MixNet: A Runtime Reconfigurable Optical-Electrical Fabric for Distributed Mixture-of-Experts Training

Xudong Liao
Hong Kong University of Science and Technology

Yijun Sun
Hong Kong University of Science and Technology

Han Tian
Hong Kong University of Science and Technology

Xinchen Wan
Hong Kong University of Science and Technology

Yilun Jin
Hong Kong University of Science and Technology

Zilong Wang
Hong Kong University of Science and Technology

Zhenghang Ren
Hong Kong University of Science and Technology

Xinyang Huang
Hong Kong University of Science and Technology

Wenxue Li
Hong Kong University of Science and Technology

Kin Fai Tse
Hong Kong University of Science and Technology

Zhizhen Zhong
Massachusetts Institute of Technology

Guyue Liu
Peking University

Ying Zhang
Meta

Xiaofeng Ye
EmbedWay

Yiming Zhang
Xiamen University

Kai Chen
Hong Kong University of Science and Technology

Abstract

Mixture-of-Expert (MoE) models outperform conventional models by selectively activating different subnets, named *experts*, on a per-token basis. This gated computation generates *dynamic* communications that cannot be determined beforehand, challenging the existing GPU interconnects that remain *static* during distributed training. In this paper, we advocate for a first-of-its-kind system, called MixNet, that unlocks topology reconfiguration *during* distributed MoE training. Towards this vision, we first perform a production measurement study and show that the MoE dynamic communication pattern has *strong locality*, alleviating the need for global reconfiguration. Based on this, we design and implement a *regionally reconfigurable high-bandwidth domain* that augments existing electrical interconnects using optical circuit switching (OCS), achieving scalability while maintaining rapid adaptability. We build a fully functional MixNet prototype with commodity hardware and a customized collective communication runtime. Our prototype trains state-of-the-art MoE models with *in-training* topology reconfiguration across 32 A100 GPUs. Large-scale packet-level simulations show that MixNet achieves performance comparable to a non-blocking fat-tree fabric while boosting the networking cost efficiency (e.g., performance per dollar) of four representative MoE models by 1.2×–1.5× and 1.9×–2.3× at 100 Gbps and 400 Gbps link bandwidths, respectively.

CCS Concepts

• **Networks → Network architectures**; **Data center networks**.

Keywords

Network Architecture, Mixture-of-Experts, Optical Circuit Switching, AI Infrastructure

1 Introduction

Mixture-of-Experts (MoE) models [9, 14, 23, 40, 43, 54, 55, 59, 97, 103, 117] have gained significant traction in the machine learning community to improve the performance of large language models (LLMs) [14, 23]. Unlike traditional methods that scale LLMs by stacking dense layers, which leads to a linear increase in computational costs as model sizes expand, MoE models utilize multiple parallel expert layers and activate only a subset of them based on the input token for each training iteration (e.g., xAI discloses that 25% weights are active in Grok-1 [14], while DeepSeek-V3 [9] only activates 37 billion parameters over total 671 billion parameters). This dynamic approach enables models to grow to large sizes without a proportional cost increase in computation.

However, such dynamic expert activations require *all-to-all* communications in and out of expert layers in each training iteration. Among parallelization strategies, *expert parallelism (EP)*, which assigns expert layers to different GPUs, requires a high volume of traffic that is comparable to that of *tensor parallelism (TP)*, and much larger than other parallelisms. Moreover, the token-specific activation of experts in EP results in *temporally non-deterministic* and *spatially non-uniform* communication patterns that vary across training iterations, challenging existing GPU interconnects.

Today's GPU interconnects contain intra-server *scale-up networks* (e.g., NVSwitch [34] or NVLink [33]) and inter-server *scale-out networks* (e.g., Ethernet or Infiniband). Both of them are currently dimensioned with uniform and static network topologies (e.g., fully-connected crossbar topology for scale-up networks [34], and Clos-style fat-tree for scale-out networks [44, 101]). When accommodating the temporal and spatial variations of MoE communication patterns, these fabrics contain over-provisioned full bisection bandwidth that is mostly under-utilized. Some recent proposals on the use of optical circuit switching (OCS) perform topology reconfiguration for spatially non-uniform traffic distributions. However, they assume stable temporal patterns such that no reconfigurations occur during the entire training process [71, 107]. As a result, these interconnect architectures face bottlenecks, leading to inefficient resource usage and slowdowns in distributed MoE training.

Therefore, to fully unlock the computational advantages of MoE models, we need to design a novel GPU interconnect fabric that is adaptable to dynamic all-to-all communication patterns at runtime. Achieving such adaptability requires the topology to be reconfigurable *during* the distributed MoE training process. This is very challenging because today's commodity OCS technologies face fundamental trade-offs between *low reconfiguration latency* (to enable reconfiguration during training) and *high scalability* (to interconnect tens of thousands of GPUs) (more details in Table 2).

To understand the problem space, we first perform a comprehensive measurement study in a production GPU cluster to investigate the real-world communication patterns of distributed MoE training. Our measurements reveal that although EP generates substantial variability during training, its dynamic range is strictly within an MoE block, creating *strong locality* for all-to-all traffic on a global scale (§3).

Based on this insight, we introduce MixNet, a novel system designed to overcome these challenges by enabling efficient topology

reconfiguration during distributed MoE training. At the core of MixNet is a *regionally reconfigurable high-bandwidth domain* based on millisecond-scale reconfigurable OCS that sits at the boundary between scale-up and scale-out networks. This design augments the existing static electrical interconnect with the capability of rapid regional reconfiguration while still preserving its scalability (§4).

MixNet contains the following key components: 1) MixNet exploits the partially predictable nature of all-to-all communications to track regional traffic demands at runtime (§5.1); 2) Based on the obtained demands, MixNet uses a greedy algorithm that generates the tailored network topology and reconfigures the OCS to realize it (§5.2); 3) With the reconfigured topology, MixNet uses a customized collective communication runtime to orchestrate inter-host DP and EP communications on EPS and regional OCS fabrics (§5.3).

To demonstrate MixNet, we build a fully functional prototype with 32 Nvidia A100 GPUs, 16 Mellanox NICs [29], a Polatis millisecond OCS [38], and an Ethernet switch. In addition, we develop a custom collective communication runtime based on Nvidia collective communications library (NCCL) [26] to support in-training topology reconfigurations. Using this prototype, we successfully demonstrated the benefits of MixNet on state-of-the-art MoE models (§6).

To evaluate the performance of MixNet at scale, we perform packet-level simulations using four representative real-world MoE models. Our results reveal that MixNet outperforms the state-of-the-art fabrics, exhibiting a training speed comparable to Rail-optimized [11] and Fat-tree [44] fabrics while significantly improving cost-efficiency. Specifically, MixNet improves networking infrastructure cost-efficiency by 1.2×–1.5× (1.9×–2.3×) compared to Fat-tree and 1.4×–1.5× (2.3×–2.4×) compared to Rail-optimized for 100Gbps (400Gbps) links. We also observe that MixNet outperforms TopoOpt [107] by up to 2.5× and supports scalability with the cluster size increasing to 30K+ GPUs (§7). When connected to co-packaged optical ports attached to GPUs, MixNet augments high-radix scale-up systems like NVL72 by 1.3× (§8).

For more information, please visit the website: https://mixnet-project.github.io/.

2 Background

In this section, we first describe MoE's model architecture and training parallelization strategies (§2.1). Then, we discuss several GPU interconnects for distributed training (§2.2).

2.1 Distributed MoE Training

MoE model architecture. MoE models contain several sequential MoE blocks [59, 97]. As shown in Figure 1a, each MoE block has an attention layer, a gate unit, and several parallel feed-forward networks (FFNs) called *experts*. The input token x is first fed into the attention layer. After that, the gate unit selects the most relevant experts based on the output of the attention layer. This is called computation-based routing and is the key to enabling MoE's sparse architecture that scales model parameters without a linear increase in computation cost. The output token, y, is the weighted sum of the outputs of all activated experts.

Data Parallelism (DP). In DP [76], the model parameters are replicated across multiple GPUs, and each GPU hosts a different
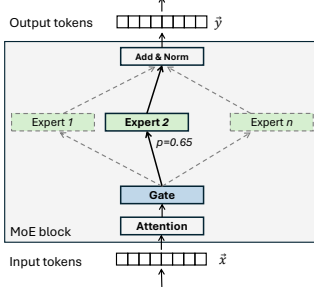
(a) An MoE block with multiple experts. In this example, the *gate* only activates *Expert 2*.
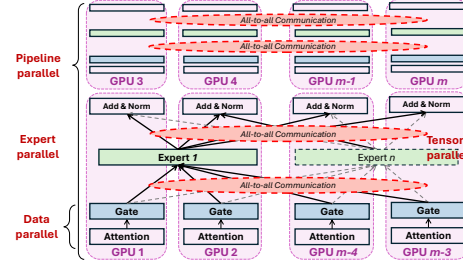
(b) Example of a hybrid parallelism that combines DP (gate), EP (parallel expert layers), PP (MoE blocks) and TP (a single expert layer).

Figure 1: Illustration of the MoE's gated expert architecture and its distributed training strategies.

Models	Mixtral	LLaMA-MoE	Qwen-MoE
Size	8×7B	6.7B	14.3B
# of MoE blocks	32	32	24
# experts	8	16	64
EP degree	8	16	16
TP degree	4	1	1
PP degree	4	4	4
Seq. len.	4096	4096	4096
Micro-batch size	8	8	8

Table 1: State-of-the-art MoE training configurations.

Commodity OCS	Port count	Reconfig. delay
Robotic (Telescent) [107]	1008×1008	Several minutes
Piezo (Polatis) [38]	576×576	10-25 ms
3D MEMS (Calient) [6, 94]	320×320	10-15 ms
2D MEMS (Google Palomar) [81]	136×136	Not reported
RotorNet (InFocus) [85, 86]	128×128	10 μs
Silicon Photonics (Lightmatter) [19]	32×32	7 μs
PLZT (EpiPhotonics) [25]	16×16	10 ns

Table 2: Tradeoff between port count and reconfiguration delay in commodity OCS technologies.

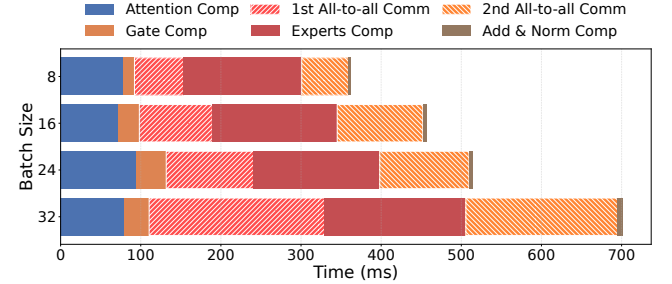**Figure 2: Traffic volume distribution of different parallelism of three state-of-the-art MoE models.**

Figure 3: [Mixtral 8×7B in production] EP's all-to-all communications occupy 33% to 55% of the total training iteration time in 400 Gbps network.

subset of training data. Because only gradients are transferred across GPUs, the traffic volume is relatively small compared to other parallelisms. In MoE models, DP is usually applied to gate units and add & norm layers (Figure 1b). DP also applies, as we create replicas of the whole model onto several different clusters.

Tensor Parallelism (TP). TP [98] is a technique to partition a layer among multiple GPUs. In distributed MoE training (Figure 1b), the expert layers are usually partitioned across different GPUs, with intermediate hidden states being transferred through collective communication primitives like broadcast, all-gather and reduce-scatter. Therefore, TP is the most communication-intensive operation, and its spatial scale is generally limited to a few GPUs in practice [98].

Pipeline Parallelism (PP). In PP [88, 98], multiple sequential stages of the model are distributed to different GPUs (Figure 1b). Therefore, only hidden activation states are transferred through point-to-point all-reduce collective communication primitives, gen-

erating the least amount of communication with deterministic volume.

Expert Parallelism (EP). In MoE models, different experts in an MoE block are allocated to different GPUs [59, 74, 77, 82] (Figure 1b). Since each GPU needs to send its local states to other experts and receive remote states from other GPUs, the dispatching of intermediate hidden states and the collection of expert outputs are performed via two all-to-all communications. EP's all-to-all communication is non-uniform and non-deterministic across different training iterations (§3).

Traffic volume of different parallelisms. We use Megatron-LM [98] to profile three state-of-the-art MoE models (Mixtral 8×7B MoE [23], Llama-MoE [117] and Qwen-MoE [40]) and measure the total amount of data transfer. The detailed model configurations are shown in Table 1. We plot the distribution of traffic volume in one MoE training iteration in Figure 2. For Mixtral 8×7B, we observe that TP generates the highest traffic volume (60% of the

(a) In the temporal dimension, the all-to-all traffic volume of each node varies across different training iterations.

(b) In spatial dimension, the all-to-all traffic volume is non-uniform across different experts.

Figure 4: [Mixtral 8×7B in production] All-to-all traffic dynamics during MoE training.

total volume). The EP generates the second-highest traffic volume (30%), leaving PP and DP contributing less than 6%. For LLaMA-MoE and Qwen-MoE, we find that EP becomes the most communication intensive (more than 80%). This is because the size of the largest layer (i.e., expert) fits into a single GPU memory.

2.2 GPU Interconnects

Scale-up fabrics using NVLink and NVSwitch. NVLink and NVSwitch are proprietary technologies provided by Nvidia to support GPU communications within a host server [28, 33]. They offer a higher bandwidth (1.8 TB/s) than PCIe (128 GB/s).

Scale-out fabrics using electrical packet switching (EPS). Ethernet- and Infiniband-based EPS has been widely adopted in data center networks with clos-style topologies [44, 64–66, 68, 101]. In such networks, data are encapsulated into packets and switched at layer 2 or above. EPS has the advantage of massive scalability to hundreds of thousands of host servers in modern data centers [44]. However, EPS networks are fixed in topology that cannot be easily reconfigured.

Scale-out fabrics using optical circuit switching (OCS). OCS is a layer-1 switching technology that creates dedicated reconfigurable optical circuits between hosts. As depicted in Table 2, today's commodity OCSes have a fundamental tradeoff between the *scalability* (in terms of port counts) and *agility* (in terms of reconfiguration delay). Technologies like the robot optical patch panel [107] scale up to thousands of ports at the cost of several minutes of reconfiguration delay. At the other end of the spectrum, waveguide-based OCS like silicon photonics [19] and PLZT [25] scores microseconds or nanoseconds latency with limited port counts.

3 Production Measurements

Unlike conventional parallelism (e.g., TP, PP, and DP) with deterministic communication patterns, EP's communications are determined by the gate unit at runtime due to the semantic heterogeneity of the input tokens. To understand the dynamics of EP traffic patterns, we profile Mixtral 8×7B [23] in a production data center, using a

hybrid parallelism that combines an EP degree of 8, TP degree of 4, PP degree of 4 at a sequence length of 4096, and micro-batch size of 8 [56].

The production fabric. We use a Certified Nvidia DGX SuperPOD platform [27] with 128 H800 GPUs and 128 ConnectX-7 400Gbps NICs in a production data center. The compute fabric is connected in a rail-optimized topology [11]. We use NCCL [26] to optimize communications on the DGX platform.

All-to-all communications within a training iteration. We first measure the time it takes for each step in Mixtral 8×7B's forward pass computation[1] and show the results in Figure 3. For the typical micro-batch size used in production (e.g., 8), we observe that the expert computation takes more than 100 ms, which is much larger than the reconfiguration latency of existing optical switches (for example, the MEMS OCS in Table 2). Therefore, it provides an opportunity to reconfigure the OCS for the second all-to-all in the expert computation phase. For backward propagation, it allows us to hide the reconfiguration latency in the attention computation of its later layer (for the second all-to-all) and expert computation period (for the first all-to-all) as the backward computation often takes more time than the forward. The results of other MoE models are presented in Appendix A.1.

All-to-all communications are temporally dynamic. Figure 4a plots the total communication volume that each expert receives in all-to-all communication in each MoE layer, which represents the activation intensity of each expert. We find that the activation intensities of each expert vary significantly across different iterations, which indicates the non-deterministic nature of the EP traffic. We also find that as training progresses, the variability of the overall communication volume among experts decreases. The decreasing variability is attributed to the use of *load balancing loss*[2]. However, even as the overall communication volumes of experts appear to converge, the sparsity of all-to-all traffic matrices persists, as illustrated in Figure 4b. Furthermore, recent advances in

---

[1]The backward pass is a reverse process of the forward pass.
[2]In MoE training, load balancing loss is commonly used to ensure that token loads are evenly distributed across all experts.
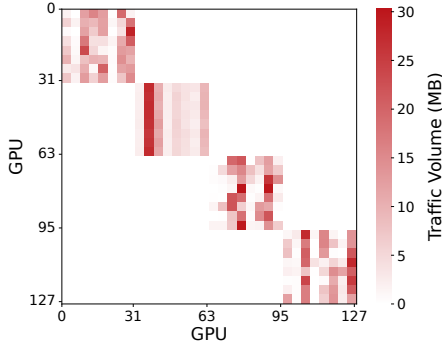
Figure 5: [Mixtral 8×7B in production] Traffic matrix of all GPUs showing strong locality.

Figure 6: The MixNet network architecture.

the ML community have introduced MoE training techniques that intentionally leave selected experts underutilized at some training stages to achieve improved model performance [52, 83]. This further highlights the dynamic communications in distributed MoE training.

All-to-all communications are spatially non-uniform. Figure 4b plots the detailed all-to-all communication matrix of a selected layer through different iterations. We observe that each traffic matrix of all-to-all communication is non-uniform, with heavy communication only between several GPU pairs. Specifically, the state-of-the-art LLM model DeepSeek-V3 reveals that explicitly creating non-uniformity in token distribution across experts while bypassing load-balancing loss[3] improves the MoE training process.

All-to-all communications have strong locality. Figure 5 shows the all-to-all communications among all the 128 GPUs during the training of Mixtral 8×7B. We observe that the EP traffic exhibits *strong locality*. This is because only the expert layers within the same MoE block need all-to-all communications, while expert layers across different MoE blocks at different PP stages do not communicate directly.

The above observations are stemmed from the inherent sparse activation characteristic of MoE layer and the gradual refinement of the gating unit during training. We note that other work in ML community [9, 75] have observed similar behaviors, suggesting that these characteristics are common across different MoE models.
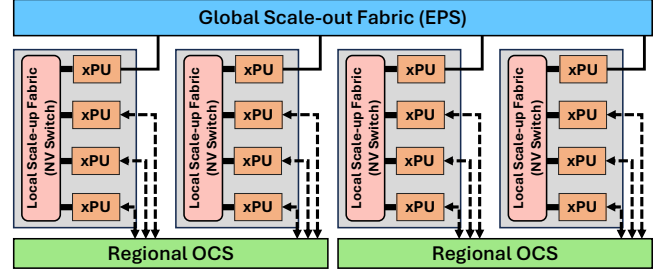
4 MixNet Architecture Design

So far, we have shown that MoE introduces unique traffic patterns that are temporally non-deterministic and spatially non-uniform. Now, an important question is *How to design a network architecture that best serves the requirements of distributed MoE training?* In this section, we first study an ideal yet practical fabric for distributed MoE training (§4.1). Then, we present our key proposal in MixNet, the regionally reconfigurable high-bandwidth domain with OCS (§4.2).

4.1 Towards An Ideal yet Practical Fabric

We start with a thought experiment from first principles on an ideal yet practical fabric for distributed MoE training.

The ideal fabric. Designing an ideal fabric for distributed MoE

training requires the best fit between the traffic patterns generated by different parallelization strategies and the interconnect technologies. In Table 3, we first summarize the requirements in terms of volume, temporal pattern, and spatial pattern for different parallelisms. Then, we review the desired fabrics, considering their bandwidth, reconfigurability, and scalability. In particular, conventional parallelisms like TP, DP, and PP require only a one-shot reconfiguration because their communication patterns are deterministic. While TP requires high bandwidth within a small number of GPUs that host an individual layer, DP and PP span more GPUs with relatively lower communication bandwidth. Notably, EP is significantly different, as its temporally non-deterministic traffic patterns require in-training topology reconfiguration, and the all-to-all communications among experts require a medium fabric radix. Therefore, the ideal fabric for MoE training should be a reconfigurable network capable of adjusting its topology as soon as traffic patterns vary across training iterations. Moreover, the topology reconfiguration should be completed before each all-to-all communication phase to avoid interrupting the computation. So, the time window left for topology reconfiguration is on the order of tens of milliseconds (based on the measurements in Figure 3).

Challenges of the ideal fabric. Realizing this ideal fabric presents significant challenges. Recall that in Table 2 we review a trade-off in commodity OCS technologies[4] between reconfiguration delay and port count. Achieving millisecond scale reconfiguration times typically limits the number of ports in today's OCS to a few hundred ports, making it difficult to scale to hundreds of thousands of nodes needed in a large MoE interconnect. In contrast, increasing the port count to support hundreds of thousands of ports results in slower reconfiguration times, failing to meet the rapid reconfiguration required within training iterations.

Landing the ideal fabric in practice. To reconcile these challenges, we leverage the key observation that despite the non-deterministic and non-uniform characteristics of MoE all-to-all traffic, there is a strong locality for traffic variations as MoE blocks are normally placed in a pipeline [9, 98]. Therefore, instead of building a globally reconfigurable OCS fabric, we propose designing several *regionally reconfigurable OCS* networks (described next in §4.2). Figure 6 depicts the network architecture of MixNet. By partitioning the network into several domains where the communication locality is strong, each regional OCS rapidly adjusts to traffic demands of

---

[3]See Figure 9 and Figure 10 in DeepSeek-V3 report [9] for more details.

[4]There are small-scale prototypes that break this tradeoff by using advanced devices (e.g., tunable lasers with arrayed waveguide grating (AWGR) [48], silicon photonics [72], etc.). They are out of the scope of this paper, which mainly focuses on commodity solutions that are readily deployable at scale.

		Traffic		Ideal Fabrics			Best Fit
	Volume	Temporal Pattern	Spatial Pattern	Bandwidth	Reconfigurability	Scalability	Interconnect Technology
DP	Low	Deterministic	Global All-Reduce	Low	Slow & One-Shot	Large	Electrical Packet Switch (Ethernet)
TP	Highest	Deterministic	Local All-Reduce	High	Slow & One-Shot	Small	Crossbar Switch (NVSwitch)
PP	Low	Deterministic	Global Point-to-Point	Low	Slow & One-Shot	Large	Electrical Packet Switch (Ethernet)
EP	High	Non-Deterministic	Regional Sparse All-to-All	High	Fast & In-Training	Medium	Circuit Switch (Optical)

Table 3: The quest for a best fit between interconnect fabric and the MoE parallelization strategies.

EP without the complexity of global reconfiguration. By implementing regionally reconfigurable OCS networks, we leverage this locality to achieve fast reconfiguration within smaller, manageable regions. This approach balances the need for reconfigurability with practical hardware limitations, effectively supporting the dynamic communication patterns of MoE training.

4.2 Regionally Reconfigurable OCS

MixNet, as the first fabric to support the in-training topology reconfiguration, highlights the core idea of building regionally reconfigurable OCS to offload dynamic EP traffic in the existing electrical fabric. By leveraging the strong locality inherent in MoE's all-to-all traffic, we partition the network into regions where communication demands are non-deterministic among expert layers. This regional approach allows for rapid reconfiguration within each partition, effectively overcoming the fundamental trade-off between reconfiguration speed and port count in OCS technology. By focusing on regional reconfigurability, MixNet achieves scalability while maintaining rapid adaptability to dynamic communication patterns of MoE training, alleviating the complexity of global network reconfiguration.
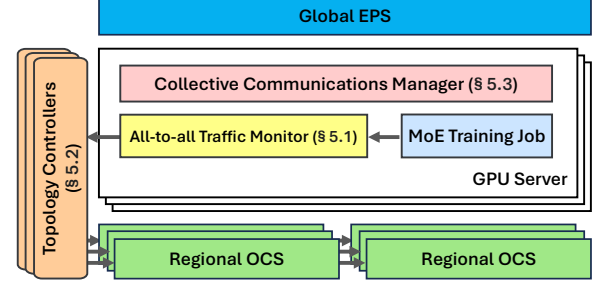
Where to deploy regionally reconfigurable OCS? To best serve the regional sparse all-to-all traffic patterns featured in distributed MoE training, a regional OCS is connected to a cluster of GPU servers where each server splits its NICs between EPS and OCS[5]. Given the fact that today's fast OCSes with millisecond-scale reconfiguration delay support up to 500 ports (Table 2) as well as a typical server contains eight NICs, a reconfigurable high-bandwidth domain interconnect through an OCS supports around 80 to 250 servers (each server assigning two to six NICs to OCS).

When to reconfigure the topology? Unlike EPS networks, which are connectionless, OCS networks are connection-oriented and require active control for reconfiguration. During topology reconfiguration, OCS networks are not available to carry packets[6]. Therefore, we need to choose the right time to start the topology reconfiguration process before the actual data transfer starts to avoid blocking training process.

How to reconfigure the topology? In MixNet, the OCS fabric is arranged into multiple isolated slices for each reconfigurable high-bandwidth region. Therefore, the regional OCS topology is controlled by its localized topology controller, which frequently collects traffic demands from the host servers. Within each train-

Figure 7: MixNet system implementation.

ing iteration are four all-to-all communications with the same or transposed traffic pattern. However, these traffic patterns are non-deterministic across different training iterations. Hence, we need to develop a mechanism to reconfigure the topology tailored to traffic patterns. The regional topology reconfiguration means that MixNet does not require a centralized topology controller, which avoids the scalability concerns of the control plane.

Towards a mixed optical-electrical fabric. MixNet seeks the best fit between the traffic patterns of the parallelization strategies and the corresponding switching technologies. Therefore, MixNet uses server-scale NVSwitch for tensor parallelism, regionally reconfigurable OCS for expert parallelism, and large-scale EPS for data parallelism and pipeline parallelism. To distribute data movement tasks on different fabrics, a new collective communication library that supports topology reconfiguration is needed.

5 MixNet System Implementation

To enable the aforementioned MixNet architecture, we design and implement several core components of the control and data planes. As shown in Figure 7, MixNet's implementation contains a traffic monitor that keeps track of the traffic demands in EP to characterize subsequent all-to-all communications (§5.1). Based on the monitored traffic demands, multiple decentralized topology controllers generate and enforce topologies for regional OCSes (§5.2). Then, a collective communication manager is responsible to steer the traffic in the MixNet fabric (§5.3). In addition, we discuss the failure handling mechanism in (§5.4).

5.1 All-to-All Traffic Characterization

In each MoE block, there are four all-to-all communication phases during each training iteration: two in the forward pass and two in the backward pass (Figure 1b). The first all-to-all communication occurs after the gate unit computation. The output from the gate unit[7]

---

[5]In the near term, MixNet's regional OCS leverages the optical transceivers through the NIC for deployment readiness. In the long term, as the co-package optics (CPO) becomes widely adopted, MixNet's regional OCS is compatible with the commodity optical I/O solutions (e.g., TeraPHY from Ayar Labs [41]) that is directly connected to the computing chip (e.g., GPU, TPU, etc.).
[6]Most commodity optical switches require tens of nanoseconds to several milliseconds or (Table 2) to reconfigure their topology.

[7]The actual output of the gate unit is the dispatching probability distribution for each token. The expert load is derived directly from the probability distribution using the *top-k* parameter.

Algorithm 1 Reconfigure OCS

1: **procedure** RECONFIGUREOCS(E, α, N, V)
   ▷ **input** E: all-to-all communication demands of experts
   ▷ **input** α: current optical degree
   ▷ **input** N: number of servers
   ▷ **input** V: server node set
   ▷ **output** S: NIC level mapping in OCS
2:    $C \leftarrow$ zero matrix of size $N \times N$
3:    $avail\_ocs[v] \leftarrow \alpha$ for $v \in V$
4:    Initialize finish time $T = \infty$ if $D[i][j] \neq 0$, otherwise $T = 0$
     ▷ Step 1: D is translated into an upper triangular matrix.
5:    D $\leftarrow$ CALCULATE_SERVER_DEMAND(E)
6:    **while** True **do**
     ▷ Step 2: Find bottleneck links
7:       (i, j) $\leftarrow$ FINDBOTTLENECKLINK(T, C, V)
8:       **if** $avail\_ocs[i] > 0$ and $avail\_ocs[j] > 0$ **then**
       ▷ Step 3: Create a link between i and j
9:          $C[i][j] \leftarrow C[i][j] + 1$ and $C[j][i] \leftarrow C[j][i] + 1$
10:         **for** $v \in \{i, j\}$ **do**
11:           $avail\_ocs[v] \leftarrow avail\_ocs[v] - 1$
12:       **else**
13:         Break
     ▷ Update the time matrix
14:       $T[i][j] \leftarrow \frac{D[i][j]}{C[i][j]}, T[j][i] \leftarrow \frac{D[j][i]}{C[j][i]}$
     ▷ Step 4: Generate OCS topology
15:    $S \leftarrow$ GETNICMAPPING(C)
16:    $S \leftarrow$ PERMUTELINKS(S)
     ▷ Step 5: Reconfigure the OCS
17:    RECONFIGUREOCS(S)
18:    **return** S

---

across the EP groups determines the traffic matrix for this communication phase. These four all-to-all traffic matrices are strictly the same or transposed due to the symmetry of the token dispatching and collection process. For communication in the first all-to-all in FP, given that the MixNet network architecture is provisioned with millisecond-scale reconfigurable OCS, there are two options. MixNet reconfigures the OCS while blocking the training process, as this reconfiguration time cannot be hidden in the computation. On the other hand, if MixNet opts to utilize a random topology or reuse topology from previous MoE layers, it cannot benefit from the efficient circuits schedule.

Besides, we observe the *partial predicabilty* of the FP's first all-to-all, which offers an opportunity to proactively reconfigure the OCS for it in advance (e.g., in attention computation phase). We offload the details of the prediction algorithm for all-to-all traffic demand in §B.1. Note that MixNet does not introduce extra demand monitoring overhead as the state-of-the-art MoE training framework already contains a mechanism to collect this information [4] to perform on-demand all-to-all transmission.

5.2 Topology Reconfiguration

In MixNet, finding an optimal topology and deriving an optical schedule is an NP-hard problem [60]. We address this challenge by introducing a lightweight greedy algorithm. The key insight is that all-to-all communication time is determined by the delay of largest transfers, which implies that the corresponding GPU pair should be allocated with more circuits. Thus, we identify the communication

pairs with the longest transmission time in each iteration and assign them with direct optical links in the OCS topology. The detailed OCS reconfiguration algorithm is shown in Algorithm 1.

Step 1: Obtain the inter-server demand (line 5). Given the predicted all-to-all communication demands, the algorithm first maps the traffic matrix to an actual inter-server communication demand with respect to the number of experts per GPU and the number of GPUs per server. Note that we provision the TX and RX bandwidth of each OCS link together, thus making the inter-server demand matrix upper triangular via adding the TX and RX demands together.

Step 2: Find bottleneck links (line 7). The algorithm then iteratively finds the bottleneck of the currently allocated links. The bottleneck link is defined as the link with the longest completion time given the demand matrix D and allocated link matrix C. We greedily calculate the bottleneck link by calculating the completion time of each link and return the server pairs with the longest completion time.

Step 3: Allocate OCS circuit (line 9-11). The algorithm first allocates the OCS link for the found bottleneck server pairs. If the OCS NICs of two servers are not fully allocated, the algorithm will assign the link for them accordingly.

Step 4: Generate OCS topology (line 15-16). The algorithm generates the topology by mapping the TX and RX NICs based on the allocated link matrix C. Note that if multiple links exist between two servers, the algorithm permutes the connection to achieve a non-uniform memory access (NUMA) optimized topology to avoid intra-host congestion. For example, if there are two links between server A and server B, the algorithm will permute the connection to ensure that the corresponding TX and RX NICs are in two different NUMA nodes for intra-host traffic forwarding (§5.3).

Step 5: Reconfigure OCS (line 17). The final step is to reconfigure the OCS cross-link connections accordingly. The topology manager leverages the TX/RX pair mappings to establish the optical path for the aforementioned pairs.

5.3 Collective Communication Allocation

With the generated topology, the next step is to allocate network traffic from different parallelisms to MixNet and generate routing schedules. In the following, we illustrate how MixNet's collective communications manager routes network traffic from various parallelisms in the data path.

TP and PP. TP traffic is limited to intra-host high bandwidth domain (e.g., the NVSwitch). PP traffic occurs across different PP stages and relies on the high-fanout EPS fabric in the MixNet architecture. As a result, there are no special configurations for these two types of parallelism.

DP. DP traffic typically spans the entire training cluster. Therefore, MixNet routes it through the EPS fabric. To further improve the communication efficiency of DP transfers, we leverage the hierarchical all-reduce algorithm [69, 104] to reduce the outbound traffic volume from each server. First, the GPUs within each server perform an intra-host reduction to aggregate the parameters to a gateway DP GPU connected to the EPS NIC. Next, all servers engage in a global ring all-reduce among the gateway DP GPUs to
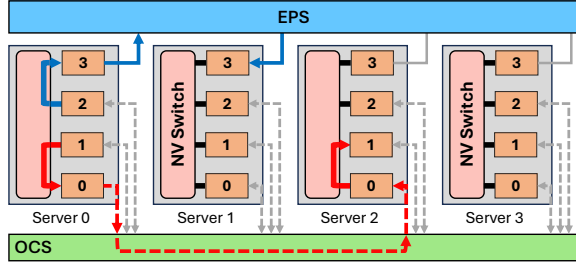
Figure 8: Routing of all-to-all communications in MixNet. For presentation simplicity, each server only contains 4 GPUs, and the TX/RX links are merged into a single link. In practice, MixNet supports the standard setup of 8 GPUs or more.

synchronize the model parameters. Finally, each server broadcasts the synchronized parameters from the gateway DP GPU to all other GPUs. The first and third stages of communication use the high-speed NVSwitch, while the second stage relies on the relatively lower-bandwidth EPS fabric. If multiple EPS NICs are available in the fabric, MixNet utilizes a multi-ring all-reduce method to fully exploit the bandwidth and reduce communication time.

Topology-aware EP. EP traffic is expected to use the regionally reconfigurable high-bandwidth domain. After reconfiguration, MixNet essentially forms a direct-connect topology for EP transfers. Based on that, MixNet uses the following steps to route the EP traffic. We illustrate this process in Figure 8, which depicts a reconfigured topology among four servers with a total of 16 GPUs (EP degree equals 16).

(1) Each GPU looks up the topology to identify its intra-server communication delegation GPU for all communication pairs. MixNet prioritizes directly connected optical circuits over EPS. For example, the delegation GPU from server 0 to server 2 is GPU 2, as they are connected with optical circuits. However, to perform communications between server 0 and server 1, they have to use GPUs that are connected in the EPS.

(2) With the gateway information, each server performs an intra-host `gather`, gathering outbound data to the corresponding delegation GPUs via NVSwitch. Note that in §5.2, we balanced the number of NICs across each NUMA node to mitigate intra-host congestion when multiple links are provisioned between a server pair. MixNet aims to distribute the traffic load across delegation NICs as evenly as possible.

(3) Each server initiates the inter-host all-to-all communication across all delegation GPUs using NICs in both the EPS and OCS fabrics.

(4) Each server performs an intra-host all-to-all communication among local experts via NVSwitch.

(5) The delegation GPUs in each server `scatter` the received all-to-all data to its final destination.

As the dataflows in steps (3) and (4) do not interfere with each other, MixNet overlaps the communication in these two steps to reduce overall completion time.

5.4 Failure Handling

There are two categories of failures in distributed MoE training: network (NIC/link) failures and GPU failures. MixNet is tolerant to both transient and permanent failures during distributed MoE training.

Network fault resilience. In practice, each server in MixNet connects to both the global EPS fabric and a regional OCS domain using multiple NICs. Specifically, the EPS-side typically uses at least two NICs per server (see §4.2), providing redundancy for packet-switched communications. Based on prior reports [91], the failure rate of a single NIC or link during a training job is approximately 0.057%, making dual EPS NICs sufficient to reduce the probability of simultaneous failure to below 0.00003%. If both EPS NICs on a server fail—a rare but possible event—MixNet reroutes traffic through the OCS domain. Specifically, traffic destined for the failed NICs is first routed optically to a healthy peer, and then relayed through that peer's functional EPS interface. Similarly, the EPS can serve as a fallback path if OCS links or ports fail. This dual-path design ensures resilient connectivity, at the cost of some additional intra-cluster forwarding overhead.

GPU fault tolerance. MixNet also supports failure recovery when one or more GPUs become unavailable. We consider two realistic failure levels:

- Single-GPU failure. If a GPU fails during training, MixNet remaps the workload to a designated backup GPU. This is aligned with the design of high-availability systems such as NVL72, which reserve spare GPUs per group for fault tolerance. In MixNet, the backup GPU may be connected via either EPS or OCS: 1) If reachable via EPS, training resumes without additional routing. 2) If reachable only via OCS, MixNet forwards traffic through a peer GPU with optical connectivity to the backup GPU after topology reconfiguration, maintaining functional interconnect through minor topology adjustments.

- Full-node failure. A complete server failure (e.g., all 8 GPUs) requires a replacement node from the global backup pool. These backup nodes connect via EPS uplinks, ensuring network connectivity without reliance on regional OCS. Upon checkpoint restoration, MixNet resumes training with minimal disruption.

Runtime reconfiguration. To maintain topology validity in the presence of failures, MixNet's decentralized topology controllers detect communication failures and regenerate the OCS topology accordingly. This involves excluding failed nodes from the candidate set and recomputing optical mappings (see Algorithm 1 in §5.2). Because MixNet relies on regional control, such reconfigurations are localized and incur minimal global disruption. We evaluate the impact of various failure scenarios in §7.5.

6 MixNet Prototype

To evaluate MixNet, we build a fully functional prototype using commodity hardware[8] capable of training state-of-the-art MoE models.

Hardware setup. Figure 9 is a picture of our prototype, which contains four commodity servers, each equipped with eight Nvidia A100 GPUs and four Mellanox ConnectX-6 100G NICs. For each server, three NICs are connected to a Polatis millisecond-scale OCS [38], while the remaining one NIC is connected to a Nvidia

---

[8]Due to Nvidia's warranty restrictions, we cannot reconfigure the topology of the DGX SuperPod used in the measurement study (§3). Hence, we use commodity servers equipped with Nvidia GPUs for the testbed.
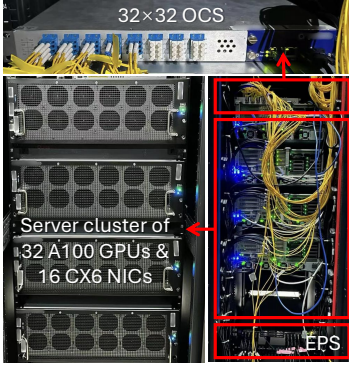
Figure 9: MixNet testbed using commodity hardware.

SN3700 Ethernet switch [35]. We use 100 Gbps QSFP28 optical transceivers and duplex LC fibers [18]. All NICs operate in RoCEv2 mode. Each server has a total of four NVLinks that connect two adjacent GPUs. Appendix C provides further details of our testbed.

Software stack. We implement MixNet's software stack using approximately 6K lines of code in C++, including the topology generator, the OCS controller, and the custom collective communication runtime supporting in-training topology reconfigurations. For DP and PP communication over the static EPS fabric, we use NCCL [26] to provide high-speed intra-host and inter-host all-reduce/point-to-point communications. For EP's all-to-all communications that involve both EPS and OCS, our custom collective communication runtime leverages RDMA for high-speed data transfer using the raw `ibverbs` library based on FuseLink [93]. We port the MixNet runtime to Python to integrate with Megatron-LM [98] for training real-world MoE models. Specifically, we have implemented communication primitives similar to those in `torch.dist` and expose them as `mixnet.all_to_all` and `mixnet.all_reduce`.

Training state-of-the-art MoE models. We use the prototype to train three state-of-the-art MoE models, and compare its performance with a baseline configuration where all the four NICs are connected to an Ethernet switch (the ideal switch baseline). Figure 10 shows that MixNet achieves comparable performance to the 4 × 100G EPS baseline. MixNet utilizes one NIC in the EPS fabric and configures the remaining three NICs in an optical circuit fabric (a total of 12 optical ports and 4 electronic ports). In contrast, the EPS baseline uses the four 100 Gbps ConnectX-6 NICs in a non-blocking EPS fabric with 16 electronic ports. MixNet's performance stems from its ability to efficiently provision high-bandwidth optical circuits for communication-intensive pairs in sparsely non-uniform all-to-all traffic, without compromising the transfer speed of DP and TP traffic. It is important to note that MixNet *does not alter the parallelization strategies* used in MoE training, but only *optimizes data transfer* through its architectural design and efficient circuit-switching algorithm. As a result, MixNet does not affect the training accuracy of MoE models.

7 Large-Scale Simulations

This section evaluates the performance of MixNet through large-scale simulation. We also present design space explorations on several factors, such as network scalability, EPS link options, and reconfiguration delays in Appendix §D.
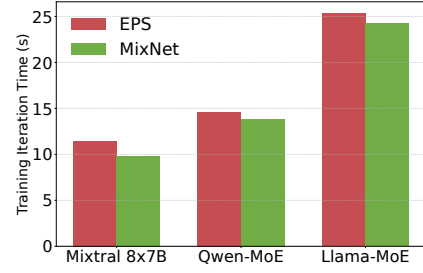

Figure 10: [Testbed] End-to-end training iteration time on our 32 GPU prototype.

7.1 Setup

Packet-level simulation methodology. The simulation process is divided into two phases. First, we develop a simulator on top of FlexFlow [12]. We extend FlexFlow to support pipeline parallelism and rectify its profiler to ensure that the profiled computation time aligns with the actual runtime on the testbed. The simulator is fed with the micro-batch size, an MoE model, and a specified parallelization strategy, and generates a task DAG that describes the computation and communication tasks for the cluster. Using this DAG, we then utilize an event-driven packet-level simulator based on `htsim` [16], which simulates packet-based communication between GPUs. The link propagation delay is set to 1 μs. We set the number of NICs and GPUs per server to 8, with each NIC having a bandwidth of B. In our setup, each server has eight GPUs, interconnected via a high-speed NVSwitch (900 GB/s), and eight NICs, reflecting typical configurations used in production environments. The training process for the MoE model is simulated across multiple iterations. The details of used models and parallelization strategies are presented in the Appendix D.1.

Simulated GPU interconnect fabrics. We compare the performance of MixNet with the following interconnects:

- **MixNet (this work).** In MixNet, each server connects two NICs to the EPS fabric using a fat-tree topology and connects the remaining six NICs to the OCS fabric by default. Following the architecture of the regionally reconfigurable high-bandwidth domain in MixNet, the optical circuit switch only needs to connect the GPUs within a single EP group, which is a maximum of 64 GPUs in our configuration. This can be easily supported by commodity OCS technologies (Table 2). MixNet blocks the network for 25 ms during the reconfiguration of the OCS for the first all-to-all communication in the forward pass and hides the reconfiguration time during computation for subsequent all-to-all communications, as discussed in §5.1.
- **Fat-tree [44].** We consider a 1:1 non-blocking *Fat-tree* network.
- **OverSub. Fat-tree.** We compare MixNet with a Fat-tree interconnect with the 3:1 over-subscription ratio.
- **Rail-optimized [11].** It has been the recommended GPU interconnect used by Nvidia. It differs from the fat-tree by connecting GPUs of the same rank to the same ToR switch, providing low latency for GPUs within the same rail.
- **TopoOpt [107].** The state-of-the-art optical interconnect that co-optimizes both model parallelization and network topology to minimize communication overhead. For TopoOpt, all NICs are *optimistically* connected via a large and flat optical patch panel.
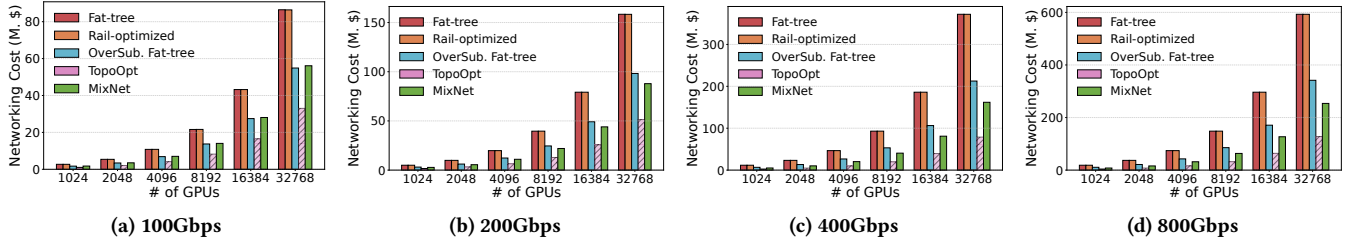
(a) 100Gbps   (b) 200Gbps   (c) 400Gbps   (d) 800Gbps

Figure 11: [Simulation] Networking cost analysis.

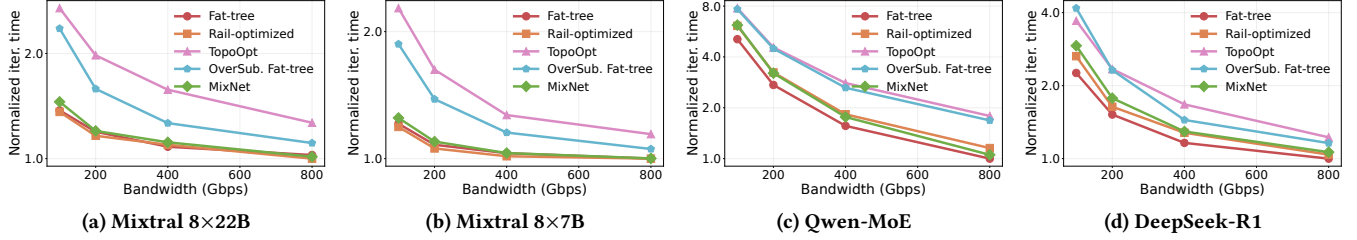(a) Mixtral 8×22B   (b) Mixtral 8×7B   (c) Qwen-MoE   (d) DeepSeek-R1

Figure 12: [Simulation] Training speed-ups in a cluster of 128 servers with 1024 GPUs.

7.2 Networking Cost Analysis

We present the cost analysis of MIXNET in Figure 11, using a common production setup where each server contains 8 GPUs, following the same methodology as [107]. The networking cost is analyzed with link bandwidths from 100 Gbps to 800 Gbps across different cluster sizes. It is important to note that we only account for the number of *actually used* switch ports in calculating the cost, as the cluster may not fit perfectly within a fat-tree/rail-optimized topology with a reasonable K. More details on the cost of each networking component can be found in Appendix D.2.

First, compared to the non-blocking Fat-tree and Rail-optimized topologies, MIXNET reduces networking costs by an average of 2.0×, as it organizes its high-bandwidth domain using OCS interconnects, which is significantly cheaper than EPS fabrics at high link bandwidth. Specifically, as shown in Figure 11c, MIXNET's OCS fabric incurs 2.3× lower cost on average than fat-tree topology at 400 Gbps.

Second, we acknowledge that MIXNET incurs slightly higher expenses than TopoOpt at the cluster size of 128 servers (1024 GPUs). This is because: 1) MIXNET requires EPS fabric to maintain global network-wide connectivity, and 2) MIXNET's high-bandwidth domains assume millisecond-level reconfigurable OCS to adapt to runtime MoE traffic, which is more expensive than TopoOpt's slowly reconfigurable patch panel. However, TopoOpt requires a multi-tier patch panel fabric to form a network of more than 1K GPUs. Achieving this requires extensive patch panel ports and expensive long-reach transceivers to compensate for the insertion loss of optical signals across multiple switching layers. As a result, it remains unclear whether TopoOpt is able to interconnect such large clusters and maintain its cost-efficiency.

7.3 Performance: Training Speed Ups

This section compares the end-to-end training iteration time of MIXNET against other interconnects across four MoE models on the cluster with 128 servers and 1024 GPUs.

Figure 12a compares the training iteration time of various interconnects for the Mixtral 8×22B model. Due to its efficient bandwidth allocation, we observe that MIXNET achieves performance very close to the ideal Fat-tree and Rail-optimized topologies. In particular, with a TP degree of 8, MIXNET provides direct optical circuits for almost all high-traffic server pairs during all-to-all communication (24 optical circuits for 8 EP participants). Compared to TopoOpt, MIXNET reduces the training iteration time by 1.5× on average, as TopoOpt's static topology cannot adapt to real-time traffic variations. In addition, we observe that MIXNET outperforms the over-subscribed fat-tree by up to 1.6×. Figure 12b shows a similar trend for the Mixtral 8×7B model, with MIXNET reducing the iteration time by 1.4× on average compared to TopoOpt. Both Mixtral models show diminishing returns from increased link bandwidth, as they are computation-bound at a micro-batch size of 8 (Figure 3). At higher bandwidths, the communication overhead shrinks, narrowing the performance gap between MIXNET and others. The results for larger batch sizes of these Mixtral models are in the Appendix D.4.

Figure 12c and Figure 12d compare MIXNET with other interconnects on Qwen-MoE (32 experts with 32-way EP) and DeepSeek-R1 (256 experts with 64-way EP), both of which use larger numbers of experts and higher EP degrees than Mixtral models. We observe that MIXNET achieves performance comparable to Rail-optimized and Fat-tree topologies, and outperforms TopoOpt by 1.5× on average in Qwen-MoE and 1.3× in DeepSeek-R1. Compared to Fat-tree, MIXNET exhibits slightly larger performance gaps at low bandwidths as the number of experts increases (e.g., 32 in Qwen-MoE vs. 8 in Mixtral 8×22B). This is because more bandwidth-intensive GPU pairs require dedicated optical circuits, which slightly exceeds MIXNET's default optical fanout under the 8-GPU setup (6 circuits for 32 EP participants). Nevertheless, owing to the sparsity of EP traffic, MIXNET narrows the performance gap as link bandwidth increases. We also observe that, unlike Qwen-MoE, MIXNET shows smaller performance gains on DeepSeek-R1 as bandwidth increases. This is because, although DeepSeek-R1 uses a higher EP degree, its larger model size results in a lower communication-to-computation ratio, thus reducing the benefit of additional bandwidth.
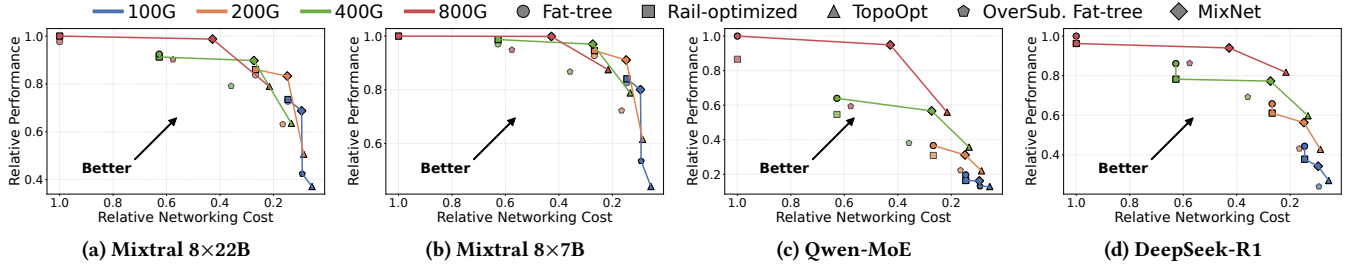
Figure 13: [Simulation] Performance-cost comparison of different interconnects on four state-of-the-art MoE models.
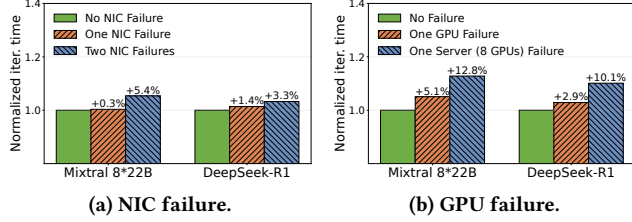

Figure 14: [Simulation] Failure resiliency of MixNet.

7.4 Cost Efficiency: Pareto Front Analysis

To better capture the trade-offs between networking cost and performance, we present the Pareto Front analysis of different interconnects in Figure 13. This approach offers a more balanced view that avoids favoring low-cost yet low-performance designs (e.g., TopoOpt or over-subscribed topologies), which may not be practically useful despite their low cost. We observe that MixNet consistently defines the Pareto Front and significantly outperforms Fat-tree and Rail-optimized across all four evaluated models in terms of cost efficiency, which is quantified as a performance-per-dollar metric (inverse of training iteration time normalized by networking cost). At 100 Gbps link bandwidth, MixNet achieves 1.2× to 1.5× higher cost-efficiency compared to Fat-tree, with Mixtral 8×7B showing the highest improvement. Moreover, MixNet outperforms Rail-optimized by 1.4× to 1.5×. At 200 Gbps link bandwidth, At 200 Gbps, the advantage grows to 1.4× to 1.8× over Fat-tree and 1.7× to 1.9× over Rail-optimized. For 400 Gbps networks, MixNet demonstrates even higher cost-efficiency gains: 2.3× for Mixtral 8×7B, 2.2× for Mixtral 8×22B. For DeepSeek-R1, MixNet improves the training cost efficiency by 2.1× compared to Fat-tree.

Notably, MixNet demonstrates strong cost-efficiency across varying link bandwidths, maintaining a 2.0×–2.4× advantage over Fat-tree and 2.2×–2.6× over Rail-optimized even at forward-looking 800 Gbps networks. These gains stem from two factors: 1) MixNet directly connects high-traffic regional GPU pairs with optical circuits, reducing the need for excessive electrical switches and optical transceivers in Fat-tree; 2) unlike Fat-tree's underutilized uniform bisection bandwidth, MixNet optimizes resource allocation to match MoE's sparse, non-uniform communication patterns, cutting hardware costs while sustaining high performance.

7.5 Failure Resiliency

Following §5.4, we evaluate the impact of different failure cases on training performance using the Mixtral 8×22B and DeepSeek-R1 models on a 1024-GPU cluster with 400 Gbps link bandwidth.

NIC failures. Figure 14a shows the training performance of MixNet

when encountering NIC failures, where indirect forwarding is employed to bypass the failed NIC. We observe that MixNet maintains acceptable performance, with only a 3.3% increase in total training time for the DeepSeek-R1 model. This minor overhead is attributed to the inherent network-wide backup policy, where EPS and OCS provide mutual fallback paths, and the intra-host scale-up domain offers sufficient forwarding bandwidth.

GPU failures. We also evaluate the impact of GPU failures, as shown in Figure 14b. For example, in the Mixtral 8×22B model, working around a single failed GPU via a regional backup GPU with OCS-based indirect forwarding leads to a 5.1% increase in total training time. This additional overhead arises because the TP communication in Mixtral 8×22B occurs between servers through the low-bandwidth scale-out fabric, rather than the original high-bandwidth intra-host scale-up domain. In the more severe case of a full server failure (all 8 GPUs), the performance degradation is higher, as all EP traffic to and from the backup GPUs has to traverse the two connected EPS NICs, as discussed in §5.4. Similarly, replacing a fully failed GPU server in the DeepSeek-R1 model results in a 6.5% performance degradation due to the constrained network connectivity for EP traffic in this scenario.

In summary, MixNet exhibits resilience to both network and GPU failures, consistently delivering acceptable performance across the evaluated scenarios.

8 Look Ahead: High-Radix Scale-Up Domains

So far, we have discussed MixNet as a production-ready system only using commodity OCS (Table 2) and networking equipment (e.g., NICs, transceivers) that are agnostic of local scale-up high-bandwidth domains. In this section, we present a look-ahead study and extend the concept of MixNet to better support the emerging trend of high-radix scale-up domains (e.g., Nvidia GB200 NVL72 system interconnects 72 GPUs in a single scale-up high-bandwidth domain)[9]. In these systems, EP's all-to-all communications consume more bandwidth in the scale-up high-bandwidth domain than that of scale-out fabrics. Therefore, different from conventional settings, we consider the port of regional OCS to be directly attached to the GPU chip, which is enabled by co-packaged optical I/O [19, 41]. Based on this, we envision MixNet evolving towards a regional OCS architecture capable of directly receiving optical signals from xPUs (e.g., GPU, TPU, NPU, etc.), as illustrated in Figure 15. Such an optical switching architecture would enable a long-reach, high-speed (e.g., 4 Tbps or more) *regional* fabric that sits at the boundary of scale-up and scale-out domains, supporting

---

[9]MixNet works with NVL72 by splitting scale-out NICs between OCS and EPS.
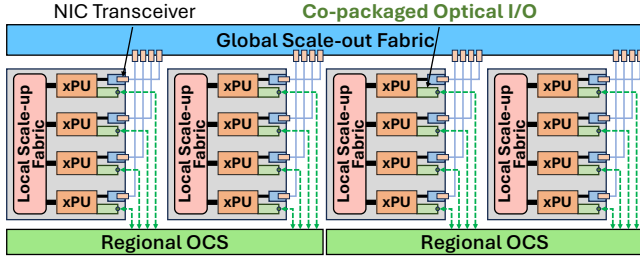
Figure 15: MixNet with co-packaged optical ports directly attached to xPUs.
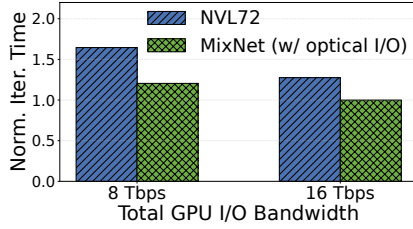

Figure 16: [Simulation] Performance comparison with a 2048-GPU cluster of NVL72 systems.

larger-scale switching fabric compared to state-of-the-art NVL72 copper-based interconnects.

Using packet-level simulations[10], we compare the training iteration time of MixNet (with optical I/O) with Nvidia GB200 NVL72 [28]. We consider a 2048-GPU cluster training the state-of-the-art MoE model, DeepSeek-V3 [9] (with an EP degree of 128, a PP degree of 16, and a micro-batch size of 240)[11]. The NVL72 system is modeled with a per-GPU NVLink bandwidth of 7.2 Tbps [28, 36] in the scale-up domain, alongside 800 Gbps Ethernet for scale-out communication. We assign 64 GPUs within each NVL72 domain to align with parallelism allocation constraints[12]. To ensure a fair comparison, we match MixNet's total GPU bandwidth in MixNet to NVL72's 8 Tbps, allocating 800 Gbps to Ethernet and splitting the remaining bandwidth equally between NVLink and MixNet (with optical I/O)'s regional OCS.

Figure 16 presents the normalized training iteration time. We observe that MixNet (with optical I/O) lowers iteration time by 1.3× compared to NVL72, as it offloads intensive cross-node communication overhead to regional reconfigurable OCS. Yet NVL72 has to use the scale-out networks for cross-node transfers. Additionally, MixNet continues to deliver performance gains even when GPU total I/O bandwidth scales to 16 Tbps in next-generation systems.

9 Discussion

Additional training parallelisms. Other potential parallelisms for LLM training, such as context parallelism [8], optimizes the computation of long sequences in the attention module. Notably, its traffic is static and does not overlap with EP's all-to-all communications. MixNet is ready to accommodate this traffic with its

---

[10]Same setup as discussed in Section 7.1
[11]Same parameters as in [9], with a larger EP size to explore larger training batch sizes and better expert capacities.
[12]In most production practices, only 64 out of 72 GPUs in NVL72 are used because training parallelisms are on the orders of 2 (e.g., 4, 8, 16, 32, etc.).

intra-host scale-up fabric or reconfigurable OCS fabric.

Support for model scaling. Currently, MoE models have two scaling trends. On the one hand, recent models like Mixtral (8×7B and 8×22B) and Grok-2 contain a small number of *huge* experts. This fits into the design of MixNet that contains multiple scale-up networks within each reconfigurable high-bandwidth domain to support huge expert layers. On the other hand, some models [9, 40, 55] choose to involve a large number of *small* experts. Therefore, the radius of the EP all-to-all communications does not grow linearly with the number of experts, as multiple small experts will be packed into one GPU for training efficiency. For example, the state-of-the-art MoE model DeepSeek-V3 [9], which uses an EP degree of 64 and a PP degree of 16, can be accommodated within multiple 64-port OCSes in an MixNet fabric.

NIC's optical fanout options. So far, we consider MixNet's NIC to be equipped with a dedicated port for each TX/RX channel connecting to the regional reconfigurable OCS. In practice, the MixNet architecture is also compatible with other optical fanout options. For example, bi-directional transceivers using optical circulators merge TX and RX ports into one fiber port [81, 90], and optical breakout cables separate multiple fiber channels within a single MPO/MTP port into individual LC connections [5].

Supporting models beyond MoE. While MixNet is primarily optimized for large-scale MoE training, it is also applicable to other non-MoE LLMs, such as GPT-3 [10] and LLaMA [20]. The core benefit of MixNet is the ability to reconfigure the network topology for non-uniform traffic patterns at a regional radius between the local scale-up networks and global scale-out networks. By blurring the boundaries of scale-up and scale-out networks using reconfigurable OCS, MixNet supports cases where non-MoE LLMs' non-uniform traffic patterns require optimized network topologies. For example, a ring-based topology serves DP all-reduce traffic in a more cost-efficient way than other topologies. Meanwhile, emerging LLM architectures in the ML community propose to build unconventional MoE-like models on top of dense LLMs that exhibit non-uniform traffic patterns at regional scales [78, 102], highlighting the need of MixNet's reconfigurable topologies with for distributed training.

Support for Multi-Tenant Training Different from conventional small-scale training jobs, running multiple concurrent MoE training jobs on the same set of GPUs is not preferred in industry practice. For example, Alibaba HPN [91] highlights that a single training job already occupies 3K GPUs. Meta shares the experience of training Llama-3 405B model with 15K GPUs [56]. DeepSeek-V3 was trained with 2K GPUs [9]. Moreover, xAI discloses that they train the Grok model with 100K GPUs [15]. Nevertheless, if needed, MixNet supports cluster-wide multi-tenancy because its regional OCS high-bandwidth domains can be reconfigured as isolated sub-networks for each small-scale tenant job.

Comparability with other OCS technologies. MixNet's regional reconfiguration design is orthogonal to the selection of OCS technologies. We note the emergence of new OCS technologies, such as iPronics [17], which offer microsecond-scale reconfiguration latencies and low per-port costs comparable to MEMS-based OCS. These advancements present a promising opportunity for MixNet to reconfigure the topology for every all-to-all communication phase in MoE training, further improving training performance

while maintaining cost efficiency. Furthermore, in the main text, we mainly discuss OCS technologies that can actively switch optical signals at the network core. Other emerging OCS techniques, such as tunable laser at endpoints (e.g., transceivers) with arrayed waveguide grating router (AWGR)-based optical switch that pushes reconfigurability into the network endpoints while leaving the network core to be passive [48, 112, 115], are also compatible with MixNet. For example, assigning dedicated wavelengths to critical expert pairs eliminates bandwidth contention for all-to-all EP communication while organizing ring-based topologies for all-reduced communication of TP. However, these AWGR and endpoint-based OCS are primarily in the research stage and not commercially ready for mass production yet, leaving a longer way to be integrated into today's computer systems.

10 Related work

Network architecture for distributed training. There has been a series of proposals on network architecture for large-scale distributed training in both industry and academia. Notably, ByteDance MegaScale [70] uses a Clos-based topology interconnecting more than 10,000 GPUs. Meta [61] shared its insights on the tuning of routing strategies, optimizing collective operations, and strengthening network resilience to design large-scale RoCE networks for AI training. Alibaba HPN [91] introduced a dual-plane network to enhance resilience to failure. Nvidia developed a rail-optimized network [11] to fully leverage the heterogeneous networking capabilities of different fabrics, which has been widely adopted in its computing clusters. Besides these industry proposals, academic researchers further proposed a rail-only design that removes the core switching layer for inter-rail GPUs, albeit at the cost of degrading cross-rail traffic performance [106].

Reconfigurable networks for distributed training. SiP-ML [72] explores silicon photonics for high-bandwidth optical interconnects for optimizing static communications from traditional DP and model-parallel communications. TopoOpt [107] proposed optimizing the network topology for distributed training jobs using one-shot optically reconfigurable networks. To the best of our knowledge, MixNet is the first to propose an optically reconfigurable network using commodity hardware for MoE training.

Reconfigurable data center networks. There has been a decades-long research agenda focused on designing reconfigurable networks for data centers [45–48, 50, 51, 57, 58, 62, 67, 79, 80, 84, 86, 87, 89, 92, 95, 105, 107–110, 113, 114]. These proposals target generic data center networks, which are not optimized to provide cost-efficient solutions for large-scale MoE training. In particular, traffic-oblivious solutions, such as RotorNet [85, 86] and Opera [84], result in suboptimal performance for MoE training, as they cannot deliver timely transfers for bandwidth-intensive all-to-all traffic. Meanwhile, the hardware innovations like Sirius [48] feature faster optical switching latency, hence allowing MixNet to achieve much faster topology reconfiguration delay. Shoal [100] proposes reconfigurable electronic circuit switches in rack-scale networks and is not unsuitable for large-scale MoE training.

OCS deployments in data centers. Google has pioneered the deployments of OCS technology in production data centers. Over the years, they have transitioned from electronic packet switching (EPS)

to hybrid optical-electrical solutions [44, 63, 68, 81, 90, 101, 118]. Early systems like Jupiter [101] used Clos topologies with EPS to achieve scalability and high bandwidth but faced limitations in power efficiency and adaptability to dynamic workloads. Subsequent innovations in Jupiter Evolving [90] introduced OCS to complement EPS in a hybrid architecture that improved cost and power efficiency. Most recently, Lightwave Fabrics from Google [81] used reconfigurable MEMS-based OCS to establish topologies for TPU supercomputers [71]. In particular, it performs one-shot topology reconfiguration prior to training such that the topology remains fixed throughout the training process. In contrast, MixNet proposes runtime topology reconfiguration during training to accommodate dynamic MoE traffic patterns. Furthermore, TPU's optical interconnect only reconfigures links between 4×4×4 cubes, while the intra-cube topology remains static (3D Torus) during OCS reconfiguration. This fixed intra-cube structure is ill-suited for dynamic all-to-all communication, which requires multi-hop forwarding in response to changing and sparse traffic demands [71].

OCS for scale-up interconnects. The regionally reconfigurable OCS design in MixNet is applicable to expand the high-bandwidth circuit-switched connectivity enabled by NVLink and NVSwitch. For example, forward-looking techniques like Lightmatter passage optical interconnect [19] benefit from MixNet by reconfiguring all GPUs within the same EP group at the chip level. This would enable high-radix on-chip photonic communication with massive bandwidth to handle the communication-intensive demands of both TP and EP in MoE training.

Emerging OCS hardware devices and systems. There are recent proposals on designing novel OCS hardware at server scale for chip-to-chip interconnects [19, 49, 96, 116]. We would like to highlight that these emerging devices require a system-level design to be practical. Similar to other system-level work [73], MixNet's regional OCS and its algorithmic designs are compatible with this vibrant line of exploration on novel OCS hardware.

11 Conclusion

This paper presented MixNet, a novel reconfigurable fabric using commodity hardware for large-scale MoE training. At the core of MixNet is the design and implementation of the regionally reconfigurable high-bandwidth domain based on distributed OCS. Through proof-of-concept prototype and large-scale packet simulations, we show that MixNet delivers distributed training performance comparable to state-of-the-art electrical and optical interconnects while significantly reducing networking cost.

Ethics: This work does not raise any ethical issues.

Acknowledgments

References

[1] [n. d.]. 100GBASE-SR4 850nm 100m DOM MPO-12/UPC MMF Optical Transceiver Module. https://www.fs.com/products/48354.html. ([n. d.]).

[2] [n. d.]. 200GBASE-SR4 850nm 100m DOM MPO-12/UPC MMF Optical Transceiver Module. https://www.fs.com/products/139696.html. ([n. d.]).

[3] [n. d.]. 400GBASE-SR4 PAM4 850nm 100m DOM MPO-12/APC MMF Optical Transceiver Module. https://www.fs.com/products/226577.html. ([n. d.]).

[4] [n. d.]. All-to-all traffic demand collection in Megatron-LM. https://github.com/NVIDIA/Megatron-LM/blob/461b06cd6d1fb4a625cebdbca499dac9484087fc/megatron/core/transformer/moe/token_dispatcher.py#L432. ([n. d.]).

[5] [n. d.]. Breakout optical cables. https://arubanetworking.hpe.com/techdocs/Switches/xcvrs/xcvr_guide/Content/Chp_overview/spl-opt-cab.htm. ([n. d.]).

[6] [n. d.]. Calient Optical Circuit Switch. www.calient.net. ([n. d.]).

[7] [n. d.]. Calix® 100-04482 Compatible TAA 10GBs XGS-PON OLT XFP Transceiver with Burst Mode (SMF, 1577nmTx/1270nmRx, SC, N1, DOM). https://www.addonnetworks.com/products/transceivers/calix/xfp/10gbase/100-04482-ao. ([n. d.]).

[8] [n. d.]. Context parallelism overview. https://docs.nvidia.com/megatron-core/developer-guide/latest/api-guide/context_parallel.html. ([n. d.]).

[9] [n. d.]. DeepSeek-V3. https://github.com/deepseek-ai/DeepSeek-V3/blob/main/DeepSeek_V3.pdf. ([n. d.]).

[10] [n. d.]. DeepSpeed. https://openai.com/index/gpt-3-apps/. ([n. d.]).

[11] [n. d.]. Doubling all2all Performance with NVIDIA Collective Communication Library 2.12. https://developer.nvidia.com/blog/doubling-all2all-performance-with-nvidia-collective-communication-library-2-12/. ([n. d.]).

[12] [n. d.]. FlexFlow. https://github.com/flexflow/FlexFlow. ([n. d.]).

[13] [n. d.]. Generic Compatible 800GBASE-SR8 QSFP-DD PAM4 850nm 50m DOM MPO-16/APC MMF Optical Transceiver Module. https://www.fs.com/products/200921.html?attribute=93760&id=3569240. ([n. d.]).

[14] [n. d.]. Grok. https://x.ai/blog/grok-1.5v. ([n. d.]).

[15] [n. d.]. Grok training cluster. https://www.windowscentral.com/software-apps/elon-musk-flaunts-the-most-powerful-training-cluster-in-the-world-that-will-transform-grok-into-the-most-powerful-ai-by-december-to-take-on-microsoft-and-openai. ([n. d.]).

[16] [n. d.]. htsim simulator. https://github.com/nets-cs-pub-ro/NDP/wiki/NDP-Simulator. ([n. d.]).

[17] [n. d.]. IPRONICS One. https://ipronics.com/ipronics-optical-networking-engine/. ([n. d.]).

[18] [n. d.]. LC UPC to LC UPC, Duplex, 2 Fibers. https://www.fs.com/products/40191.html. ([n. d.]).

[19] [n. d.]. LightMatter. https://lightmatter.co/products/passage/. ([n. d.]).

[20] [n. d.]. LLaMA. https://llama.meta.com. ([n. d.]).

[21] [n. d.]. Mixtral 8x22B. https://huggingface.co/mistralai/Mixtral-8x22B-Instruct-v0.1. ([n. d.]).

[22] [n. d.]. Mixtral 8x7B-Instruct-v0.1. https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1. ([n. d.]).

[23] [n. d.]. Mixtral of experts. https://mistral.ai/news/mixtral-of-experts/. ([n. d.]).

[24] [n. d.]. MSN4700-WS2FC, NVIDIA® Mellanox Spectrum-3 Based 32-Port Ethernet L3 Data Center Switch, 32 x 400Gb QSFP-DD. https://www.colfaxdirect.com/store/pc/viewPrd.asp?idproduct=4135. ([n. d.]).

[25] [n. d.]. Nano-Second Speed PLZT Photonics. http://epiphotonics.com/products.html. ([n. d.]).

[26] [n. d.]. NVIDIA Collective Communications Library (NCCL). https://developer.nvidia.com/nccl. ([n. d.]).

[27] [n. d.]. NVIDIA DGX SuperPOD. https://www.nvidia.com/en-us/data-center/dgx-superpod/. ([n. d.]).

[28] [n. d.]. NVIDIA GB200 NVL72 Delivers Trillion-Parameter LLM Training and Real-Time Inference. https://developer.nvidia.com/blog/nvidia-gb200-nvl72-delivers-trillion-parameter-llm-training-and-real-time-inference/. ([n. d.]).

[29] [n. d.]. NVIDIA Mellanox MCX515A-CCAT ConnectX®-5 EN Network Interface Card, 100GbE Single-Port QSFP28. https://www.fs.com/products/119648.html. ([n. d.]).

[30] [n. d.]. NVIDIA Mellanox MCX653105A-HDAT ConnectX®-6 InfiniBand/VPI Adapter Card 200GbE/HDR, Single-Port QSFP56. https://www.fs.com/products/168437.html. ([n. d.]).

[31] [n. d.]. NVIDIA Mellanox MCX75310AAS-NEAT ConnectX®-7 InfiniBand/VPI Adapter Card 400GbE/NDR, Single-Port OSFP. https://www.fs.com/products/212161.html. ([n. d.]).

[32] [n. d.]. NVIDIA Mellanox Spectrum-4 SN5600 800G 64-Port 51.2Tb/s 2U Data Center Switch. https://fireowls.com/products/920-9n42f-00ri-7c0-nvidia-spectrum-sn5600-ethernet-switch?srsltid=AfmBOoqC5-OQefz1MpAi_2QkzW2tnaCTXA_xGtgoj9b3NphWUXw8dBrl. ([n. d.]).

[33] [n. d.]. NVIDIA NVLink and NVLink Switch. https://www.nvidia.com/en-us/data-center/nvlink/. ([n. d.]).

[34] [n. d.]. NVIDIA NVSwitch Technical Overview. https://images.nvidia.com/content/pdf/nvswitch-technical-overview.pdf. ([n. d.]).

[35] [n. d.]. NVIDIA Spectrum SN3700. https://marketplace.nvidia.com/en-us/enterprise/networking/sn3700/. ([n. d.]).

[36] [n. d.]. NVLink Switch Specifications. https://www.nvidia.com/en-us/data-center/nvlink/?ncid=no-ncid#nvlink-switch-specifications. ([n. d.]).

[37] [n. d.]. Polatis Optical Circuit Switch. https://www.polatis.com/series-7000-384x384-port-software-controlled-optical-circuitswitch-sdn-enabled.asp. ([n. d.]).

[38] [n. d.]. Polatis Optical Switches. http://www.polatis.com/series-7000-384x384-port-software-controlled-optical-circuit-switch-sdn-enabled.asp. ([n. d.]).

[39] [n. d.]. PON/Burst-mode Transceivers. https://oesolutions.com/product-type/burst-mode-transceivers/. ([n. d.]).

[40] [n. d.]. Qwen1.5-MoE-A2.7B. https://huggingface.co/Qwen/Qwen1.5-MoE-A2.7B. ([n. d.]).

[41] [n. d.]. Rethinking AI Architectures with Optical I/O. https://www.windowscentral.com/software-apps/elon-musk-flaunts-the-most-powerful-training-cluster-in-the-world-that-will-transform-grok-into-the-most-powerful-ai-by-december-to-take-on-microsoft-and-openai. ([n. d.]).

[42] [n. d.]. Telescent G4 Network Topology Manager. https://www.telescent.com/products. ([n. d.]).

[43] [n. d.]. XVERSE-MoE-A36B MoE base model. https://huggingface.co/xverse/XVERSE-MoE-A36B. ([n. d.]).

[44] Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat. 2008. A scalable, commodity data center network architecture. In *ACM SIGCOMM Computer Communication Review*, Vol. 38. ACM New York, NY, USA, 63–74.

[45] Daniel Amir, Nitika Saran, Tegan Wilson, Robert Kleinberg, Vishal Shrivastav, and Hakim Weatherspoon. 2024. Shale: A Practical, Scalable Oblivious Reconfigurable Network. In *Proceedings of the ACM SIGCOMM 2024 Conference (ACM SIGCOMM '24)*. https://doi.org/10.1145/3651890.3672248

[46] Daniel Amir, Tegan Wilson, Vishal Shrivastav, Hakim Weatherspoon, and Robert Kleinberg. 2023. Poster: Scalability and congestion control in oblivious reconfigurable networks. In *Proceedings of the ACM SIGCOMM 2023 Conference*. 1138–1140.

[47] Daniel Amir, Tegan Wilson, Vishal Shrivastav, Hakim Weatherspoon, Robert Kleinberg, and Rachit Agarwal. 2022. Optimal oblivious reconfigurable networks. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. 1339–1352.

[48] Hitesh Ballani, Paolo Costa, Raphael Behrendt, Daniel Cletheroe, Istvan Haller, Krzysztof Jozwik, Fotini Karinou, Sophie Lange, Kai Shi, Benn Thomsen, et al. 2020. Sirius: A flat datacenter network with nanosecond optical switching. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*. 782–797.

[49] Darius Bunandar, Shashank Gupta, Jessie Rosenberg, Clifford Chao, Kuang Liu, and Nicholas C Harris. 2024. Optical communication substrate using glass interposer. (Oct. 24 2024). US Patent App. 18/638,820.

[50] Kai Chen, Ankit Singla, Atul Singh, Kishore Ramachandran, Lei Xu, Yueping Zhang, Xitao Wen, and Yan Chen. 2012. OSA: An Optical Switching Architecture for Data Center Networks with Unprecedented Flexibility. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*. USENIX Association.

[51] Li Chen, Kai Chen, Zhonghua Zhu, Minlan Yu, George Porter, Chunming Qiao, and Shan Zhong. 2017. Enabling Wide-Spread Communications on Optical Fabric with MegaSwitch. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*. USENIX Association, Boston, MA, 577–593. https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/chen

[52] Tianyu Chen, Shaohan Huang, Yuan Xie, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. 2022. Task-specific expert pruning for sparse mixture-of-experts. *arXiv preprint arXiv:2206.00277* (2022).

[53] Kari Clark, Hitesh Ballani, Polina Bayvel, Daniel Cletheroe, Thomas Gerard, Istvan Haller, Krzysztof Jozwik, Kai Shi, Benn Thomsen, Philip Watts, et al. 2018. Sub-nanosecond clock and data recovery in an optically-switched data centre network. In *2018 European Conference on Optical Communication (ECOC)*. IEEE, 1–3.

[54] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan

Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhichena Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. (2025). arXiv:cs.CL/2501.12948 https://arxiv.org/abs/2501.12948

[55] DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jin Chen, Jingyang Yuan, Junjie Qiu, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruizhe Pan, Runxin Xu, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Size Zheng, T. Wang, Tian Pei, Tian Yuan, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Liu, Xin Xie, Xingkai Yu, Xinnan Song, Xinyi Zhou, Xinyu Yang, Xuan Lu, Xuecheng Su, Y. Wu, Y. K. Li, Y. X. Wei, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Zheng, Yichao Zhang, Yiliang Xiong, Yilong Zhao, Ying He, Ying Tang, Yishi Piao, Yixin Dong, Yixuan Tan, Yiyuan Liu, Yongji Wang, Yongqiang Guo, Yuchen Zhu, Yuduan Wang, Yuheng Zou, Yukun Zha, Yunxian Ma, Yuting Yan, Yuxiang You, Yuxuan Liu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhewen Hao, Zhihong Shao, Zhiniu Wen, Zhipeng Xu, Zhongyu Zhang, Zhuoshu Li, Zihan Wang, Zihui Gu, Zilin Li, and Ziwei Xie. 2024. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. (2024). arXiv:cs.CL/2405.04434 https://arxiv.org/abs/2405.04434

[56] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783* (2024).

[57] Nathan Farrington, Alex Forencich, George Porter, P-C Sun, Joseph E Ford, Yeshaiahu Fainman, George C Papen, and Amin Vahdat. 2013. A multiport microsecond optical circuit switch for data center networking. *IEEE Photonics Technology Letters* 25, 16 (2013), 1589–1592.

[58] Nathan Farrington, George Porter, Sivasankar Radhakrishnan, Hamid Hajabdolali Bazzaz, Vikram Subramanya, Yeshaiahu Fainman, George Papen, and Amin Vahdat. 2010. Helios: a hybrid electrical/optical switch architecture for modular data centers. In *Proceedings of the ACM SIGCOMM 2010 Conference (SIGCOMM '10)*. Association for Computing Machinery. https://doi.org/10.1145/1851182.1851223

[59] William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research* 23, 120 (2022), 1–39.

[60] Klaus-Tycho Foerster, Manya Ghobadi, and Stefan Schmid. 2018. Characterizing the algorithmic complexity of reconfigurable data center architectures. In *Proceedings of the 2018 Symposium on Architectures for Networking and Communications Systems*. 89–96.

[61] Adithya Gangidi, Rui Miao, Shengbao Zheng, Sai Jayesh Bondu, Guilherme Goes, Hany Morsy, Rohit Puri, Mohammad Riftadi, Ashmitha Jeevaraj Shetty, Jingyi Yang, et al. 2024. RDMA over Ethernet for Distributed Training at Meta Scale. In *Proceedings of the ACM SIGCOMM 2024 Conference*. 57–70.

[62] Thomas Gerard, Kari Clark, Adam Funnell, Kai Shi, Benn Thomsen, Philip Watts, Krzysztof Jozwik, Istvan Haller, Hugh Williams, Paolo Costa, et al. 2021. Fast and uniform optically-switched data centre networks enabled by amplitude caching. In *2021 Optical Fiber Communications Conference and Exhibition (OFC)*. IEEE, 1–3.

[63] Dan Gibson, Hema Hariharan, Eric Lance, Moray McLaren, Behnam Montazeri, Arjun Singh, Stephen Wang, Hassan MG Wassel, Zhehua Wu, Sunghwan Yoo, et al. 2022. Aquila: A unified, low-latency fabric for datacenter networks. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*. 1249–1266.

[64] Albert Greenberg, James R Hamilton, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, David A Maltz, Parveen Patel, Sushant Sengupta, Jennifer Rexford, et al. 2009. VL2: A scalable and flexible data center network. In *ACM SIGCOMM Computer Communication Review*, Vol. 39. ACM New York, NY, USA, 51–62.

[65] Chuanxiong Guo, Hui Wu, Kai Tan, Lei Shiy, Yongguang Zhang, and Songnian Lu. 2009. Bcube: A high performance, server-centric network architecture for modular data centers. In *ACM SIGCOMM Computer Communication Review*, Vol. 39. ACM New York, NY, USA, 63–74.

[66] Chuanxiong Guo, Hui Wu, Kai Tan, Lei Shiy, Yongguang Zhang, and Songnian Lu. 2010. Dcell: A scalable and fault-tolerant network structure for data centers. In *ACM SIGCOMM Computer Communication Review*, Vol. 40. ACM New York, NY, USA, 63–74.

[67] Navid Hamedazimi, Zafar Qazi, Himanshu Gupta, Vyas Sekar, Samir R Das, Jon P Longtin, Himanshu Shah, and Ashish Tanwer. 2014. Firefly: A reconfigurable wireless data center fabric using free-space optics. In *Proceedings of the 2014 ACM conference on SIGCOMM*. 319–330.

[68] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jon Zolla, Urs Hölzle, Stephen Stuart, and Amin Vahdat. 2013. B4: experience with a globally-deployed software defined wan. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM (SIGCOMM '13)*. 3–14. https://doi.org/10.1145/2486001.2486019

[69] Yimin Jiang, Yibo Zhu, Chang Lan, Bairen Yi, Yong Cui, and Chuanxiong Guo. 2020. A unified architecture for accelerating distributed {DNN} training in heterogeneous {GPU/CPU} clusters. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*. 463–479.

[70] Ziheng Jiang, Haibin Lin, Yinmin Zhong, Qi Huang, Yangrui Chen, Zhi Zhang, Yanghua Peng, Xiang Li, Cong Xie, Shibiao Nong, et al. 2024. {MegaScale}: Scaling large language model training to more than 10,000 {GPUs}. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. 745–760.

[71] Norm Jouppi, George Kurian, Sheng Li, Peter Ma, Rahul Nagarajan, Lifeng Nai, Nishant Patil, Suvinay Subramanian, Andy Swing, Brian Towles, et al. 2023. Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*. 1–14.

[72] Mehrdad Khani, Manya Ghobadi, Mohammad Alizadeh, Ziyi Zhu, Madeleine Glick, Keren Bergman, Amin Vahdat, Benjamin Klenk, and Eiman Ebrahimi. 2021. SiP-ML: high-bandwidth optical network interconnects for machine learning training. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*. 657–675.

[73] Abhishek Vijaya Kumar, Arjun Devraj, Darius Bunandar, and Rachee Singh. 2024. A case for server-scale photonic connectivity. In *Proceedings of the 23rd ACM Workshop on Hot Topics in Networks*. 290–299.

[74] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668* (2020).

[75] Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. 2023. Merge, then compress: Demystify efficient SMoe with hints from its routing policy. *arXiv preprint arXiv:2310.01334* (2023).

[76] Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, et al. 2020. Pytorch distributed: Experiences on accelerating data parallel training. *arXiv preprint arXiv:2006.15704* (2020).

[77] Wenxue Li, Xiangzhou Liu, Yuxuan Li, Yilun Jin, Han Tian, Zhizhen Zhong, Guyue Liu, Ying Zhang, and Kai Chen. 2024. Understanding communication characteristics of distributed training. In *Proceedings of the 8th Asia-Pacific Workshop on Networking*. 1–8.

[78] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. 2024. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947* (2024).

[79] He Liu, Feng Lu, Alex Forencich, Rishi Kapoor, Malveeka Tewari, Geoffrey M. Voelker, George Papen, Alex C. Snoeren, and George Porter. 2014. Circuit Switching Under the Radar with REACToR. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*. USENIX Association, Seattle, WA, 1–15. https://www.usenix.org/conference/nsdi14/technical-sessions/presentation/liu_he

[80] He Liu, Matthew K Mukerjee, Conglong Li, Nicolas Feltman, George Papen, Stefan Savage, Srinivasan Seshan, Geoffrey M Voelker, David G Andersen, Michael Kaminsky, et al. 2015. Scheduling techniques for hybrid circuit/packet networks. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*. 1–13.

[81] Hong Liu, Ryohei Urata, Kevin Yasumura, Xiang Zhou, Roy Bannon, Jill Berger, Pedram Dashti, Norm Jouppi, Cedric Lam, Sheng Li, Erji Mao, Daniel Nelson, George Papen, Mukarram Tariq, and Amin Vahdat. 2023. Lightwave Fabrics: At-Scale Optical Circuit Switching for Datacenter and Machine Learning Systems. In *Proceedings of the ACM SIGCOMM 2023 Conference (ACM SIGCOMM '23)*. https://doi.org/10.1145/3603269.3604836

[82] Juncai Liu, Jessie Hui Wang, and Yimin Jiang. 2023. Janus: A unified distributed training framework for sparse mixture-of-experts models. In *Proceedings of the ACM SIGCOMM 2023 Conference*. 486–498.

[83] Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. 2024. Not All Experts are Equal: Efficient Expert Pruning and Skipping for Mixture-of-Experts Large Language Models. *arXiv preprint arXiv:2402.14800* (2024).

[84] William M Mellette, Rajdeep Das, Yibo Guo, Rob McGuinness, Alex C Snoeren, and George Porter. 2020. Expanding across time to deliver bandwidth efficiency and low latency. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. 1–18.

[85] William M Mellette, Alex Forencich, Rukshani Athapathu, Alex C Snoeren, George Papen, and George Porter. 2024. Realizing RotorNet: Toward Practical Microsecond Scale Optical Networking. In *Proceedings of the ACM SIGCOMM 2024 Conference*. 392–414.

[86] William M Mellette, Rob McGuinness, Arjun Roy, Alex Forencich, George Papen, Alex C Snoeren, and George Porter. 2017. Rotornet: A scalable, low-complexity, optical datacenter network. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. 267–280.

[87] William Maxwell Mellette, Glenn M Schuster, George Porter, George Papen, and Joseph E Ford. 2016. A scalable, partially configurable optical switch for data center networks. *Journal of Lightwave Technology* 35, 2 (2016), 136–144.

[88] Deepak Narayanan, Aaron Harlap, Amar Phanishayee, Vivek Seshadri, Nikhil R Devanur, Gregory R Ganger, Phillip B Gibbons, and Matei Zaharia. 2019. PipeDream: Generalized pipeline parallelism for DNN training. In *Proceedings of the 27th ACM symposium on operating systems principles*. 1–15.

[89] George Porter, Richard Strong, Nathan Farrington, Alex Forencich, Pang Chen-Sun, Tajana Rosing, Yeshaiahu Fainman, George Papen, and Amin Vahdat. 2013. Integrating microsecond circuit switching into the data center. *ACM SIGCOMM Computer Communication Review* 43, 4 (2013), 447–458.

[90] Leon Poutievski, Omid Mashayekhi, Joon Ong, Arjun Singh, Mukarram Tariq, Rui Wang, Jianan Zhang, Virginia Beauregard, Patrick Conner, Steve Gribble, Rishi Kapoor, Stephen Kratzer, Nanfang Li, Hong Liu, Karthik Nagaraj, Jason Ornstein, Samir Sawhney, Ryohei Urata, Lorenzo Vicisano, Kevin Yasumura, Shidong Zhang, Junlan Zhou, and Amin Vahdat. 2022. Jupiter evolving: transforming google's datacenter network via optical circuit switches and software-defined networking. In *Proceedings of the ACM SIGCOMM 2022 Conference (SIGCOMM '22)*. https://doi.org/10.1145/3544216.3544265

[91] Kun Qian, Yongqing Xi, Jiamin Cao, Jiaqi Gao, Yichi Xu, Yu Guan, Binzhang Fu, Xuemei Shi, Fangbo Zhu, Rui Miao, Chao Wang, Peng Wang, Pengcheng Zhang, Xianlong Zeng, Eddie Ruan, Zhiping Yao, Ennan Zhai, and Dennis Cai. 2024. Alibaba HPN: A Data Center Network for Large Language Model Training. In *Proceedings of the ACM SIGCOMM 2024 Conference (ACM SIGCOMM '24)*.

[92] Arslan Sajid Raja, Sophie Lange, Maxim Karpov, Kai Shi, Xin Fu, Raphael Behrendt, Daniel Cletheroe, Anton Lukashchuk, Istvan Haller, Fotini Karinou, Benn Thomsen, Krzysztof Jozwik, Junqiu Liu, Paolo Costa, Tobias Jan Kippenberg, and Hitesh Ballani. 2021. Ultrafast optical circuit switching for data centers using integrated soliton microcombs. *Nature Communications* 12, 1 (Oct. 2021). https://doi.org/10.1038/s41467-021-25841-8

[93] Zhenghang Ren, Yuxuan Li, Zilong Wang, Xinyang Huang, Wenxue Li, Kaiqiang Xu, Xudong Liao, Yijun Sun, Bowen Liu, Han Tian, Junxue Zhang, Mingfei Wang, Zhizhen Zhong, Guyue Liu, Ying Zhang, and Kai Chen. 2025. Enabling Efficient GPU Communication over Multiple NICs with FuseLink. In *19th USENIX Symposium on Operating Systems Design and Implementation (OSDI 25)*. 91–108.

[94] R Ryf, J Kim, JP Hickey, A Gnauck, D Carr, F Pardo, C Bolle, R Frahm, N Basavanhally, C Yoh, et al. 2001. 1296-port MEMS transparent optical crossconnect with 2.07 petabit/s switch capacity. In *OFC 2001. Optical Fiber Communication Conference and Exhibit. Technical Digest Postconference Edition (IEEE Cat. 01CH37171)*, Vol. 4. IEEE, PD28–PD28.

[95] Nitika Saran, Daniel Amir, Tegan Wilson, Robert Kleinberg, Vishal Shrivastav, and Hakim Weatherspoon. 2024. Semi-Oblivious Reconfigurable Datacenter Networks. In *Proceedings of the 23rd ACM Workshop on Hot Topics in Networks*. 150–158.

[96] Tae Joon Seok, Niels Quack, Sangyoon Han, Richard S Muller, and Ming C Wu. 2016. Large-scale broadband digital silicon photonic switches with vertical adiabatic couplers. *Optica* 3, 1 (2016), 64–70.

[97] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538* (2017).

[98] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2020. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism. (2020). arXiv:cs.CL/1909.08053

https://arxiv.org/abs/1909.08053

[99] Chenchen Shou, Guyue Liu, Hao Nie, Huaiyu Meng, Yu Zhou, Yimin Jiang, Wenqing Lv, Yelong Xu, Yuanwei Lu, Zhang Chen, Yanbo Yu, Yichen Shen, Yibo Zhu, and Daxin Jiang. 2025. InfiniteHBD: Building Datacenter-Scale High-Bandwidth Domain for LLM with Optical Circuit Switching Transceivers. (2025). arXiv:cs.NI/2502.03885 https://arxiv.org/abs/2502.03885

[100] Vishal Shrivastav, Asaf Valadarsky, Hitesh Ballani, Paolo Costa, Ki Suh Lee, Han Wang, Rachit Agarwal, and Hakim Weatherspoon. 2019. Shoal: A network architecture for disaggregated racks. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. 255–270.

[101] Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, Anand Kanagala, Jeff Provost, Jason Simmons, Eiichi Tanda, Jim Wanderer, Urs Hölzle, Stephen Stuart, and Amin Vahdat. 2015. Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (SIGCOMM '15)*. https://doi.org/10.1145/2785956.2787508

[102] Sainbayar Sukhbaatar, Olga Golovneva, Vasu Sharma, Hu Xu, Xi Victoria Lin, Baptiste Rozière, Jacob Kahn, Daniel Li, Wen-tau Yih, Jason Weston, et al. 2024. Branch-Train-MiX: Mixing Expert LLMs into a Mixture-of-Experts LLM. *arXiv preprint arXiv:2403.07816* (2024).

[103] Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115* (2024).

[104] Xinchen Wan, Hong Zhang, Hao Wang, Shuihai Hu, Junxue Zhang, and Kai Chen. 2020. Rat-resilient allreduce tree for distributed machine learning. In *Proceedings of the 4th Asia-Pacific Workshop on Networking*. 52–57.

[105] Guohui Wang, David G Andersen, Michael Kaminsky, Konstantina Papagiannaki, TS Eugene Ng, Michael Kozuch, and Michael Ryan. 2010. c-Through: Part-time optics in data centers. In *Proceedings of the ACM SIGCOMM 2010 Conference*. 327–338.

[106] Weiyang Wang, Manya Ghobadi, Kayvon Shakeri, Ying Zhang, and Naader Hasani. 2024. Rail-only: A Low-Cost High-Performance Network for Training LLMs with Trillion Parameters. (2024). arXiv:cs.NI/2307.12169 https://arxiv.org/abs/2307.12169

[107] Weiyang Wang, Moein Khazraee, Zhizhen Zhong, Manya Ghobadi, Zhihao Jia, Dheevatsa Mudigere, Ying Zhang, and Anthony Kewitsch. 2023. TopoOpt: Co-optimizing Network Topology and Parallelization Strategy for Distributed Training Jobs. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. USENIX Association, Boston, MA, 739–767. https://www.usenix.org/conference/nsdi23/presentation/wang-weiyang

[108] Tegan Wilson, Daniel Amir, Nitika Saran, Robert Kleinberg, Vishal Shrivastav, and Hakim Weatherspoon. 2024. Breaking the VLB Barrier for Oblivious Reconfigurable Networks. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. 1865–1876.

[109] Tegan Wilson, Daniel Amir, Vishal Shrivastav, Hakim Weatherspoon, and Robert Kleinberg. 2023. Extending optimal oblivious reconfigurable networks to all n. In *2023 Symposium on Algorithmic Principles of Computer Systems (APOCS)*. SIAM, 1–16.

[110] Yiting Xia, Mike Schlansker, TS Eugene Ng, and Jean Tourrilhes. 2015. Enabling Topological Flexibility for Data Centers Using {OmniSwitch}. In *7th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 15)*.

[111] Kaiqiang Xu, Decang Sun, Hao Wang, Zhenghang Ren, Xinchen Wan, Xudong Liao, Zilong Wang, Junxue Zhang, and Kai Chen. 2025. Design and Operation of Shared Machine Learning Clusters on Campus. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1 (ASPLOS '25)*. Association for Computing Machinery, New York, NY, USA, 295–310. https://doi.org/10.1145/3669940.3707266

[112] Xiaohui Ye, SJ Ben Yoo, and Venkatesh Akella. 2012. AWGR-based optical topologies for scalable and efficient global communications in large-scale multiprocessor systems. *Journal of Optical Communications and Networking* 4, 9 (2012), 651–662.

[113] Yufang Yu, Nan Hua, Zhizhen Zhong, Jialong Li, Ruijie Luo, Zelin Zheng, and Xiaoping Zheng. 2017. Fast-Reconfigurable Optical Interconnect Architecture Based on Time-Synchronized Node Coordination for High Performance Computing, In Asia Communications and Photonics Conference. *Asia Communications and Photonics Conference*, S4C.6. https://doi.org/10.1364/ACPC.2017.S4C.6

[114] Mingyang Zhang, Jianan Zhang, Rui Wang, Ramesh Govindan, Jeffrey C Mogul, and Amin Vahdat. 2021. Gemini: Practical reconfigurable datacenter networks with topology and traffic engineering. *arXiv preprint arXiv:2110.08374* (2021).

[115] Shicheng Zhang, Xuwei Xue, Bingli Guo, Yixuan Li, Wenzhe Li, Shikui Shen, Haoze Qian, Xiaojie Yin, Buzheng Wei, Guojun Yuan, et al. 2024. Fast-tunable Graphene-based AWGR for Deep Learning Training Networks. In *Proceedings of the 1st SIGCOMM Workshop on Hot Topics in Optical Technologies and Applications in Networking*. 14–20.

[116] Xiaosheng Zhang, Ming Chiang A Wu, Andrew S Michaels, and Johannes Henriksson. 2022. Beam-steering system based on a MEMS-actuated vertical-coupler array. (Sept. 13 2022). US Patent 11,441,353.

[117] Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. 2024. LLaMA-MoE: Building Mixture-of-Experts from LLaMA with Continual Pre-training. *arXiv preprint arXiv:2406.16554* (2024). https://arxiv.org/abs/2406.16554

[118] Yazhou Zu, Alireza Ghaffarkhah, Hoang-Vu Dang, Brian Towles, Steven Hand, Safeen Huda, Adekunle Bello, Alexander Kolbasov, Arash Rezaei, Dayou Du, et al. 2024. Resiliency at Scale: Managing {Google's}{TPUv4} Machine Learning Supercomputer. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. 761–774.

Appendix

Appendices are supporting material that has not been peer-reviewed.

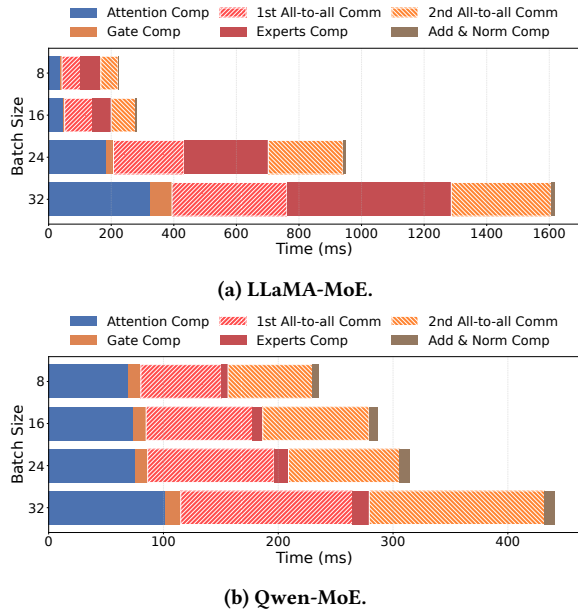## A   Production Measurement Details

A.1   Profiling of MoE Models

(a) LLaMA-MoE.

(b) Qwen-MoE.

Figure 17: Timeline of MoE models.

Figure 17 presents the profiling results of an MoE layer for LLaMA-MoE and Qwen-MoE. EP communication constitutes a more significant portion in these models compared to Mixtral models. For instance, in LLaMA-MoE, the two all-to-all communication phases account for 42%–58% of the iteration time. In comparison, EP communication dominates even more in Qwen-MoE, reaching up to 68%.

A.2   Non-uniform token distribution in trained MoE model

We measured the all-to-all token distribution in the forward pass of pre-trained Mixtral 8×7B [22], as shown in Figure 18. We observe that the number of tokens dispatched to each expert is non-uniform and varies across different MoE blocks, which argues for a necessary mechanism to adapt to the dynamic traffic in EP even when the model has largely converged.
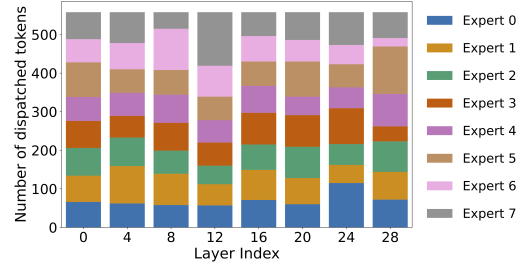

Figure 18: [Mixtral 8×7B in production] Non-uniform token distribution across MoE blocks.
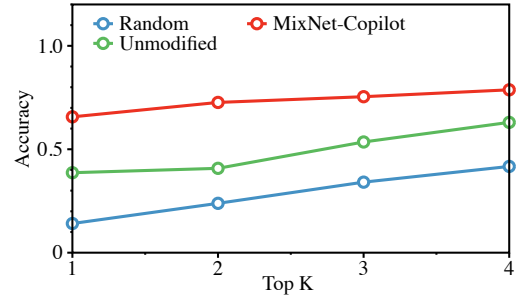

Figure 19: Average prediction accuracy of MixNet-Copilot.

B   Implementation Details

B.1   Traffic Demand Prediction

MixNet aims to handle the first all-to-all communication in the forward pass with a predictive approach. By default, the OCS topology for this initial communication is either randomly generated (e.g., for the first all-to-all in the first layer) or remains unchanged from the previously used topology (e.g., the first all-to-all in the second layer). The traffic demand prediction algorithm predicts the *conditional probability* of the traffic matrix, denoting the conditional probability of a token gated to expert j given that it is gated to the expert i in the last layer. With the conditional probability matrix and the empirical token distribution in the previous layer, we can predict the traffic distribution in the current layer.

Matrix Estimation: For each layer, MixNet estimates the conditional probability matrix with the traffic demand records in recent iterations. Focusing on the recent expert load distributions, we employ a weighted average within a fixed window of traffic records in time series. For each layer, the optimization objective is to minimize the square error between the predicted load distribution and the ground truth (for simplicity, we omit the layer index here):

$$\min_{P} \quad \sum_{i=1}^{k} w_i \cdot \Sigma_i \left((Y_i - PX_i)^2 \right), \qquad (1)$$

where k is the window size. The transition matrix P is of size $N \times N$, representing the conditional probability of the current layer's expert load distribution, given the expert load distribution of the previous layer. X_i, Y_i are normalized expert load distribution vectors of two neighboring layers. Each element in P is constrained to be in the range $[0, 1]$, and the sum of each column is constrained to be 1 to ensure that P is a valid probability matrix.
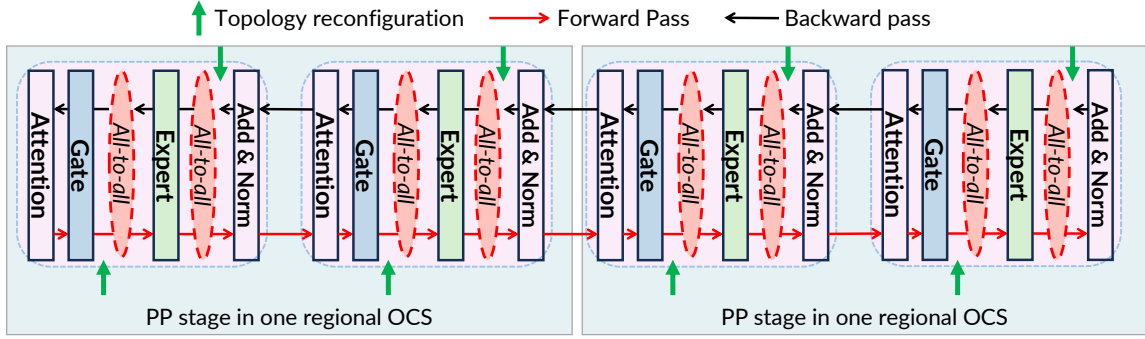
Figure 20: Reconfiguration timeline during runtime.

We employ the Sequential Least Squares Programming (SLAP) method for optimization, as it is suitable for nonlinear problems with linear constraints. The algorithm is implemented via the `scipy.optimize` library in Python. During the inference, with the load distribution in layer i given, we can predict the expert load distribution for the next layer in advance for its first all-to-all communication.

We name this method MixNet-Copilot. Figure 19 compares the prediction accuracy of MixNet-Copilot against the aforementioned methods, i.e., the randomly assigned token distribution (uniform bandwidth allocation), and the unmodified token distribution from the previous layers (unchanged topology) on collected traces from measurements. Top K accuracy measures whether MixNet is able to find the top-k activation-intensive experts. We find that MixNet-Copilot exhibits significantly higher accuracy than other counterparts, which implies that MixNet-Copilot can find the most intensive pairs in all-to-all communication with high probability. Therefore, MixNet-Copilot offers an opportunity to proactively reconfigure the topology for the FP's first all-to-all in advance.

B.2 Details of Topology Reconfigurations

Figure 20 details the runtime reconfiguration methodology of MixNet. Each MoE layer involves four all-to-all communication phases. As discussed in §5.1, MixNet can deterministically characterize the traffic patterns for the second all-to-all communication in the forward pass (FP) and both all-to-all communications in the backward pass (BP).

Consequently, MixNet conceptually reconfigures the topology twice per MoE layer—once during FP and once during BP. However, for the first all-to-all communication in FP, MixNet cannot fully characterize the traffic matrix in advance due to the absence of runtime information at that stage of the iteration. To tackle this issue, MixNet performs an inaccurate reconfiguration based on partial estimates or reuses the topology from the previous layer. It then recalibrates the topology for the second all-to-all in FP with minor OCS adjustments, ensuring a more accurate configuration for subsequent communication phases.

C ProtoType Details

Prototype profiling. We profiled the overall reconfiguration turnaround time of our OCS, and the results are shown in Figure 21.
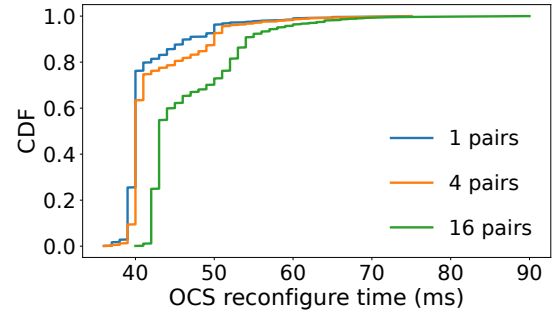

Figure 21: [Testbed] Reconfiguration delay across different number of pairs.
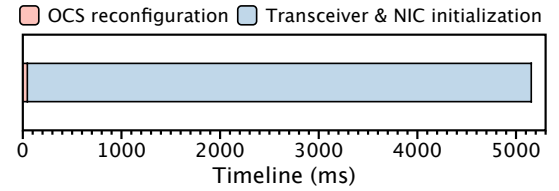

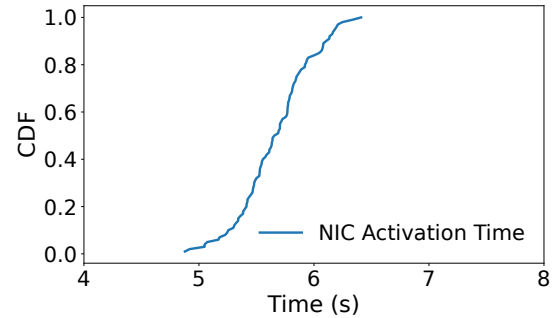**Figure 22: [Testbed] Overall timeline of one OCS control.**

Figure 23: [Testbed] CDF of time elapsed from OCS reconfiguration completion to NIC becoming active.

The OCS is controlled by issuing TL1 commands over Ethernet. We observed that as the number of pairs increases, the reconfiguration time slightly rises. The average reconfiguration time is approximately 41.44 ms for 1 pair, 42.44 ms for 4 pairs, and 46.75 ms for 16

pairs. The 99th percentile reconfiguration times are around 60 ms for 1 pair, 62 ms for 4 pairs, and 68 ms for 16 pairs. Notably, 99% of the reconfiguration times are under 70 ms, which is acceptable for MoE training, given the relatively long expert computation times typically used in practice (e.g., 122 ms for a batch size of 16).

Figure 22 illustrates the overall timeline from issuing an OCS reconfiguration control command to the successful completion of an RDMA send, providing a detailed view of the control process. The process consists of two main stages: (1) the control server sends a reconfiguration command to the OCS; (2) the transceiver and NIC initialize the physical link and set up the network device. Our observations indicate that the overall turnaround time of one reconfiguration is predominantly influenced by the physical link initialization and NIC device initialization.

We further plot the CDF of the time elapsed from the OCS re-configuration completion to the NIC becoming active. The results are shown in Figure 23. The average NIC activation time is approx-imately 5.67 s and the 99 percentile is around 6.33 s. These findings align with previous observations from [85], highlighting that some commodity transceivers and NICs are not well optimized for fast reconfiguration. We have discussed this issue with a transceiver vendor, who confirmed that the observed multi-second NIC re-activation latency is not a fundamental limitation, but rather a consequence of current commercial transceiver modules not being optimized for fast reconfigurable optical switching in datacenter en-vironments. Note that the burst-mode transceiver (e.g., [7, 39]) has already been deployed with passive optical networks (PONs) in ac-cess networks, which are designed to handle intermittent upstream transmissions with fast CDR locking and signal recovery. There-fore, extending datacenter transceivers with burst-mode features supported in PON transceivers is an engineering problem rather than an architectural barrier. In particular, these engineering efforts include: 1) Classical receiver-side CDR circuits require continuous data streams to maintain CDR state. To prevent loss-of-signal (LOS) during OCS reconfiguration, the optical transceiver can be config-ured into a local Tx/Rx loopback mode [99] prior to the switching event, ensuring that the receiver continues to observe a valid signal throughout the transition. 2) After the OCS reconfiguration, the CDR logic can be optimized using fast-locking CDR designs (e.g., [53]), which enable rapid recovery and re-synchronization upon detection of the new signal.

As a result, we currently exclude this NIC activation time to calculate the actual training time in MixNet testbed experiments. Due to the limited number of GPUs, we cannot train the full MoE models as shown in Table 1. We only run 7 layers of Mixtral 8×7B, 16 layers of LLaMA-MoE, and 12 layers of Qwen-MoE.

D Large-Scale Simulation Details

D.1 Simulated MoE Models and Parallelization Strategies

We simulate the training process of four MoE models: Mixtral 8×22B [21], Mixtral 8×7B [23], Qwen-MoE [40] and DeepSeek-R1 [54]. For Mixtral 8×22B, we use a hybrid parallelism that com-bines an EP degree of 8, TP degree of 8, PP degree of 8 at a sequence length of 4096, and micro-batch size of 8. For DeepSeek-R1, we follow the default training parallelisms in [54] with 64-way EP and

Link Band-width	Trans-ceiver ($)	NIC ($)	Elec. switch port ($)	OCS port ($)	Patch panel port ($)
100 Gbps	99 [1]	659 [29]	187 [107]	520 [37]	100 [42]
200 Gbps	239 [2]	1079 [30]	374 [107]	520 [37]	100 [42]
400 Gbps	659 [3]	1499 [31]	1090 [24]	520 [37]	100 [42]
800 Gbps	1399 [13]	2248[1][31]	1400 [32]	520 [37]	100 [42]

[1] Conservatively estimated as 1.5 times the price of 400G NIC, as 800G products are not yet commercially available.

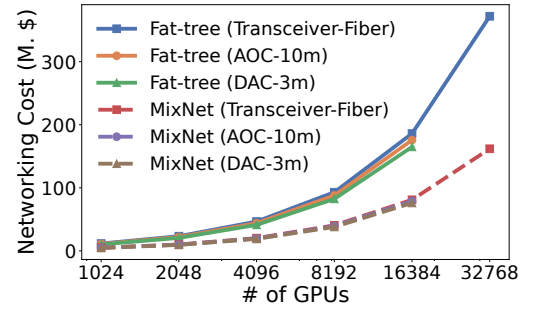**Table 4: Cost of network components.**

Figure 24: [Simulation] Cost comparison of different EPS links at 400 Gbps bandwidth.

16-way PP. For other models, we reuse the same configurations in Table 1.

D.2 Cost Analysis Details

Table 4 lists the costs of network components used in §7.2. We reuse the prices for electronic switches at 100G, 200G as well as for NICs, OCS ports, and patch panel ports from TopoOpt [107], and we add the prices of transceivers, NICs, and electronic switch ports for 400 Gbps and 800 Gbps link accordingly. We also follow the same methodology as in TopoOpt when calculating the fiber costs.

D.3 Different EPS Link Options

The OCS portion of MixNet requires optical transceivers with pluggable fibers to allow optical switching. For the EPS part of MixNet, especially short-reach rack-scale links between the servers and ToR switches, Direct Attach Copper (DAC) cables or Active Optical Cables (AOC) are more cost-effective alternatives to optical transceivers plus fibers (typically used for long-reach links). We analyze the cost implications of these link options in Figure 24. The results show that replacing the EPS links with DAC or AOC slightly reduce the costs for both fat-tree interconnect and MixNet. Most importantly, the cost effectiveness of MixNet is orthogonal to the choices of EPS links, and maintains significant cost advantages over fat-tree topology. For example, with 400 Gbps DAC cables option in a 4096-GPU cluster, MixNet achieves 2.2× lower total cost compared to fat-tree topology.
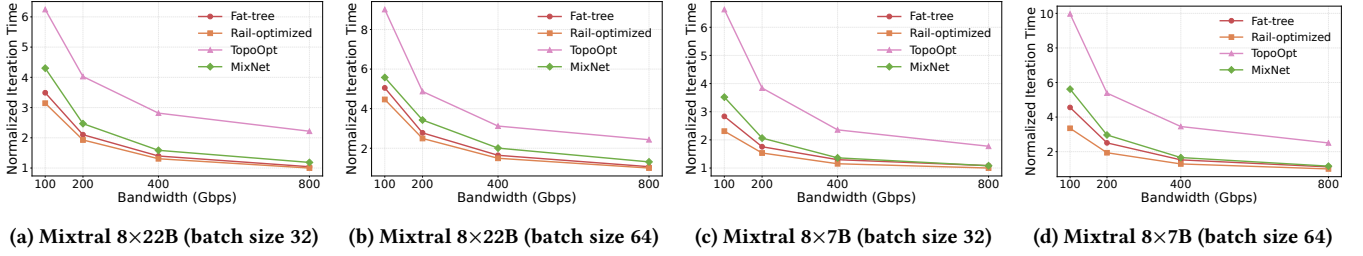
(a) Mixtral 8×22B (batch size 32)   (b) Mixtral 8×22B (batch size 64)   (c) Mixtral 8×7B (batch size 32)   (d) Mixtral 8×7B (batch size 64)

Figure 25: [Simulation] Training speed ups of Mixtral models with large batch sizes.

(a) Performance comparison.
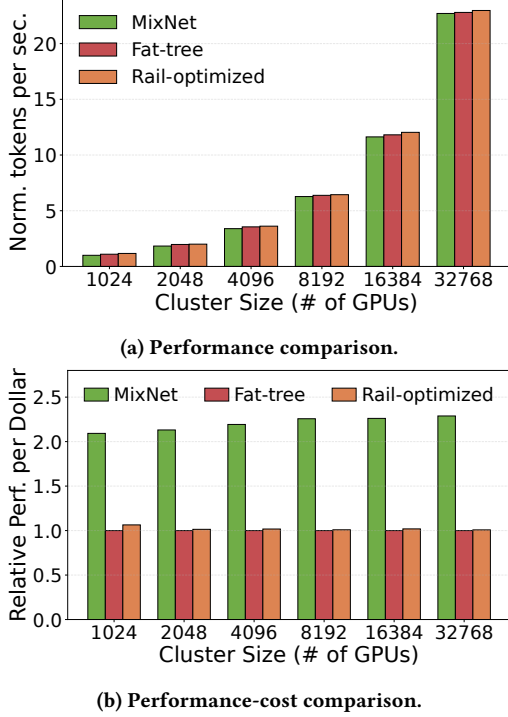

(b) Performance-cost comparison.

Figure 26: [Simulation] Scalability analysis of MixNet with different cluster sizes.

D.4 Training Speed Ups of Mixtral Models with Larger Batch Sizes

We further evaluate MixNet's performance with larger batch sizes using two Mixtral MoE models (Mixtral 8×7B and Mixtral 8×22B). For each model, we test batch sizes of 32 and 64 across varying network bandwidths (100-800 Gbps). As shown in Figure 25, MixNet consistently outperforms TopoOpt under all configurations. Specifically, MixNet achieves an average speedup of 1.8× for Mixtral-8x7B with a batch size of 32 and 2.0× with a batch size of 64, as training becomes more communication-intensive compared to the settings in Figure 12. Furthermore, we observe that as link bandwidth increases, MixNet's performance gradually approaches that of Fat-tree and Rail-optimized architectures.

D.5 Scalability

We demonstrate the scalability of MixNet in Figure 26a. The Mixtral 8×7B model is evaluated at 400 Gbps bandwidth, with the
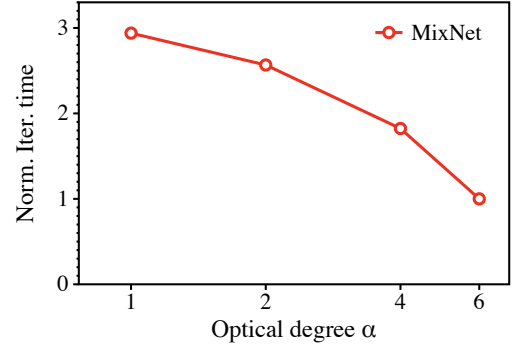
**Figure 27: [Simulation] Impact of optical degree α in MixNet.**

cluster size varying from 128 servers to 4,096 servers, covering up to 32768 GPUs. MixNet demonstrates scalability by fundamentally relaxing the port limits of OCS through the design of several decentralized regionally reconfigurable domains, allowing it to scale similarly to a fat-tree topology. Our results show that MixNet scales effectively with increasing number of GPUs, achieving training throughput comparable to both non-blocking Fat-tree and Rail-optimized topologies in terms of tokens processed per second. We further present the performance-cost comparison in Figure 26b, which shows that MixNet consistently achieves a superior performance-cost trade-off—approximately 2× higher performance-per-dollar—compared to Fat-tree and Rail-optimized topologies as the number of GPUs increases. This suggests that MixNet maintains the training cost-effectiveness even as the cluster size grows.

D.6 Impact of Optical Degree

We show the impact of the optical degree on MixNet's performance in Figure 27. We evaluate the Mixtral 8×22B model on a cluster of 128 servers with 100 Gbps link bandwidth. The optical degree α in MixNet is varied to adjust its connectivity in the OCS. We reduce the bandwidth of each electronic port when increasing their number, to ensure a cost-equivalent comparison. Our findings show that, as the optical degree increases, MixNet further reduces iteration time, as more communication-intensive GPU pairs can be provisioned with dedicated high-bandwidth optical circuits.

D.7 Impact of Reconfiguration Latency

To investigate MixNet's sensitivity to OCS reconfiguration latency, we evaluate the Mixtral 8×22B model on a cluster of 128 servers
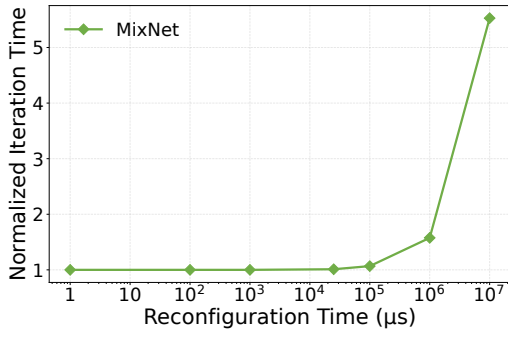
Figure 28: [Simulation] Impact of reconfiguration latency.

with 400 Gbps link bandwidth, varying the reconfiguration latency from 1 μs to 10 s. Figure 28 shows the normalized iteration time. MɪxNᴇᴛ assumes the use of a millisecond-scale reconfigurable OCS (25 ms) in its current implementation. We observe that further reducing reconfiguration latency does not yield significant performance gains, as the OCS reconfiguration process can already be fully hidden during the computation phase. However, provisioned with microsecond-scale reconfigurable OCS, MɪxNᴇᴛ can enable fully accurate topology reconfigurations for the first all-to-all communication in the forward pass (FP), resulting in marginal performance improvements in this specific phase. On the other hand, when the reconfiguration latency exceeds 1000 ms, performance degrades obviously, as the OCS reconfiguration may not be hidden and starts to block the training process. As a result, MɪxNᴇᴛ does not perform well with second-scale reconfigurable OCS systems.