neural network accelerators with specialized hash encoding engines.

VIII. Discussion and Future Work

Discussion on the Impact of Chiplet to Design Scaling. Using chiplets can reduce the number of chips needed when designing systems to handle larger models. Chiplets provide higher bandwidth between chips in the advanced package, as noted in [42]. This high bandwidth enables the integration of a buffer in the I/O module, which connects to the four computing chips in our multi-chip system with a higher bandwidth than that of the off-chip connections, as shown in Fig. 14(a). This buffer allows the computing chips to be temporally reused with a low off-package bandwidth by caching data within the package instead of performing off-package data exchanges. This setup helps avoid the need to spatially add more chips simply to accommodate larger model sizes when off-chip bandwidth is limited. However, the newly introduced buffer incurs additional overhead. To quantitatively assess this overhead, Fig. 14(b) demonstrates the relationship between the model size and the required area of the I/O module, aiming to maintain an off-package bandwidth of 0.6 GB/s. As we can see, the I/O module area needs to increase significantly when scaling up the model size. Therefore, a potential future direction for this work is to find an optimal balance between off-chip communication, silicon area in different modules, and runtime performance across various multi-chip systems (e.g., PCB-based and chiplet-based) for 3D reconstruction tasks.

Discussion on the Impact of 3D Stacked Memory to Design Scaling. 3D stacked memory can help boost single-chip performance thus helping reduce the number of chips needed for multi-chip systems. It can also help reduce fabrication costs for the IO Module. Based on the post-layout results, approximately 50% of our chip's Feature Interpolation Module's area is occupied by SRAMs, and the critical path of the design is now a long wire that crosses the SRAM block. Using 3D stacked memory can move this memory to another dimension [23], effectively doubling the core count within the same area and resolving this critical path issue. Thus, with 3D stacked memory, both the area efficiency and clock frequency can be improved. With this enhancement in single-chip performance, we can reduce the number of chips needed for multi-chip configurations. Moreover, the stacked memory die could be reused between the computing chips and the IO module, thus further reducing the tapeout cost for the chiplet design. One potential future direction of this work is to study an optimal silicon area arrangement for logic and memory, considering reuse in multi-chip systems with 3D stacked memory for 3D reconstruction tasks.

IX. Conclusions

We propose, design, and validate an end-to-end and scalable 3D reconstruction and rendering acceleration framework. The techniques integrated in the single-chip accelerator are validated through a silicon prototype in 28nm CMOS, while the multi-chip system is proven in concept through a PCB prototype. To the best of our knowledge, this work is the first to achieve both instant ($\leq$ 2 seconds) 3D reconstruction and real-time ($\geq$ 30 FPS) rendering, while requiring only 0.6 GB/s of memory bandwidth.

Acknowledgments

This work was supported by Silicon Creations, the NSF Energy, Power, Control, and Networks (EPCN) program (Award ID: 2346091), the NSF Computing and Communication Foundations (CCF) program (Award ID: 2434166 and 2312758), and CoCoSys, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

REFERENCES

[1] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5470–5479.

[2] M. Q. Blog, "Tackling telepresence: 'spatial' delivers collaborative computing on oculus quest," https://www.meta.com/nl-nl/blog/quest/tackling-telepresence-spatial-delivers-collaborative-computing-on-oculus-quest, 2020.

[3] Cadence, "Cadence Genus&Innovus," 2022, https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/synthesis/genus-synthesis-solution.html, accessed 2022-05-20.

[4] Cadence, "Cadence Genus&Innovus," 2022, https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/synthesis/genus-synthesis-solution.html, accessed 2022-05-20.

[5] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *European Conference on Computer Vision (ECCV)*, 2022.

[6] Z. Chen, T. Funkhouser, P. Hedman, and A. Tagliasacchi, "Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures," in *The Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[7] N. Du, Y. Huang, A. M. Dai, S. Tong, D. Lepikhin, Y. Xu, M. Krikun, Y. Zhou, A. W. Yu, O. Firat, B. Zoph, L. Fedus, M. Bosma, Z. Zhou, T. Wang, Y. Wang, K. Webster, M. Pellat, K. Robinson, K. Meier-Hellstern, T. Dukea, L. Dixon, K. Zhang, Q. V. Le, Y. Wu, Z. Chen, and C. Cui, "Glam: Efficient scaling of language models with mixture-of-experts," in *International Conference on Machine Learning*. PMLR, 2022, pp. 5547–5569.

[8] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 5232–5270, 2022.

[9] Y. Feng and K. Ma, "Chiplet actuary: A quantitative cost model and multi-chiplet architecture exploration," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, 2022, pp. 121–126.

[10] Y. Fu, Z. Ye, J. Yuan, S. Zhang, S. Li, H. You, and Y. Lin, "Gen-nerf: Efficient and generalizable neural radiance fields via algorithm-hardware co-design," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, ser. ISCA '23. Orlando, FL, USA: Association for Computing Machinery, 2023.

[11] C. Geng, S. Peng, Z. Xu, H. Bao, and X. Zhou, "Learning neural volumetric representations of dynamic humans in minutes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8759–8770.

[12] M. Giordano, K. Prabhu, K. Koul, R. M. Radway, A. Gural, R. Doshi, Z. F. Khan, J. W. Kustin, T. Liu, G. B. Lopes, V. Turbiner, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, G. Lallement, B. Murmann, S. Mitra, and P. Raina, "Chimera: A 0.92-tops, 2.2-tops/w edge ai accelerator with 2-mbyte on-chip foundry resistive ram for efficient training and inference," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 4, pp. 1013–1026, 2022.

[13] D. Han, J. Ryu, S. Kim, S. Kim, and H.-J. Yoo, "2.7 metavrain: A 133mw real-time hyper-realistic 3d-nerf processor with 1d-2d hybrid-neural engines for metaverse on mobile devices," in *2023 IEEE International Solid- State Circuits Conference (ISSCC)*, 2023, pp. 50–52.

[14] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[15] T. Jia, Y. Ju, R. Joseph, and J. Gu, "Ncpu: An embedded neural cpu architecture on resource-constrained low power devices for real-time end-to-end performance," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2020, pp. 1097–1109.

[16] J. Lee, K. Choi, J. Lee, S. Lee, J. Whangbo, and J. Sim, "Neurex: A case for neural rendering acceleration," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, 2023, pp. 1–13.

[17] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "Gshard: Scaling giant models with conditional computation and automatic sharding," *arXiv preprint arXiv:2006.16668*, 2020.

[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 *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, ser. ICCAD '22. New York, NY, USA: Association for Computing Machinery, 2022.

[19] C. Li, B. Wu, A. Pumarola, P. Zhang, Y. Lin, and P. Vajda, "Ingeo: Accelerating i nstant n eural scene reconstruction with noisy geo metry priors," in *European Conference on Computer Vision*. Springer, 2022, pp. 686–694.

[20] C. Li, B. Wu, P. Vajda *et al.*, "Mixrt: Mixed neural representations for real-time nerf rendering," *arXiv preprint arXiv:2312.11841*, 2023.

[21] R. Li, H. Gao, M. Tancik, and A. Kanazawa, "Nerfacc: Efficient sampling accelerates nerfs," *arXiv preprint arXiv:2305.04966*, 2023.

[22] S. Li, C. Li, W. Zhu, B. Yu, Y. Zhao, C. Wan, H. You, H. Shi, and Y. Lin, "Instant-3d: Instant neural radiance field training towards on-device ar/vr 3d reconstruction," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, ser. ISCA '23. Orlando, FL, USA: Association for Computing Machinery, 2023.

[23] W. Li, M. Manley, J. Read, A. Kaul, M. S. Bakir, and S. Yu, "H3datten: Heterogeneous 3-d integrated hybrid analog and digital compute-in-memory accelerator for vision transformer self-attention," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2023.

[24] X. Li and J. Cai, "Robust transmission of jpeg2000 encoded images over packet loss channels," in *2007 IEEE International Conference on Multimedia and Expo*. IEEE, 2007, pp. 947–950.

[25] M.-S. Lin, C.-C. Tsai, C.-H. Hsieh, W.-H. Huang, Y.-C. Chen, S.-C. Yang, C.-M. Fu, H.-J. Zhan, J.-Y. Chien, S.-Y. Li, Y.-H. Chen, C.-C. Kuo, S.-P. Tai, and K. Yamada, "A 16nm 256-bit wide 89.6 gbyte/s total bandwidth in-package interconnect with 0.3 v swing and 0.062 pj/bit power in info package," in *2016 IEEE Hot Chips 28 Symposium (HCS)*. IEEE, 2016, pp. 1–32.

[26] Mark Crovella, "Gaussian elimination to solve linear functions," https://www.cs.bu.edu/fac/crovella/cs132-book/L03RowReductions.html, accessed 2023-11-01.

[27] I. Meta Platforms, "Oculus quest 2," 2021, https://www.oculus.com/experiences/quest/, accessed 2021-08-01.

[28] 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," in *ECCV*, 2020.

[29] R. B. Miller, "Response time in man-computer conversational transactions," in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, 1968, pp. 267–277.

[30] M. Mubarik, R. Kanungo, T. Zirr, and R. Kumar, "Hardware acceleration of neural graphics," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, ser. ISCA '23. Orlando, FL, USA: Association for Computing Machinery, 2023.

[31] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530127

[32] F. F.-H. Nah, "A study on tolerable waiting time: how long are web users willing to wait?" *Behaviour & Information Technology*, vol. 23, no. 3, pp. 153–163, 2004.

[33] NVIDIA Inc., "Jeston Xavier NX Series Modules," 2022, https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-nx/, accessed 2022-06-01.

[34] NVIDIA LLC., "GeForce RTX 2080 TI Graphics Card — NVIDIA," 2021, https://www.nvidia.com/en-me/geforce/graphics-cards/rtx-2080-ti/, accessed 2020-09-01.

[35] NVIDIA LLC., "Jetson Nano Developer Kit," 2021, https://developer.nvidia.com/embedded/jetson-nano-developer-kit, accessed 2020-09-01.

[36] M. Pharr, W. Jakob, and G. Humphreys, *Physically based rendering: From theory to implementation*. MIT Press, 2023.

[37] J. W. Poulton, J. M. Wilson, W. J. Turner, B. Zimmer, X. Chen, S. S. Kudva, S. Song, S. G. Tell, N. Nedovic, W. Zhao, S. R. Sudhakaran, C. T. Gray, and W. J. Dally, "A 1.17-pj/b, 25-gb/s/pin ground-referenced single-ended serial link for off-and on-package communication using a process-and temperature-adaptive voltage regulator," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 43–54, 2018.

[38] C. Reiser, R. Szeliski, D. Verbin, P. Srinivasan, B. Mildenhall, A. Geiger, J. Barron, and P. Hedman, "Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–12, 2023.

[39] C. Riquelme, J. Puigcerver, B. Mustafa, M. Neumann, R. Jenatton, A. Susano Pinto, D. Keysers, and N. Houlsby, "Scaling vision with sparse mixture of experts," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8583–8595, 2021.

[40] S. Sabour, S. Vora, D. Duckworth, I. Krasin, D. J. Fleet, and A. Tagliasacchi, "Robustnerf: Ignoring distractors with robust losses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20 626–20 636.

[41] Samsung, "Samsung Galaxy S24 Ultra Features & Highlights Samsung US," 2024, https://www.samsung.com/us/smartphones/galaxy-s24-ultra/, accessed 2024-03-03.

[42] Y. S. Shao, J. Clemons, R. Venkatesan, B. Zimmer, M. Fojtik, N. Jiang, B. Keller, A. Klinefelter, N. Pinckney, P. Raina, S. G. Tell, Y. Zhang, W. J. Dally, J. Emer, C. T. Gray, B. Khailany, and S. W. Keckler, "Simba: Scaling deep-learning inference with multi-chip-module-based architecture," in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO '52. New York, NY, USA: Association for Computing Machinery, 2019, p. 14–27. [Online]. Available: https://doi.org/10.1145/3352460.3358302

[43] A. Shokrollahi, D. Carnelli, J. Fox, K. Hofstra, B. Holden, A. Hormati, P. Hunt, M. Johnston, J. Keay, S. Pesenti, R. Simpson, D. Stauffer, A. Stewart, G. Surace, A. Tajalli, O. T. Amiri, A. Tschank, R. Ulrich, C. Walter, F. Licciardello, Y. Mogentale, and A. Singh, "10.1 a pin-efficient 20.83 gb/s/wire 0.94 pj/bit forwarded clock cnrz-5-coded serdes up to 12mm for mcm packages in 28nm cmos," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2016, pp. 182–183.

[44] H. E. Sumbul, T. F. Wu, Y. Li, S. S. Sarwar, W. Koven, E. Murphy-Trotzky, X. Cai, E. Ansari, D. H. Morris, H. Liu, D. Kim, and E. Beigne, "System-level design and integration of a prototype ar/vr hardware featuring a custom low-power dnn accelerator chip in 7nm technology for codec avatars," in *2022 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 2022, pp. 01–08.

[45] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa, "Nerfstudio: A modular framework for neural radiance field development," in *ACM SIGGRAPH 2023 Conference Proceedings*, ser. SIGGRAPH '23, 2023.

[46] J. Tang, H. Zhou, X. Chen, T. Hu, E. Ding, J. Wang, and G. Zeng, "Delicate textured mesh recovery from nerf via adaptive surface refinement," *arXiv preprint arXiv:2303.02091*, 2022.

[47] N. Tatarchuk, S. Lagarde, and A. Benyoub, "Towards filmic quality at 30 fps: Real-time ray tracing for practical game engine pipelines," in *Game Developers Conference*, 2019.

[48] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W. Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi, T. Simon, C. Theobalt, M. Niessner, J. T. Barron, G. Wetzstein, M. Zollhoefer, and V. Golyanik, "Advances in neural rendering," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022, pp. 703–735.

[49] N. Thomos, N. V. Boulgouris, and M. G. Strintzis, "Optimized transmission of jpeg2000 streams over wireless channels," *IEEE Transactions on image processing*, vol. 15, no. 1, pp. 54–67, 2005.

[50] USB IF, "USB 3.2 Revision 1.1 - June 2022," 2022, https://www.usb.org/document-library/usb-32-revision-11-june-2022, accessed 2022-06-03.

[51] R. Wang, B. Yu, J. Marco, T. Hu, D. Gutierrez, and H. Bao, "Real-time rendering on a power budget," *ACM Trans. Graph.*, vol. 35, no. 4, jul 2016. [Online]. Available: https://doi.org/10.1145/2897824.2925889

[52] L. Yariv, P. Hedman, C. Reiser, D. Verbin, P. P. Srinivasan, R. Szeliski, J. T. Barron, and B. Mildenhall, "Bakedsdf: Meshing neural sdfs for real-time view synthesis," *arXiv preprint arXiv:2302.14859*, 2023.

[53] Y. Zhao, Z. Li, Y. Fu, Y. Zhang, C. Li, C. Wan, H. You, S. Wu, X. Ouyang, V. Boominathan *et al.*, "I-flatcam: A 253 fps, 91.49 μj/frame ultra-compact intelligent lensless camera for real-time and efficient eye tracking in vr/ar," in *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 2022, pp. 108–109.

[54] Y. Zhao, Y. Zhang, Y. Fu, X. Ouyang, C. Wan, S. Wu, A. Banta, M. M. John, A. Post, M. Razavi *et al.*, "e-g2c: A 0.14-to-8.31 μj/inference nn-based processor with continuous on-chip adaptation for anomaly detection and ecg conversion from egm," in *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 2022, pp. 252–253.