2025 USENIX Annual Technical Conference

NetKeeper: Enhancing Network Resilience with Autonomous Network Configuration Update on Traffic Patterns and Anomalies

Zhaoyang Wan[1] , Rongxin Han[1] , Haifeng Sun[1], Qi Qi[1] , Zirui Zhuang[1] , Bo He[1] ,
Liang Zhang[2] , Jianxin Liao[1] , Jingyu Wang[1]
[1]*State Key Laboratory of Networking and Switching Technology,*
Beijing University of Posts and Telecommunications
[2]*Huawei Technologies Co., Ltd*

nirc.top

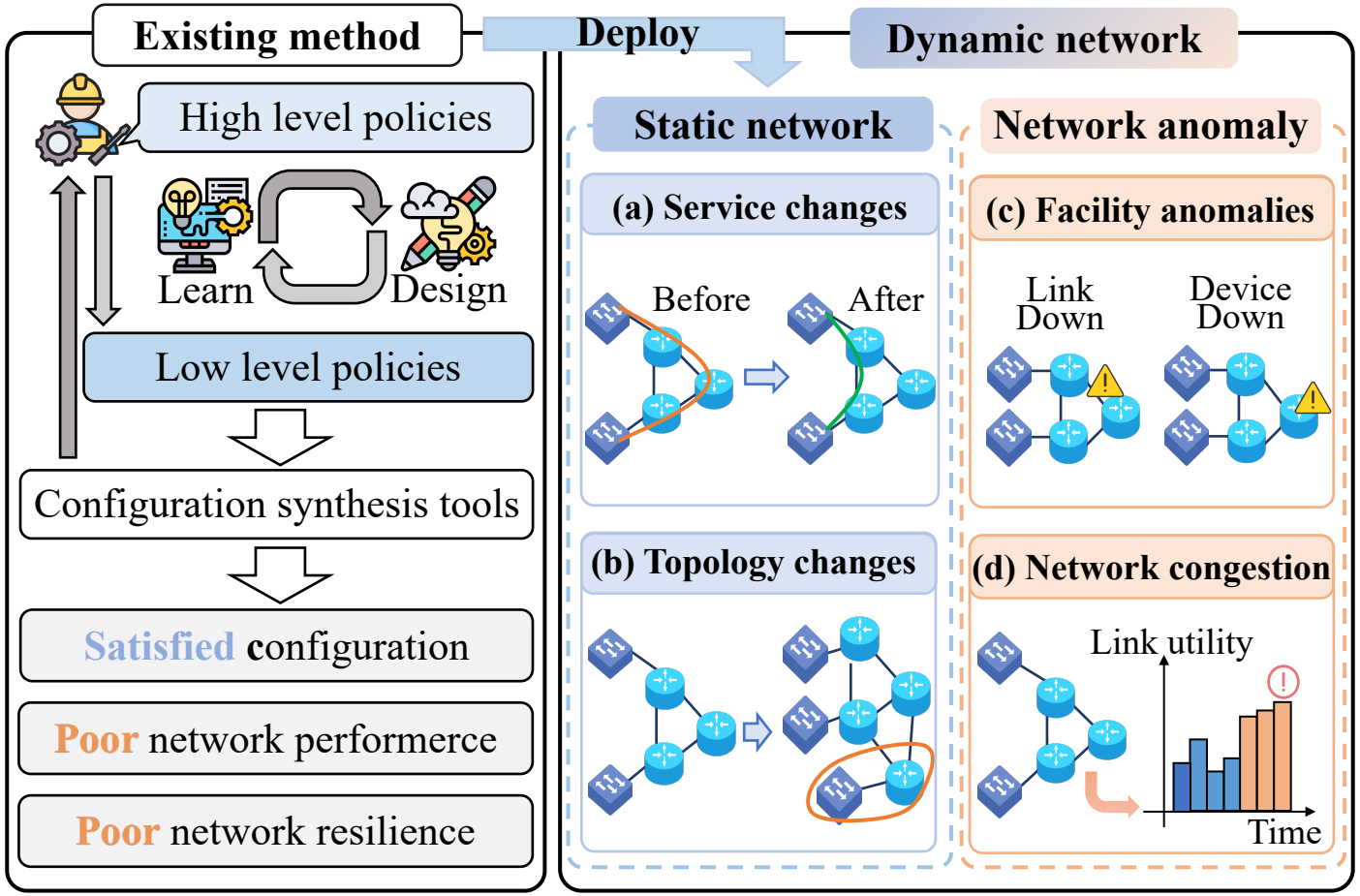
NETWORK INTELLIGENCE RESEARCH CENTER

Outline

- Background
- Challenges
- NetKeeper
- Evaluation
- Conclusion

Network Configuration Update

- **Configuration update based on intents in dynamic network under different scenarios**

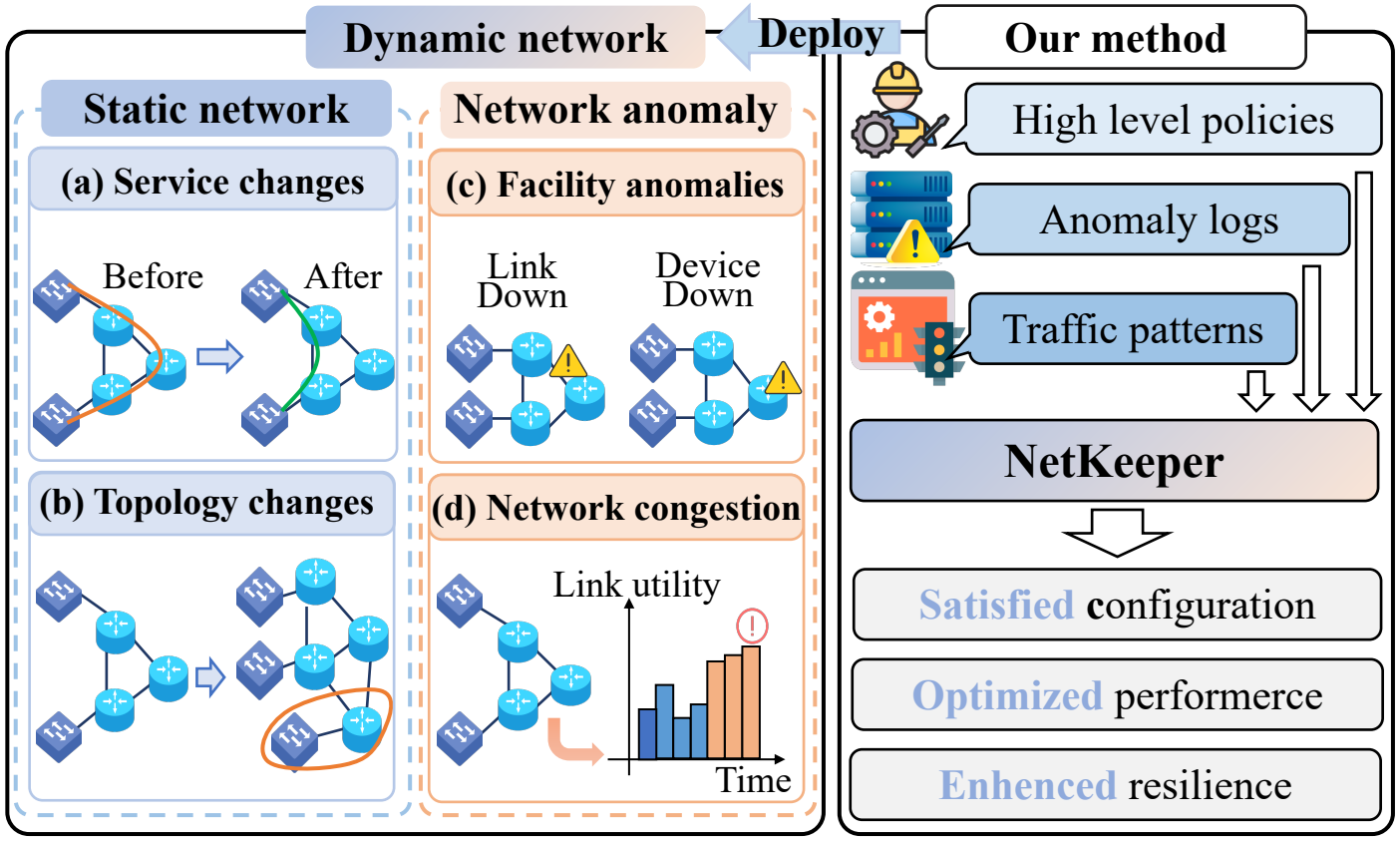


Satisfied configuration in static network ? Not enough !

Better network performance with high resilience in dynamic network !

Challenges under autonomous network

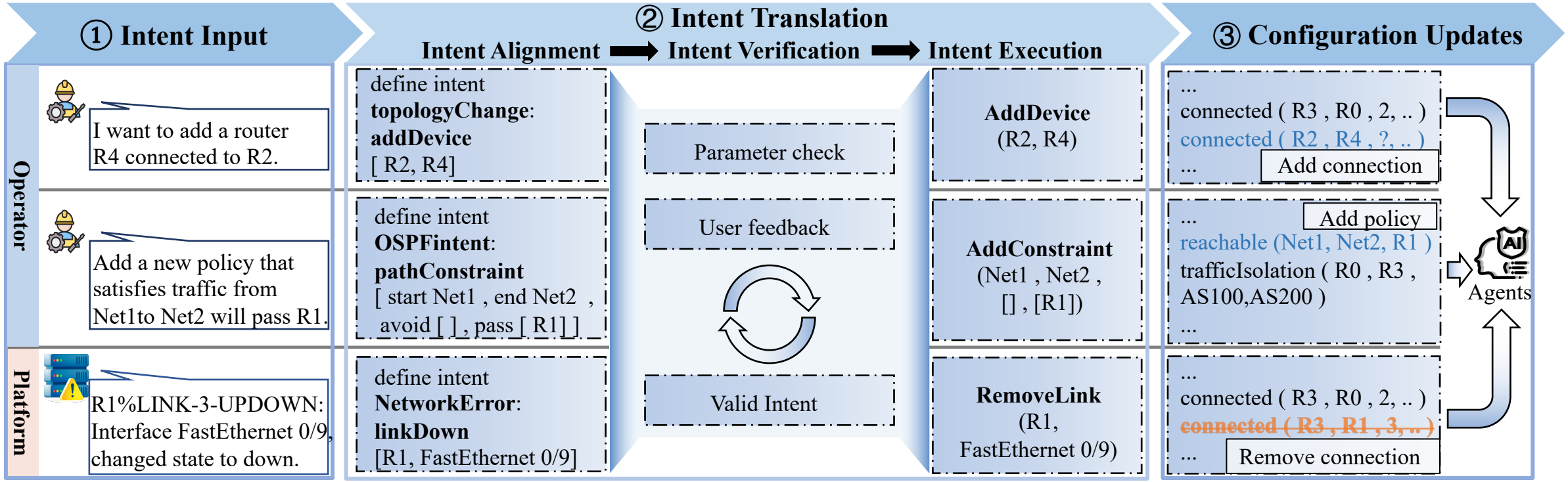
- **Multimodal intent integration**
- **Traffic-driven resilience enhancement**
- **Dynamic adaptive automation**



Find the way towards autonomous network ~

System Overview

- **Intent Translation Interface for Both Operators and Network Management Platforms**
- **Goal-Oriented, Traffic-Aware Configuration Update Model based on Multi-Agent**
- **An End-to-end Autonomous Configuration Update Framework for Dynamic Network**



Natural language or anmoly logs → DSL → API → New configuration

System Overview

❖ Optimization Objectives

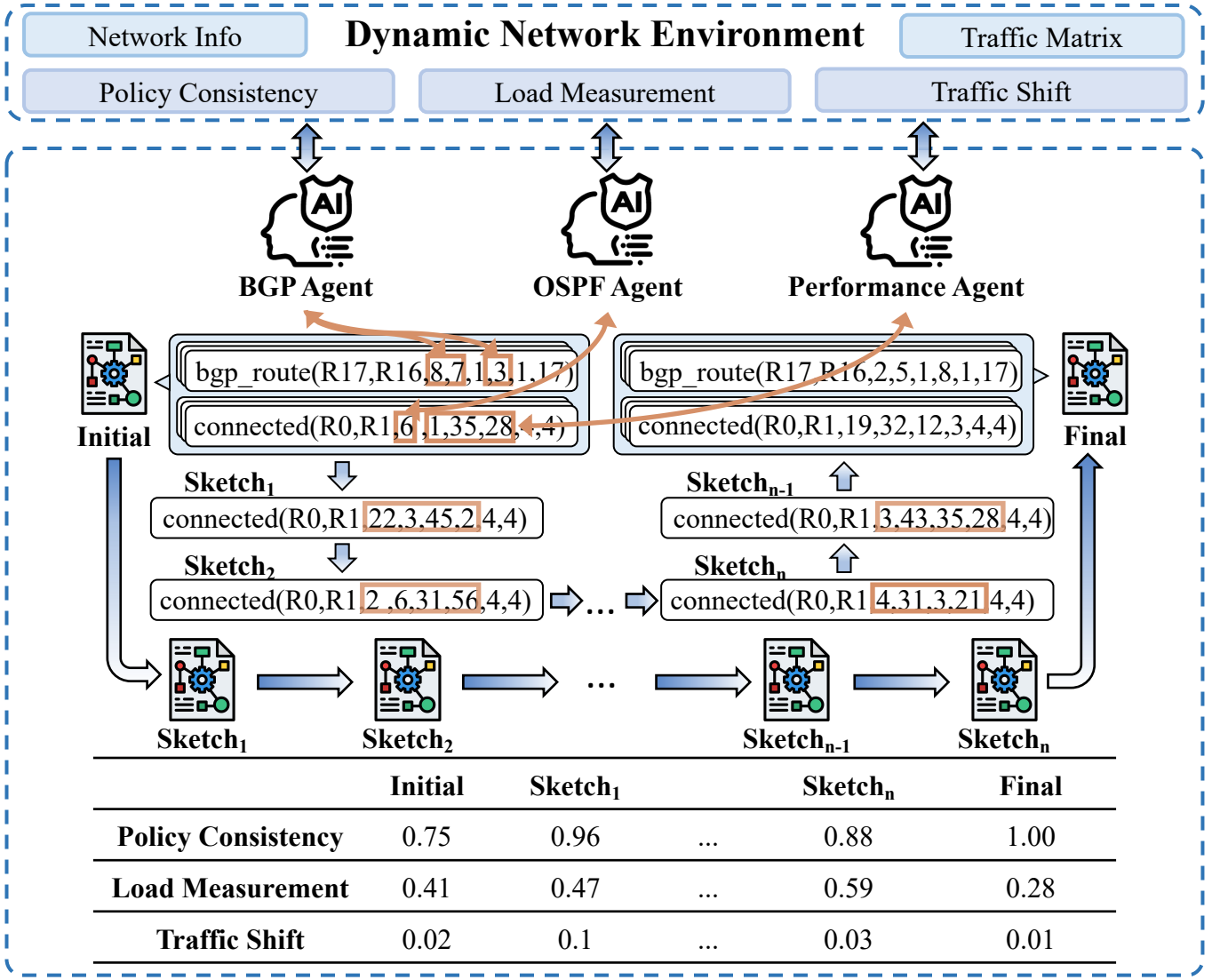
- Policy Consistency (Primary)
- Load Measurement
- Traffic Shift

❖ Multi-Agent System

- Specialized Agents for Different Tasks
- Coordination and Joint Optimization

❖ Working Environment

- Dynamic/Static Network Environment
- Real-Time Interaction
- Adaptive Evaluation



	Initial	Sketch$_1$		Sketch$_n$	Final
Policy Consistency	0.75	0.96	...	0.88	1.00
Load Measurement	0.41	0.47	...	0.59	0.28
Traffic Shift	0.02	0.1	...	0.03	0.01

Intent Translation

❖ Alignment for Intents from Operators and Management Platform

#	Natural language intent
1	Configure IP address 127.39.236.169 and subnet mask 255.255.255.248 for the Ethernet34 port of R55.
2	Set the OSPF Hello packet interval on R5 to 20 ms.
3	Configure access control for R45's own traffic, performing a deny operation on traffic from R19.
4	Remove R6 from the global network.
5	Adjust port weights to form a route from R2 to R8, avoiding R6 and R1 while including the routing nodes R5 and R3.

From Operators (Northbound)

#	Network anomaly log
6	R1%LINK-3-UPDOWN: Interface Fe 0/9 change state to down.
7	HOST_X%Node down: ping timeout (0 / 5).
8	R1%LINK-3-CONGESTION: Interface Gi1/0/1 output queue full, packets dropped: 1532.

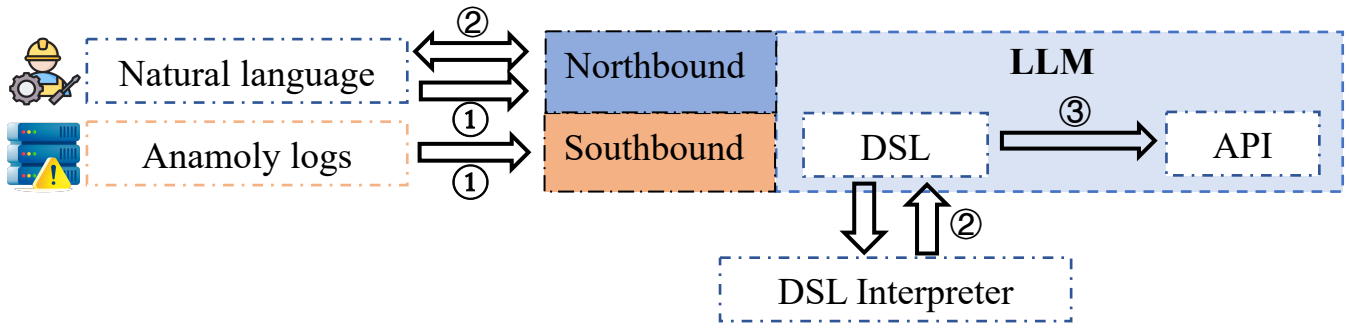
From Management Platform (Southbound)

Intent = {
Network Devices,
Network Protocols,
Proto col Options,
Network Rules,
Attributes,
Forwading Polices }

❖ Intent Validation via a DSL Derived from the Intent Model

Name	Type	Parameters
assignIp	Basic Config	port/subnet/mask
staticRouteBasic	Static Route	subnet/mask/next_hop
generateAcl	ACL	ip/subnet/action
assignAcl	ACL	router/interface/aclItem
establishNeighbor	OSPF/BGP	ip/mask/area
setPortWeight	OSPF	router/interface/weight
pathConstraint	OSPF/BGP	start/end/avoid/pass
assignAsNumber	BGP	asNumber
bgpPolicy	BGP	policies

(1) Part of DSL-supported operations.



(2) The process of intent translation.

Task Goals

❖ Goals

- **Higher Policy Consistency *(Primary)***

$$\pi = \frac{\sum_{p_i \in P} Satisfy(p_i)}{|P|}.$$

Three kinds polices are considered:

(1) forward(A, B, C) : Traffic from A to B must be forwarded to its neighbor C.

(2) reachable(A, B, C) : Traffic from A to B must pass by C.

(3) isolation(A, B, C, D) : One of forward(A, B, C) or forward(A, B, D) must be true at the same time.

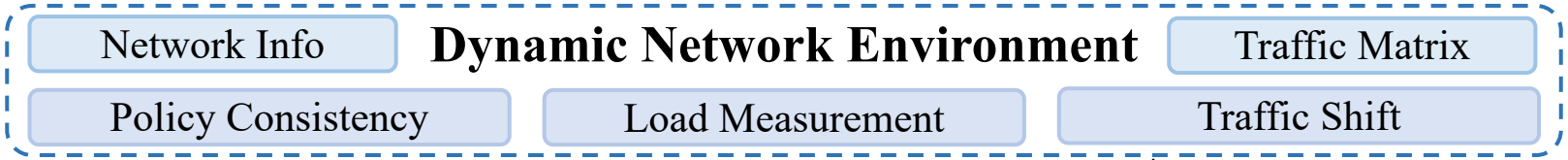
- **Lower Load Measurement** (Based on link utilization)

- **Lower Traffic Shift**

$$\tau = \frac{1}{|ND| \cdot |PF|} \sum_{nd \in ND} \sum_{pf \in PF} \begin{cases} 1, & \text{if } nh_{t-1}(nd, pf) \neq nh_t(nd, pf), \\ 0, & \text{otherwise.} \end{cases}$$

Method

❖ Multi-agent Deep Reinforcement Learning



Description	Parameter	Description	Parameter
OSPF weight	α	Local preference (LP)	γ_1
Bandwidth	β_1	AS path length (AS)	γ_2
Capacity	β_2	Multi-exit discriminator (MED)	γ_3
Queue length	β_3		

Table 3: Parameters of OSPF, link attributes and BGP.

- By assigning specialized agents to different parameter types and enabling their collaboration, our multi-agent approach simplifies complex configuration updates and ensures optimal overall network performance.

Environment

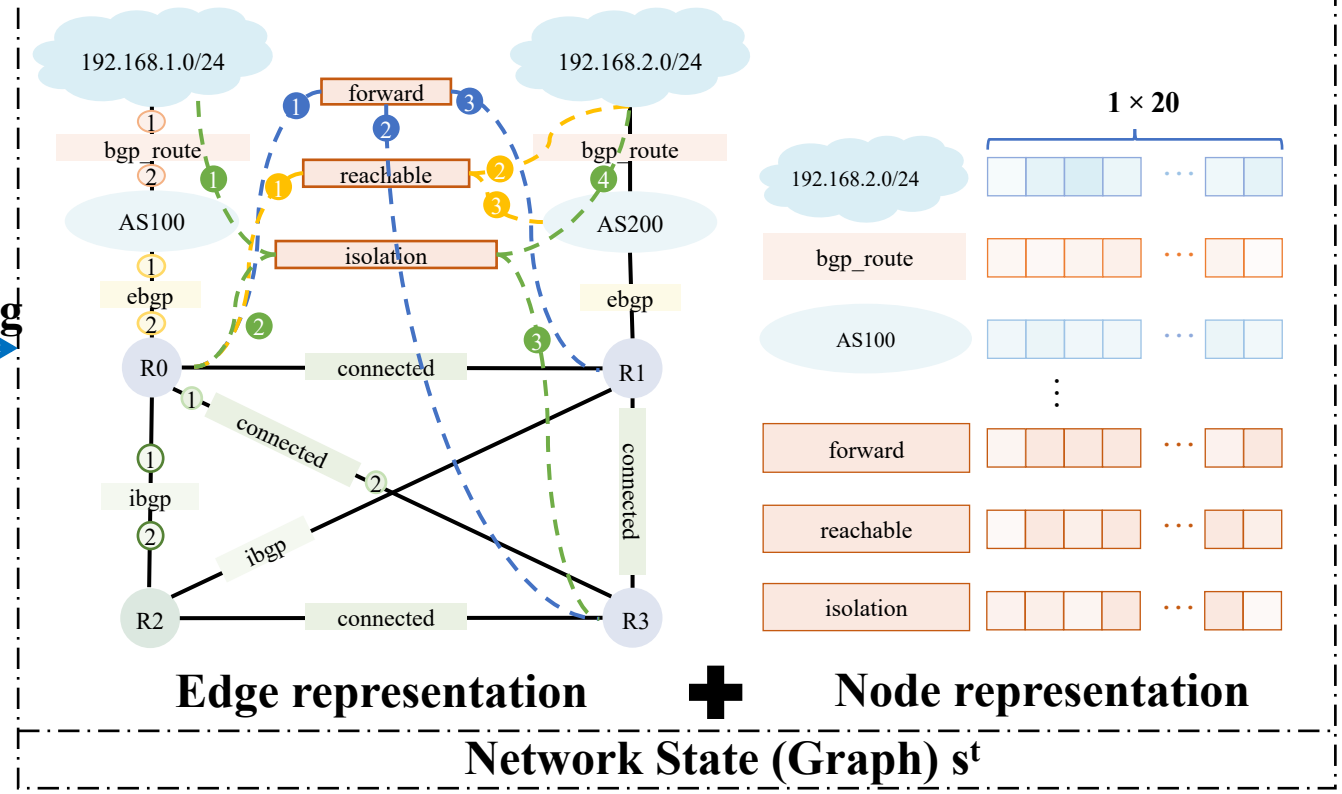
State (Graph)

- Nodes : Including router, network, external, and route_reflector
- Topology and Protocol information : Including ibgp, ebgp, connected, and bgp_route
- Routing polices: Including forward, isolation, and reachable

router (R0)
router (R1)
router (R3)
network (192.168.1.0/24)
network (192.156.2.0/24)
external (AS100)
external (AS200)
route_reflector (R2)
connected (R2 , R3 , 22 , 3 , 45 , 2 , 4 , 4)
connected (R3 , R0 , 2 , 6 ,31 , 56 , 5 , 2)
connected (R3 , R1 , 3 , 43 , 35 , 28 , 7 , 1)
connected (R1 , R0 , 4 , 31 , 3 , 21 , 2 , 8)
ibgp (R2 , R0)
ibgp (R2 , R1)
ebgp (R0 , AS100)
ebgp (R1 , AS200)
bgp_route (AS100 , 192.168.1.0/24 , 1 , 5 , 1 , 11 , 1 , 8)
bgp_route (AS200 , 192.168.2.0/24 , 23 ,13 , 1 , 33 , 1 , 8)
forward (R0 , R1 , R3)
isolation (R0 , R3 , 192.168.2.0/24 , 192.168.1.0/24)
reachable (R0 , AS200 , 192.168.2.0/24)

Network Sketch

Embedding

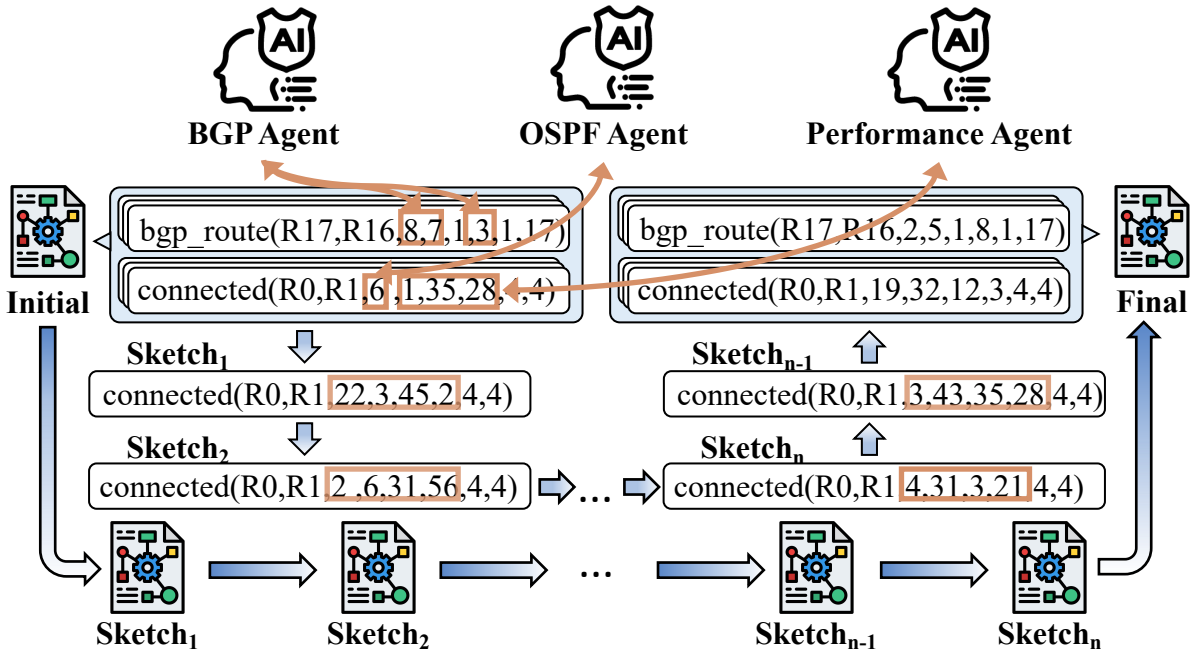


Network State (Graph) s^t

Environment

❖ Action

- Based on the previous network sketch, the action is a combination of outputs from different agents and is directly used to modify the attributes of the nodes.



Algorithm 1: State Update Process

Input: Current state $s = (V, E)$, Joint action a
Output: New state s'

Initialize s' as a copy of s;
for *each action* $a_{i,k} \in a$ **do**
  $i_k \leftarrow$ index corresponding to parameter type k;
  for *each node* j *in* V' **do**
    $V'_j[i_k] \leftarrow a_{i,k[j]}$;
  end
end
return $s' = (V', E)$;

Environment

❖ Reward Function

- Different agents adopt tailored reward functions according to their specific configuration tasks, with dynamic terms designed to ensure policy consistency as the primary objective before optimizing other goals such as load balancing and traffic shifting.

$$R(s,a)_{A_{OSPF}} = R(s,a)_{pol} + R(s,a)_{perf},$$

$$R(s,a)_{A_{BGP}} = R(s,a)_{pol},$$

$$R(s,a)_{A_{perf}} = R(s,a)_{perf}.$$

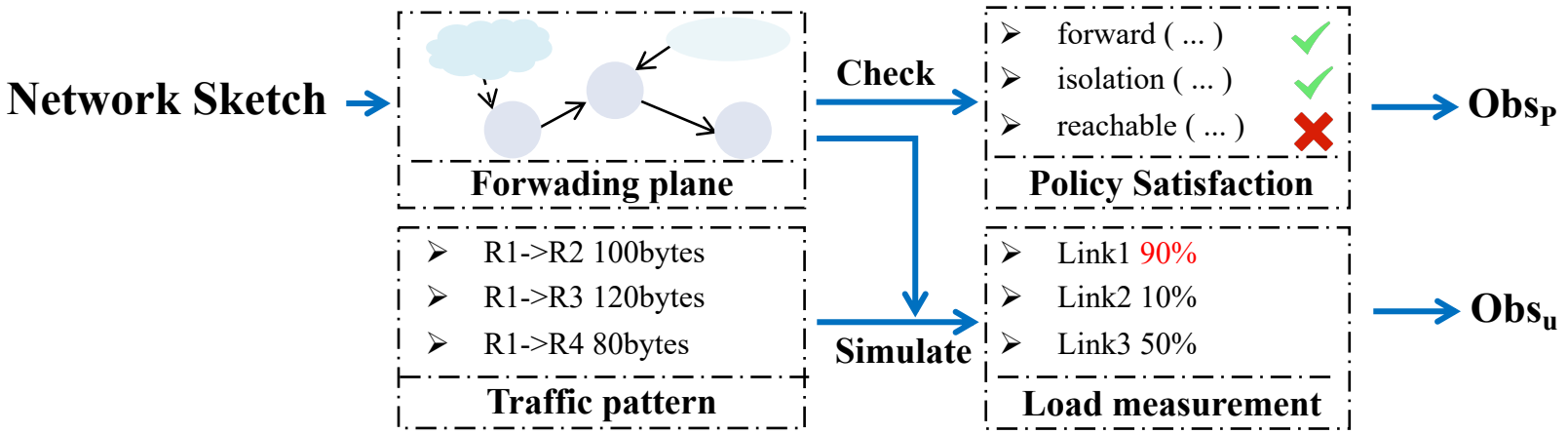
$$R(s,a)_{pol} = K \cdot \pi + R_s + R_d,$$

$$R_s = \begin{cases} 0, & \text{if } t = 0 \text{ or } a_t \neq a_{t-1}, \\ R_s - 1, & \text{if } a_t = a_{t-1}, \end{cases}$$

$$R_d = \begin{cases} 0, & \text{if } t = 0, \\ \sum_{p_i \in P}(Satisfy(p_i)_t - Satisfy(p_i)_{t-1}), & \text{if } t > 0. \end{cases}$$

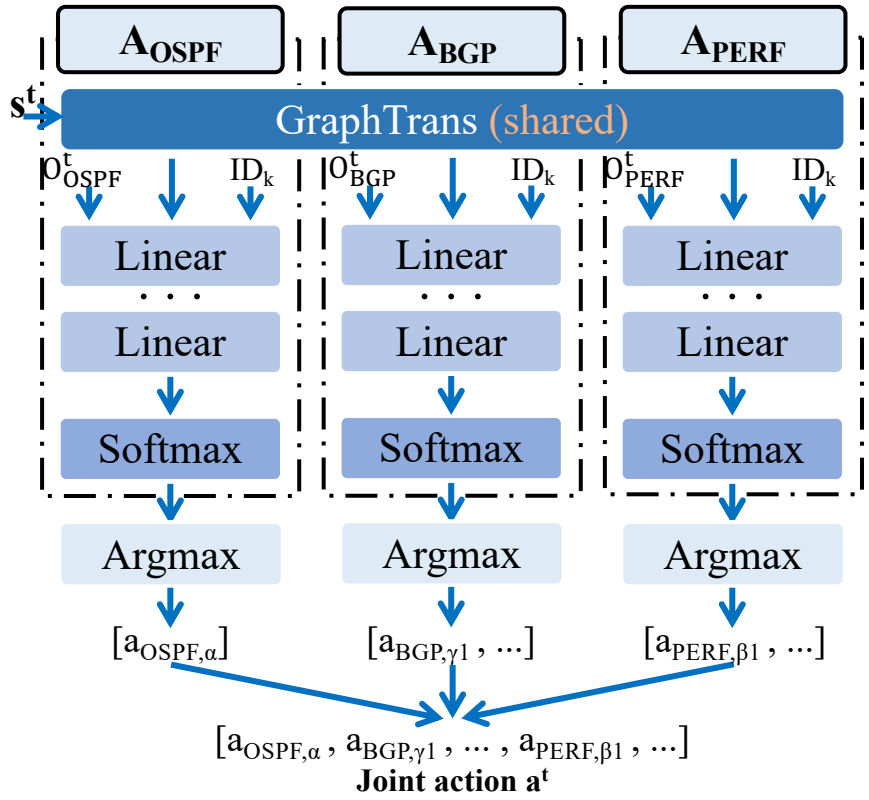
$$R(s,a)_{perf} = K \cdot ((1-\rho) + (1-\tau)).$$

❖ Observation From Environment

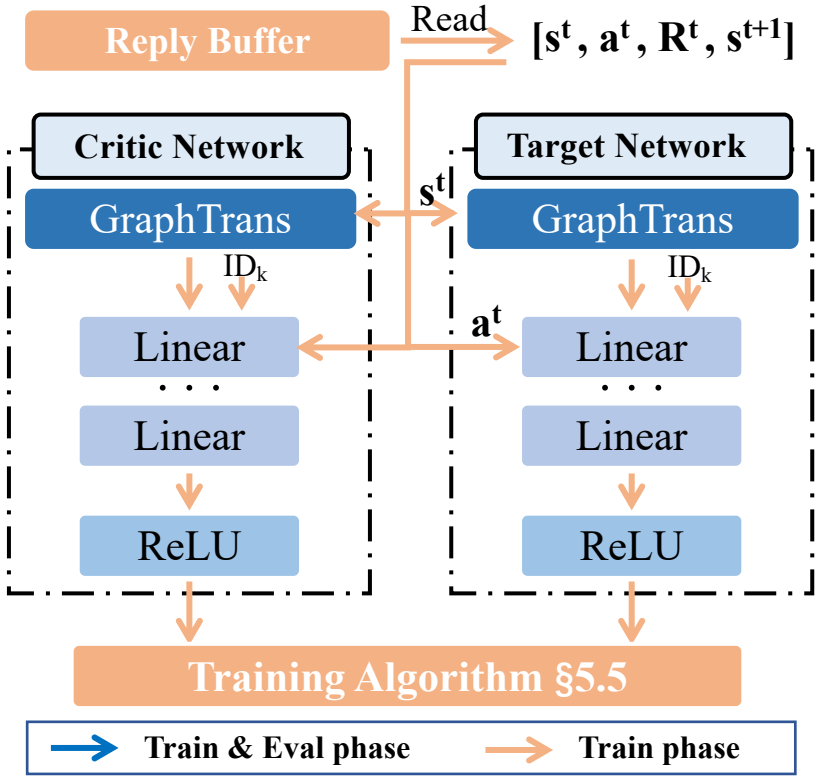


Configuration Update

❖ Actor-Critic Network



(a) Actor network



(b) Critic network

- Counter-factual multi-agent (COMA) is used to train those agents and networks.

Evaluation

❖ Experiment Setup

- Dataset

  (1) 11,000 instances mapping network intent to DSL to API calls.

  (2) 1,024 sample instances from Topology Zoo which contains 258 real-world network topologies.

- Environment for DRL :

  (1) Python-based network simulation environment.

- Experiment Parameter

  (1) Intent translation : ChatGLM-6B

  (2) Configuration update : an encoder with 8 layers of GCN and 8 layers of transformer encoder, a decoder with 4 linear layers.

- Metrics and Comparisons

  (1) Intent translation accuracy, policy consistency, configuration update time, link utilization, and traffic shift.

  (2) NetComplete (SMT), Graph attention network (GAT) and NetRen.

Parameter	Value
Episodes (EP)	10
Timestep (TM)	200
Replay Buffer Size	100
Batch Size	16
Discount Factor (γ)	0.85
Initial Exploration Rate	1.0 (decaying)
Final Exploration Rate	0.01
Exploration Decay Rate	0.99 per timestep
Soft Target Update Interval	16 timestep
K in reward function	10
Disobey reward in reward function	-1

Table 4: Parameters in COMA algorithm.

Parameter	Value	Parameter	Value
OSPF weight (α)	[1, 64]	Local preference (γ_1)	[1, 64]
Bandwidth (β_1)	(64, 128]	AS path length (γ_2)	[1, 64]
Capacity (β_2)	(64, 128]	Multi-exit discriminator (γ_3)	[1, 64]
Queue length (β_3)	(64, 128]	Packet size (q_s)	5
Lose rate (M_{Loss})	[10, 50] %		

Table 5: Parameters of network.

Evaluation

❖ Intent Translation Accuracy

- An average precision of 96%, recall of 94%, and an F1 score of 0.95 for combined IT2API (Intent to API) tasks with DSL and feedback.
- An improvement of 11% in precision, 14% in recall, and 0.12 in F1 score for NL2API (Natural language to API) over the baseline (without feedback or DSL).
- An improvement of 10% in precision, 12% in recall, and 0.11 in F1 score for AL2API (anomaly logs to API) over the baseline.
- An average feedback iteration time of 2 seconds, with most errors resolved in a single round.

Task	Feedback	DSL	Precision	Recall	F1
NL2API	✗	✗	83%	79%	0.81
NL2API	✓	✗	88%	83%	0.85
NL2API	✗	✓	92%	90%	0.91
NL2API	✓	✓	94%	93%	0.93
AL2API	✗	✗	90%	88%	0.89
AL2API	✓	✗	95%	94%	0.95
AL2API	✗	✓	97%	96%	0.96
AL2API	✓	✓	100%	100%	1.00
IT2API	✓	✓	**96%**	**94%**	**0.95**

Table 6: Intent translation under three different tasks with different conditions.

Evaluation

❖ Configuration Quality

- An average policy consistency of **99.6%** achieved by *NetKeeper*.
- An average policy consistency improvement of **1.3%**, **10.4%**, and **1.2%** over SMT, GAT, and NetRen, respectively.

Task	Topology Scale	Forward				Reachable				Isolation				Overall			
		SMT	GAT	NetRen	Ours	SMT	GAT	NetRen	Ours	SMT	GAT	NetRen	Ours	SMT	GAT	NetRen	Ours
3 × 2	S	0.98	1.00	1.00	**1.00**	0.98	0.96	1.00	**1.00**	0.99	0.58	1.00	**1.00**	0.98	0.86	1.00	**1.00**
	M	0.99	1.00	1.00	**1.00**	0.98	0.88	1.00	**1.00**	1.00	0.75	0.94	**1.00**	0.99	0.87	0.98	**1.00**
	L	0.99	1.00	1.00	**1.00**	0.99	1.00	1.00	**1.00**	0.98	0.79	0.94	**1.00**	0.98	0.89	0.98	**1.00**
3 × 8	S	0.98	0.96	1.00	**1.00**	0.98	0.96	0.98	**1.00**	0.99	0.79	1.00	**1.00**	0.98	0.89	0.99	**1.00**
	M	0.99	0.98	1.00	**1.00**	0.99	0.98	0.98	**1.00**	0.99	0.75	1.00	**1.00**	0.99	0.91	0.99	**1.00**
	L	0.98	0.98	1.00	**1.00**	0.99	0.95	1.00	**1.00**	0.98	0.75	0.96	**1.00**	0.99	0.89	0.98	**1.00**
3 × 16	S	0.98	0.94	0.99	**1.00**	0.99	0.98	0.95	**1.00**	0.98	0.78	0.96	**1.00**	0.98	0.91	0.98	**1.00**
	M	0.99	0.98	0.99	**1.00**	0.99	0.95	0.95	**1.00**	0.99	0.73	0.97	**1.00**	0.99	0.91	0.98	**1.00**
	L	0.99	0.97	0.99	**1.00**	0.99	0.96	0.99	**1.00**	0.99	0.79	0.94	**1.00**	0.99	0.92	0.98	**0.99**

Table 7: Comparison of average policy consistency under tasks of varying complexity.

Evaluation

❖ Configuration Quality

- Continuous improvement in policy consistency as timesteps increase: **96.2%** (5 steps), **98.7%** (20 steps), and **99.6%** (100 steps).
- Average policy consistency of **99.6%** within **4.07** seconds and a configuration synthesis timeout rate as low as **0.04%**.
- Ability to exceed NetRen's policy satisfaction rate (**98%**) in no more than 8 iterations (approx. 3.2 seconds).

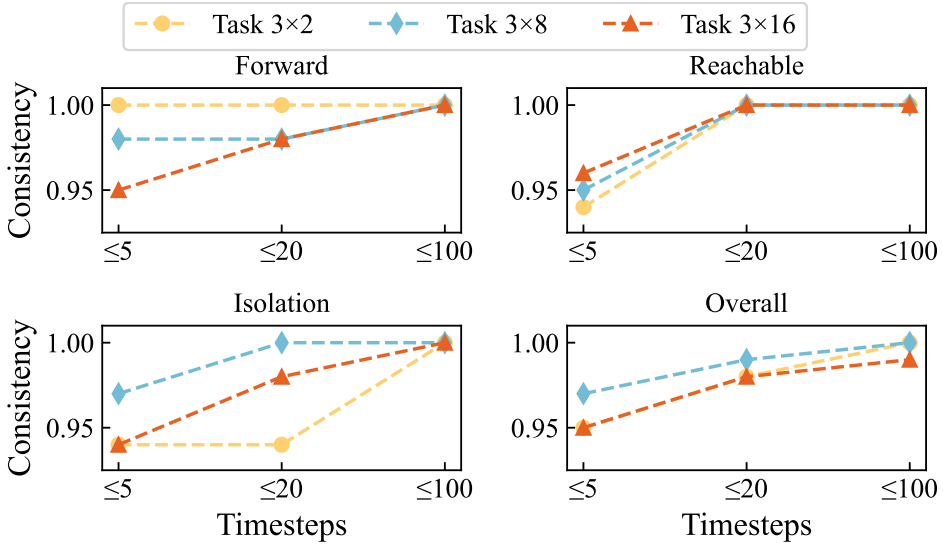


Figure 7: Average policy consistency within limited timesteps.

Task	Topology Scale	SMT(s)	GAT(s)	NetRen(s)	Ours(s)
3 × 2	S	7.68	2.62	0.37	**0.31**
	M	98.38	4.37	0.41	**0.34**
	L	1329.32	13.41	1.26	**0.51**
3 × 8	S	58.09	4.26	0.64	**0.42**
	M	148.41	5.89	0.81	**0.47**
	L	1500+	17.35	**2.73**	3.01
3 × 16	S	283.16	5.25	1.13	**1.04**
	M	317.24	6.39	**1.78**	2.08
	L	1500+	20.71	**3.16**	4.07

Table 8: comparison of average time consuming under tasks of varying complexity (1500+ means timeout).

Evaluation

❖ Load Measurement

- An average network congestion mitigation improvement of **6.3%**, **3.9%**, and **5.8%** in ideal, normal, and high-load scenarios respectively, compared to NetRen, with a maximum congestion mitigation of **18.0%**, **15.0%**, and **26.0%**.
- An overall network performance improvement of **5.3%** over baseline methods.

Topology Scale	Traffic Range (bytes)	Ideal				Normal				High-load			
		Random	OSPF	NetRen	Ours	Random	OSPF	NetRen	Ours	Random	OSPF	NetRen	Ours
S	1~2	0.89	0.49	0.02	**0.01**	0.64	0.76	0.03	**0.02**	>1.1	>1.1	0.06	0.06
	2~4	0.70	0.96	0.05	0.05	1.04	0.99	0.06	**0.04**	>1.1	>1.1	0.11	**0.10**
	4~8	>1.1	>1.1	0.09	**0.08**	>1.1	>1.1	0.10	0.10	>1.1	>1.1	0.26	**0.20**
	8~16	>1.1	>1.1	0.23	**0.22**	>1.1	>1.1	0.25	**0.22**	>1.1	>1.1	0.48	0.48
M	1~2	0.86	>1.1	0.06	**0.04**	0.60	0.84	0.04	0.04	>1.1	>1.1	0.09	0.09
	2~4	>1.1	>1.1	0.18	**0.11**	>1.1	>1.1	0.10	**0.08**	>1.1	>1.1	0.16	**0.15**
	4~8	>1.1	>1.1	0.43	**0.25**	>1.1	>1.1	0.29	**0.21**	>1.1	>1.1	0.37	0.40
	8~16	>1.1	>1.1	0.40	**0.38**	>1.1	>1.1	0.40	0.40	>1.1	>1.1	0.73	**0.60**
L	1~2	>1.1	>1.1	0.10	**0.09**	>1.1	>1.1	0.05	0.05	>1.1	>1.1	0.13	**0.07**
	2~4	>1.1	>1.1	0.18	**0.15**	>1.1	>1.1	0.12	0.12	>1.1	>1.1	0.25	**0.18**
	4~8	>1.1	>1.1	0.43	**0.32**	>1.1	>1.1	0.29	**0.23**	>1.1	>1.1	0.49	**0.43**
	8~16	>1.1	>1.1	0.78	**0.66**	>1.1	>1.1	0.68	**0.53**	>1.1	>1.1	1.06	**0.80**

Table 9: Comparison of maximum link utility under three scenarios and four traffic patterns

Evaluation

❖ Traffic Shift

- An average traffic shift of **5.9%** with *NetKeeper*, compared to **13.9%** with NetRen.
- An average reduction in traffic shift of **8.7%**, with a maximum reduction of **34.7%** over NetRen.
- Achieves a high policy consistency of **99.9%**, versus **85.8%** with NetRen, during network configuration updates.

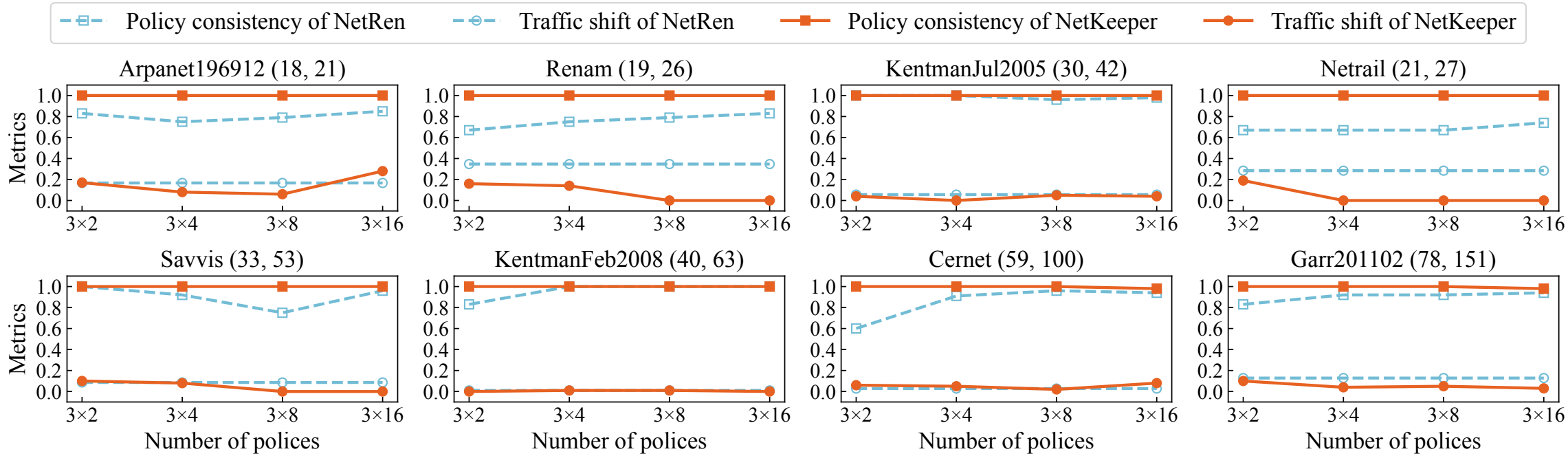


Figure 8: Traffic shift during configuration updates and policy consistency with policy count increases. Each subplot is titled with the topology name, its corresponding number of node and edge.

Evaluation

❖ Adaptability To Dynamic Network

- NetKeeper delivers full or near-full (**≥98%**) policy satisfaction during varied dynamic network events, adapts rapidly (within **30 timesteps**) to real-world topology changes, and sustains network optimization and stability even under policy conflicts or component failures.

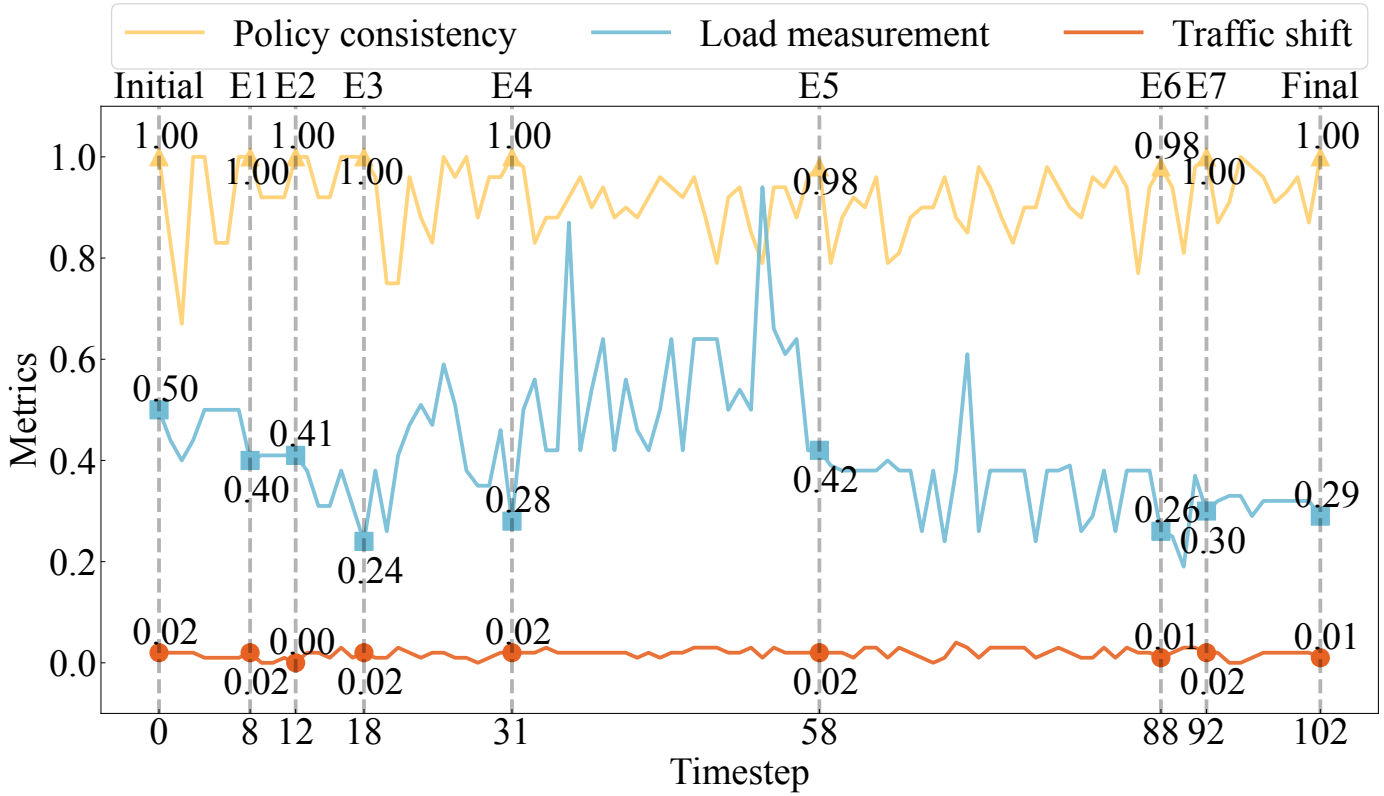


Figure 9: *NetKeeper*'s performance in dynamic network: Policy consistency, Load measurement and Traffic shift.

Evaluation

❖ Deployment in Production Network

- Achieves 94.8% correct reconfiguration rate in a large-scale enterprise network deployment, with 2.1% false negatives and 3.1% false alarms.
- Improves network resilience by 39.4% (reduction in congestion events) during real-world congestion scenarios.
- Reduces manual interventions by 51.5% through natural language interface and automation.

```
2024-10-04 17:05:38 step:9 policy_consistency: 0.77 max_utility: 0.11 traffic_shift: 0.11
2024-10-04 17:05:39 step:10 policy_consistency: 0.91 max_utility: 0.11 traffic_shift: 0.11
2024-10-04 17:05:41 step:11 policy_consistency: 0.79 max_utility: 0.14 traffic_shift: 0.00
2024-10-04 17:05:42 step:12 policy_consistency: 0.89 max_utility: 0.14 traffic_shift: 0.00
2024-10-04 17:05:44 step:13 policy_consistency: 0.91 max_utility: 0.14 traffic_shift: 0.11
2024-10-04 17:05:45 step:14 policy_consistency: 0.72 max_utility: 0.14 traffic_shift: 0.11
2024-10-04 17:05:47 step:15 policy_consistency: 0.96 max_utility: 0.14 traffic_shift: 0.09
2024-10-04 17:05:48 step:16 policy_consistency: 1.00 max_utility: 0.14 traffic_shift: 0.02
```

Figure 10: Network optimization logs in production network.

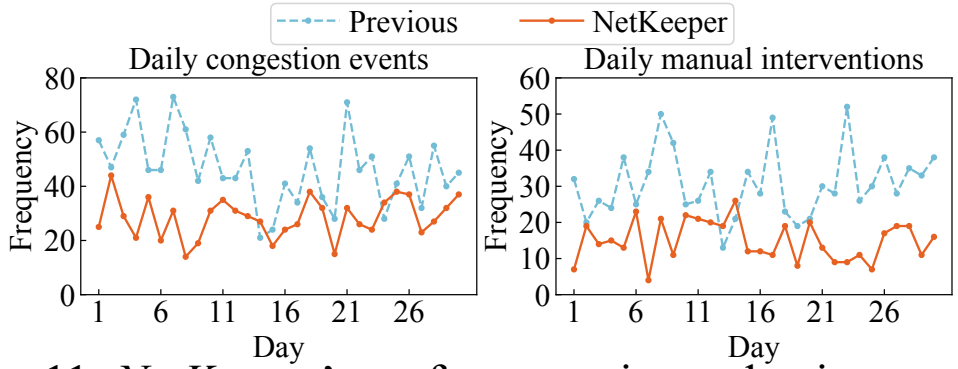


Figure 11: *NetKeeper*'s performance in production network.

Conclusion

❖ Highlights and Future Work

- A bidirectional intent translation interface is introduced for automated policy management and anomaly response.
- A traffic-aware, goal-driven multi-agent update model is designed for autonomous network resilience.
- NetKeeper is proposed for self-managed, consistent, and efficient autonomous network configuration under anomalies.
- Enabling autonomous policy generation and developing fine-grained, application-aware traffic profiling are key future directions.

Thank You !

Any questions?

Contact me: Zhaoyang Wan

Email: wanzhaoyang1@bupt.edu.cn