DYNAMIC ROUTING WITH MIXTURE OF EXPERTS FOR DISTRIBUTED SKETCH MEASUREMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Sketch enables line rate network measurement on the programmable data plane but suffers from resource constraints, making distributed deployment necessary. In distributed sketch measurement, dynamic routing planning (DRP) directly determines traffic distribution across sketches. Unbalanced loads cause sketch overflow and accuracy degradation. However, existing DRP methods face limitations in multi-objective planning, model interpretability, and timely model response. This paper proposes a MoE-based DRP framework with three core components. First, a MoE model comprising four expert networks and one gating network enables multi-objective planning (e.g., bandwidth, path length and switch overhead). Second, a teacher-student distillation framework injects domain knowledge into the MoE model, making each decision traceable to specific experts for interpretability. Third, a decision cache stores model decisions for recurring traffic states, reducing model response latency. Experiments on BMv2 show that our framework improves distributed sketch measurement accuracy by 6.7% and reduces maximum data plane overhead by 6.6%.

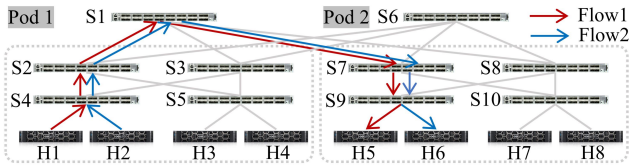
1 INTRODUCTION

As a probabilistic data structure, sketches offer trade-offs between resources and accuracy (Zheng et al., 2022). Through efficient deployment on the programmable data plane, sketches enable line rate network measurement, and support a variety of measurement functions (e.g., Top-K detection, and heavy hitter detection) (Gu et al., 2024).

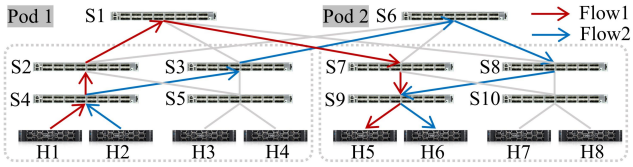
However, programmable data planes face severe resource constraints (i.e., typically tens of MB of memory and 12 stages for computation) (Zhou et al., 2022). As traffic volumes continue to grow, the storage and computation bottlenecks prevent a single sketch from accurately measuring all network traffic. Therefore, distributed sketch deployment becomes necessary, where multiple data plane switches cooperate to collectively obtain complete network measurement results, while maintaining balanced traffic distribution across switches (Gu et al., 2024). This trend naturally leads to the need for dynamic routing planning (DRP), which efficiently coordinates routing decisions among switches to enhance overall measurement performance (Wang et al., 2023).

Through broad research, existing DRP methods can be mainly categorized into two classes (i.e., local static planning (LSP) methods, and constraint-based global planning (CGP) methods). (1) LSP methods plan routing through local device information instead of global planning objectives. For example, ECMP utilizes hash calculation to evenly distribute traffic among available paths without considering the real-time network status (Jamil et al., 2025). Dijkstra only focus on path length and tends to cause core link congestion (Adel et al., 2023). Thus, LSP methods cannot achieve multi-objective planning, resulting in limited DRP results. (2) CGP methods formulate DRP as integer programming problems and can conduct multi-objective planning. For instance, joint routing and sketch configuration (JRS) takes into account both the path length and the sketch deployment (Chang et al., 2023). Rounding-based routing and sketch configuration (R-RSC) simultaneously plans bandwidth and sketch measurement (Zhai et al., 2020). However, CGP methods must repeatedly execute their optimization procedures as network states change, resulting in nonnegligible online planning latency (Pavlitska et al., 2023). Directly replacing these procedures with a black-box learning model can reduce latency but makes individual routing decisions difficult to trace. Consequently, there is an

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Routing strategy w/o DRP



(b) Routing strategy w/ DRP

Figure 1: Motivation for DRP in distributed sketch measurement

urgent need for a DRP approach that supports multi-objective planning, traceable learned decisions, and timely model response.

Luckily, we observe that the emerging mixture of experts (MoE) architecture presents new opportunities to address the above limitations. By employing multiple expert networks for parallel decision-making integrated through a gating mechanism, MoE offers inherent advantages for multi-objective planning in DRP. This MoE architecture has already demonstrated strong performance across diverse domains (e.g., unmanned aerial vehicles and computer vision) (Kong et al., 2025b).

Motivated by our observations, this paper proposes a MoE-based DRP framework. The key idea is to achieve a MoE model that generates multi-objective planning with interpretable DRP strategies and timely model response. Specifically, (1) for multi-objective planning, we design a MoE model that comprises four expert networks and one gating network. Each expert network consists of three neural layers, with a total parameter count of 140,000. We coordinately consider the flow size and the planning sequence to ensure stable results. (2) For model interpretability, we adopt a teacher-student training framework to transfer domain expert knowledge into the neural network. And each expert in our MoE model corresponds to a specific network state. (3) For timely model response, we deploy a decision caching method to reduce the model response latency on large flows. Implemented on a fat-tree topology using BMv2, our proposed framework improves distributed sketch measurement accuracy by 6.7%, and reduces maximum data plane overhead by 6.6%.

The main contributions of this paper include:

- We propose a MoE-based framework for DRP in distributed sketch measurement.
- We achieve a MoE model with four experts for multi-objective planning.
- We utilize a teacher-student training framework to transfer domain expert knowledge into the neural network for model interpretability.
- We implement decision caching to ensure timely model response.
- Experimental results demonstrate that our method improves distributed sketch measurement performance and reduces data plane overhead.

2 Related Work

Existing DRP methods can be broadly categorized into LSP methods and CGP methods. As for LSP methods, ECMP hashes flow key information (e.g., packet five-tuples) and evenly distributes flows across available paths (Jamil et al., 2025). However, ECMP cannot incorporate real-time network status into its decisions. Dijkstra finds the shortest path by minimizing hop count (Adel et al., 2023). While simple, it tends to aggregate traffic onto core links, creating congestion bottlenecks. If large flows flood into a single sketch, the data plane resources can be rapidly exhausted, leading to degraded measurement results (Zhou et al., 2022). Therefore, LSP methods only considers limited constraints and cannot conduct multi-objective planning in DRP.

To overcome the limitations of LSP methods, CGP methods have been proposed specifically for distributed sketch measurement. JRS adopts a decoupled greedy strategy that it first selects the K

shortest paths for a flow, then chooses the sketch with the least overhead for measurement (Chang et al., 2023). Nevertheless, JRS handles candidate-path generation and sketch selection in two separate steps, and therefore cannot jointly optimize path length and sketch-resource balance. R-RSC deploys multiple sketches on a single data plane switch, and uses different sketches for measuring specific flows. It plans sketch measurement and link bandwidth by formulating the problem as integer programming with stochastic rounding (Zhai et al., 2020). More critically, existing CGP methods require repeated online planning after network-state changes, resulting in nonnegligible response latency (Pavlitska et al., 2023). Although a learned model can amortize this planning process, it must preserve the relationship between network conditions and routing decisions. Consequently, there is a pressing need for a DRP method capable of multi-objective planning with model interpretability and timely model response.

3 Motivation, Goals, and Challenges

Motivation. We provide an example to illustrate the motivation of our work. As shown in Fig. 1, a topology with ten switches and eight servers is divided into two pods (Chang et al., 2023). Two flows (i.e., Flow1 and Flow2) travel from server H1/H2 to server H5/H6, respectively. In Fig. 1(a), both flows follow the same path, competing for link bandwidth and sketch resources. Expanding to large volume of traffic may cause sketch overflow, reducing measurement performance. In contrast, Fig. 1(b) shows a better routing strategy that disperses the two flows, achieving load balancing in both bandwidth and sketch measurement. This example demonstrates that DRP can affect distributed sketch measurement performance.

Goals. This paper aims to design an MoE-based DRP framework for distributed sketch measurement with three core goals. (1) Multi-objective planning. We need to jointly consider link bandwidth, path length, and switch overhead for more effective planning (Zhai et al., 2020). (2) Interpretable DRP strategy. Each decision should be traced back to specific expert networks and the gating mechanism. (3) Timely model response. We need to generate routing strategies quickly to keep pace with the packet receiving speed (Kong et al., 2025a).

Challenges. Achieving the above goals faces three key challenges. (1) Complexity of multi-objective optimization. This complexity arises from two aspects. First, network features include both topology and network status, which have different scales and meanings. Directly using them as model inputs leads to high dimension and training difficulty. Second, some objectives inherently conflict (e.g., pursuing the shortest path harms link bandwidth). (2) Stability of routing strategies. Due to the Zipf distribution of network traffic, large flows are few but contain major information (Zhai et al., 2020). MoE model cannot reroute a large flow solely due to a slight load difference, because it may cause route oscillation (i.e., sharp traffic fluctuations on both old and new paths). (3) Real-time requirement. Standard MoE models are complex and have nonnegligible inference latency (Kong et al., 2025b). Simply reducing model complexity risks performance loss.

4 Problem Statement

Sketches are maintained by a list of counter arrays and hash functions. As for a sketch containing c counter arrays with b buckets to capture and store the information of d flows F, where $F = \{f_1, f_2, ..., f_d\}$. For each counter array c_i, the sketch uses an independent hash function h_i to calculate the five-tuples of packets, so as to find the index of the exact bucket in this counter array. However, due to the limited storage resources of switches, b is not infinite. When a large number of flows flood into a single sketch, some flows may be stored in the same bucket due to hash collisions (i.e., different packets having the same hash result), resulting in degraded accuracy (Zhou et al., 2022). Therefore, distributed sketch measurement is proposed to alleviate this situation. By deploying sketches on different switches in the network, the number of overall incoming packets drops, thereby reducing the hash collision rate. Under these circumstances, the measurement effect relies on DRP, which determines traffic distribution.

We consider a network topology $G = (S, L)$. Specifically, $S = \{sw_1, sw_2, ..., sw_m\}$ means m switches, and $L = \{l_1, l_2, ..., l_n\}$ reflects the n links between switches. The goal of DRP is to find the most suitable path p for each flow $f \in F$, where p contains a series of switches and links (e.g., $p_i = \{sw_1, l_1, sw_3, l_2, sw_2\}$). For each flow f, we construct a set of candidate paths $\mathcal{P}_f$ between

its source and destination. We use a binary variable $x_{f,p}$ to indicate whether flow f is assigned to path $p \in \mathcal{P}_f$. Accordingly, F_{sw} denotes the flows whose selected paths traverse switch sw. For optimal DRP, we simultaneously consider the flow size s_f, the bandwidth w_l of each link $l \in L$, the overhead o_{num} and o_{size} (i.e., the total number and size of flows) of each switch $sw \in S$, the path length len, the sketch size ss, and the hash collision rate r. All the key parameters are averaged and normalized to ensure fairness in the training process. We can additionally increase the weight coefficients to enable the model to support different planning goals. Therefore, we can define the DRP problem from (1) to (6).

$$w = \frac{1}{|L|} \sum_{l \in L} \sum_{f \in F} \sum_{\substack{p \in \mathcal{P}_f: \\ l \in p}} s_f x_{f,p} \tag{1}$$

$$o_{num} = \frac{1}{|S|} \sum_{sw \in S} |F_{sw}| \tag{2}$$

$$o_{size} = \frac{1}{|S|} \sum_{sw \in S} \left(\sum_{f \in F_{sw}} s_f \right) \tag{3}$$

$$len = \frac{1}{|F|} \sum_{f \in F} |p_f| \tag{4}$$

$$ss = c \cdot b \tag{5}$$

$$r = \frac{1}{|S|} \sum_{sw \in S} \left\{ 1 - \left(1 - \frac{1}{b} \right)^{c \cdot |F_{sw}|} \right\} \tag{6}$$

Based on the above metrics, we formulate DRP as a constrained multi-objective optimization problem. Let $\hat{w}$, $\hat{o}_{num}$, $\hat{o}_{size}$, $\widehat{len}$, and $\hat{r}$ denote the normalized forms of the corresponding metrics. The routing objective is defined as:

$$X^* = \arg\min_X \quad \lambda_w \hat{w}(X) + \lambda_n \hat{o}_{num}(X) + \lambda_s \hat{o}_{size}(X) + \lambda_l \widehat{len}(X) + \lambda_r \hat{r}(X) \tag{7}$$

$$\text{s.t.} \quad \sum_{p \in \mathcal{P}_f} x_{f,p} = 1, \quad \forall f \in F, \quad \sum_{f \in F} \sum_{\substack{p \in \mathcal{P}_f: \\ l \in p}} s_f x_{f,p} \leq w_l, \quad \forall l \in L, x_{f,p} \in \{0, 1\}$$

The coefficients λ_w, λ_n, λ_s, λ_l, and λ_r are nonnegative and sum to one. They allow operators to adjust the planning preference without changing the model architecture. Since the sketch size ss is fixed after deployment, it affects the objective through the collision rate rather than serving as an independent routing variable. This formulation provides the optimization target for the teacher model and the supervision signal for the student model.

5 MoE-based DRP

5.1 Architecture Overview

We employ a teacher-student framework for training the MoE model, as shown in Fig. 2. The teacher model is a knowledge-based model implemented following the MoE principles (Ghosh et al., 2024), and it generates optimal DRP based on domain expert knowledge. The student model is a lightweight MoE model, which learns from the outputs of the teacher model. This teacher-student framework accelerates convergence and enhances model interpretability.

5.2 Knowledge-based Teacher Model

As shown in Fig. 2(a), the teacher model treats each switch as an expert that perceives network status via in-band network telemetry (INT). For each candidate path, we normalize the metrics defined in §4 (i.e., link bandwidth, path length, and switch overhead) and combine them via a weighted sum. Weights are set by domain experts to ensure interpretability. The path with the highest score is selected, and corresponding switches update their forwarding tables. After routing each flow, we immediately refresh affected link bandwidth and switch overhead. A CM-sketch then evaluates measurement performance, providing feedback for student model training. Unlike prior CGP methods,

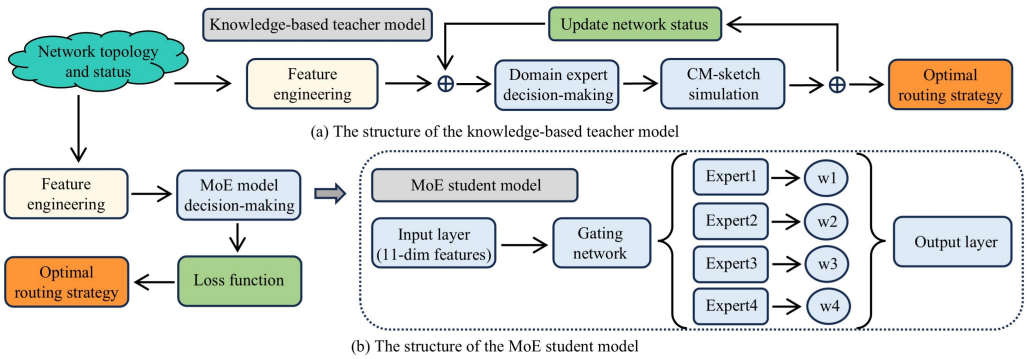


Figure 2: The teacher-student model training framework

our teacher model considers richer constraints and follows the MoE principle of assigning distinct tasks to specific experts.

The teacher and student define experts at different levels. A teacher expert corresponds to a physical switch and provides the local bandwidth and sketch-overhead contributions of a candidate path. These local contributions are aggregated with path length to generate the path-level teacher score $q_T(f, p)$. In contrast, a student expert does not correspond to a particular switch. Instead, it is designed to capture recurring routing-state patterns characterized by different combinations of bandwidth pressure, sketch overhead, and path cost. During distillation, the path-level teacher scores transfer the joint effects of multiple switch experts to the student, while the gating network determines which state-specific student experts should dominate the current decision. This cross-level design allows a fixed number of student experts to process paths containing different switches, without making the model size dependent on the topology scale.

One may issue that updating the remaining network resources after allocating each flow could lead to different DRP results depending on the order of flow input, potentially causing significant variations between consecutive allocations (i.e., route oscillation). To ensure stable DRP, we first sort the flows from largest to smallest before feeding the flow information into the model, and the teacher model prioritizes large flows. This is because network traffic follows a Zipf distribution, and large flows have a significant impact on the network status (Zhai et al., 2020).

Furthermore, some DRP methods suggest considering the departure of flows from the topology. However, we argue that in real-world networks, large flows that contain most network information are typically long-lived (Zhou et al., 2026). The brief presence and departure of small flows cause negligible influence on the network resources. Based on the above facts, we do not consider the flow departure situation during model training, thereby accelerating the model response speed.

5.3 MoE Student Model

In this section, we introduce our lightweight MoE model, as shown in Fig. 2(b). The overall architecture comprises the following core components. (1) Input feature layer. The MoE model takes an 11-dimensional feature vector as input. This vector includes 2-dimensional topological features (i.e., normalized information of the source and destination pods), 1-dimensional flow feature (i.e., the normalized value of flow size), and 8-dimensional network status features (i.e., the overhead of switches and the link bandwidth between switches). (2) Gating network. This network adopts a three-layer fully connected structure. The first layer consists of 64 neurons and employs ReLU activation. The second layer uses 32 neurons with ReLU activation. The output layer uses four neurons with Softmax activation. The gating network is responsible for dynamically computing the weight distribution for each expert network according to the input features. (3) Expert networks. The model contains four independent experts, each employing an identical three-layer architecture. The hidden layer has 128 neurons with ReLU activation, and the output layer has one neuron with Sigmoid activation. (4) Output fusion layer. It performs a weighted fusion of the outputs from expert networks, using the weights from the gating network, to generate the final DRP decision.

The student model evaluates candidate paths instead of directly encoding a path identifier as its output. For each flow f and candidate path $p \in \mathcal{P}_f$, we construct an 11-dimensional feature vector $\mathbf{z}_{f,p}$ using the source and destination information, flow size, and the current bandwidth and switch

overhead along p. Let $g_e(\mathbf{z}_{f,p})$ and $h_e(\mathbf{z}_{f,p})$ denote the gating weight and output of expert e, respectively. The student score and final routing decision are:

$$q_S(f,p) = \sum_{e=1}^{E} g_e(\mathbf{z}_{f,p}) h_e(\mathbf{z}_{f,p}) \tag{8}$$

$$p_f^* = \arg\max_{p \in \mathcal{P}_f} q_S(f,p), \quad E = 4 \tag{9}$$

Let $J(X)$ denote the weighted objective in (8), and let $X_{<f}$ denote the assignments of the flows processed before f. Since all objective components are normalized to $[0,1]$ and the weight coefficients sum to one, the teacher score is defined as:

$$q_T(f,p) = 1 - J(X_{<f} \cup \{f \to p\}) \tag{10}$$

Therefore, $q_T(f,p) \in [0,1]$, and a larger teacher score represents a lower routing cost. This definition is consistent with both the Sigmoid outputs of the student experts and the maximization used for final path selection.

To interpret a decision, the control plane records the individual contribution $g_e(\mathbf{z}_{f,p})h_e(\mathbf{z}_{f,p})$ of each expert. After training, each expert is further profiled using the average network state of the samples for which it receives the largest gating weight. The resulting profiles indicate whether an expert is mainly activated by bandwidth pressure, sketch overhead, path length, or a balanced network state. Therefore, a selected path can be traced to both the activated experts and the network conditions represented by these experts, rather than relying on the gating weights alone.

The training process is divided into two stages. (1) Knowledge distillation stage. For each flow, the teacher model generates DRP strategies and sketch measurement results based on the network status. We use the outputs as supervisory signals. (2) Imitation learning stage. The student model learns domain expert knowledge by minimizing the difference between its predicted output and the teacher's target value. We use the mean squared error (MSE) as the loss function. To prevent certain experts from being overused while others are neglected, we introduce a load balancing term into the loss function. For a mini-batch $\mathcal{B}$, we first calculate the average gating mass $\bar{g}_e$ assigned to experts. We then penalize its deviation from the uniform expert utilization $1/E$, where $E = 4$ is the number of experts. The total loss is defined in (13), where $\alpha = 0.01$ controls the load balancing term.

$$\bar{g}_e = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{z} \in \mathcal{B}} g_e(\mathbf{z}) \tag{11}$$

$$loss_{bal} = E \sum_{e=1}^{E} \left(\bar{g}_e - \frac{1}{E} \right)^2 \tag{12}$$

$$loss = loss_{mse} + \alpha loss_{bal} \tag{13}$$

Knowledge distillation does not require the student to exactly reproduce every teacher score. Let p_T^* be the path selected by the teacher for flow f, and define the teacher decision margin as:

$$\Delta_f = q_T(f, p_T^*) - \max_{p \in \mathcal{P}_f \setminus \{p_T^*\}} q_T(f,p) \tag{14}$$

Suppose that the student score satisfies $|q_S(f,p) - q_T(f,p)| \leq \epsilon$ for every candidate path. If $\Delta_f > 2\epsilon$, the student and teacher necessarily select the same path. This follows because the approximation error can reduce the score difference between the best and second-best paths by at most 2ϵ. Therefore, the student preserves the teacher decision whenever the teacher has a clear routing preference, while avoiding the online cost of repeatedly executing the knowledge-based optimization.

The teacher model is used only during offline training to generate supervisory path scores. Applying it directly at runtime would require evaluating all candidate paths and sequentially updating the affected network states for every new flow. The student model converts this repeated decision process into an amortized path-scoring function. Its gating network selects different expert combinations according to the current network state, allowing the same model to represent different trade-offs among bandwidth, path length, and sketch overhead. Therefore, the student is not designed to outperform the optimization objective of the teacher. Instead, it preserves clear teacher decisions under the condition described above, while providing a reusable and traceable model for online DRP.

5.4 Decision Caching

To provide timely model response, we maintain a decision cache for recurring flows. Each cache entry uses the flow five-tuple as its lookup key and stores the selected output port together with a

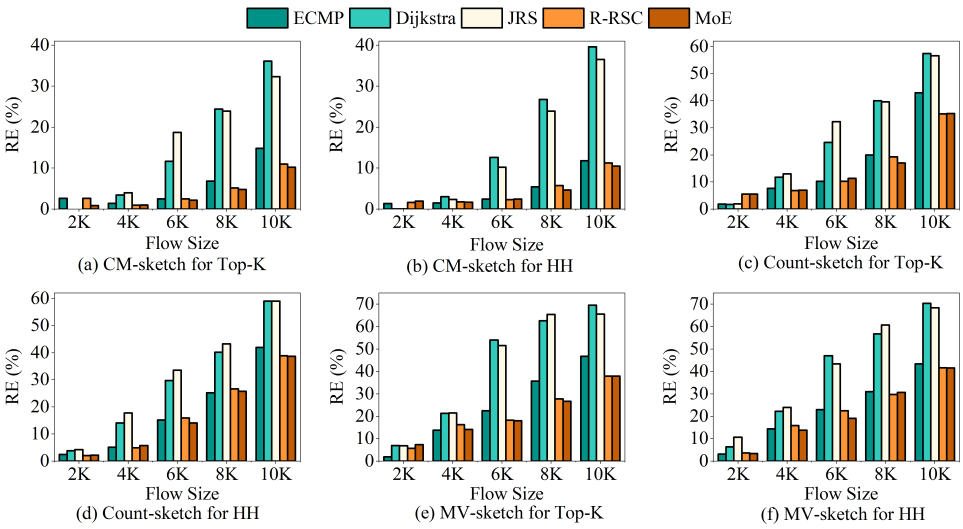


Figure 3: (Exp#1) Distributed sketch measurement results under DRP

compact network-state version and an expiration time. When a packet arrives, the data plane first searches the cache. A cached decision is reused only when the entry has not expired and its state version matches the current version. Otherwise, the request is treated as a cache miss and sent to the control plane for model inference.

The control plane generates the state version from the normalized bandwidth and switch-overhead features used by the MoE model. It increments the version when the maximum change of these features exceeds a threshold δ, or when a topology or forwarding policy change occurs. Corresponding stale entries are therefore invalidated without comparing complete feature vectors in the data plane. The time-based expiration mechanism further handles gradual network changes that do not individually exceed δ. After inference, the new routing decision and current state version are written into the cache. This design allows subsequent packets of a large flow to bypass model inference, while preventing decisions generated under outdated network states from being repeatedly reused.

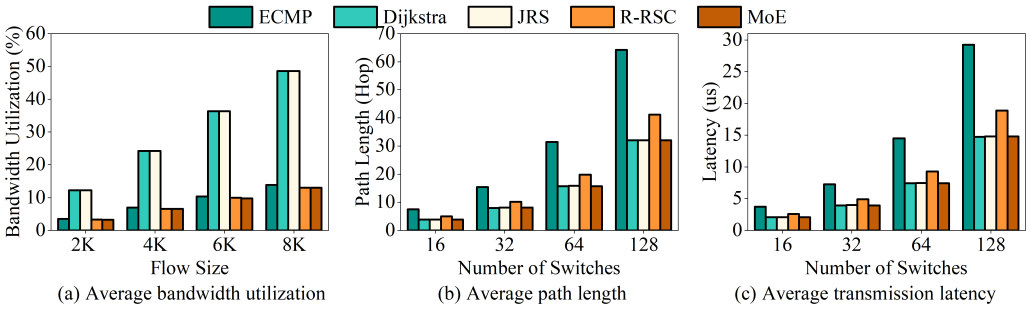
6 Implementation

We implement the proposed MoE-based DRP framework on programmable switches, which contain a control plane that enables the deployment of learning-based models, and a data plane that allows for line rate packet processing. As for control plane implementation, the MoE model is trained offline using Python 3.8. The teacher model generates supervisory signals based on domain knowledge, and the student learns to mimic its outputs. After training, the student MoE model is deployed in the control plane for online inference. As for data plane implementation, we use P4-16 (Zhou et al., 2022). The decision cache is achieved through a register that stores up to 64 recent routing decisions, keyed by the flow five-tuple, with the corresponding output port as the value.

When a packet enters the programmable switch, the data plane first checks the decision cache. If a hit occurs, the switch directly forwards subsequent packets of that flow according to the cached port, bypassing the model inference. A hit is accepted only when the cached state version matches the current version; otherwise, the packet follows the cache-miss processing path. This significantly reduces model response latency for large flows with multiple packets. If the cache misses, the switch uses the `digest` primitive to encapsulate the five-tuple and send it to the control plane, where the MoE model computes a routing decision and updates the cache with this new entry (i.e., evicting the least recently used entry if necessary).

7 Evaluation

Testbed Environment We conduct experiments on a K = 4 fattree topology with 20 switches and 16 servers, which is widely used in data center networks (Chen et al., 2023). The bandwidth between



(a) Average bandwidth utilization (b) Average path length (c) Average transmission latency

Figure 4: (Exp#2) Impact on link bandwidth utilization, path length and transmission latency

switches and the switch port capacity are both set to 10 Gbps. We utilize the ICS-NAD dataset (Zhou et al., 2026) as background traffic and process the source and destination addresses of each flow to fit our experimental topology. The traffic distribution is also processed to consist of 5% large flows (i.e., exceeding 70% of the maximum flow size (MFS)), 10% medium flows (i.e., 30% to 70% of the MFS), and 85% small flows (i.e., 0% to 30% of the MFS). In the following experiments, we utilize ECMP and Dijkstra as baselines and compare our MoE model with the state-of-the-art JRS and R-RSC methods.

(Exp#1) Distributed sketch measurement effect. We test distributed sketch measurement under five different DRP methods through Top-K detection and heavy hitter detection, which are two common sketch measurement applications (Chen et al., 2023). We implement them by deploying three typical sketches (i.e., CM-sketch, Count-sketch, and MV-sketch) following the setup in (Chen et al., 2023), and we send 2000, 4000, ..., 10000 flows in the network as background traffic. As the number of flows increases, the switch overhead rises, resulting in an increased hash collision rate and accuracy degradation. DRP can influence distributed sketch measurement by adjusting the load balance of links and switches. Sketch measurement results are represented by relative error (RE). Other parameters (e.g., F1 score) show similar trends but are omitted here due to space constraints.

Fig. 3(a) and (b) show the CM-sketch results of Top-K detection and heavy hitter detection. Correspondingly, Fig. 3(c) and (d) reflect Count-sketch, and Fig. 3(e) and (f) reflect MV-sketch. Among the three sketches, our method basically achieves the smallest RE in both applications, as we consider the switch overhead and load balancing, resulting in a more uniform traffic distribution. In contrast, the Dijkstra and JRS methods cause traffic aggregation on certain core paths. The R-RSC method, which considers sketch deployment, achieves a lower RE than the random DRP of ECMP. Experimental results demonstrate that our MoE-based method can effectively maintain high sketch measurement performance.

(Exp#2) Bandwidth utilization, path length and transmission latency. We measured the bandwidth usage of five DRP methods. A concentrated routing strategy causes certain paths to encounter high bandwidth utilization, while others remain largely unused. Following (Exp#1), we again send five flow volumes and compare the average bandwidth utilization.

As shown in Fig. 4, Dijkstra and JRS show significantly higher bandwidth usage. In contrast, our method and the R-RSC method consistently maintain lower bandwidth utilization rates. We save 6.2% average usage compared with ECMP, demonstrating a good load balancing effect. Although our data is slightly higher than the R-RSC when facing large traffic, R-RSC does not consider path length, allowing some flows to take longer paths for a more balanced traffic distribution, which will be discussed in (Exp#3). Experimental results indicate that our method achieves load balancing and lower bandwidth utilization.

We further measure the path length and end-to-end transmission latency of DRP strategies. Since the fattree topology employs a three-layer structure (i.e., edge, aggregation, and core layers), all flows have a fixed 6-hop path from source to destination. Therefore, we adopt a ring topology in this experiment, where each switch is connected only to its two neighbors, forming a circular structure (Chang et al., 2023). For a topology with x switches, the path length can vary from one hop to $x/2$ hops, sufficiently reflecting the difference in path length. We conduct four experiments with x = 16, 32, 64, and 128.

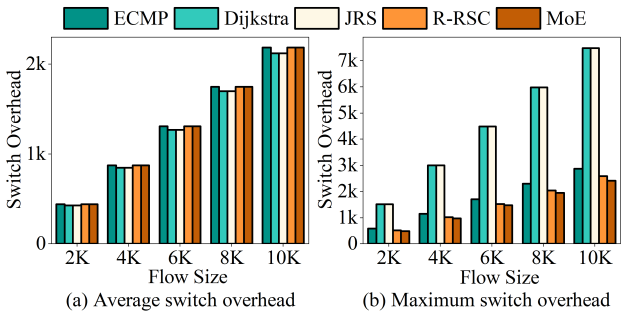


Figure 5: (Exp#3) Impact on the switch overhead

Method	Stage	SRAM(%)	ALU(%)	Latency(us)
MoE	4	0.73	6.25	436.74
R-RSC	12	6.56	64.58	243.55

Table 1: (Exp#4) Switch resource utilization and response latency

Fig. 4(b) shows the average hops under five DRP methods, and Fig. 4(c) shows the average end-to-end transmission latency. Since ECMP uses hash functions to randomly select paths, its result is close to the expectation value (i.e., x/2). Our MoE-based method, along with the Dijkstra and JRS methods, achieves short paths and low latency. Conversely, the R-RSC method, which does not consider path length, adopts 20.4% longer paths and 21.7% more latency than our method when x = 128, showing our multi-objective planning ability.

(Exp#3) Switch overhead. We measure the switch overhead (i.e., the number of flows that switches need to handle). Since sketches store different flows in different counters, this metric is crucial for sketch measurement, as it can impact the hash collision rate (Zhou et al., 2022). Following (Exp#1), we transmit five different flow volumes, and we compare the average and maximum switch overhead.

As shown in Fig. 5(a), the average switch overhead differs slightly, and the relatively lower results for Dijkstra and JRS come from their path length priority (i.e., some switches have zero load). However, Fig. 5(b) highlights the superiority of our MoE-based method, which saves 6.6% switch overhead compared with state-of-the-art methods, indicating the well consideration of switch overhead.

(Exp#4) Switch resource utilization and per-flow response latency. We compare these metrics with R-RSC, which improves measurement accuracy by deploying multiple sketches on one switch. Specifically, we deploy CM-sketch, and R-RSC deploys three mainstream sketches utilized in (Exp#1). We further test the model response latency for evaluating timeliness.

As shown in Table 1, our MoE-based method significantly reduces hardware resource overhead. R-RSC utilizes all the stage resources when it deploys three sketches, indicating that the assumption of enhancing the measurement effect by deploying multiple sketches is not valid. We also save 88.9% SRAM and 90.3% ALU, compared with R-RSC. Regarding per-flow response latency, although the MoE model undoubtedly requires more time to generate DRP strategies, by utilizing model optimization in (§5.4), our per-flow response latency is limited to only 436 µs. This latency achieves the same order of magnitude as the state-of-the-art DRP method and remains far below the millisecond-level requirement for timely network transmission (Deng et al., 2022). Experimental results demonstrate that our MoE model supports practical sketch deployment and enables timely model response.

8 Conclusion

This paper presents a MoE-based DRP framework that effectively achieves multi-objective planning, model interpretability, and timely model response for distributed sketch measurement. Future work will extend the proposed framework to diverse programmable platforms and emerging network architectures to enhance generalization ability.

9 AI USE STATEMENT

In this work, we used generative AI tools solely for language polishing and editing of the manuscript. We have not used generative AI tools for study design, experimental design, data collection, data analysis, code generation, or interpretation of results, and other tasks with required disclosure are not applicable to this work. Additionally, we used generative AI tools for improving grammar, clarity, and readability of the text. We have reviewed all AI-assisted work. For example, all AI-polished text was checked and edited by the authors for accuracy, originality, and preservation of the intended meaning. No AI tools were used to generate research ideas, design experiments, analyze data, or produce scientific conclusions. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Salsabil Adel, Raghda Samir, Irene Samy, and Tawfik Ismail. Integrated trust-clustering and dijkstra routing algorithms for energy-efficient wsns. In *2023 17th International Conference on Telecommunications (ConTEL)*, pp. 1–6, 2023.

You-Cheng Chang, Wei-Te Lu, and Kate Ching-Ju Lin. Joint routing and sketch configuration for multi-switch cooperative network measurement. In *2023 IEEE Global Communications Conference (GLOBECOM)*, pp. 4320–4325, 2023.

Xiang Chen et al. Eliminating control plane overload via measurement task placement. *IEEE/ACM Transactions on Networking*, 31(4):1717–1731, 2023.

Libing Deng et al. A survey of real-time ethernet modeling and design methodologies: From AVB to TSN. *ACM Computing Surveys (CSUR)*, 55(2):1–36, 2022.

Saptarshi Ghosh et al. Intelligent routing as a service, a flexible routing framework for knowledge-defined networks. In *2024 IFIP Networking Conference*, pp. 5–13, 2024.

Liyuan Gu et al. Per-flow network measurement with distributed sketch. *IEEE/ACM Transactions on Networking*, 32(1):411–426, 2024.

Hasibul Jamil et al. Flowtracer: A tool for uncovering network path usage imbalance in ai training clusters. In *2025 IEEE International Conference on Communications (ICC)*, pp. 6988–6993, 2025.

Justin H. Kong et al. Joint routing and resource allocation in covert multi-flow heterogeneous networks. *IEEE Communications Letters*, pp. 1–1, 2025a.

Rui Kong et al. Serving moe models on resource-constrained edge devices via dynamic expert swapping. *IEEE Transactions on Computers*, 74(8):2799–2811, 2025b.

Svetlana Pavlitska, Christian Hubschneider, Lukas Struppek, and J. Marius Zöllner. Sparsely-gated mixture-of-expert layers for cnn interpretability. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, 2023.

Xingfu Wang et al. A dynamic opportunistic routing protocol for asynchronous duty-cycled wsns. *IEEE Transactions on Sustainable Computing*, 8(3):314–327, 2023.

Yutong Zhai et al. Joint routing and sketch configuration in software-defined networking. *IEEE/ACM Transactions on Networking*, 28(5):2092–2105, 2020.

Hao Zheng et al. Rethinking fine-grained measurement from software-defined perspective: A survey. *IEEE Transactions on Services Computing*, 15(6):3649–3667, 2022.

Xun Zhou et al. A dataset collected in real-world industrial control systems for network attack detection. *Scientific Data*, 13(1), 2026.

Zhengyan Zhou et al. Sketchguide: Reconfiguring sketch-based measurement on programmable switches. In *2022 IEEE 30th International Conference on Network Protocols (ICNP)*, pp. 1–11, 2022.