

LLM-as-a-Judge for Dynamic Resource Allocation in QoT-Aware Elastic Optical Networks

Talles S. Magalhães, Awlana Santana, Danilo Loureiro Carneiro, Fabricio R. L. Lobato, Carlos Natalino, João C. W. A. Costa, and Paolo Monti

Abstract—Elastic optical networks (EONs) require online routing, modulation, and spectrum assignment (RMSA) under spectrum-continuity and physical-layer constraints. This paper evaluates compact large language models (LLMs) as judges that select among quality of transmission (QoT)-feasible actions generated by candidate heuristics. Candidate actions are decoded, QoT-checked, deduplicated, and summarized through physical and spectral descriptors. A frozen shortlist ranker then supplies up to three candidates to the LLM judge. On the *Nobel-EU* topology, three zero-shot, temperature-zero LLM judges are compared with fixed heuristics and a shortlist ranker-only baseline from 100 to 300 Erlang. Blocking is rare at low loads, but under congestion, Llama-3.1 8B has the lowest observed blocking, with 77.8% and 42.0% reductions relative to the baseline at 250 and 300 Erlang, respectively. Diagnostics show that most calls contain multiple candidates and that LLM selections often differ from the ranker’s top candidate.

Keywords—elastic optical networks, routing, modulation, and spectrum assignment, quality of transmission, LLM-as-a-Judge.

I. INTRODUCTION

Elastic optical networks (EONs) mitigate fixed-grid inefficiencies under heterogeneous bandwidth demands. Their finer spectral granularity and adaptive configuration of spectrum, modulation, and bit rate allow optical resources to be matched more closely to traffic requirements, improving spectral efficiency and operational flexibility [1]. This flexibility, however, also makes online control more difficult because each new connection request must be provisioned under coupled routing, spectrum, and physical-layer constraints [2].

In dynamic EON operation, resource assignment is commonly formulated as the routing, modulation, and spectrum assignment (RMSA) problem [2]. A feasible decision must jointly select a route, a modulation format, and a contiguous spectral region while preserving continuity across links and satisfying transmission-quality constraints [3]. Raw spectrum availability alone is not sufficient because quality of transmission (QoT), amplified spontaneous emission (ASE) noise, nonlinear interference (NLI), and reach limitations all affect whether a candidate lightpath is admissible [3]. Repeated arrivals and departures of connections progressively fragment the spectrum, reducing long-term usability of available slots and increasing the connection blocking rate [4].

Talles S. Magalhães, Awlana Santana, Danilo Loureiro Carneiro, Fabricio R. L. Lobato, and João C. W. A. Costa are with the Institute of Technology, Federal University of Pará, Belém-PA, Brazil, e-mail: talles.magalhaes@itec.ufpa.br. Carlos Natalino and Paolo Monti are with the Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden.

The online nature of RMSA makes fixed heuristics the most practical choice in many operating regimes [2]. These heuristics encode distinct local biases: first-fit and last-fit differ in spectrum placement, load-balancing spreads traffic more evenly, highest-SNR prioritizes QoT margins over spectrum efficiency, and lowest-fragmentation attempts to preserve future spectral contiguity. Each bias can improve performance in some states, but none is uniformly beneficial across all network states [4]: as load increases and fragmentation, route pressure, and QoT constraints evolve, the locally preferable heuristic can also change.

This work formulates candidate selection as a large language model (LLM)-as-a-Judge arbitration task. The LLM does not propose lightpaths itself: it reviews network-state metrics and candidate summaries and selects one of the admissible actions produced by the fixed heuristics. Three LLM judges are compared with fixed heuristics and the ranker-only baseline.

II. RELATED WORK

In online QoT-aware RMSA, the coupling of spectrum allocation and physical-layer impairments makes arbitration among heuristic-generated candidates a dynamic multi-objective decision problem [3]. At each request arrival, the controller must trade off immediate spectrum utilization and QoT margins against the preservation of long-term spectral contiguity [4]. Classical RMSA heuristics provide computationally inexpensive compromises, but their fixed biases are analogous to static scalarization in conventional multi-objective optimization [5]. Under dynamic traffic, fixed preferences can become misaligned with the network state, motivating adaptive arbitration [6].

Spectrum fragmentation, route contention, and QoT margins evolve with connection arrivals and departures, producing a broad, non-stationary state space. Although machine learning (ML) has been studied for optical network management [7], supervised ML is difficult to apply directly to online RMSA because training data must cover diverse loads, occupancy patterns, and physical-layer conditions [8]. Moreover, if labels represent optimal actions, generating them requires repeatedly solving the NP-hard RMSA problem for sampled states [2], making dataset construction computationally expensive and difficult to scale.

Language models have been studied as constrained decision-support components in network control, including knowledge-free network management [9] and adaptive resource allocation under changing objectives [10]. Broader applications

are surveyed in [11], [12]. Recent work incorporated an LLM into reinforcement-learning-based RMSA as a high-level adaptation mechanism, guiding reward design according to network objectives and conditions [8]. These studies indicate that structured network telemetry can support language-model decisions within constrained action spaces.

LLM-as-a-Judge evaluation uses a language model to score, compare, or rank candidate outputs, reducing the amount of expert assessment required in open-ended evaluations [13]. We adapt that role to online control. The judge receives only QoT-feasible RMSA actions and returns one candidate, so lightpath generation remains outside the model.

Each request triggers a decision, making inference latency an operational constraint. We therefore use compact LLM judges and bound the action space with candidate heuristics and QoT filtering. LLM arbitration over QoT-feasible RMSA candidates under dynamic traffic remains largely unexplored. Related work selects LLMs for network-automation tasks under joint quality-efficiency constraints [14]. In this work, we adapt the LLM-as-a-Judge paradigm to online heuristic selection in QoT-aware RMSA.

III. LLM JUDGE FOR QOT-AWARE RMSA

As illustrated in Fig. 1, the end-to-end pipeline separates the optical environment from the LLM judge. The agent receives structured JSON payloads and returns a candidate selection, while feasibility checks and physical-layer constraints remain enforced by the environment. This separation keeps invalid lightpaths outside the LLM action space and allows direct comparison with fixed heuristics and the ranker-only baseline.

At each request arrival, candidates are represented by compact physical and spectral indicators. The QoT margin M_{QoT} is the available generalized signal-to-noise ratio (GSNR) minus the GSNR threshold of the selected modulation format. We use the clipped margin $M_c = \min(\max(M_{\text{QoT}}, 0), 3)$, capped at 3 dB, so candidates are not favored for excess QoT headroom. The path-common free-slot ratio ρ_{free} measures the fraction of spectrum slots simultaneously available across all links of the candidate route. The slot ratio ρ_{slot} compares candidate slot demand with the minimum for the same request. Following fragmentation-aware RMSA descriptors [15], route pressure (R_p) summarizes congestion and spectral misalignment along the route:

$$R_p = \frac{u_{\max} + \bar{u} + c_r + (1 - r_{ssr})}{4}. \quad (1)$$

Here, $u_{\max}$ and $\bar{u}$ are the maximum and mean link utilization along the candidate route, c_r is a normalized route-cut indicator, and r_{ssr} is the residual spectral spread. Larger R_p denotes greater congestion or spectral misalignment.

Local damage (D_l) captures fragmentation around the selected spectrum block:

$$D_l = \frac{b_a + (1 - l_b) + (1 - f_l) + n_b}{4}. \quad (2)$$

The variables b_a , l_b , f_l , and n_b denote the number of affected local free blocks, left residual block size, local free-spectrum fraction, and number of neighboring residual blocks,

respectively, all normalized to $[0, 1]$. Larger D_l values indicate stronger local fragmentation.

These descriptors summarize online RMSA trade-offs without exposing the full network state.

A. Candidate Generation and QoT Filtering

For each request, five RMSA heuristics generate candidate actions in a fixed order: first-fit, load-balancing, highest-SNR, last-fit, and lowest-fragmentation. Each encoded action specifies a route, modulation format, and initial spectrum slot. Duplicate encoded actions are merged and attributed to the first source in this order.

A candidate is QoT-feasible when the calculated GSNR is greater than or equal to the threshold required by the selected modulation format [3], [16]. The LLM therefore receives only decoded, deduplicated, QoT-checked candidates and does not arbitrate over infeasible or ill-formed actions.

B. Shortlist Ranker and Arbitration Protocol

When multiple QoT-feasible candidates remain, a frozen shortlist ranker constructs a compact set of actions for the LLM. The ranker is not intended to solve RMSA globally. It ranks candidates by slot efficiency, route relief, fragmentation preservation, common free spectrum along the route, and QoT robustness. Alone, it forms the ranker-only baseline.

The ranker maps the descriptors into five normalized components: $E_s = 1/\rho_{\text{slot}}$, $E_r = 1 - R_p$, $E_d = 1 - D_l$, $E_f = \rho_{\text{free}}$, and $E_q = \min(M_c, 1.5)/1.5$. Using these components, the frozen shortlist ranker assigns candidate c the following score:

$$Q(c) = w_s E_s(c) + w_r E_r(c) + w_d E_d(c) + w_f E_f(c) + w_q E_q(c) - P(c). \quad (3)$$

The frozen weights are $w_s = 0.24$, $w_r = 0.34$, $w_d = 0.12$, $w_f = 0.18$, and $w_q = 0.12$. They were obtained by normalizing the feature importances of a random forest trained to predict blocking from separate preliminary 250-Erlang runs and were frozen for all model comparisons.

$P(c)$ penalizes redundant and dominated options:

$$P(c) = \lambda_{\text{dom}} I_{\text{dom}}(c) + \lambda_{\text{route}} I_{\text{route}}(c) + \lambda_{\text{same}} I_{\text{same}}(c). \quad (4)$$

The frozen penalty weights are $\lambda_{\text{dom}} = 0.06$, $\lambda_{\text{route}} = 0.10$, and $\lambda_{\text{same}} = 0.12$. The I terms are binary indicator functions for candidates dominated by another QoT-feasible option (I_{dom}), same-slot alternatives with worse route pressure and no local-damage improvement (I_{route}), and same-path modulation variants that consume at least two additional slots (I_{same}). The shortlist retains up to three high-scoring candidates and shuffles their order to mitigate LLM position bias. The LLM receives neither $Q(c)$ nor shortlist ranks. Algorithm 1 summarizes the online flow. For each request, the five heuristics generate candidate actions, which are decoded, checked against the QoT threshold, and deduplicated. Requests with no valid candidate are rejected, while a single remaining candidate is executed directly. When multiple candidates remain, the ranker builds a shortlist of up to three candidates

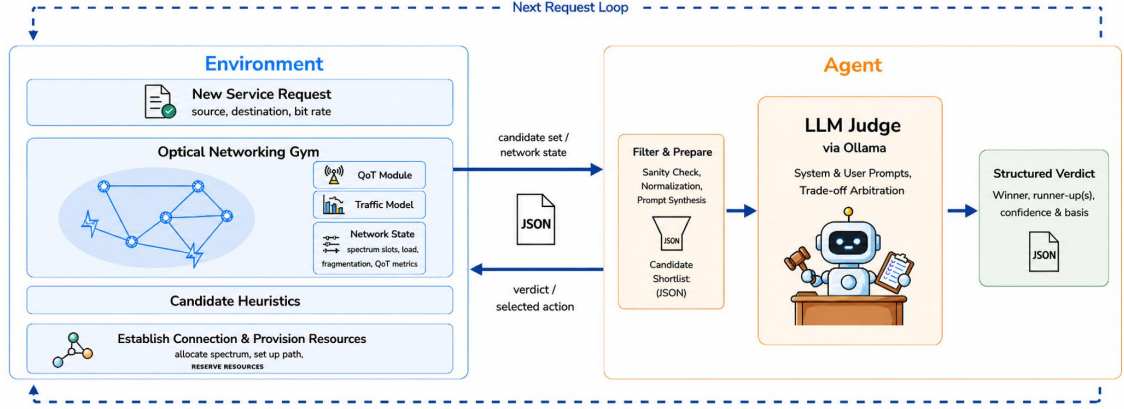


Fig. 1. System architecture of the proposed LLM-as-a-Judge approach.

for LLM arbitration. If the LLM output is invalid or cannot be parsed, the controller executes the top candidate under the frozen shortlist ranker.

Algorithm 1: Online LLM judge step.

Input: request r , network state s , heuristic set H
Output: selected RMSA action or rejection

```

1  $C \leftarrow \emptyset$ 
2 foreach  $h \in H$  do
3    $a \leftarrow h(s, r)$ 
4    $c \leftarrow \text{decode route, modulation, and slot from } a$ 
5   if  $c$  is not a rejection and reaches modulation QoT threshold then
6      $C \leftarrow C \cup \{c\}$ 
7  $C \leftarrow \text{deduplicate } C \text{ by encoded action}$ 
8 if  $C = \emptyset$  then
9   return a rejection action
10 if  $|C| = 1$  then
11   execute the single candidate without the LLM
12 else
13    $S \leftarrow \text{shortlist } C \text{ based on ranker } Q \text{ and shuffle order}$ 
14    $y \leftarrow \text{LLM}(\text{payload}(r, s, S))$ 
15   if  $y$  identifies a valid candidate in  $S$  then
16     execute selected candidate  $y$ 
17   else
18     execute fallback candidate  $q = \arg \max_{c \in S} Q(c)$ 
    
```

C. Judge Payload

Listing 1. LLM judge prompt payload.

```

{
  "request": {
    "services_processed": 0,
    "source": "Vienna",
    "destination": "Lyon",
    "bit_rate_gbps": 100,
    "network_state": {
      "episode_service_blocking_rate": 0.0,
      "network_util_mean": 0.0,
      "free_slots_ratio": 1.0,
      "candidates": [
        {
          "candidate_id": "TJMM",
          "route": {
            "path_hops": 4,
            "path_length_km": 1295.38,
            "mod": "64QAM",
            "slot": 0,
            "slots": 2,
            "metrics": {
              "qot_feasible": true,
              "qot_margin": 1.21,
              "local_dmg": 0.41,
              "route_press": 0.0013
            }
          },
          "candidate_id": "Q2J6",
          "...": "..."
        }
      ]
    },
    "pairwise_deltas": [
      {
        "candidate_id": "TJMM",
        "vs_candidate_id": "Q2J6",
        "delta_required_slots": 0,
        "delta_route_pressure_score": -0.012,
        "delta_local_damage_score": 0.08,
        "delta_qot_margin_clipped_db": 0.42
      }
    ],
    "prompt_context": {
      "progress": 0.0,
      "congestion": "open",
      "risk": "low"
    }
  }
}
    
```

The controller serializes the selected candidates and network state into a compact JSON payload (Listing 1) with request data, aggregate state indicators, candidate attributes, QoT/fragmentation descriptors, pairwise deltas, and regime bands. Candidate identifiers are short neutral tokens assigned after deduplication and shuffling. They do not encode heuristic names, shortlist-ranker positions, or alphabetical order. Pairwise deltas summarize relative differences between candidates without exposing the baseline decision. If the LLM output cannot be parsed or identifies an invalid action, the controller executes the top candidate q under the frozen shortlist ranker.

IV. SIMULATION METHODOLOGY

A. Network Scenario and Physical-Layer Model

The simulations use Optical Networking Gym for dynamic lightpath provisioning over the European *Nobel-EU* topology [17]. Transmission feasibility is evaluated using GSNR:

$$\text{GSNR} = \frac{P_{\text{ch}}}{P_{\text{ASE}} + P_{\text{NLI}}}. \quad (5)$$

Here, P_{ch} is the channel power and P_{ASE} is the accumulated ASE noise power [3]. The nonlinear contribution is $P_{\text{NLI}} = P_{\text{SCI}} + P_{\text{XCI}}$, estimated by the enhanced Gaussian noise model, with P_{SCI} and P_{XCI} denoting self-channel and cross-channel interference power, respectively [16]. Table I summarizes the simulation parameters.

Connection arrivals follow a Poisson process, holding times are exponentially distributed, and source–destination pairs and bit rates are sampled uniformly. All methods use the same seed sequence at each load, and each reported blocking rate is the mean across ten 1,000-request episodes.

B. Models, Baselines, and Inference Setup

The benchmark compares three LLM judges with five fixed heuristics and the ranker-only baseline. *Nemotron-3-Nano* 4B is an NVIDIA model for local inference [18]. *Llama-3.1* 8B is an open-weight instruction-following model [19]. *GPT-OSS 20B* is the largest model in this comparison [20]. All runs use zero-shot decoding at temperature 0, without chain-of-thought generation or model-specific prompt tuning. Inference ran on

TABLE I
SIMULATION SETUP.

Parameter	Value
Loads	100, 150, 200, 250, 300 Erlang
Episodes per load	10
Requests per episode	1000
k -shortest paths	5
Spectrum slots	320
Bit rates	40, 100, 400 Gbps
Slot bandwidth / total band	12.5 GHz / 4 THz
Mean holding time	10800 s
Launch power / margin	0 dBm / 0 dB
Span / attenuation / noise figure	80 km / 0.2 dB/km / 4.5 dB
Modulations	BPSK, QPSK, 8QAM, 16QAM, 32QAM, 64QAM

a workstation with two RTX 5090 GPUs, 192 GB of RAM, and an AMD Ryzen 9 9900X CPU.

C. Evaluation Metrics

Blocking rate (B) is defined as $B = N_{\text{blocked}}/N_{\text{requests}}$, where N_{blocked} and N_{requests} denote the numbers of blocked and total requests, respectively. Choice distribution $P(h)$ describes how often each heuristic source h appears in the admitted decisions. It is computed from the number of admitted connections selected from heuristic h , denoted by N_h , relative to the total number of admitted connections N_{selected} .

We report diagnostics of the LLM arbitration process: agreement, shortlist collapse, and runtime. Let y_i be the candidate selected by the LLM for request i and q_i the top candidate under the frozen ranker Q . Agreement is the fraction of judge decisions for which $y_i = q_i$. Shortlist collapse is the fraction of requests in which a single candidate survives QoT filtering and deduplication and is executed directly without the LLM (Algorithm 1). Runtime is the mean inference time per decision, measured from the start timestamp t_i^{start} to the end timestamp t_i^{end} of each LLM call. Because the LLM judge augments rather than replaces RMSA heuristics, this response time measures its added operational latency. Fig. 4 summarizes these diagnostics in four panels: (a) agreement, (b) the shortlist size distribution, (c) collapse, and (d) runtime.

V. RESULTS AND DISCUSSION

A. Blocking Performance Across Load Regimes

Fig. 2 compares the blocking rates of the three LLM judges, five fixed heuristics, and the ranker-only baseline from 100 to 300 Erlang. At 100, 150, and 200 Erlang, nearly all methods have zero blocking, with only isolated nonzero cases. Blocking is too rare in this range to distinguish them.

Blocking rates separate at 250 Erlang. Llama-3.1 8B has the lowest blocking in the reported runs, 87.5% below the best fixed heuristic, while all three LLM judges outperform the ranker-only baseline. At 300 Erlang, Llama-3.1 8B remains 38.9% below the strongest fixed heuristic and 42.0% below the baseline. GPT-OSS 20B also remains below both references, by 20.0% and 24.0%, respectively.

In these runs, LLM arbitration can outperform fixed heuristics and the ranker-only baseline while retaining deterministic candidate generation and QoT filtering.

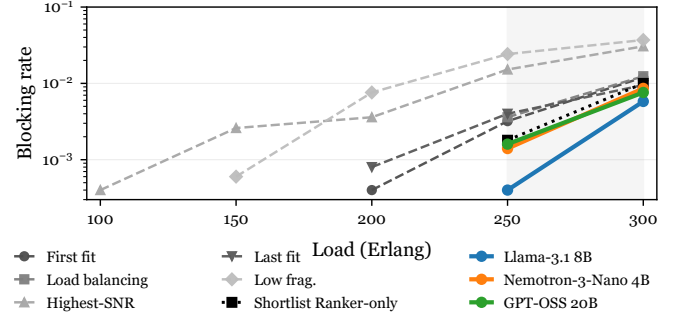


Fig. 2. Blocking rate from 100 to 300 Erlang.

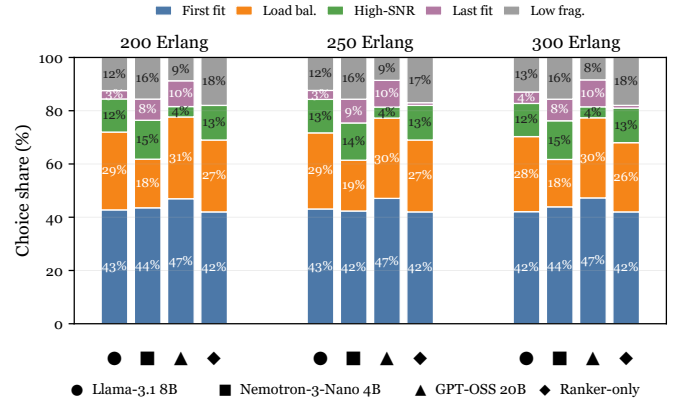


Fig. 3. Heuristic-source distribution at 200, 250, and 300 Erlang.

Fig. 3 compares the heuristic-source distributions of the four selection methods at 200, 250, and 300 Erlang. At 250 Erlang, Llama-3.1 8B, Nemotron-3-Nano 4B, and GPT-OSS 20B reduced blocking by 77.8%, 22.2%, and 11.1%, respectively, relative to the ranker-only baseline. First-fit remained the most frequent source, accounting for 42–47% of the choices. The load-balancing shares were 29%, 19%, and 30% for Llama, Nemotron, and GPT-OSS, respectively, compared with 27% for the baseline; the corresponding highest-SNR shares were 13%, 14%, and 4%, compared with 13%. Llama also selected last-fit less frequently than the other LLMs (3% versus 9% and 10%). These distributions are consistent with its lower blocking being associated with combining route relief through load balancing with QoT robustness through highest-SNR, whereas Nemotron placed less emphasis on load balancing and GPT-OSS placed less emphasis on QoT.

B. Judge Behavior and Inference Latency

Fig. 4(a) shows that the LLM judges often select a candidate other than the frozen shortlist ranker's top choice. Llama-3.1 8B reaches the highest mean agreement, at 66.8%, while GPT-OSS 20B and Nemotron-3-Nano 4B remain at 59.8%.

Fig. 4(b) shows that most LLM calls contain three candidates, with a smaller but persistent share of two-candidate cases. Because the controller never invokes the LLM with fewer than two candidates (Algorithm 1), the collapse rates in Fig. 4(c) reflect requests in which the LLM is bypassed entirely. Up to 300 Erlang, the maximum collapse rates are

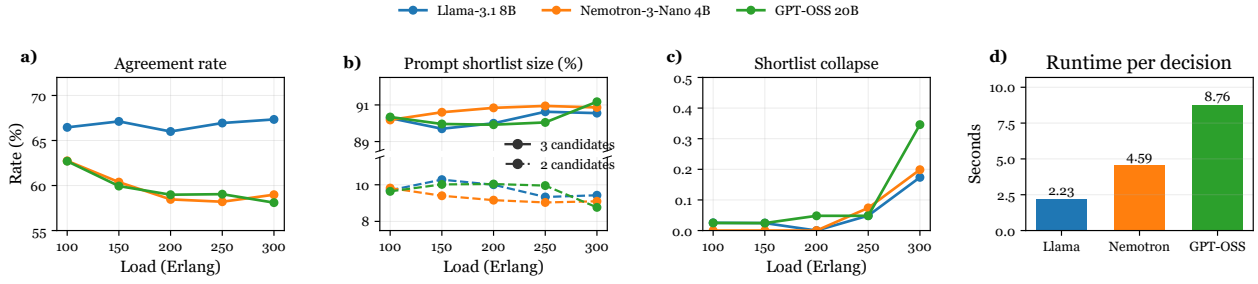


Fig. 4. Judge metrics from 100 to 300 Erlang.

0.17% for Llama-3.1 8B, 0.20% for Nemotron-3-Nano 4B, and 0.35% for GPT-OSS 20B. The ranker-only baseline uses the same candidate-generation stage, so these diagnostics separate shortlist availability from final LLM arbitration.

Fig. 4(d) reports mean inference times of 2.23 s for Llama-3.1 8B, 4.59 s for Nemotron-3-Nano 4B, and 8.76 s for GPT-OSS 20B. Runtime did not track model size, which may reflect serving and architectural differences. These latencies motivate further optimization before live control-plane deployment.

VI. CONCLUSION

We evaluated LLM-as-a-Judge arbitration over QoT-feasible, heuristic-generated RMSA candidates. Three zero-shot LLM judges were compared with five fixed heuristics and the ranker-only baseline. Blocking was too rare to distinguish methods at low loads. At 250 and 300 Erlang, Llama-3.1 8B yielded the lowest blocking, 77.8% and 42.0% below the baseline. Relative to the best fixed heuristic, the corresponding reductions were 87.5% and 38.9%. Mean agreement ranged from 59.8% to 66.8%, maximum shortlist collapse did not exceed 0.35%, and mean inference latency ranged from 2.23 to 8.76 s per decision.

Future work will examine cost and energy efficiency, multi-band EONs, and comparisons with ML selectors and multi-agent solutions.

Code and simulation configuration: <https://github.com/Talles64/optical-networking-gym/tree/SBRT-2026>.

ACKNOWLEDGEMENTS

This work was supported by the National Council for Scientific and Technological Development (CNPq) and Amazon Sustainability Institute with Science and Innovation (iSACI).

REFERENCES

- [1] M. Jinno, H. Takara *et al.*, "Spectrum-efficient and scalable elastic optical path network: Architecture, benefits, and enabling technologies," *IEEE Commun. Mag.*, vol. 47, 2009.
- [2] B. C. Chatterjee, N. Sarma *et al.*, "Routing and spectrum allocation in elastic optical networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 17, 2015.
- [3] H. Beyranvand and J. A. Salehi, "A quality-of-transmission aware dynamic routing and spectrum assignment scheme for future elastic optical networks," *J. Lightw. Technol.*, vol. 31, 2013.
- [4] B. C. Chatterjee, S. Ba *et al.*, "Fragmentation problems and management approaches in elastic optical networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, 2018.
- [5] S. Kaddani, D. Vanderpooten *et al.*, "Weighted sum model with partial preference information: Application to multi-objective optimization," *Eur. J. Oper. Res.*, vol. 260, 2017.
- [6] A. R. Adekoya and M. Helbig, "Decision-maker's preference-driven dynamic multi-objective optimization," *Algorithms*, vol. 16, no. 11, p. 504, 2023.
- [7] Y. Zhang, J. Xin *et al.*, "Overview on routing and resource allocation based machine learning in optical networks," *Opt. Fiber Technol.*, vol. 60, 2020.
- [8] S. Easwaran and M. Shadaram, "Smart resource utilization in elastic optical network using generative AI," in *2025 IEEE 4th International Conference on Computing and Machine Intelligence (ICMI)*, 2025, pp. 901–905.
- [9] H. Lee, M. Kim *et al.*, "Large language models for knowledge-free network management: Feasibility study and opportunities," *IEEE Access*, vol. 13, pp. 187 092–187 106, 2025.
- [10] H. Noh, B. Shim *et al.*, "Adaptive resource allocation optimization using large language models in dynamic wireless environments," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 10, pp. 16 630–16 635, 2025.
- [11] C. Liu, X. Xie *et al.*, "Large language models for networking: Workflow, advances, and challenges," *IEEE Network*, vol. 39, no. 5, pp. 165–172, 2025.
- [12] G. O. Boateng, H. Sami *et al.*, "A survey on large language models for communication, network, and service management: Application insights, challenges, and future directions," *IEEE Communications Surveys & Tutorials*, vol. 28, pp. 527–566, 2026.
- [13] L. Zheng, W.-L. Chiang *et al.*, "Judging LLM-as-a-judge with MT-bench and chatbot arena," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.
- [14] K. Rezaei, O. Ayoub *et al.*, "Human-grounded evaluation of large language models for optical network automation," in *International Conference on Transparent Optical Networks (ICTON)*, Jul. 2026.
- [15] Y. Yin, M. Zhang *et al.*, "Fragmentation-aware routing, modulation and spectrum assignment algorithms in elastic optical networks," in *OFC/NFOEC*, 2013.
- [16] M. R. Zefreh, F. Forghieri *et al.*, "Accurate closed-form real-time EGN model formula leveraging machine-learning over 8500 thoroughly randomized full C-band systems," *J. Lightw. Technol.*, vol. 38, 2020.
- [17] C. Natalino, T. Magalhães *et al.*, "Optical networking gym: An open-source toolkit for benchmarking resource assignment problems in optical networks," *J. Opt. Commun. Netw.*, vol. 16, 2024.
- [18] NVIDIA, "Nemotron-3-Nano-4B," https://developer.download.nvidia.com/assets/ace/model_card/Nemotron_3_Nano_4B.pdf, 2026.
- [19] A. Grattafiori, A. Dubey *et al.*, "The Llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.
- [20] OpenAI, "GPT-OSS-120b & GPT-OSS-20b model card," *arXiv preprint arXiv:2508.10925*, 2025.