

Parade: Unified and Data-Driven Prefix-AS Semantics for BGP Anomaly Detection

Yunze Wei
Tsinghua University
Beijing, China

Xingang Shi[✉]
Tsinghua University
Beijing, China
Zhongguancun Laboratory
Beijing, China
shixg@cernet.edu.cn

Minghao Zhang
Tsinghua University
Beijing, China

Zhaozhen Wang
Tsinghua University
Beijing, China

Qiang Ma
Zhongguancun Laboratory
Beijing, China

Han Zhang
Tsinghua University
Beijing, China
Zhongguancun Laboratory
Beijing, China

Yahui Li
Tsinghua University
Beijing, China

Xia Yin
Tsinghua University
Beijing, China
Zhongguancun Laboratory
Beijing, China

Abstract

The global routing system remains vulnerable to various routing anomalies. State-of-the-art detection systems are constrained by their disjoint modeling of prefix-AS interactions, heavy reliance on unreliable non-native priors within their core modeling, and lossy abstraction of propagation topologies.

We propose Parade, a novel framework for prefix-AS joint hyperbolic representation learning. Parade aggregates prefixes into propagation profiles to ensure scalability and semantic consistency and embeds them together with ASes into a unified hyperbolic space using a BGP-native, data-driven encoder. By employing an asymmetric modeling strategy that maintains ASes as stable semantic anchors while representing profiles as multi-vector sets, it utilizes a stochastic winner-take-all masking to disentangle semantic polysemy in complex routing scenarios. This enables the encoder to capture both prefix-AS affinities and profile co-propagation patterns without relying on external priors such as business relationships or statistical features. Leveraging this encoder, we implement Parade as a high-performance system capable of monitoring global routing changes with high fidelity and detecting anomalies by measuring semantic deviations. Evaluation on real-world incidents and simulations demonstrates that Parade achieves superior precision, sensitivity, and performance in identifying not only large-scale events but also stealthy anomalies. For simulated path manipulations involving non-core networks, Parade maintains recall above 98.3%. The stratified evaluation further identifies core ASes as challenging because of their ubiquitous transit roles.

CCS Concepts

• Security and privacy → Network security; Intrusion/anomaly detection and malware mitigation; • Computing methodologies → Learning latent representations.

Keywords

Border Gateway Protocol (BGP); Routing Anomaly Detection; Hyperbolic Representation Learning

ACM Reference Format:

Yunze Wei, Xingang Shi, Minghao Zhang, Zhaozhen Wang, Qiang Ma, Han Zhang, Yahui Li, and Xia Yin. 2026. Parade: Unified and Data-Driven Prefix-AS Semantics for BGP Anomaly Detection. In *Proceedings of the 2026 ACM SIGSAC Conference on Computer and Communications Security (CCS '26)*, November 15–19, 2026, The Hague, Netherlands. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3830454.3846639>

1 Introduction

The Border Gateway Protocol (BGP) serves as the *de facto* standard for inter-domain routing, coordinating the exchange of prefix reachability among Autonomous Systems (ASes). However, BGP lacks intrinsic mechanisms to authenticate routing information, leaving the global routing vulnerable to various anomalies such as route hijacks, route leaks, and path manipulations. Although several security mechanisms have been proposed, including Route Origin Validation (ROV) [8], BGPsec [27], and ASPA [4], they suffer from *limited adoption* due to operational overhead and compatibility concerns, and their universal adoption remains a distant prospect [10, 11, 15, 28]. Consequently, recent route incidents [7, 14, 16] continue to demonstrate the persistent exposure of the routing system to both malicious attacks and inadvertent misconfigurations.

In the absence of universally deployed cryptographic protections, routing legitimacy is fundamentally determined by whether observed routing behaviors are consistent with the underlying *routing semantics* [12]. Such semantics are not explicitly encoded in BGP messages. Instead, they emerge implicitly from the propagation patterns of prefixes across ASes [38], shaped by routing policies, business relationships, and operational practices. Therefore, detecting routing anomalies requires understanding how prefixes are expected to propagate and identifying deviations from these established semantic patterns. In practice, such deviations manifest as inconsistencies in the semantic relationships between prefixes and ASes, as reflected in their propagation behaviors.



This work is licensed under a Creative Commons Attribution 4.0 International License. *CCS '26, The Hague, Netherlands*

© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2871-6/2026/11
<https://doi.org/10.1145/3830454.3846639>

However, existing detection systems remain limited in capturing such semantics in a unified and reliable manner [12, 13, 22, 23, 33, 36–38, 42]. They directly embed unreliable external priors (e.g., expert-defined features, business relationships, or geolocation) into their core representation space or pattern-learning processes. Moreover, they often adopt disjoint analytical views that treat prefixes and ASes separately, leaving prefix-AS interactions insufficiently modeled. These limitations prevent them from accurately representing context-dependent semantic relationships and reduce their effectiveness in detecting anomalies, especially in complex scenarios like Multi-Origin AS (MOAS) and Traffic Engineering (TE).

To fill this gap, we propose Parade, a novel framework enabling Prefix-AS joint Representation learning for BGP Anomaly DEtection. Parade addresses the semantic complexity of global routing with an *encoder* through two core innovations: First, it introduces a profile-based asymmetric modeling strategy that captures prefix-AS interactions by aggregating prefixes with identical propagation behaviors into *propagation profiles*, which serve as atomic semantic units. ASes are modeled as stable semantic anchors, while profiles are represented as *multi-vector sets* to capture semantic polysemy. Second, Parade applies a data-driven hyperbolic representation learning approach to jointly embed prefixes and ASes within a unified semantic space. The encoder leverages the exponential volume expansion of the hyperbolic space to accurately preserve the hierarchical structure of the routing system [12, 26, 31]. It then distills routing semantics into structured relationships constrained by *hierarchical affinity* and *co-propagation affinity* from native BGP data, which aligns ASes with profiles based on their relationship strengths and guides the semantic closeness of profiles.

We then implement Parade as a high-performance BGP anomaly detection system. The system features a global propagation monitor with an *edge-centric propagation graph* design and an efficient *shared-memory IPC* data processing pipeline. This architecture enables the high-fidelity maintenance of the propagation graph for all prefixes, providing real-time inputs to the encoder for inference. By leveraging the learned routing semantics and relationships, Parade quantifies the consistency of semantic relationships between prefixes and ASes in route updates, pinpointing significant deviations in semantic relationships. Finally, we incorporate a lightweight, context-aware post-hoc filtering module that suppresses candidates with strong consistency evidence and annotates operationally explainable deviations (e.g., IP leasing or DDoS mitigation).

Our analysis demonstrates that Parade effectively captures the hierarchical structure of ASes and the semantic relationships between prefixes and ASes in a data-driven manner. In historical-event evaluation, Parade covers more anomaly categories among the evaluated systems. Beyond pinpointing attackers and victims across historical datasets, our system maintained recall above 98.3% across simulated path manipulations involving regional and edge ASes. The decline is concentrated in the most central tier, while the adjacent tier remains close to 90%, reflecting the structural ambiguity created by ubiquitous core transit. Furthermore, a representative case study underscores our data-driven advantage: while multiple platforms raised alarms regarding Venezuela’s state-owned ISP preceding Operation Absolute Resolve [6, 21], Parade correctly identified them as false positives stemming from erroneous relationship inferences.

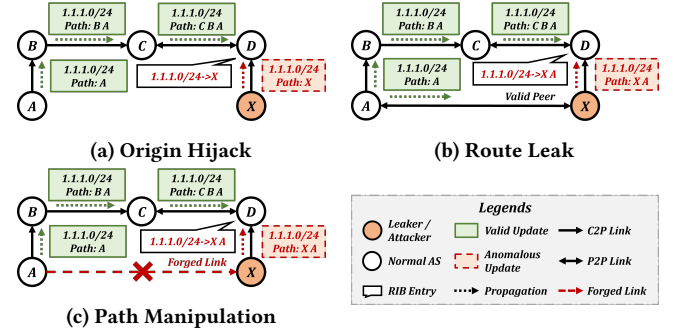


Figure 1: Illustration of BGP Anomalies.

This conclusion was subsequently corroborated by an independent analysis conducted by a security firm.

In summary, we make the following contributions:

- We propose a hyperbolic encoder for joint prefix-AS representation learning. By constructing a unified embedding space, it quantifies the intrinsic semantic consistency between prefixes and ASes, effectively disentangling complex routing behaviors without relying on external priors.
- We implement Parade, a high-performance BGP anomaly detection system. Powered by an edge-centric propagation monitor and data-driven inference, Parade achieves fine-grained detection with high-fidelity topology maintenance.
- We conduct extensive evaluations on historical incidents, simulations, and deployments, including collector-coverage and end-to-end sensitivity analyses. We have open-sourced our implementation¹ to facilitate future research.

2 Background and Motivation

2.1 BGP Anomalies

The BGP routing system is vulnerable to various routing anomalies. Despite the standardization of defenses [4, 5, 8, 27], their deployment remains partial [11, 15, 28] and these vulnerabilities persist. As illustrated in Figure 1, we consider three types of anomalies.

Route Origin Hijack. A route origin hijack occurs when an AS illegitimately originates a prefix that it is not authorized to announce. It falsifies the source of reachability information and may appear as either an exact-prefix or sub-prefix hijack.

Route Leak. A route leak preserves the legitimate origin but propagates routing announcements beyond their intended policy scope. Such events typically violate inter-domain export policies and can redirect traffic through unintended ASes.

Path Manipulation. Path manipulation alters the AS_PATH attribute to construct seemingly plausible but unauthorized paths [19, 23]. Unlike origin hijacks, these attacks may preserve the correct origin and therefore evade defenses that validate only route origins.

Since route leaks and path manipulation both preserve the announced origin while introducing abnormal propagation paths, we refer to them collectively as *path-related anomalies* in our dataset construction and evaluation.

¹<https://github.com/weiyz23/CCS26-Parade>.

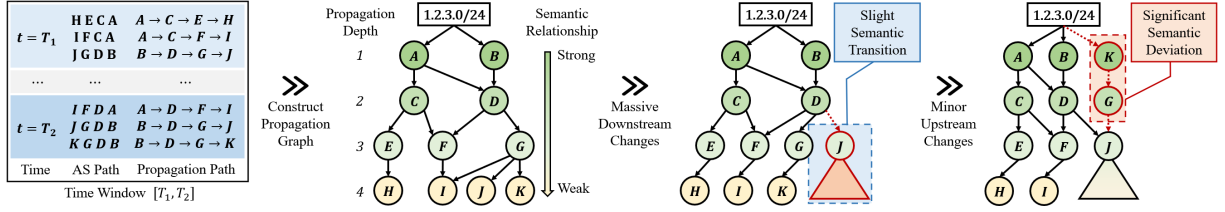


Figure 2: An example of data-driven routing semantics. Observed routing paths are aggregated into a propagation graph, where semantic relationships between prefixes and ASes are depicted. As the propagation depth increases, this relationship tends to weaken. Thus, massive downstream changes may not cause critical semantic alterations, while minor upstream changes can lead to significant semantic deviations.

2.2 Routing Semantics

Inter-domain routing exhibits implicit semantics that govern how prefixes are propagated across ASes. These semantics emerge from prefix-AS interactions shaped by routing policies, business relationships, and operational practices but are neither explicitly encoded in BGP messages nor robustly enforced by existing mechanisms. We ground such semantics in prefix-AS propagation behaviors, expressed as semantic relationships under diverse routing contexts.

Role-Induced and Stable AS Semantics. The semantics of an AS are shaped by its roles in propagating prefixes, rather than derived from explicitly modeled attributes. These roles often correlate with network scale, business type, and hierarchical position [12]. Such roles tend to be *relatively stable* within a tactical time window (e.g., daily), except for transiently registered ASes.

Contextual and Polysemous Prefix Semantics. The semantics of a routing prefix are defined by its propagation patterns. The routing paths encode how the prefix is disseminated and implicitly capture its underlying semantics. Thus, prefixes with similar semantics tend to exhibit similar propagation patterns, leading to co-propagation behavior. Complex routing practices such as Anycast and MOAS may expose a prefix to multiple propagation patterns to different routing contexts, resulting in semantic polysemy [34].

As illustrated in Figure 2, *propagation graphs* encapsulate semantic relationships between prefixes and ASes by capturing AS roles and prefix dissemination patterns. Upstream ASes (e.g., transit ASes) generally have stronger influence on a prefix’s propagation, leading to stronger prefix-AS semantic relationships. Therefore, route updates can be assessed by measuring how they alter propagation patterns and prefix-AS semantic relationships. This propagation-centric view enables data-driven reasoning over native BGP data without relying on out-of-band information.

2.3 Motivation: Semantic Deviation

The fundamental challenge in BGP anomaly detection is not merely identifying critical routing changes but determining whether such changes are *semantically consistent* with the underlying ecosystem. While many semantic transitions arise from benign operational practices, others introduce deviations that may indicate anomalies.

Routing Anomalies as Semantic Deviations. Routing anomalies often manifest as deviations in prefix-AS semantic relationships. The severity of such deviations depends on semantic inconsistency rather than the scale of topological changes [23]. As

shown in Figure 2, small changes in upstream nodes can cause substantial shifts in semantic relationships, while large-scale changes downstream may have minimal impact on semantic consistency. This motivates detection systems that quantify semantic deviations in addition to monitoring topological changes. Existing detection paradigms lack a unified, comprehensive understanding of routing semantics, leading to two fundamental limitations.

(1) Dependence on External Priors in Core Modeling. Current detection systems often explicitly inject external priors into their core learning or feature extraction processes to approximate routing semantics [12, 22, 23, 33, 38, 42]. Such priors, including inferred business relationships [9] or heuristic assumptions such as geolocation and peering status [32], may be incomplete, stale, error-prone [17, 20], or even adversarially polluted [14, 19], undermining semantic modeling and anomaly detection.

(2) Lack of Unified Semantic Modeling. The legitimacy of a routing update is inherently a *relational and joint property* of the prefix, the AS, and their *context-dependent interaction*. This is especially critical in complex TE scenarios, where an AS may apply different policies to different prefixes, even under the same origin. Existing approaches often adopt a disjoint view: they either focus on AS-level features [12, 13, 22, 23, 33, 42] or prefix-level statistics [38], or analyze them separately [29]. Such designs collapse diverse prefix-specific behaviors into aggregated representations, making it difficult to distinguish legitimate, context-specific policy variations from genuine semantic deviations.

Design Goals. Based on the above analysis, an effective data-driven anomaly detection system based on routing semantics should satisfy four key requirements:

G1: Unified Semantic Representation. The system should jointly model prefixes and ASes to capture context-dependent interaction patterns across diverse routing policies.

G2: Native Data-Driven Semantics. It should learn routing semantics directly from native BGP data, without injecting external priors into the core semantic model.

G3: High-Fidelity Propagation Monitoring. It must preserve fine-grained propagation patterns to detect semantic deviations caused by subtle routing changes.

G4: Practical False Positive Mitigation. It needs to incorporate mechanisms to reduce false positives from legitimate operational events such as IP leasing or DDoS mitigation.

To fulfill these design goals, Parade adopts a two-part architecture: a joint representation learning encoder (addressing G1 & G2)

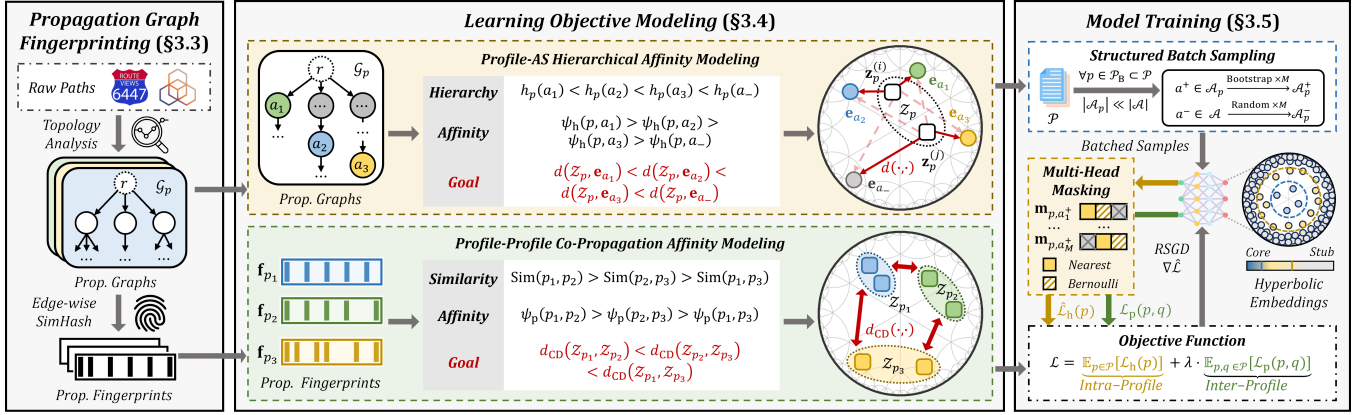


Figure 3: Overview of the hyperbolic encoder. It transforms BGP data into propagation graphs and fingerprints to optimize hierarchical and co-propagation affinities. The model is trained via structured sampling and masking for robust embeddings.

that constructs stable semantics offline and a high-performance system pipeline (addressing G3 & G4) that computes semantic deviations online and applies post-hoc filtering to suppress or contextualize detected candidates for operator investigation.

3 Joint Representation Learning Encoder

3.1 Overview

To operationalize semantic relationships and their deviations, we construct an encoder that maps prefix-AS semantics into a latent geometric space, where distances encode semantic affinity. The encoder captures joint, context-dependent, and potentially polysemous semantics (G1) from observed BGP propagation behavior without explicitly injecting external priors into the core representation (G2), and uses hyperbolic space to model hierarchical and scale-free structures of the global routing system.

To reduce redundancy and capture semantic regularities, Parade first performs propagation graph fingerprinting (§3.3), aggregating prefixes with identical behaviors into *profiles* as atomic units for modeling prefix semantics. Parade extracts two key features within each graph: (1) *propagation depth*, which captures the hierarchical position of an AS in propagating the profile; and (2) a *propagation fingerprint*, a compact vector that encodes the propagation graph for efficient comparisons across profiles.

The encoder applies two semantic affinity constraints under an asymmetric modeling strategy (§3.4). ASes are represented as single vectors serving as stable semantic anchors, while profiles are modeled as multi-vector embeddings inherited by their associated prefixes. This design captures polysemous prefix-AS semantics across different propagation contexts. *Hierarchical affinity* aligns the semantic distance between a profile and an AS with their vertical separation in the routing propagation hierarchy. *Co-propagation affinity* pulls profiles with similar propagation fingerprints together while pushing dissimilar ones apart, capturing horizontal behavioral similarities among propagation profiles.

Finally, the encoder is trained by integrating structured batch sampling and a stochastic winner-take-all masking mechanism

(§3.5), improving training efficiency while learning robust embeddings for profiles and ASes.

3.2 Formulation and Preliminaries

Hyperbolic Semantic Space. We model the latent semantic space as an n -dimensional hyperbolic manifold, where geometric distances encode the strength of semantic relationships between routing entities (profiles and ASes). Specifically, shorter distances indicate stronger semantic affinity, while larger distances reflect weaker or anomalous relationships. This interpretation enables the quantification of semantic deviation through geometric measurements in the latent space.

The encoder operates in an n -dimensional Poincaré ball $\mathbb{D}^n = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| < 1\}$, equipped with the distance metric:

$$d_{\mathbb{D}}(\mathbf{u}, \mathbf{v}) = \cosh^{-1} \left(1 + 2 \frac{\|\mathbf{u} - \mathbf{v}\|^2}{(1 - \|\mathbf{u}\|^2)(1 - \|\mathbf{v}\|^2)} \right). \quad (1)$$

It is well-suited for modeling routing semantics due to its ability to represent hierarchical and scale-free structures. Its exponential volume growth allows entities in the routing hierarchy to be embedded with minimal distortion [26, 31].

Propagation Fingerprint. A fingerprint $f_p \in \{0, 1\}^L$ abstracts the routing patterns of a profile p . Let $\mathcal{R}_p$ be the set of propagation paths for prefixes within p , where each path $\pi = \langle a_1, \dots, a_{|\pi|} \rangle$ is a reversed AS_PATH starting from the origin a_1 . We define the connection set C_p containing all directed links in $\mathcal{R}_p$:

$$C_p = \{(u, v) \mid \exists \pi \in \mathcal{R}_p, \exists k, \text{ s.t. } \pi[k] = u \wedge \pi[k+1] = v\}.$$

f_p aggregates the links in C_p , serving as a compact abstraction of the propagation graph. It provides the basis for measuring co-propagation affinity between profiles.

Affinity Relations. We seek to preserve two types of semantic relations in the latent space, where these relations serve as proxies for semantic consistency between routing entities:

- **Hierarchical Affinity** $\psi_h(p, a)$ denotes the vertical semantic proximity between a profile p and an AS a . High affinity implies that a plays a more influential role in the propagation of p , typically being closer to the origin or acting as a direct provider.

- **Co-Propagation Affinity** $\psi_p(p, q)$ denotes the horizontal semantic similarity between profiles. Profiles sharing substantial path segments exhibit higher affinity, indicating similar routing semantics under comparable propagation contexts.

AS Embedding. Let $\mathcal{A}$ be the set of ASes. We define a mapping $g_{\mathcal{A}} : \mathcal{A} \rightarrow \mathbb{D}^n$ that assigns each AS $a \in \mathcal{A}$ a vector $\mathbf{e}_a \in \mathbb{D}^n$. This embedding encodes the AS's semantic role in the routing system, reflecting its hierarchical position and its influence on prefix propagation across the observed routing topology.

Multi-Vector Profile Embedding. Given the profile set $\mathcal{P}$, we define a one-to-many mapping $g_{\mathcal{P}} : \mathcal{P} \rightarrow 2^{\mathbb{D}^n}$. Each profile p is represented by a set $\mathcal{Z}_p = \{\mathbf{z}_p^{(1)}, \dots, \mathbf{z}_p^{(K)}\} \subseteq \mathbb{D}^n$ of K latent vectors. This multi-vector approach allows the model to capture the polysemous nature of routing semantics arising from complex scenarios (e.g., MOAS or Anycast). Each prefix inherits the embedding of its associated profile, ensuring consistency between prefix-level observations and profile-level representations.

3.3 Propagation Graph Fingerprinting

3.3.1 Propagation Graph Construction. We construct propagation graphs from a combination of Routing Information Base (RIB) snapshots and UPDATE (UPD) messages within a bounded time window. RIB snapshots provide stable and consolidated routing states, while UPD messages capture transient routing dynamics.

Profile Aggregation. To ensure scalability while preserving routing semantics, we aggregate prefixes with identical AS path sets into propagation profiles, which serve as atomic semantic units for subsequent modeling. Each profile represents a group of prefixes that are indistinguishable in terms of their observed propagation behavior within one time window. For brevity, a profile's propagation and origin ASes denote those of its underlying prefixes.

Topology Generation. For each profile p , we construct an initial propagation graph from the connection set C_p . Let $\mathcal{A}_p = \{a \mid \exists \pi \in \mathcal{R}_p, a \in \pi\}$ denote the set of *semantically related ASes* involved in p 's propagation; conversely, any $a \notin \mathcal{A}_p$ is considered to have no observed semantic relationship with p . Due to misconfigurations or merging data from different time points, loops may exist in the initial graph. We apply Tarjan's algorithm [39] to contract strongly connected components (SCCs) into super-nodes and ensure a directed acyclic graph (DAG). Each node S_u in the DAG may contain one or more ASes, and the directed edges represent the observed propagation flow between ASes within different nodes. Finally, a virtual root S_p^* is introduced to represent the profile and unify the graph structure; we add a directed edge (S_p^*, S_v) for every node S_v that contains the origin AS of p .

Depth Calculation. We assign a propagation depth $h_p(S_u)$ to each node S_u to quantify its hierarchical status, which reflects the relative semantic influence of ASes in propagating profile p . By performing a topological sort on the DAG [24], we compute $h_p(S_u)$ as the length of the longest path from S_p^* to S_u . For simplicity, we use $h_p(a)$ to denote the propagation depth of an AS a within S_u , and all ASes within a node share the same depth. This longest-path formulation effectively mitigates artifacts from AS prepending, thereby restoring the true hierarchical structure.

3.3.2 Propagation Fingerprinting. Standard graph matching metrics (e.g., graph edit distance) are computationally prohibitive for

large-scale routing data. To efficiently approximate co-propagation similarity at scale, we propose a *depth-weighted SimHash-based* fingerprinting scheme that compresses each propagation graph into a fixed-length representation $\mathbf{f}_p \in \{0, 1\}^L$ (where $L = 128$), as detailed in our additional technical report [1].

We treat directed edges as local features weighted by their topological depth. We employ a composite hash function $H : \mathcal{E} \rightarrow \{0, 1\}^L$ to map each edge $(u, v) \in C_p$ to the feature space. An accumulator $\mathbf{x} \in \mathbb{R}^L$ aggregates these features, weighted by the source node's depth $h_p(S_u)$ and a decay factor α :

$$\mathbf{x}[k] \leftarrow \mathbf{x}[k] + (-1)^{1-H(u,v)[k]} \cdot e^{-\alpha h_p(S_u)}. \quad (2)$$

The fingerprint is obtained via $\mathbf{f}_p[k] = \mathbb{I}(\mathbf{x}[k] > 0)$. Such a design ensures that edges closer to the origin (with smaller depth) contribute more significantly, preserving the core propagation patterns.

We define the *similarity* between the fingerprints of profiles p, q using a truncated Hamming-based similarity:

$$s(p, q) = \max \left(0, 1 - \frac{2}{L} \sum_{k=0}^{L-1} (\mathbf{f}_p[k] \oplus \mathbf{f}_q[k]) \right). \quad (3)$$

3.4 Learning Objectives and Affinity Modeling

We model semantic relationships as affinity constraints, which are translated into geometric distances and optimized via a composite loss function, enforcing *vertical hierarchical alignment* and *horizontal semantic clustering* across propagation profiles.

We extend the hyperbolic distance to the multi-vector setting. For convenience, we overload $d(\cdot, \cdot)$ to denote hyperbolic distances involving sets: for a vector $\mathbf{x}$ and a vector set $\mathcal{Z}$, we define

$$d(\mathbf{x}, \mathcal{Z}) = \min_{\mathbf{z} \in \mathcal{Z}} d_{\mathbb{D}}(\mathbf{x}, \mathbf{z}). \quad (4)$$

Distances between two vector sets are defined via the symmetric Chamfer distance

$$d_{\text{CD}}(\mathcal{Z}_1, \mathcal{Z}_2) = \frac{1}{2K} \left(\sum_{\mathbf{z} \in \mathcal{Z}_1} d(\mathbf{z}, \mathcal{Z}_2) + \sum_{\mathbf{z} \in \mathcal{Z}_2} d(\mathbf{z}, \mathcal{Z}_1) \right). \quad (5)$$

3.4.1 Hierarchical Affinity Loss. The hierarchical objective $\mathcal{L}_h(p)$ models the intra-profile topology. It aligns the geometric arrangement of ASes with propagation depth $h_p(\cdot)$, such that embedding distances reflect the relative semantic influence of ASes in the propagation hierarchy of the corresponding profile.

We define a generic pairwise ranking loss function $\ell(\cdot)$ with margin ξ :

$$\ell(u, v; \mathcal{Z}_p, \xi) = \text{ReLU}(d(\mathbf{e}_u, \mathcal{Z}_p) - d(\mathbf{e}_v, \mathcal{Z}_p) + \xi). \quad (6)$$

Using this, we decompose the hierarchical loss into two terms: $\mathcal{L}_h(p) = \mathcal{L}_m(p) + \mu \cdot \mathcal{L}_r(p)$.

Reachability Distinction Margin ($\mathcal{L}_m$). The model separates ASes with semantic relationships ($\mathcal{A}_p$) from unrelated ASes ($\mathcal{A}'_p = \mathcal{A} \setminus \mathcal{A}_p$), ensuring that related ASes are embedded closer to the profile than unrelated ones. Given a strict margin ξ_m :

$$\mathcal{L}_m(p) = \sum_{a^+ \in \mathcal{A}_p} \sum_{a^- \in \mathcal{A}'_p} \ell(a^+, a^-; \mathcal{Z}_p, \xi_m). \quad (7)$$

Depth-Aware Ranking ($\mathcal{L}_r$). Within the related set $\mathcal{A}_p$, the embedding should preserve the propagation order. ASes with smaller

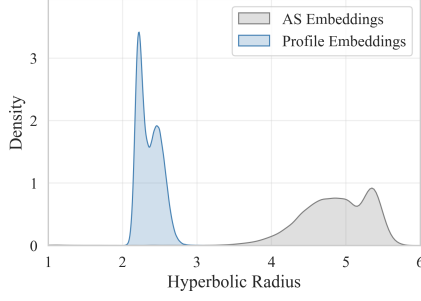


Figure 4: Hyperbolic radius distributions of AS and profile embeddings.

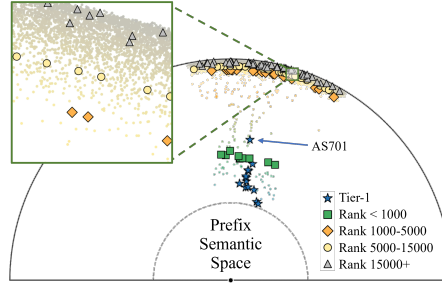


Figure 5: The 2D UMAP projection of AS embeddings in the Poincaré disk, colored by AS Tier levels.

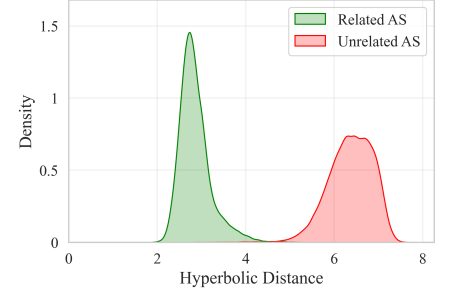


Figure 6: The PDF of hyperbolic distances between profile and AS embeddings for positive versus negative pairs.

h_p have stronger semantic influence and are expected to be embedded closer to the profile, preserving the upstream-downstream ordering in propagation. We employ a smaller margin $\xi_r < \xi_m$ to encode this relative "upstream-downstream" flow while keeping the graph structure compact:

$$\mathcal{L}_r(p) = \sum_{u,v \in \mathcal{A}_p} \mathbb{I}[h_p(u) < h_p(v)] \cdot \ell(u, v; \mathcal{Z}_p, \xi_r). \quad (8)$$

3.4.2 Co-Propagation Affinity Loss. While the hierarchical loss structures the vertical topology, the co-propagation affinity loss $\mathcal{L}_p$ captures the horizontal semantic relationships between profiles. We utilize the fingerprint similarity $s(p, q)$ as a soft weight to guide the clustering of profiles. For profiles p, q , we define a similarity-weighted contrastive loss:

$$\begin{aligned} \mathcal{L}_p(p, q) = & s(p, q) \cdot d_{CD}(\mathcal{Z}_p, \mathcal{Z}_q)^2 \\ & + (1 - s(p, q)) \cdot (\text{ReLU}(\xi_p - d_{CD}(\mathcal{Z}_p, \mathcal{Z}_q)))^2. \end{aligned} \quad (9)$$

This formulation dynamically balances attraction and repulsion: highly similar profiles are pulled closer, while dissimilar profiles are pushed apart beyond a margin. This prevents representation collapse and preserves semantic distinguishability of profiles.

3.4.3 Overall Objective. The final objective combines intra-profile hierarchical constraints and inter-profile co-propagation constraints:

$$\mathcal{L} = \mathbb{E}_{p \in \mathcal{P}} [\mathcal{L}_h(p)] + \lambda \cdot \mathbb{E}_{p, q \in \mathcal{P}} [\mathcal{L}_p(p, q)]. \quad (10)$$

3.5 Model Training

To efficiently learn semantic relationships, we adopt a structured training strategy that balances representation quality and computational efficiency while mitigating semantic mode collapse in multi-vector embeddings. This strategy integrates *structured batch sampling* to approximate the distribution of semantic relationships and a *stochastic multi-vector masking* mechanism to balance semantic specialization and diversity.

3.5.1 Structured Batch Sampling. Global optimization over the full adjacency structure is computationally prohibitive due to memory and I/O constraints. Moreover, dynamic batch generation on sparse graphs often incurs high I/O overhead. To address this, we implement a decoupled sampling strategy that exploits the high sparsity of the routing topology ($|\mathcal{A}_p| \ll |\mathcal{A}|$).

In each epoch, we construct positive (semantically related) and negative (unrelated) AS sets for each p in the profile batch $\mathcal{P}_B$:

- **Positive Sampling** ($\mathcal{A}_p^+ \subseteq \mathcal{A}_p$): We uniformly sample M semantically related ASes from $\mathcal{A}_p$ (with replacement to ensure tensor alignment). These samples provide positive signals that capture hierarchical relationships within the profile's propagation graph.
- **Negative Sampling** ($\mathcal{A}_p^- \subseteq \mathcal{A}$): We uniformly sample $3M$ ASes from the global set $\mathcal{A}$ to approximate pairs without relationships.

Sampling analysis verifies substantial coverage and an average false-negative rate below 0.5%, and full results are provided in our additional technical report [1]. Since RIB snapshots and UPDATE messages differ in stability, we associate each positive profile-AS sample with a source-dependent confidence weight w . RIB-derived samples are treated as high-confidence observations ($w = 1.0$), whereas UPDATE-derived samples are assigned a lower weight ($w = 0.2$) to reflect their transient nature.

3.5.2 Stochastic Multi-Vector Masking. We represent each profile as a multi-vector set $\{\mathbf{z}_p^{(1)}, \dots, \mathbf{z}_p^{(K)}\}$ to capture polysemous routing semantics across different propagation contexts. However, standard min-pooling optimization may lead to semantic mode collapse, where multiple semantic modes are merged into a single dominant representation instead of retaining distinct propagation contexts.

To mitigate this, we introduce a masking mechanism that balances competitive specialization and stochastic exploration to enable the disentanglement of semantic polysemy. For a profile p and a target positive AS a^+ , let k^* be the index of the "expert" head closest to a^+ . We generate a confidence-aware mask $\mathbf{m}_{p,a^+} \in \{0, 0.2, 1.0\}^K$:

$$\mathbf{m}_{p,a^+}[k] = \begin{cases} w_{p,a^+}, & \text{if } k = k^* \quad (\text{Specialization}) \\ \beta_k, & \text{otherwise} \quad (\text{Exploration}) \end{cases}, \quad (11)$$

where $\beta_k \sim \text{Bernoulli}(\epsilon)$ allows non-optimal heads to update with a small probability (e.g., $\epsilon = 0.2$). This enforces specialization of the closest sub-vector (winner-take-all), while stochastic updates encourage exploration and prevent inactive representations.

3.5.3 Optimization Objective. We adapt the hierarchical objective to the weighted masked multi-vector setting. Let $\hat{\ell}(u, v; \mathbf{z}, \xi)$ denote the vector-wise pairwise ranking loss (distinct from the set-wise loss in §3.4.1). According to Equation 7, the *masked margin loss*

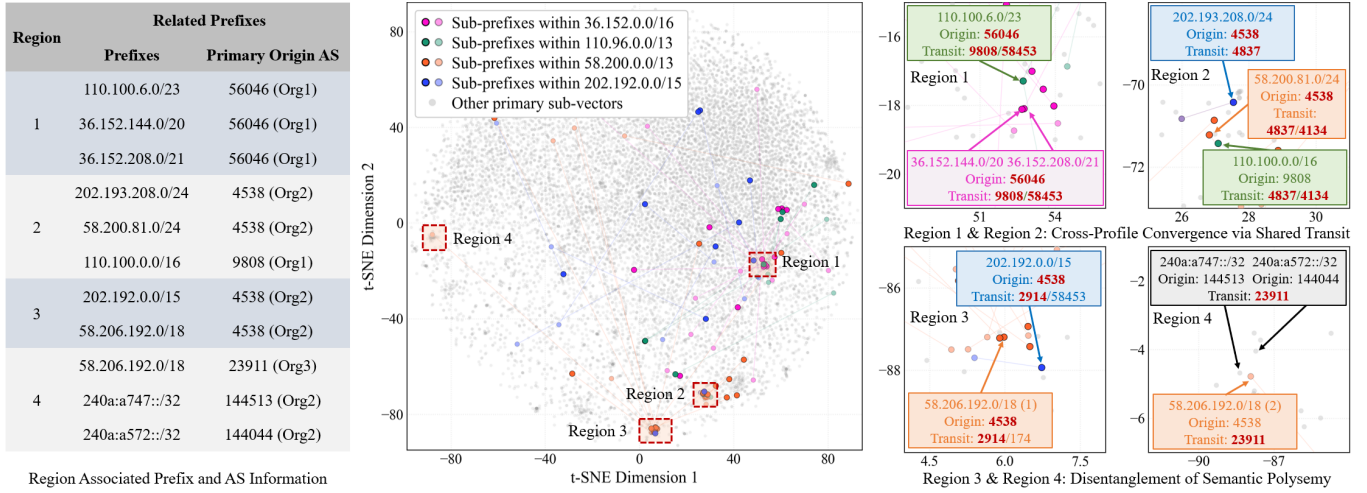


Figure 7: Visualization of learned profile embeddings projected by t-SNE. The table summarizes representative prefixes and their associated origin ASes for each region. The central plot shows the global profile embedding distribution. Regions 1 and 2 illustrate cross-profile semantic convergence driven by shared transit ASes, while Regions 3 and 4 highlight the disentanglement of semantic polysemy enabled by the multi-vector representation. AS names have been anonymized.

aggregates contributions from active heads:

$$\hat{\mathcal{L}}_m(p) = \sum_{a^+ \in \mathcal{A}_p^+} \sum_{a^- \in \mathcal{A}_p^-} \sum_{k=1}^K \mathbf{m}_{p,a^+}[k] \cdot \hat{\ell}(a^+, a^-; \mathbf{z}_p^{(k)}, \xi_p).$$

The final objective function for a batch combines the masked margin loss, the ranking loss, and the co-propagation affinity loss:

$$\hat{\mathcal{L}} = \mathbb{E}_{p \in \mathcal{P}_B} [\hat{\mathcal{L}}_m(p) + \mu \cdot \mathcal{L}_r(p)] + \lambda \cdot \mathbb{E}_{p,q \in \mathcal{P}_B} [\mathcal{L}_p(p, q)]. \quad (12)$$

We optimize this objective using Riemannian Stochastic Gradient Descent via the geopt library.

3.6 Result Analysis

3.6.1 Global Geometry of the Embedding Space. We first examine whether the learned hyperbolic space captures the structural properties of the routing system.

Radial Disentanglement. Figure 4 shows the distribution of hyperbolic radii for AS and profile embeddings. A clear separation emerges: profile embeddings concentrate near the center, while AS embeddings occupy the outer regions. This emergent layout achieves effective *semantic disentanglement*, minimizing the semantic ambiguity between AS and prefix representations.

Data-Driven Hierarchy. While AS embeddings generally occupy larger radii than profiles, relative radial differences within AS embeddings still reflect their hierarchical roles. As illustrated in Figure 5, Tier-1 ASes cluster near the center of the Poincaré disk, while stub ASes fan out toward the boundary. Notably, AS701 is positioned at a larger hyperbolic radius, reflecting its weaker connectivity and less frequent appearance in routing paths compared to other Tier-1 ASes. This demonstrates that the encoder reconstructs routing hierarchy [12] directly from native BGP data without external supervision, benefiting from the exponential volume growth of hyperbolic space, which preserves semantic distinguishability even among stub ASes.

3.6.2 Semantic Relationship Separation. We next evaluate whether the embedding preserves semantic relationships defined by the learning objectives. Figure 6 compares the hyperbolic distances between profiles and their related ASes versus unrelated ASes. Related ASes consistently lie closer to their corresponding profiles, while unrelated ASes are pushed to the outer regions. The two distributions are well separated, indicating that the encoder establishes a clear semantic boundary between related and unrelated ASes. This confirms that the model effectively captures semantic consistency in prefix-AS relationships, and suggests that semantic distance can serve as a reliable anomaly indicator.

3.6.3 Case Study: Profile Semantics in Embedding Space. To provide an interpretable view of the learned profile semantics, we visualize profile embeddings using t-SNE and analyze representative regions in Figure 7, providing a qualitative view of whether geometric proximity is consistent with propagation-dependent routing semantics.

Cross-profile convergence via shared transit. Regions 1 and 2 show that prefixes from different address blocks and origin ASes can be embedded closely when their propagation structures share key transit ASes. In Region 1, prefixes associated with different prefix groups converge due to shared transit ASes such as AS9808 and AS58453. Similarly, Region 2 shows that prefixes with different origins are placed close when their propagation paths share transit structures. These cases indicate that profile semantics are primarily shaped by propagation behavior rather than prefix address proximity or origin identity alone.

Disentanglement of semantic polysemy. Regions 3 and 4 demonstrate how the multi-vector profile representation captures multiple propagation contexts. The prefix 58.206.192.0/18 appears in two distinct regions: one sub-vector is associated with propagation through AS2914/AS174, while another is associated with AS23911. This separation is consistent with MOAS-like routing behavior, where the same prefix can be propagated through different origin

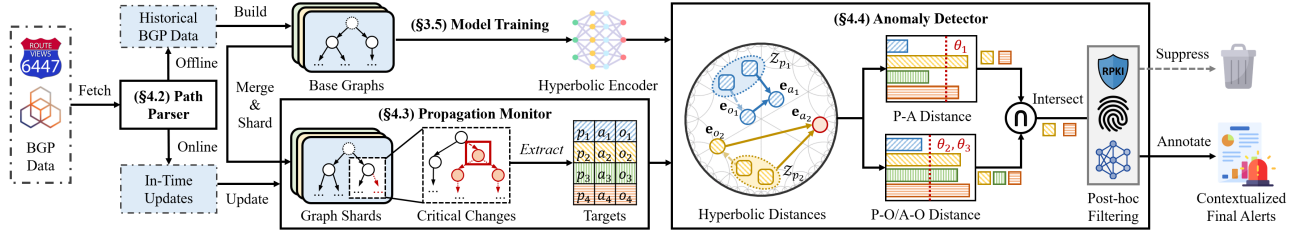


Figure 8: Overview of the Parade pipeline. The detector identifies suspicious candidates via hyperbolic distances and applies post-hoc filtering to suppress or annotate them into contextualized final alerts.

or transit contexts. This further confirms that Parade disentangles semantic modes according to propagation patterns.

Overall, these results show that the learned embedding space not only captures structural and semantic properties of routing behavior, but also provides a discriminative foundation for distance-based anomaly detection. Further analyses of the learned semantics and ablation results are provided in [1], with the latter further justifying our default use of hyperbolic embeddings with $K = 2$.

4 Anomaly Detection with Parade

4.1 System Overview

The workflow is engineered as a high-throughput, low-latency, and high-fidelity pipeline (G3). As illustrated in Figure 8, Parade comprises three modular components: the *path parser*, the *propagation monitor*, and the *anomaly detector*.

The *path parser* acts as the ingestion engine, normalizing BGP data into standardized routing entries and transmitting them downstream via a high-performance IPC mechanism. The *propagation monitor* tracks the real-time state of global routing propagation. It consumes routing entries to update the propagation graphs and identifies topology changes. By analyzing graph changes, it extracts candidate tuples comprising the affected prefix, the critical AS, and the origin AS. The *anomaly detector* projects candidate tuples into hyperbolic space to quantify semantic deviations. Subsequently, a *post-hoc filtering* module suppresses candidates with strong consistency evidence from RPKI validation or propagation-fingerprint preservation, and annotates remaining candidates using auxiliary operational signals to aid operator interpretation.

4.2 Path Parser

The path parser is designed as a high-performance ingestion engine, employing multi-level optimizations to efficiently parse and transmit massive volumes of BGP data.

Unified Stream Abstraction. We construct an abstraction layer based on libbgpstream to mask the heterogeneity of data sources (e.g., RouteViews, RIPE RIS) and underlying formats (MRT, BMP). The parser normalizes offline RIB dumps and real-time update messages into a compact, fixed-size BGPREcord structure. It automatically resolves aggregated attributes (e.g., AS Sets/Sequences), ensuring the semantic integrity of the topological information delivered to the propagation monitor for graph construction.

Shared-Memory Zero-Copy IPC. To eliminate inter-process communication bottlenecks, we implement a lock-free ring buffer

based on shared memory. By leveraging atomic operations to manage queue states, this design facilitates zero-copy interaction between the parser and the monitor, effectively removing overheads associated with mutex locks and context switching. With a 128 MB buffer, the system can cache over 524,000 entries, providing high throughput and resilience against traffic bursts.

Memory Alignment and Batch Processing. To further enhance efficiency, the system utilizes HugePages (2MB) to reduce TLB miss overhead. Furthermore, read/write operations strictly adhere to a memory-aligned batch processing pattern, employing padding to align with page size when necessary. This design maximizes memory locality and prevents performance penalties arising from cross-page access during batched routing-data processing.

4.3 Propagation Monitor

The propagation monitor serves as the middleware connecting data ingestion and anomaly detection. It is tasked with two critical responsibilities: maintaining the global propagation graphs and extracting key topological changes.

4.3.1 Fine-Grained Sharded Graph Storage. To maintain complete propagation graphs for massive prefixes within limited memory while supporting concurrent updates, we adopt a hash-based sharding architecture with independently synchronized graph partitions.

Edge-Centric Storage. To optimize memory usage while preserving high-fidelity propagation graphs, the storage is organized around edges. The core system maintains an ASEdgeInfo structure, which treats each AS adjacency as an edge, storing all prefixes carried by that edge alongside their latest timestamps. Through compact hash mapping, this design eliminates redundant storage of topological structures, enabling the efficient in-memory storage and querying of the complete global propagation graph.

Sharding Architecture. The global routing graph is partitioned into 256 independent shards and each edge is uniformly mapped to a shard. Each shard maintains independent read/write locks, allowing multiple threads to operate on different shards in parallel. This design significantly minimizes lock contention, ensuring high throughput during multi-core parallel updates.

4.3.2 Critical AS Extraction. The monitor consumes batches of BGPRECORDs from the ring buffer and performs differential analysis. New edges that induce changes in the propagation graph are flagged for further investigation. To pinpoint potentially responsible ASes from massive routing changes, we propose a critical AS extraction algorithm based on the *upstream hypothesis*.

Upstream Hypothesis. We observe that many routing anomalies induce a topological convergence pattern around the AS or AS link that introduces the abnormal path segment. Such announcements may create more attractive paths, causing downstream ASes to adopt the new routes and generating a cascade of updates that topologically converges toward the responsible AS or link. Therefore, topology changes induced by potential attacks often manifest as an upstream convergence pattern in the propagation graph.

Extraction Algorithm. For each affected prefix, we continuously track propagation changes and construct a subgraph containing newly added AS links within each time window. We first apply the algorithm described in §3.3 to calculate the propagation depth. To efficiently analyze reachability, we employ bitmap optimization, mapping the ancestor and descendant sets of each node to bitsets for rapid calculation.

For each connected component in the DAG, we perform two iterative passes (top-down and bottom-up) to calculate the upstream coverage $\text{Cov}_{\text{up}}(v)$ and downstream coverage $\text{Cov}_{\text{down}}(v)$ for each node v where $h_p(v) < 8$. These metrics quantify the proportion of ancestors and descendants reachable from v , respectively. We then compute a criticality score:

$$\text{Crit}(v) = \log_2(h_p(v) + 1) \cdot (\text{Cov}_{\text{up}}(v))^3 \cdot (\text{Cov}_{\text{down}}(v))^2.$$

The logarithmic term favors nodes with deeper topological levels. $\text{Cov}_{\text{up}}(v)$ is assigned a cubic weight to prioritize upstream convergence, thereby filtering out leaf nodes that may exhibit high downstream coverage merely by chance.

If $\text{Crit}(v) \geq 1.0$, node v is identified as a critical node, suggesting it serves as the primary convergence point within the connected component. Consequently, the monitor extracts the tuple $\langle p, a, o \rangle$ for node v and all its upstream nodes, where p denotes the affected prefix, a the critical AS, and o the corresponding origin AS. This tuple is forwarded to the anomaly detector for semantic embedding analysis. This process effectively prunes benign route flapping noise and significantly reduces the computational overhead for subsequent distance-based verification and post-hoc filtering.

4.4 Anomaly Detector

The anomaly detector receives candidate tuples $\langle p, a, o \rangle$ from the propagation monitor and performs semantic consistency verification. Leveraging the pre-trained encoder, this module first identifies suspicious candidates through geometric distance analysis and then applies post-hoc filtering to suppress or annotate them before generating contextualized alerts for operator investigation.

4.4.1 Semantic Distance-based Decision. To quantify the abnormality of topology changes, we transform them into continuous geometric distance metrics. For each candidate tuple, the system retrieves the profile embedding set $\mathcal{Z}_p$, the critical AS embedding $\mathbf{e}_a$, and the origin AS embedding $\mathbf{e}_o$. We define three key semantic distances:

- **Propagation Consistency:** $D_1 = d(\mathbf{e}_a, \mathcal{Z}_p)$. This metric characterizes the legitimacy of the critical AS a carrying prefix p based on historical propagation patterns.
- **Origination Consistency:** $D_2 = d(\mathbf{e}_o, \mathcal{Z}_p)$. This metric measures the plausibility of the origin AS o announcing prefix p , reflecting their semantic proximity.

- **AS-Role Consistency:** $D_3 = d_{\mathbb{D}}(\mathbf{e}_a, \mathbf{e}_o)$. This metric quantifies the role-induced semantic compatibility between AS a and o , detecting potential violations in their routing policies.

These distance metrics are grounded in the empirical observations reported in § 3.6.2 and in [1], where semantically consistent prefix-AS and origin relationships consistently exhibit smaller hyperbolic distances than unrelated pairs. All candidate tuples are scored against the historical profile reference constructed before the suspicious update is applied, and a newly observed more-specific prefix inherits the most-specific available historical parent profile. If prefixes that previously shared a profile diverge, the changed link or segment is evaluated against that pre-change reference, and profile reassignment occurs after scoring. The reference is immutable within each 24-hour model interval, and daily retraining consumes observations after current-interval scoring completes. This ordering isolates current detection from reference updates. Sustained changes across multiple retraining rounds are discussed in §6.

We identify anomalies by checking if these distances violate semantic consistency bounds defined by dynamic thresholds Θ . The decision logic adapts to two scenarios:

- **Standard Path Verification ($a \neq o$):** When the critical AS acts as a transit node, we enforce strict incompatibility corroborated by multi-dimensional evidence:

$$(D_1 > \theta_1) \wedge ((D_2 > \theta_2) \vee (D_3 > \theta_3)) . \quad (13)$$

- **Self-Origination Verification ($a = o$):** When the critical AS claims to be the origin, D_3 is undefined. To prevent false negatives caused by the *hyperbolic centrality bias* of core ASes, we introduce a norm-based sensitivity factor $\gamma(\mathbf{e}_a)$ to adaptively calibrate the decision boundary:

$$(D_1 > \gamma(\mathbf{e}_a) \cdot \theta_1) \vee (D_2 > \gamma(\mathbf{e}_a) \cdot \theta_2) . \quad (14)$$

For each distance, we derive $\theta = \min(\max(\tau_{\text{err}}, \tau_{\text{pos}}), \tau_{\text{neg}})$ from historical positive and random negative pairs: τ_{err} minimizes the sum of Type I and II errors, while τ_{pos} and τ_{neg} are the positive 99.95th and negative 1.6th percentiles. The radial factor γ is 0.8 below AS norm 0.8, increases linearly to 1.0 at norm 0.95, and remains 1.0 beyond it; detailed derivations are provided in [1].

4.4.2 Post-hoc Semantic Filtering. Certain legitimate operational events may also induce significant deviations. To mitigate such false positives, Parade applies a post-hoc filtering module to suspicious candidates produced by the distance-based detector.

Strong Consistency-based Suppression. We use two types of high-confidence evidence to suppress benign candidates. First, for origin-related candidates, we perform ROA validation on the candidate $\langle p, o \rangle$ pair. If the announcement is covered by a valid ROA, the candidate is treated as legitimate and suppressed. Second, for traffic-engineering cases, we check whether the prefix remains aligned with historical propagation fingerprints associated with the AS. If the updated route preserves the established fingerprint, the event is considered propagation-consistent and suppressed. Unlike external-prior-based rules, this fingerprint-based suppression relies only on native BGP-derived propagation semantics and therefore does not contaminate the learned semantic encoder.

Operational Context Annotation. For the remaining candidates, we identify representative operational patterns that may

explain large semantic deviations without providing sufficient evidence for direct suppression:

- **Intra-Organization Adjustments (Affiliation Correlation).** Routing responsibilities may shift among ASes operated by the same organization. We detect such transitions by examining organizational affiliations between the involved ASes and their upstreams, where ownership changes do not necessarily alter the higher-level propagation context.
- **Defensive Mitigation (Blackhole Routing Identification).** Defensive mechanisms such as DDoS mitigation may introduce aggressive route announcements with blackhole-routing community attributes. We identify these events by detecting BGP community attributes associated with blackhole routing.
- **Ownership Delegation (Re-Announcement Recognition).** Prefixes may be leased or temporarily delegated. We identify such cases by detecting total withdrawal followed by re-announcement after a significant gap, which suggests an operational transition rather than a persistent inconsistency.

These annotations are useful for operator interpretation but do not provide the same level of consistency as RPKI validation or fingerprint preservation. Therefore, Parade does not directly discard candidates matched only by these patterns. Instead, it reports them as contextualized alerts with operational annotations.

4.4.3 Contextualized Alert Aggregation and Ranking. After post-hoc filtering, Parade aggregates the remaining candidates into contextualized final alerts. Candidates suppressed by RPKI validation or fingerprint-based propagation consistency are excluded from the final alert set, while candidates matched only by auxiliary operational signals are retained with contextual annotations.

To enhance alert interpretability, we aggregate candidates based on the critical AS and its upstream ASes. Multiple anomalous samples occurring within the same time window and converging to the same critical AS are grouped into one report. We treat these critical ASes as candidate root causes. For a critical AS a , if all anomalous paths share the same upstream neighbor b and b is also identified as a critical AS, we attribute the report to b 's path adjustment rather than generating redundant alerts for downstream ASes.

Finally, aggregated alerts are ranked by the maximum D_1 among candidates. Since D_1 measures the semantic inconsistency between the affected prefix and the critical AS, this ranking prioritizes alerts with the most severe prefix-AS semantic violations. Each operator-facing report includes the decomposed $D_1/D_2/D_3$ scores, the changed upstream segment and suspicious AS path, the affected prefix scope, and the post-hoc evidence or operational annotations used during filtering.

5 Evaluation

In this section, we evaluate Parade using both real-world and simulated datasets. First, we benchmark Parade against state-of-the-art systems using recent incidents to verify its detection capabilities in realistic scenarios (§5.1). Next, we employ simulations to validate its robustness across a broader spectrum of anomaly types (§5.2). Finally, we present case studies from actual deployments, highlighting Parade's practical utility in combining data-driven semantic detection with post-hoc context (§5.4).

Baselines. We select the state-of-the-art methods from recent years as comparison baselines:

- **BEAM** [12]: A representation learning-based method that relies on *business relationships* to learn AS embeddings for identifying routing anomalies in path changes.
- **DFOH** [23]: A system that detects fake paths by extracting *multi-dimensional statistical features* from newly appeared AS links and utilizing weakly supervised learning.
- **Ares** [38]: An unsupervised method that clusters propagation graphs and detects significant changes via weighted edit distance based on link frequency and geolocation.
- **GraphBGP** [41]: A dynamic graph-learning method that outputs a window-level anomaly decision and ranks ASes by anomaly scores for culprit identification.
- **A+T**: *Argus* [37] detects anomalies by checking whether path segments in updates are absent from historical data. Subsequently, *Themis* [34] is further applied to eliminate false positives arising from legitimate MOAS events.
- **GraphDist**: Our non-learning baseline, which replaces Parade's hyperbolic distances with unweighted shortest-path distances on the historical AS graph while sharing the same profiles, candidates, post-hoc filters, and alert aggregation.

For a profile p with historical propagation AS set V_p and origin set O_p , GraphDist computes $D_1^G = \min_{v \in V_p} d_G(a, v)$, $D_2^G = \min_{v \in O_p} d_G(o, v)$, and $D_3^G = d_G(a, o)$ for the same candidate $\langle p, a, o \rangle$ used by Parade. Unseen or unreachable ASes receive $d_{\max} + 1$; discrete decision thresholds are derived from historical positive and negative pairs without tuning on the replayed incidents.

Experiment Setup. All our experiments were conducted on a Dell PowerEdge R760 server running Ubuntu 24.04 LTS. The hardware configuration consists of Intel Xeon Gold 6426Y CPUs and an NVIDIA GeForce RTX 4090 GPU (24GB VRAM). The proposed framework was implemented using C++17 and Python 3.12.

For baseline methods with public artifacts, we utilized the official code and recommended configurations. For A+T, whose implementation is not publicly available, we contacted the authors and used the historical detection results they provided for comparison. For historical baseline comparisons, we uniformly used data from all BGP collectors of RouteViews [35] to ensure fairness. For DFOH, as the official implementation was non-functional, we consulted the authors and applied necessary patches to the source code to ensure proper execution. Parade was trained using only 24 hours of historical RIB snapshots and 7 days of UPD messages prior to the start of each evaluated event.

Simulated evaluation sets. We separate historical-event replay, aggregate mixed-anomaly evaluation, and PM-specific diagnosis. For aggregate detection metrics, we construct a fixed set of 6,000 simulated anomalies from the January 1, 2026 RIB: 2,000 origin hijacks (OH), 2,000 path manipulations (PM), and 2,000 route leaks (RL). For each anomaly, we sample four legitimate routes from the same RIB, producing 24,000 normal samples and a 4:1 normal-to-anomalous class composition. The matched normal routes follow the same source/transit detection branch, propagation depth, and AS-rank tier as the corresponding anomaly. We repeat only the matched-normal sampling five times while keeping the 6,000 anomalies fixed, and metrics are computed per resampling and then

Table 1: Comparative results on historical anomalies, grouped by origin hijack (OH), route leak (RL), and path manipulation (PM). Each cell reports Id./Pfx/Alm: culprit identification (✓: correct; ✗: applicable but failed to detect or correctly attribute the event), unique target prefixes after event-window aggregation, and aggregated alarm reports divided by event duration in hours. A dash (–) denotes an unsupported anomaly type, unavailable historical results, or an unreported metric. Pfx measures detector-visible scope; Alm measures report volume. For a fair comparison, outputs from all methods are deduplicated by available prefix and/or AS path within each time window before alarm aggregation.

Type	Dataset	Culprit AS	Parade	GraphDist	GraphBGP	BEAM	DFOH [†]	Ares	A+T [‡]
OH	Rostelecom	12389	✓/542/15.3	✓/6702/42.0	✗/–/–	✗/0/4.0	–	✗/8/6.0	–
	Vodafone	55410	✓/282/19.3	✓/231/73.3	✗/–/–	✗/42/141.3	–	✗/11/3.3	✓/759/283.0
	Celer bridge [*]	209243	✓/1/12.3	✗/0/54.8	✗/–/–	✗/0/1.8	–	✓/1/1.8	✗/0/77.0
RL	Safehost	21217	✓/910/9.0	✓/137/50.6	✗ [‡] /–/–	✗/74/18.4	–	✗/87/14.4	–
	TIC	49666	✓/82/9.3	✓/7783/26.8	✗/–/–	✗/0/5.3	–	✗/0/2.8	✗/0/74.8
	Voxility	3223	✓/147/21.3	✓/32/39.8	✗/–/–	✗/14/2.3	–	✓/25/4.5	✓/223/1250
	Cloudflare [*]	13335	✓/28/20.0	✓/1/32.6	✗/–/–	✗/0/0.5	–	✗/0/0.0	✗/0/5704
PM	Klayswap [*]	6461/9457 [†]	✓/2/9.8	✓/18/38.0	✗/–/–	✗/0/1.0	✓/2/3.8	✗/0/0.4	✗/0/150.8

^{*} Small-scale incident; details are provided in [1].

[†] DFOH targets path manipulation introducing a new AS link; OH and RL are outside its scope. A+T results prior to 2020 are unavailable according to the authors.

[‡] GraphBGP flags the Safehost time window as route leak, but AS21217 is absent from its Top-10 AS ranking. GraphBGP also does not report Pfx or Alm.

Table 2: Performance comparison using datasets on January 1, 2026. Costs are measured per 15-minute update interval.

Method	Peak Memory (GB)	Preprocess (h) [*]	Process+Detect (s) [*]
Parade	30.8	1.53	4.6 + 3.2
BEAM	3.7	~ 10	28.9
DFOH	5.3	~ 1	8.0 + 1.8
Ares	45.4	~ 1	37.1
A+T [†]	5.2	~ 10	406.1

^{*} Excluding data download time for fair comparison.

[†] Provided by the authors and used only as a rough reference.

averaged. The same anomaly set and five matched-normal sample sets are reused across all collector-coverage levels. In contrast, the tier–depth PM heatmaps use a separate PM-only diagnostic set and are not aggregated into the mixed-set results.

5.1 Comparison on Real-World Events

We evaluated Parade against state-of-the-art methods using diverse real-world BGP incidents. Adopting the taxonomy in §2.1, we categorized these events into origin hijack, route leak, and path manipulation, encompassing both widespread disruptions and subtle, small-scale anomalies. We select publicly reported incidents whose time windows, culprit ASes, and affected prefixes or origins can be cross-validated against public BGP data, independently of Parade’s outcomes. The benchmark includes large-scale and targeted small-scale events, and the corresponding dataset details are provided in [1]. For validation, we manually verified alerts against reported affected prefixes and ASes for small-scale events. A method was considered successful in culprit AS identification only if it explicitly identified the reported culprit AS. Missed detections or incorrect attribution were marked as failures only when the anomaly type was supported by the method.

Table 1 presents the comparative results. We compare methods within each anomaly category because their detection objectives

and applicable signals differ. DFOH is evaluated only for PM because its OH and RL entries are out of scope. Parade covers more anomaly categories among the evaluated systems. For GraphBGP, we count attribution as correct when the culprit appears among its Top-10 AS scores. It flags only the Safehost window and does not rank any benchmark culprit in the Top 10.

Origin Hijacks (OH). Large-scale origin changes are readily exposed by origin- or path-novelty checks, so the main distinction in these events is culprit attribution and sensitivity to small scope. The baselines detect portions of the Vodafone event, and A+T identifies its culprit, while several systems misattribute or miss the other OH cases. Parade identifies the culprit in all three events, including the single-prefix Celer Bridge incident. We therefore use OH as a coverage and attribution check; RL and PM provide the more discriminative comparisons of path-related detection capabilities.

Route Leaks (RL). The four RL events directly compare systems on propagation-policy changes. Parade and GraphDist identify the culprit in all four, while Ares and A+T succeed only on Voxility among the completed baseline entries. GraphDist produces substantially broader target scopes or higher alarm rates on several events, most visibly TIC, while Parade retains correct attribution with a smaller operator-facing report volume. For Safehost, Parade surfaces 910 prefixes from 37 origin ASes, compared with 74 prefixes for BEAM and 87 for Ares. These Pfx counts represent detector-visible scope because public incident reports provide only coarse impact estimates.

Path Manipulation (PM). Klayswap is the only public PM event in the benchmark with sufficient route-level ground truth. Parade, DFOH, and GraphDist all detect it, whereas BEAM, GraphBGP, Ares, and A+T do not identify the culprit. This single event establishes replayability but provides limited evidence for ranking the three successful methods. Section 5.2 therefore compares Parade, DFOH, and GraphDist under the same tier/depth path-manipulation simulation. For Klayswap attribution, AS6461 is the suspicious origin and AS9457 is its upstream on the same path.

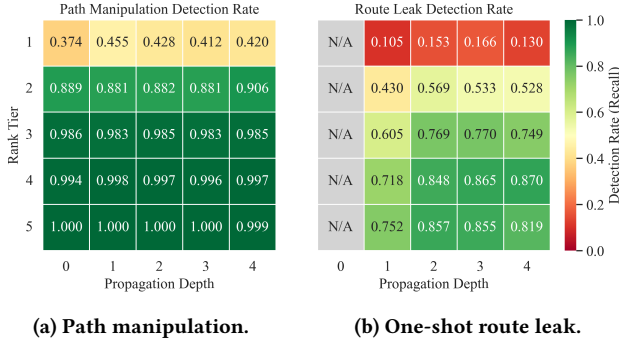


Figure 9: Detection recall of Parade against simulated anomalies, stratified by the propagation depth of the anchor AS (a_u) and the rank tier of the attacker or recipient AS (a_d).

Moreover, although GraphDist detects most of these events, it exhibits substantial false-positive risk on small-scale incidents. For Klayswap, it reports 18 prefixes although only two were affected; for the single-prefix Celer Bridge event, it fails to detect the affected prefix while still generating 54.8 alarms per hour. In contrast, Parade correctly identifies both incidents and reports exactly the documented affected prefixes, with no off-target prefixes observed.

Table 2 reports processing performance on January 1, 2026 data. Across deterministic nested inputs of 15–74 collectors, parser and dataset-building time grows from approximately 10 to 41 minutes, while early-stop training time varies with convergence. Full results are provided in [1]. At the full input, this complete preprocessing workflow takes 1.53 hours, and all measured scales remain below two hours. The online processor and detector take 4.6 and 3.2 seconds per 15-minute update batch, respectively. The processor and monitor reach stage-wise memory peaks of 30.8 GB and 27.7 GB.

5.2 Evaluation on Simulated Events

Path manipulations and small-scale route leaks are inherently more difficult to detect than origin hijacks. Given the scarcity of recent ground truth for these anomalies, we use tier–depth diagnostic simulations based on real-world BGP data from January 1, 2026. These diagnostic results are reported separately from the aggregate mixed-set metrics defined above.

Path Manipulation. For a victim prefix p , we select an existing upstream AS a_u from its propagation tree. We then designate an attacker AS a_d that shares no legitimate connection with a_u and inject a fake link (a_u, a_d) into the propagation path of p and then propagate the resulting manipulated route announcement.

One-Shot Route Leak. We select a leaker AS a_u from the propagation graph of a prefix p and identify a neighbor a_d such that propagating p from a_u to a_d violates the valley-free principle [18] based on CAIDA relationship datasets [9]. These inferred relationships are used only to construct simulation labels; neither the encoder nor online inference consumes them.

In both tier–depth diagnostics, the topological deviation converges at (a_u, a_d). We stratified the results by the *propagation depth* of a_u and the *rank tier* of a_d . ASes were categorized into five ranking tiers: [1, 500], (500, 2000], (2000, 5000], (5000, 15000], and >15000.

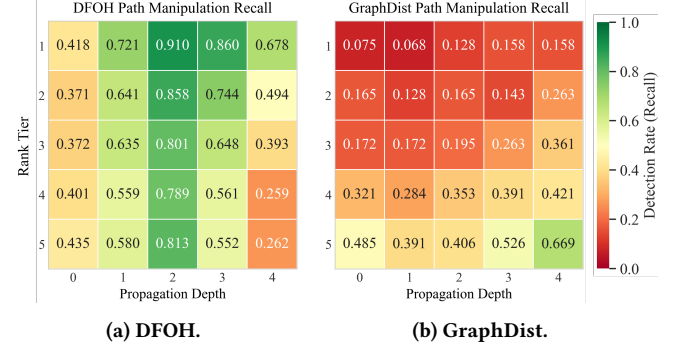


Figure 10: Path-manipulation recall of DFOH and GraphDist under the same simulation setup as Figure 9.

We generated 2,000 samples for each diagnostic scenario. Note that for route leaks, depth 0 corresponds to the origin AS and is therefore inapplicable (N/A) in this route-leak diagnostic setting.

PM-specific diagnostic set. The tier–depth PM heatmap is evaluated on a separate path-manipulation-only set. These samples are used only for the PM heatmaps of Parade, DFOH, and GraphDist and are excluded from the 4:1 mixed evaluation set and all aggregate collector-coverage results reported for the mixed anomaly set.

Figure 9a illustrates path manipulation detection. All Tier 3–5 cells (Rank > 2000) achieve 98.3–100% recall, which defines the scope of the 98.3% result. Tier 2 remains stable at 88.1–90.6% across propagation depths. The pronounced decline appears only in Tier 1. Tier-1 recall ranges from 37.4% to 45.5% because core ASes appear in many legitimate propagation contexts and occupy central regions of the learned space; inserting such an AS creates a weaker semantic displacement than inserting a regional or edge AS.

Figure 10 provides the controlled DFOH and GraphDist comparisons. Across Tiers 2–5, Parade achieves 0.967 macro-average recall, compared with 0.558 for DFOH, and improves every non-Tier-1 tier/depth cell. DFOH is stronger in Tier 1 because its link-level statistical features can identify an unusual new adjacency even when the participating AS is globally central. The two methods therefore capture complementary signals: Parade models prefix-specific propagation affinity, while DFOH emphasizes local link novelty. GraphDist loses additional resolution because shortest-path distances concentrate in the small-world AS graph, especially around the densely connected core. Separately, on the mixed evaluation set containing all three anomaly categories, GraphDist and Parade achieve aggregate recalls of 0.468 and 0.854, respectively, with MCC values of 0.638 and 0.898.

Intentional hijacks by established core carriers are uncommon in prior observations of persistent offenders [40], so the strong regional/edge results cover the more typical attacker population. Core-AS sensitivity remains operationally relevant for accidental leaks and compromised networks, motivating future fusion of semantic and link-novelty signals.

Figure 9b presents results for one-shot route leaks. The trend mirrors path manipulation, with Tier-1 detection being particularly challenging due to their extensive peering, which renders policy violations plausible from BGP propagation observations alone. This

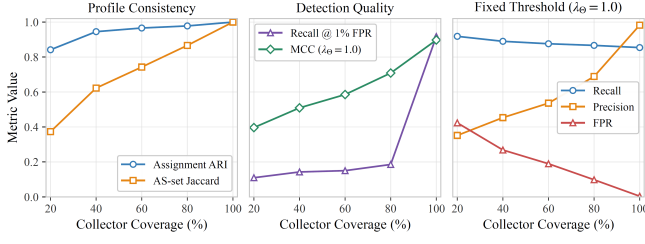


Figure 11: Sensitivity to collector coverage. Profile consistency relative to the 100%-coverage reference, measured by assignment ARI and mean AS-set Jaccard over prefixes observed. Overall detection quality on the mixed anomaly set, measured by recall at 1% FPR and MCC.

one-shot setting is a deliberately difficult stress test, while real-world leaks typically involve multiple prefixes. The results of the *TIC* (Rank 134) and *Cloudflare* (Rank 82) incidents confirm that Parade can successfully detect leaks initiated by core-level ASes in practice.

5.3 Robustness and Sensitivity

Incomplete observations. Using the January 1, 2026 RIB, we identify 74 RouteViews/RIS collectors and construct deterministic, source- and region-balanced nested subsets containing 15, 30, 44, 59, and 74 collectors. Figure 11 evaluates coverage at three levels: profile construction, overall detection quality, and fixed-threshold detection. All detection results pool OH, PM, and RL.

At 20% coverage, assignment ARI remains 0.84, but mean AS-set Jaccard falls to 0.37, indicating stable coarse partitioning with incomplete profile membership. As coverage increases to 100%, recall at 1% FPR rises from 0.110 to 0.918 and fixed-threshold MCC from 0.396 to 0.898. At $\lambda_0 = 1.0$, FPR drops from 42.36% to 0.38% and mixed-set precision rises from 0.352 to 0.983, while recall changes from 0.919 to 0.854. Thus, sparse coverage mainly inflates recall by over-flagging both anomalies and unseen legitimate relationships; full coverage produces approximately 38 false alerts per 10,000 normal routes in the mixed evaluation set.

Threshold sensitivity. The dynamic thresholds (§4.4) are selected from historical positive/negative distance distributions, accounting for latent legitimate routes in the sampled data. A label-based sensitivity sweep over the five matched-normal resamplings finds a statistical balance at threshold scale $\lambda_\Theta = 0.85$ (recall 0.906, precision 0.979 on the mixed evaluation set). The deployed default $\lambda_\Theta = 1.0$ prioritizes a conservative operator load and reduces one-day aggregated alerts from 824 to 322 (recall 0.854, precision 0.983 on the same mixed set). The sweep characterizes the sensitivity-alert-volume trade-off; full results are provided in [1].

Figure 12 reports the end-to-end detection sensitivity to the hyperparameters. Under the default configuration ($d = 24$, $K = 2$), Parade achieves 0.854 recall, 0.983 precision, 0.378% FPR, and an MCC of 0.898. The aggregate recall is primarily limited by route-leak detection: the recalls for origin hijacks, path manipulations, and route leaks are 0.962, 0.923, and 0.678, respectively.

Across the two sweeps, the aggregate metrics remain within narrow ranges: 0.851–0.870 recall, 0.982–0.983 precision, 0.363–0.399%

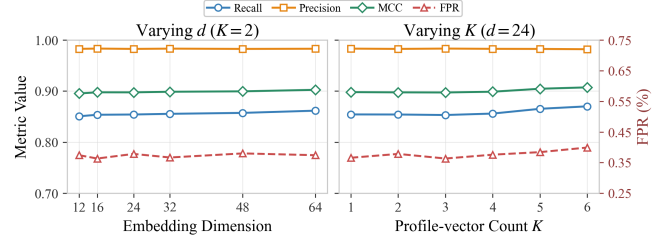


Figure 12: End-to-end detection sensitivity to embedding dimension d and profile-vector count K .

FPR, and 0.895–0.907 MCC. Thus, within the evaluated ranges, d and K have limited influence on downstream detection performance, while their more pronounced trade-offs concern representation capacity and computational cost.

Independent leak validation. For 2,000 CAIDA-constructed one-shot leaks, Parade achieves 0.678 recall overall, 0.701 on the 385 ASPA-covered samples, and 0.710 on the 376 paths independently classified as ASPA Invalid. ASPA-Invalid is a subset of ASPA-covered; ASPA-Unknown is a validation outcome rather than the complement of coverage, and includes nine covered but inconclusive paths in addition to uncovered paths. The recalls differ by only 3.2 points. This consistency follows from the experiment design: the encoder learns from BGP observations, while CAIDA and ASPA define or validate the evaluation subsets; further details are provided in [1].

5.4 Case Studies

Continuous deployment. We ran the online pipeline on RouteViews/RIS feeds from June 11–25, 2026 with daily retraining. It produced 3,285 raw candidates, of which 1,444 remained after strong consistency filtering; operators prioritized 194 for manual investigation using the changed path, historical propagation, RPKI/IRR status, organizational records, and public incident evidence. The retained stream also contained questionable operational practices, including leaked private ASNs, internal IPv4 /32 or IPv6 /64 routes, and RPKI/IRR-inconsistent transfers between related organizations. These counts define the operational alert funnel and validation procedure. The manual investigation prioritized the highest-ranked alerts; unconfirmed retained alerts remain *unresolved* rather than being labeled false positives.

Representative cases. We demonstrate the advantages of Parade’s data-driven approach in reducing false positives. We deployed Parade in a real-world setting from January 1 to January 7, 2026. During this period, multiple external reports flagged an anomaly involving Venezuela’s state-owned ISP, CANTV (AS8048) on January 2 [6, 21], coinciding with “Operation Absolute Resolve”. However, Parade classified the event as normal network fluctuation. Further investigation revealed that the AS link involving AS8048, which was suspected of violating business relationships, had been historically stable. The observed changes were not caused by AS8048 but originated downstream from AS52320. This conclusion was subsequently validated by a reputable security firm through an independent, comprehensive analysis. This highlights that detection methods relying on prior knowledge are susceptible

to misinterpretation, whereas Parade effectively filters such noise through native semantics.

This robustness is further evidenced by the analysis of the *Vodafone* route hijack event [2, 3]. Parade reported significantly fewer affected prefixes compared to public reports. Our manual verification indicated that the excluded prefixes belonged to Vodafone’s subnetworks (e.g., AS38266, AS12353). Since these networks legitimately rely on the parent AS for transit, their propagation patterns did not exhibit significant deviations, justifying Parade’s decision.

6 Discussion

Limitations. The efficacy of the data-driven approach is intrinsically linked to the density and diversity of routing observations. We attempted to reproduce less recent events, such as the 2017 *Rostelecom* incident [30]. Our evaluation reveals degradation in capability, attributed to sparse data coverage during that era. Our controlled collector ablation shows that reduced visibility primarily raises false positives and changes recall non-monotonically: at 20% coverage, many legitimate AS relationships are unseen and therefore also appear anomalous. The reference model is refreshed daily, so a sufficiently persistent manipulation may eventually influence a later baseline. The immutable pre-update reference prevents same-update contamination; baseline protection across retraining rounds remains future work. The 24-hour cadence is an operational trade-off between adaptation and reference stability, not a theoretically optimal interval. Each daily model is trained independently, and we do not claim stability of raw embedding coordinates across days.

Future Work. We plan to extend Parade in two directions. From a defensive standpoint, we aim to correlate the learned control-plane semantics with data-plane measurements to verify whether routing anomalies result in actual reachability failures or traffic blackholing. This cross-layer verification is expected to significantly reduce false positives in deployments. From an analytical standpoint, we intend to develop a fine-grained classifier based on Parade’s embeddings to automatically categorize attack types (e.g., distinguishing route leaks from malicious hijacks). This will facilitate automated incident response and more targeted mitigation strategies.

7 Related Work

Heuristic and Statistical Approaches. Traditional systems relying on statistical heuristics or expert-defined rules [13, 36, 37, 40] often struggle to adapt to the dynamic nature of BGP. DFOH [23] analyzes link-level statistical features, while Ares [38] clusters propagation graphs. These approaches suffer from *lossy topological abstraction*: DFOH is confined to local link-level views, while Ares resorts to coarse clustering, which can obscure subtle topological shifts caused by stealthy anomalies.

Graph-Based and Representation Learning Approaches. Recent works leverage Graph Neural Networks (GNNs) to capture routing semantics [12, 22, 33], including GraphBGP’s dynamic attributed AS graph [41]. However, these paradigms typically adopt a *disjoint analytical perspective*. Methods like BEAM [12] and BGP-Shield [42] focus primarily on AS-level topology, treating prefixes as passive identifiers. This yields representations that are *invariant* across prefixes, failing to capture the heterogeneity of per-prefix routing policies. Although FIRE [29] attempts to bridge this gap,

it still relies on aggregated statistical features, falling short of a unified, high-fidelity representation of propagation dynamics. Complementary work shows that latent hyperbolic geometry can recover routing-relevant paths even from substantially incomplete AS graphs [25]. Our collector ablation quantifies the effect of missing observations on prefix-specific anomaly decisions.

8 Conclusion

In this paper, we presented Parade, a framework for prefix-AS joint hyperbolic representation learning. By embedding routing entities into a unified space, Parade captures the hierarchical and co-propagation affinities of the routing system. Our asymmetric modeling strategy resolves semantic polysemy in complex scenarios using only native BGP RIBs and updates.

We implemented Parade as a high-performance system capable of maintaining high-fidelity propagation topologies and conducting real-time anomaly detection. Historical evaluation demonstrates coverage across anomaly categories. Across simulated path manipulations involving regional and edge ASes, Parade maintains recall above 98.3%. Its main sensitivity boundary lies at the Internet core, where broadly connected ASes are semantically compatible with many prefixes. These results suggest that data-driven semantic modeling is promising for practical BGP anomaly detection and understanding the routing ecosystem.

Acknowledgments

This work was supported by Zhongguancun Laboratory, Beijing, China. This paper was edited for grammar and fluency using OpenAI’s ChatGPT (GPT-5.5 and GPT-5.6).

References

- [1] 2026. Parade: Additional Technical Report for Extended Evaluation Details and Supplementary Results. <https://github.com/weiyz23/CCS26-Parade/blob/main/technical-report.pdf>. Additional technical report.
- [2] Aftab Siddiqui. 2021. A Major BGP Hijack by AS55410-Vodafone Idea Ltd. <https://manrs.org/2021/04/a-major-bgp-hijack-by-as55410-vodafone-idea-ltd/>.
- [3] Alessandro Improta and Luca Sani. 2021. Vodafone Idea BGP leak: global routing system must implement MANRS. <https://www.catchpoint.com/blog/vodafone-idea-bgp-leak>.
- [4] Alexander Azimov, Eugene Bogomazov, Randy Bush, Keyur Patel, Job Snijders, and Kotikalapudi Sriram. 2025. BGP AS_PATH Verification Based on Autonomous System Provider Authorization (ASPA) Objects. <https://datatracker.ietf.org/doc/draft-ietf-sidrops-aspa-verification/24/>.
- [5] Alexander Azimov, Eugene Bogomazov, Randy Bush, Keyur Patel, and Kotikalapudi Sriram. 2022. Route Leak Prevention and Detection Using Roles in UPDATE and OPEN Messages. <https://www.rfc-editor.org/info/rfc9234>.
- [6] Bryton Herdes. 2026. A closer look at a BGP anomaly in Venezuela. <https://blog.cloudflare.com/bgp-route-leak-venezuela/>.
- [7] Bryton Herdes and Tom Strickx. 2026. Route leak incident on January 22, 2026. <https://blog.cloudflare.com/route-leak-incident-january-22-2026/>.
- [8] Randy Bush and Rob Austein. 2025. The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 2. <https://datatracker.ietf.org/doc/draft-ietf-sidrops-8210bis/23/>. Work in Progress.
- [9] CAIDA. 2026. The CAIDA AS Relationships Dataset. <https://www.caida.org/catalog/datasets/as-relationships/>.
- [10] Wenqi Chen, Zhiliang Wang, Dongqi Han, Chenxin Duan, Xia Yin, Jiahai Yang, and Xingang Shi. 2022. ROV-MI: Large-Scale, Accurate and Efficient Measurement of ROV Deployment. In *Proc. NDSS*.
- [11] Yihao Chen, Qi Li, Ke Xu, Zhuotao Liu, and Jianping Wu. 2026. Understanding the Stealthy BGP Hijacking Risk in the ROV Era. In *Proc. NDSS*.
- [12] Yihao Chen, Qilei Yin, Qi Li, Zhuotao Liu, Ke Xu, Yi Xu, Mingwei Xu, Ziqian Liu, and Jianping Wu. 2024. Learning with Semantics: Towards a Semantics-Aware Routing Anomaly Detection System. In *Proc. USENIX Security*. 5143–5160.
- [13] Yutao Dong, Qing Li, Richard O Sinnott, Yong Jiang, and Shutao Xia. 2021. ISP self-operated BGP anomaly detection based on weakly supervised learning. In *Proc. IEEE ICNP*. IEEE, 1–11.

- [14] Doug Madory. 2023. A Brief History of the Internet's Biggest BGP Incidents. <https://nanog.org/stories/articles/a-brief-history-of-the-internets-biggest-bgp-incidents/>.
- [15] Doug Madory. 2024. RPKI ROV Deployment Reaches Major Milestone. <https://www.kentik.com/blog/rpki-rov-deployment-reaches-major-milestone/>.
- [16] Doug Madory. 2025. Beyond Their Intended Scope: DDoS Mitigation Leak. <https://manrs.org/2025/04/ddos-mitigation-leak-kentik/>.
- [17] Ben Du, Katherine Izhikevich, Sumanth Rao, Guatam Akiwate, Cecilia Testart, Alex C Snoeren, and KC Claffy. 2023. Irregularities in the Internet Routing Registry. In *Proc. ACM IMC*. 104–110.
- [18] Lixin Gao and Jennifer Rexford. 2001. Stable Internet routing without global coordination. *IEEE/ACM Trans. Netw.* 9, 6 (2001), 681–692.
- [19] Alessandro Giaconia, Muoi Tran, Laurent Vanbever, and Stefano Vissicchio. 2025. Is Crunching Public Data the Right Approach to Detect BGP Hijacks? *arXiv preprint arXiv:2507.20434* (2025).
- [20] Vasileios Giotsas, Matthew Luckie, Bradley Huffaker, and KC Claffy. 2014. Inferring complex AS relationships. In *Proc. ACM IMC*. 23–30.
- [21] Graham Helton. 2026. News: There Were BGP Anomalies During The Venezuela Blackout. <https://loworbitsecurity.com/radar/radar16/>.
- [22] Kevin Hoarau, Pierre Ugo Tournoux, and Tahiry Razafindralambo. 2022. Bgnn: Detection of BGP anomalies using graph neural networks. In *Proc. IEEE ISCC*. IEEE, 1–6.
- [23] Thomas Holterbach, Thomas Alfroy, Amreesh Phokeer, Alberto Dainotti, and Cristel Pelsser. 2024. A system to detect Forged-Origin BGP hijacks. In *Proc. USENIX NSDI*. 1751–1770.
- [24] Arthur B Kahn. 1962. Topological sorting of large networks. *Commun. ACM* 5, 11 (1962), 558–562.
- [25] Maksim Kitsak, Alexander Ganin, Ahmed Elmokashfi, Hongzhu Cui, Daniel A. Eisenberg, David L. Alderson, Dmitry Korkin, and Igor Linkov. 2023. Finding shortest and nearly shortest path nodes in large substantially incomplete networks by hyperbolic mapping. *Nature Communications* 14, 1 (2023), 186. doi:10.1038/s41467-022-35181-w
- [26] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. 2010. Hyperbolic geometry of complex networks. *Phys. Rev. E* 82, 3 (2010), 036106.
- [27] Matt Lepinski and Kotikalapudi Sriram. 2017. BGPsec Protocol Specification. <https://www.rfc-editor.org/info/rfc8205>
- [28] Weitong Li, Zhexiong Lin, Md Ishtiaq Ashiq, Emile Aben, Romain Fontugne, Amreesh Phokeer, and Taejoong Chung. 2023. RoVista: Measuring and analyzing the route origin validation (ROV) in RPKI. In *Proc. ACM IMC*. 73–88.
- [29] Yizhi Li, Jiang Li, Jiahao Cao, Renjie Xie, Yangyang Wang, and Mingwei Xu. 2024. Poster: Few-Shot Inter-Domain Routing Threat Detection with Large-Scale Multi-Modal Pre-Training. In *Proc. ACM CCS*. 4970–4972.
- [30] Nick Kephart. 2017. Rostelecom Route Leak Targets E-Commerce Services. <https://www.thousandeyes.com/blog/rostelecom-route-leak-targets-ecommerce-services>.
- [31] Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. *Adv. Neural Inf. Process. Syst.* 30 (2017).
- [32] PeeringDB. 2026. PeeringDB. <https://www.peeringdb.com/>.
- [33] Songtao Peng, Jiaqi Nie, Xincheng Shu, Zhongyuan Ruan, Lei Wang, Yunxuan Sheng, and Qi Xuan. 2022. A multi-view framework for BGP anomaly detection via graph attention network. *Comput. Netw.* 214 (2022), 109129.
- [34] Lancheng Qin, Dan Li, Ruifeng Li, and Kang Wang. 2022. Themis: Accelerating the detection of route origin hijacking by distinguishing legitimate and illegitimate MOAS. In *Proc. USENIX Security*. 4509–4524.
- [35] RouteViews. 2026. University of Oregon RouteViews Project. <https://www.routeviews.org/routeviews/>.
- [36] Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas Dimitropoulos, Danilo Cicalese, Alistair King, and Alberto Dainotti. 2018. ARTEMIS: Neutralizing BGP hijacking within a minute. *IEEE/ACM Trans. Netw.* 26, 6 (2018), 2471–2486.
- [37] Xingang Shi, Yang Xiang, Zhiliang Wang, Xia Yin, and Jianping Wu. 2012. Detecting prefix hijackings in the Internet with Argus. In *Proc. ACM IMC*. 15–28.
- [38] Yinxiang Tao, Chengwan Zhang, Changqing An, Shuying Zhuang, Jilong Wang, and Congcong Miao. 2025. Ares: Comprehensive Path Hijacking Detection via Routing Tree. In *Proc. USENIX Security*. 803–821.
- [39] Robert Tarjan. 1972. Depth-first search and linear graph algorithms. *SIAM J. Comput.* 1, 2 (1972), 146–160.
- [40] Cecilia Testart, Philipp Richter, Alistair King, Alberto Dainotti, and David Clark. 2019. Profiling BGP serial hijackers: capturing persistent misbehavior in the global routing table. In *Proc. ACM IMC*. 420–434.
- [41] Zheng Wu, Yanbiao Li, Xin Wang, Zulong Diaio, Weibei Fan, Fu Xiao, and Gaogang Xie. 2025. GraphBGP: BGP Anomaly Detection Based on Dynamic Graph Learning. *IEEE Trans. Inf. Forensics Security* 20 (2025), 9864–9877. doi:10.1109/TIFS.2025.3607239
- [42] Heng Zhao, Ruoyu Wang, Tianhang Zheng, Qi Li, Bo Lv, Yuyi Wang, and Wenliang Du. 2025. From Topology to Behavioral Semantics: Enhancing BGP Security by Understanding BGP's Language with LLMs. *arXiv preprint arXiv:2511.14467* (2025).

A Ethical Considerations

Our research is designed to enhance the security and resilience of the Internet's routing system. To achieve this, our methodology relies exclusively on passively collected routing data (e.g., RIBs and update messages). We strictly adhere to the data usage policies of RouteViews, RIPE RIS, and CAIDA. Thus, our system involves no active probing or interference with live network traffic.

Furthermore, this study does not involve sensitive information or confidential data. All prefixes and ASes explicitly discussed in our representation learning analysis are derived from public routing and registration data, and their ownership or organizational information is obtained from publicly available sources such as WHOIS and RIPEstat. All BGP anomalies mentioned in this paper are publicly documented, and the case study validations were conducted with the explicit consent of the collaborating security firm which remains anonymous to prevent any potential disclosure.

Finally, we ensured proper authorization and academic integrity regarding the baseline methods evaluated. For closed-source systems (e.g., A+T), we officially obtained the detection results directly from the respective authors. For temporarily nonfunctional baselines (e.g., DFOH), we were explicitly granted permission to patch their source code to conduct fair comparisons.

B Open Science

To support transparency and reproducibility, we provide the Parade artifacts through the project's public GitHub repository:

<https://github.com/weiyz23/CCS26-Parade>

The provided repository contains the following components:

- **Source Code:** The complete C++ and Python implementation of Parade, including the whole pipeline from data preprocessing to model training and anomaly detection.
- **Scripts:** We provide scripts for the entire experimental workflow, allowing for reproducibility of the results presented in the paper and any custom analyses.
- **Documentation:** A comprehensive README.md detailing system requirements, installation steps, and commands to reproduce the core experimental results.
- **Additional Technical Report:** A standalone report containing dataset details, fingerprinting and sampling analyses, additional representation results, ablations, and evaluation protocols [1].