NETCAST: LEARNING COMPACT LATENT REPRESENTATIONS FROM NETWORK TRACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Network foundation models promise reusable representations for diverse traffic analysis tasks, but existing models can produce concentrated embedding spaces and show weak alignment with interpretable traffic statistics, raising questions about whether their representations capture meaningful traffic behavior. We translate these limitations into two concrete design objectives: learning behavioral structure through future burst-state prediction and encouraging representation diversity through distributional regularization. Guided by these objectives, we introduce NetCast, which predicts future burst states from causal flow history using a shared encoder that combines packet content with traffic metadata and a state-distribution regularizer that encourages variation across encoded bursts. We evaluate NetCast across four traffic datasets through centered kernel alignment (CKA) with measured network statistics and intrinsic analyses of embedding geometry. NetCast achieves the highest reported mean CKA of 0.104 and the lowest average pairwise cosine similarity of 0.55 among the compared trained models, while reducing angular concentration on every dataset. A regularizer ablation further supports the contribution of distributional regularization to this geometric improvement. Together, these results demonstrate the potential of burst-state prediction with distributional regularization to produce better-structured network representations that more closely reflect measurable traffic behavior.

1 INTRODUCTION

Network foundation models learn reusable representations from packet traces to support application identification, intrusion detection, and traffic analysis (Ahmad et al., 2021; Lin et al., 2022; Zhao et al., 2023; Guthula et al., 2023). When traffic is encrypted, these models must rely on visible protocol fields, timing, and flow behavior (Akbari et al., 2021). These signals reflect how a network exchange unfolds: applications generate traffic, transport protocols control delivery, and queues and competing traffic change its pace. A useful representation should capture these behaviors and preserve differences between traffic observations.

Classification accuracy alone does not show whether a model has learned such a representation. Trustee shows that models can achieve strong results by exploiting patterns specific to a dataset (Jacobs et al., 2022). The Intrinsic Evaluation Framework (IEF) examines representations more directly and finds that evaluated network foundation models often map different observations to similar vectors, show weak correspondence with expert-defined traffic features, and struggle to distinguish underlying network conditions (Beltiukov et al., 2025). These findings suggest two requirements: representations should capture understandable properties of traffic, and they should keep different observations distinguishable. We use both requirements to assess representation quality.

The order of traffic bursts provides a way to learn these properties. The direction, size, and timing of past bursts reveal how an exchange is progressing. Predicting future bursts therefore encourages a model to retain useful information from past bursts. UNUM's work on learning network states also motivates using past observations to understand changing network behavior (Chen et al., 2026). Prediction alone, however, is not enough. If the encoder maps every burst to nearly the same vector, those vectors become easy to predict but carry little useful information. Our central idea is therefore to *predict future burst representations from past bursts while keeping burst representations diverse*.

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

NetCast implements these two goals through burst prediction and network state regularization. A shared encoder maps each burst's packet content and traffic metadata to a vector, and a history model uses past burst representations to predict future burst representations. The prediction task encourages the encoder to capture how an exchange develops, while the regularizer preserves differences among burst representations. Figure 1 compares this approach with masked packet reconstruction.

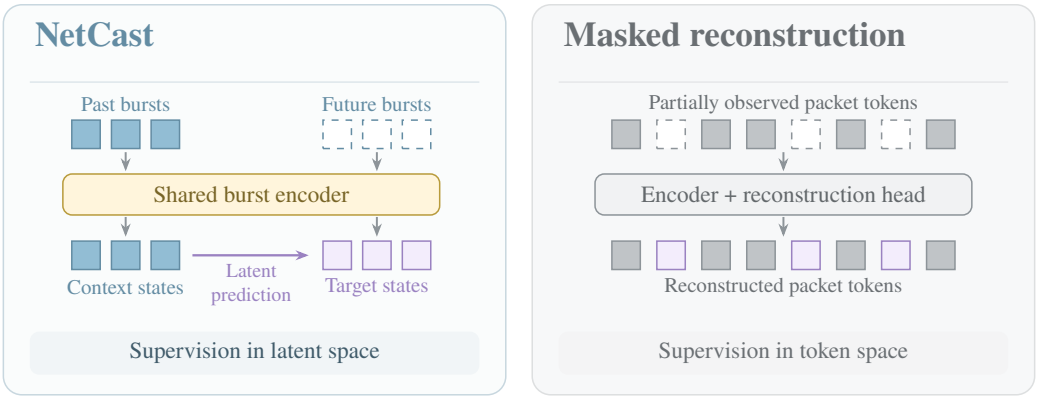


Figure 1: **Predicting burst representations versus reconstructing packet content.** NetCast uses past burst representations to predict future burst representations produced by a shared encoder. Masked reconstruction instead predicts missing tokens from packet content.

We evaluate representation quality on Crossmarkets, CIC-IDS2017, CAIDA, and MAWI. We use Centered Kernel Alignment (CKA) to measure how closely the learned representations correspond to measured traffic statistics. We also examine whether representation vectors point in similar directions, using mean pairwise cosine similarity, and whether a few dimensions dominate that similarity, using Mean Cosine Contribution (MCC).

NetCast achieves the highest reported mean CKA: 0.104, compared with 0.102 for netFound-small and 0.093 for netFound-base. This advantage is small and comes mainly from MAWI; netFound performs better on the other three datasets. The results for vector similarity are more consistent. NetCast has the lowest mean pairwise cosine similarity on all four datasets, with an average of 0.55, compared with 0.73 and 0.68 for the two netFound models. Its MCC is 0.01 on every dataset. A controlled comparison with and without the regularizer further shows that regularization makes the vectors less concentrated in the same direction. Together, these measurements provide our main evidence of improved representation quality. Application and intrusion classification offer additional checks that the representations remain useful in practice.

Our contributions are as follows: First, we introduce a pretraining method that predicts future burst representations from past bursts while encouraging burst representations to remain diverse. Second, we evaluate what the representations capture and how they preserve differences between observations, finding a small improvement in average alignment with traffic statistics and a consistent reduction in directional concentration. Third, we show the regularizer's role in this reduction through a controlled ablation and verify the representations' practical usefulness on traffic-analysis tasks.

2 Network Observations and Learning Requirements

2.1 Traffic as an Evolving, Partially Observed Process

A flow is $X = (B_1, \ldots, B_T)$, where $B_t = (P_t, m_t)$ combines packet-derived tokens P_t with direction, timing, volume, and protocol metadata m_t. Burst order t follows the exchange; elapsed time is encoded in m_t. Protocol-aware inputs and operational metadata preserve complementary aspects of traffic behavior (Guthula et al., 2023; Beltiukov et al., 2026). Application phase, transport feedback, queues, and cross-traffic jointly shape these observations. A conceptual hidden state s_t

expresses this dependence:

$$s_{t+1} \sim p(s_{t+1} \mid s_t), \qquad B_t \sim p(B_t \mid s_t). \tag{1}$$

These relations connect evolving conditions to observable bursts. Repeated exchanges and changing pace supply historical evidence about network state (Chen et al., 2026).

2.2 From representation diagnostics to training requirements

Diagnostic studies motivate three requirements. **Preserve behavioral evidence:** protocol content and burst metadata should remain available together, while preprocessing and evaluation control identity and timestamp shortcuts (Arp et al., 2020; Zhao et al., 2025). **Connect observations over time:** histories should retain information useful for anticipating later bursts, including temporal effects of hidden context highlighted by IEF (Beltiukov et al., 2025). **Maintain distinguishable states:** heterogeneous observations should occupy a distributed latent space so that prediction operates on meaningful differences. These requirements guide input organization, causal prediction, and latent distribution regularization, respectively.

2.3 Learning target and evaluation

NetCast encodes each burst as z_t and summarizes the observed history $B_{1:t}$ as c_t. The training target is the next K burst embeddings, $z_{t+1:t+K}$. Context and targets share an encoder and projector; only observed states enter the causal predictor. Both sets of latents receive distributional regularization (Figure 2). Our primary evaluation examines the resulting representations directly: mean CKA measures correspondence with operational traffic statistics, while cosine geometry and dimensional dominance characterize how observations occupy the latent space. A regularizer ablation tests its effect on angular concentration. Downstream classification provides an auxiliary check of practical utility.

3 Method

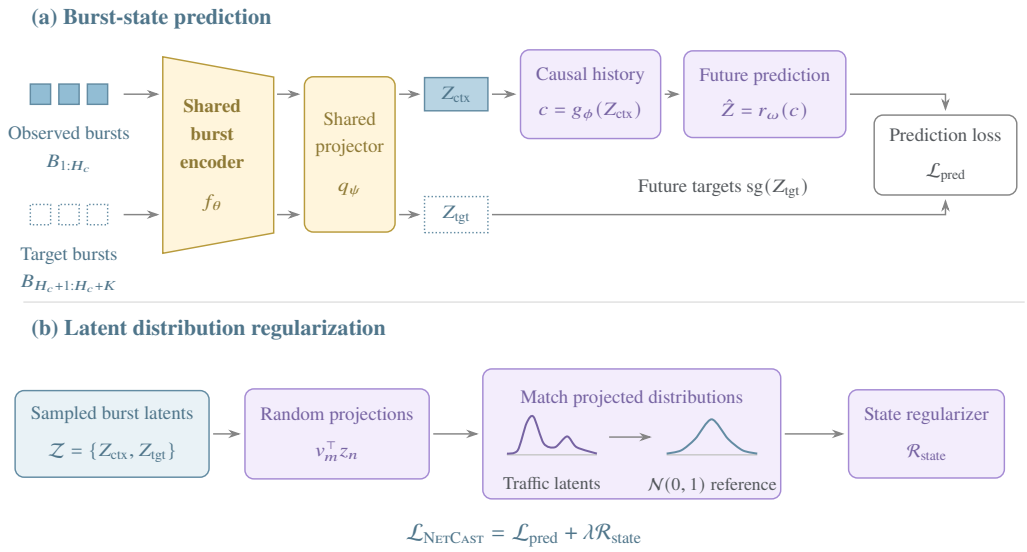


Figure 2: **Training pipeline.** Context and target bursts pass through the same encoder and projector. The history branch predicts subsequent targets without attending to future burst tokens. Latent distribution regularization acts on both sets of burst latents alongside the prediction loss.

3.1 FROM PACKET OBSERVATIONS TO BURST STATES

Each input flow contains at most T bursts, each with at most L packet-derived tokens and a prepended `CLS` token. We use the same flow processing and tokenization as netFound (Guthula et al., 2023). Metadata include direction, byte and packet counts, inter-arrival time, flow duration, and transport protocol. Burst features are aligned with token positions, and inter-arrival times use a millisecond-level discretization. Payload stripping, when enabled, retains protocol-header tokens and metadata.

For token $p_{t,j}$ at position j in burst t, the encoder input is

$$u_{t,j} = E_{\text{tok}}(p_{t,j}) + E_{\text{pos}}(j) + E_{\text{meta}}(m_{t,j}) + E_{\text{proto}}(\pi_t), \tag{2}$$

where π_t denotes the transport protocol and the embedding maps share a common output dimension. Self-attention operates within each burst (Vaswani et al., 2017; Zhao et al., 2023; Guthula et al., 2023):

$$h_t = \big[f_\theta(u_{t,0:L})\big]_{\texttt{CLS}}, \qquad z_t = q_\psi(h_t) \in \mathbb{R}^d. \tag{3}$$

Padding positions are excluded from the representation. The vector h_t summarizes a burst observation; the projector q_ψ supplies its prediction target z_t. Cross-burst context is introduced by the causal history model below.

3.2 PREDICTING THE CONTINUATION OF A FLOW

A sampled training window contains H_c context bursts followed by K target bursts. A causal Transformer summarizes the context, and a prediction head maps this summary to the future targets:

$$c_t = g_\phi(z_{1:t}), \quad t \le H_c, \qquad \hat{Z}_{H_c+1:H_c+K} = r_\omega(c_{H_c}). \tag{4}$$

Here g_ϕ denotes the causal context computation and c_{H_c} its output used by the head. The predictor receives no future burst tokens or target latents. Its loss is

$$\mathcal{L}_{\text{pred}} = \frac{1}{K}\sum_{k=1}^{K} \|\hat{z}_{H_c+k} - \text{sg}(z_{H_c+k})\|_2^2, \tag{5}$$

where sg stops gradients through the target in this term. Prediction in representation space (Assran et al., 2023; Bardes et al., 2024; Assran et al., 2025) avoids requiring a decoder to reproduce individual packet fields. For traffic, the training signal links the observed exchange to its continuation at burst granularity, while the burst encoder retains packet content and timing context.

3.3 NETWORK-STATE DISTRIBUTION REGULARIZATION

Preserving the states that drive a flow's continuation. The observations in Section 2 distinguish stages of an exchange: similar packet content can occur during a sustained transfer or after an application pause, while a direction change can mark a response. The latent space must preserve these differences as the temporal objective learns their continuation.

NetCast couples two requirements: *temporal predictability within a flow* and *distributional diversity across encoded burst observations*. The first trains the relation between a history and its continuation. The second gives those predictions a non-degenerate state space. In particular, shrinking both encoded and predicted states by a factor α shrinks their squared prediction error by α^2, even as differences between observed exchange phases vanish. Our regularizer opposes this contraction by assigning the latent distribution a reference scale and spread. This separation of temporal structure from distributional structure also reflects the distinction emphasized in network trace modeling (Lu et al., 2026).

Regularizing traffic states at burst granularity. Let $\mathcal{W}$ index the valid context and target bursts in the sampled training windows, with (i,t) denoting burst t of flow i and $N = |\mathcal{W}|$. Their empirical latent distribution is

$$\widehat{P}_{\text{burst}} = \frac{1}{N}\sum_{(i,t)\in\mathcal{W}} \delta_{z_{i,t}}, \qquad z_{i,t} = q_\psi(h_{i,t}), \tag{6}$$

where δ_z is a point mass at z and $h_{i,t}$ encodes packet content together with the burst metadata. Each sample therefore corresponds to a concrete observed stage of a flow. Including context and target bursts regularizes both sides of the temporal prediction task in the coordinates defined by the shared encoder and projector.

Measuring variation across traffic latents. For a unit direction $v \in \mathbb{S}^{d-1}$, the scalar $v^\top z_{i,t}$ probes a combination of encoded traffic attributes. Its empirical characteristic function is

$$\widehat{\varphi}_v(u) = \frac{1}{N} \sum_{(i,t)\in\mathcal{W}} \exp\big(\mathrm{i}u\, v^\top z_{i,t}\big)\,. \tag{7}$$

This function summarizes how the sampled burst states are distributed along that latent direction; u indexes variation in latent coordinates rather than elapsed network time. Random directions probe the joint representation without assigning individual coordinates to hand-selected metrics such as byte count or inter-arrival time. A constant burst encoding has $|\widehat{\varphi}_v(u)| = 1$, exposing a representation that has erased differences among traffic observations.

A common scale for heterogeneous burst states. We use a standard normal reference for each projection and penalize distributional mismatch:

$$\mathcal{R}_{\text{state}}(\widehat{P}_{\text{burst}}) = \frac{N}{M} \sum_{m=1}^{M} \int_{\mathbb{R}} \left| \widehat{\varphi}_{v_m}(u) - e^{-u^2/2} \right|^2 w(u)\, du, \tag{8}$$

where $v_1, \ldots, v_M$ are sampled unit directions and w is a nonnegative integration weight. The reference places the learned traffic coordinates on a common scale. At the ideal distributional target, every unit projection has zero mean and unit variance, so the latent covariance is the identity. This encourages use of the available latent dimensions when encoding heterogeneous timing, volume, direction, and protocol observations. The Gaussian target describes this representation geometry; the observed packet measurements retain their own distributions. We evaluate the integral with finite integration samples and average over the sampled directions.

The joint objective is

$$\mathcal{L}_{\text{NetCast}} = \underbrace{\mathcal{L}_{\text{pred}}}_{\text{within-flow continuation}} + \lambda\, \underbrace{\mathcal{R}_{\text{state}}(\widehat{P}_{\text{burst}})}_{\text{burst-state distribution}}, \tag{9}$$

where λ balances continuation accuracy and the distribution of encoded traffic states. The prediction term supplies temporal order; the regularizer supplies a reference geometry for its context and targets. Together they train a compact space in which burst states remain distributed while their evolution is predictable. Sections 4.4 and 4.3 evaluate the resulting traffic-statistic alignment and latent geometry. The regularizer ablation isolates its effect on angular concentration; auxiliary classification results appear in Section 4.5.

4 Evaluation

We evaluate representation quality through two complementary forms of intrinsic evidence: alignment with interpretable traffic statistics and the geometry of the learned embeddings. The four-dataset mean CKA summarizes alignment, while cosine similarity and Mean Cosine Contribution (MCC) characterize angular concentration and dimensional dominance. We then ablate the state regularizer to examine its geometric effect. Downstream classification serves as an auxiliary check of representation utility.

4.1 Pretraining data

We pretrain NetCast on the identical Campus 1 training split used by netFound (Guthula et al., 2023), comprising approximately 1.8 million flows. Both models use the same corpus, preprocessing pipeline, and tokenization scheme. This common input pipeline anchors the netFound comparison in shared packet observations, allowing us to examine the representations learned by the different models. NetCast has 48M parameters; model size provides context for the comparison, while intrinsic representation quality is the primary outcome.

Campus 1 consists of passive traces from a campus border gateway. The timestamp-based training split contains 70% of the approximately 2.6 million flows remaining after filtering. NetCast is pretrained on four Nvidia A10 GPUs using the burst-state objective in Section 3. The corpus is shared with permission. Collection procedures, privacy protections, and implementation details are given in Appendix A.

4.2 Datasets and evaluation design

Intrinsic representation evaluation. We analyze embeddings on four datasets spanning application traffic, security benchmarks, and operational networks. **Crossmarkets** contains 46,179 flows across 210 application classes with a long-tailed distribution (Ren et al., n.d.); we remove timestamp-based shortcuts in TCP options (Arp et al., 2020). **CIC-IDS2017** includes benign traffic and multiple attack categories, including DDoS, brute force, and infiltration (Sharafaldin et al., 2018; Liu et al., 2022). We also analyze 1.5 million anonymized backbone flows from **CAIDA** (CAIDA, 2024) and 1 million sampled flows from **MAWI** (Cho et al., 2000). These operational collections broaden the evaluation beyond labeled classification benchmarks. Each dataset contributes equally to the reported mean CKA and mean cosine similarity.

Comparisons and supporting evaluation. We compare ET-BERT, YaTC, NetMamba, netFound-small, and netFound-base in the intrinsic analyses; Appendix C reports netFound-large separately. The regularizer ablation holds the remaining NetCast training setup fixed. For auxiliary classification, we use Crossmarkets and CIC-IDS2017 together with **ISCXVPN-2016**, which contains 9,536 flows across 17 application classes under VPN conditions (Draper-Gil et al., 2016). These task scores assess whether the representations retain practical utility alongside the intrinsic improvements.

Table 1: Embedding geometry: mean pairwise cosine similarity (↓). Avg. is the arithmetic mean across four datasets. Random is the isotropic reference. Colors rank trained models only: green/bold = lowest, orange = second-lowest distinct, red = highest.

Model	Crossmarkets	CIC-IDS2017	CAIDA	MAWI	Avg.
Random	0.00	0.00	0.00	0.00	0.00
ET-BERT	0.88	0.74	0.87	0.85	0.84
YaTC	0.85	0.85	0.86	0.85	0.85
NetMamba	0.93	0.92	0.99	0.99	0.96
netFound-small	0.68	0.65	0.79	0.78	0.73
netFound-base	0.63	0.60	0.73	0.77	0.68
NetCast (ours)	**0.52**	**0.50**	**0.53**	**0.64**	**0.55**

4.3 Embedding geometry and network-state regularization

Design. We next examine whether the alignment result is accompanied by a less concentrated latent space, as motivated by the state-distribution requirement from Section 3.3. Mean pairwise cosine similarity measures the angular concentration of embeddings: lower positive values indicate a representation that is less concentrated in a shared direction. Table 1 compares the full pretrained models; Table 2 then isolates the effect of $\mathcal{R}_{\text{state}}$ within NetCast. This separates the cross-model observation from the intervention on the proposed training objective.

Global anisotropy. NetCast has the lowest mean cosine similarity among the trained models on each of the four datasets. On Crossmarkets, its 0.52 compares with 0.68 for small, 0.63 for base, and 0.85–0.93 for the other traffic models. On CIC-IDS2017, it reaches 0.50 compared with 0.60 for base. The same pattern extends to operational traffic: CAIDA decreases from 0.73 for base to 0.53 for NetCast, and MAWI decreases from 0.77 to 0.64. The average is 0.55 for NetCast, compared with 0.68 for base and 0.73 for small. Thus the geometric difference appears across both task-labeled and operational traffic collections.

Dimensional dominance. Dimensional analysis provides a complementary view through Mean Cosine Contribution (MCC), which quantifies the contribution of the dominant embedding dimension to cosine similarity. YaTC's MCC values are 0.25, 0.22, 0.21, and 0.19 on Crossmarkets, CIC-IDS2017, CAIDA, and MAWI, respectively; NetCast reports 0.01 on all four. Together, mean cosine similarity and MCC characterize two aspects of the latent space: concentration in a shared direction

and concentration of similarity in one coordinate. NetCast exhibits reduced angular concentration and low single-dimension dominance. These are complementary intrinsic advantages: similarity is less concentrated in a common direction and less dominated by one coordinate. Lower cosine similarity alone does not establish informative representations, as the random reference illustrates; its value here is assessed together with traffic-statistic alignment.

Effect of the regularizer. We remove $\mathcal{R}_{\text{state}}$ from Equation 9 while keeping the remaining training setup fixed. Table 2 reports this paired comparison separately from the full-model analysis. Adding the regularizer reduces cosine similarity on every dataset, with the largest absolute reductions of 0.12 on CIC-IDS2017 and CAIDA. The decreases on Crossmarkets and MAWI are both 0.02. The effect therefore spans an application benchmark, an intrusion benchmark, and two operational collections.

The paired comparison attributes the reduction in angular concentration to $\mathcal{R}_{\text{state}}$. It connects the distributional objective to the measured geometric improvement. This ablation evaluates cosine geometry; it does not isolate the regularizer's effect on CKA.

Table 2: Ablation of network-state regularization. Values are mean pairwise cosine similarities (↓); the last row is the reduction obtained by adding $\mathcal{R}_{\text{state}}$. Green and bold mark the lower value in each paired comparison.

Configuration	Crossmarkets	CIC-IDS2017	CAIDA	MAWI
NetCast w/o $\mathcal{R}_{\text{state}}$	0.54	0.62	0.65	0.66
NetCast w/ $\mathcal{R}_{\text{state}}$	**0.52**	**0.50**	**0.53**	**0.64**
Reduction	0.02	0.12	0.12	0.02

4.4 Alignment with network statistics

Design. We use Centered Kernel Alignment (CKA) to examine the structure shared by model embeddings and CICFlowMeter features (Kornblith et al., 2019; Canadian Institute for Cybersecurity, n.d.; Beltiukov et al., 2025). These features summarize packet sizes, direction, timing, and transport behavior. CKA assesses correspondence with these interpretable traffic attributes without making them reconstruction targets during pretraining. Table 3 reports dataset-level values and their equally weighted mean, our aggregate measure of traffic-statistic alignment.

Table 3: CKA alignment with CICFlowMeter features (↑). Avg. is the arithmetic mean across datasets. Random and Oracle denote idealized zero-alignment and self-alignment references. Colors rank trained models by average CKA only: green/bold = highest, orange = second-highest distinct, red = lowest.

Model	Crossmarkets	CIC-IDS2017	CAIDA	MAWI	Avg.
Random	0.000	0.000	0.000	0.000	0.000
Oracle (ideal)	1.000	1.000	1.000	1.000	1.000
ET-BERT	0.012	0.064	0.033	0.026	0.034
YaTC	0.098	0.092	0.014	0.070	0.069
NetMamba	0.047	0.042	0.030	0.051	0.043
netFound-small	0.154	0.149	0.046	0.059	0.102
netFound-base	0.154	0.143	0.045	0.030	0.093
NetCast (ours)	0.138	0.134	0.034	0.108	**0.104**

Highest reported mean alignment. NetCast achieves a mean CKA of 0.104, compared with 0.102 for netFound-small, 0.093 for netFound-base, 0.069 for YaTC, 0.043 for NetMamba, and 0.034 for ET-BERT. The netFound-large comparison in Appendix C reports 0.094. Thus, NetCast has the

highest reported average across the evaluated trained models. The margin over the strongest mean-CKA baseline is small: 0.002 at the reported precision. We interpret this as a favorable aggregate alignment result alongside the geometric evidence below; these point estimates do not establish statistical significance.

Where the aggregate advantage comes from. NetCast's MAWI CKA is 0.108, compared with 0.059 for small, 0.030 for base, and 0.070 for YaTC. This gain drives the four-dataset average: netFound-small and netFound-base have higher alignment on Crossmarkets, CIC-IDS2017, and CAIDA. The result therefore supports an aggregate advantage across the evaluated traffic collections, with different models retaining distinct alignment profiles. On MAWI, stronger alignment also accompanies lower angular concentration, linking the two complementary criteria for representation quality.

Interpretable feature associations. Appendix D reports each model's five most aligned CICFlowMeter features. NetCast's strongest associations include minimum backward packet length (0.513), minimum forward segment size (0.405), maximum backward packet length (0.383), maximum packet length (0.373), and initial forward-window bytes (0.345). These associations connect the representation to bidirectional packet-size structure and transport initialization. CKA measures shared structure with the selected statistics; it does not establish causal recovery of hidden network state or direct encoding of individual features.

4.5 Downstream verification

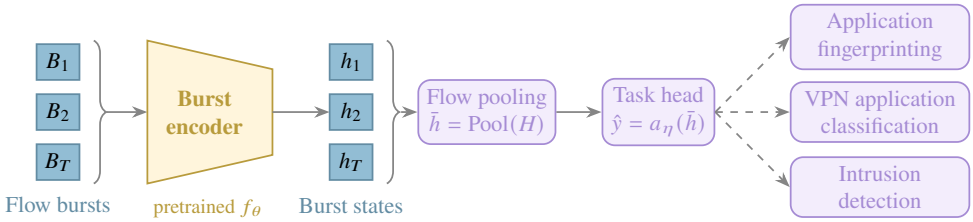


Figure 3: **Downstream adaptation.** The pretrained burst encoder supplies hidden states that are pooled into a flow representation for a task-specific classifier.

Table 4: Downstream verification. NetCast and netFound use the same pretraining data and processing pipeline. Crossmarkets reports Acc@10, while ISCXVPN-2016 and CIC-IDS2017 report F1. Avg. denotes the unweighted descriptive mean across the three task scores. Colors rank models by average performance only: green/bold = highest, orange = second-highest distinct, and red = lowest.

Model	Application identification		Intrusion detection	Avg. ↑
	Crossmarkets Acc@10 ↑	ISCXVPN-2016 F1 ↑	CIC-IDS2017 F1 ↑	
Curtains	20.64	66.85	99.75	62.41
nPrintML (Holland et al., 2021)	64.83	84.10	99.93	82.95
ET-BERT	35.62	77.57	99.94	71.04
YaTC	58.13	83.84	99.92	80.63
netFound-small (53M)	47.27	89.20	99.96	78.81
netFound-base (225M)	60.66	91.75	99.97	84.13
NetCast (48M, ours)	60.64	92.53	99.99	**84.39**

Design. We use application identification and intrusion detection to check the practical utility of the learned representations. The pretrained burst encoder supplies pooled flow representations to task-specific classifiers (Figure 3). Table 4 reports Crossmarkets Acc@10 and ISCXVPN-2016 and CIC-IDS2017 F1, all as percentages. The netFound comparisons share pretraining data and input processing; the remaining models provide additional benchmark context.

Practical utility alongside intrinsic improvements. NetCast improves over the similarly sized netFound-small on all three tasks, including 60.64 versus 47.27 Acc@10 on Crossmarkets and 92.53 versus 89.20 F1 on ISCXVPN-2016. Relative to netFound-base, it reaches nearly identical Crossmarkets performance (60.64 versus 60.66) and higher VPN F1 (92.53 versus 91.75), using 48M versus 225M parameters. CIC-IDS2017 scores are near saturation for both models (99.99 versus 99.97), providing limited discrimination. These results confirm useful transfer to application and security tasks. The principal evidence for improved representation quality remains the combination of mean CKA, reduced angular concentration, and low dimensional dominance.

5 Related work

Traffic representation learning and evaluation. ET-BERT, YaTC, netFound, and Lens learn traffic representations using packet context, masked prediction, or protocol structure (Lin et al., 2022; Zhao et al., 2023; Guthula et al., 2023; Li et al., 2024). NetMamba explores efficient modeling with a state-space architecture (Wang et al., 2024), while NetBench benchmarks traffic classification and generation (Qian et al., 2024). NetCast evaluates how well burst representations capture traffic statistics and preserve differences between observations. Comparisons with netFound use the same pretraining corpus and input processing.

Traffic generation and network dynamics. NetGPT and TrafficGPT use generative pretraining for traffic understanding and generation (Meng et al., 2023; Qu et al., 2024), while TrafficLLM and NetLLM adapt language-model methods to traffic and networking tasks (Cui et al., 2025; Wu et al., 2024). NetDiffusion and NetReplica synthesize traffic for protocol-constrained generation and domain adaptation (Jiang et al., 2023; Daneshamooz et al., 2025). Other work focuses on how traffic evolves: UNUM estimates network state from measurement histories (Chen et al., 2026), and CascadeNet generates flow time series that preserve temporal patterns (Lu et al., 2026). NetCast learns from passive packet traces, using past bursts to predict future burst representations.

Self-supervised representation learning. Self-supervised learning extracts useful representations from unlabeled data by constructing supervision from the data itself. Within this line of research, joint-embedding methods learn contextual relationships directly in feature space (Assran et al., 2023; Bardes et al., 2024; Assran et al., 2025; Balestriero and LeCun, 2025). I-JEPA learns semantic image representations by inferring the features of masked target regions from visible context, with spatial masking encouraging the capture of high-level image structure (Assran et al., 2023). V-JEPA extends this approach to video, learning features that support both appearance and motion understanding without pixel reconstruction or text supervision (Bardes et al., 2024). V-JEPA 2 scales this learning framework to larger video datasets, improving visual understanding and action anticipation while enabling robotic planning through action-conditioned post-training (Assran et al., 2025).

6 Conclusion

NetCast learns network representations by predicting future burst states from causal flow history under a state-distribution regularizer. Its central result is a favorable combination of intrinsic properties: the highest reported mean CKA across the evaluated traffic datasets, consistently lower angular concentration, and low single-dimension dominance. The mean-CKA advantage is modest and driven by MAWI, while the geometric improvement holds across all four datasets. Removing the regularizer increases angular concentration, connecting this component of the objective to the observed geometry. Auxiliary classification results confirm that the representations remain useful for traffic-analysis tasks. Together, the evidence supports burst-state prediction with distributional regularization as an approach to improving network representation quality.

AI USE STATEMENT

The authors defined the research questions, interventions, experimental design, and interpretation of results. Generative AI tools assisted with writing, language editing. The authors are responsible for verifying the measurements, analysis, and final manuscript.

ETHICS STATEMENT

Network representation learning can support network management and intrusion detection, but it may also enable traffic surveillance, user fingerprinting, and network reconnaissance. The campus network traces used in this work were shared under institutional authorization and remain subject to the data owner's release policy. Packet truncation and address anonymization reduce exposure but do not eliminate behavioral privacy risks. We report results only in aggregate. These risks, together with the associated access restrictions, should be considered when using or releasing models trained on operational network traffic; Appendix B describes the corresponding data safeguards.

REPRODUCIBILITY STATEMENT

We describe the shared pretraining split and processing pipeline in Section 4.1, the learning objective in Section 3, and implementation and evaluation details in Appendix A. The private pretraining corpus contains sensitive user network information and is therefore shared under a permission controlled access policy; obtaining the data requires authorization from the data owner. The supplementary material includes implementation code. The complete codebase and model checkpoints will be released upon acceptance.

REFERENCES

Zeeshan Ahmad, Adnan Shahid Khan, Cheah Wai Shiang, Johari Abdullah, and Farhan Ahmad. Network Intrusion Detection System: A Systematic Study of Machine Learning and Deep Learning Approaches. Transactions on Emerging Telecommunications Technologies, 32(1):e4150, 2021. URL https://doi.org/10.1002/ett.4150.

Iman Akbari, Mohammad A. Salahuddin, Leni Ven, Noura Limam, Raouf Boutaba, Bertrand Mathieu, Stephanie Moteau, and Stephane Tuffin. A Look Behind the Curtain: Traffic Classification in an Increasingly Encrypted Web. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 5(1), Article 4, 2021. URL https://doi.org/10.1145/3447382.

Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. Dos and Don'ts of Machine Learning in Computer Security, 2020. URL https://arxiv.org/abs/2010.09470. Published at USENIX Security 2022.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture, 2023. URL https://arxiv.org/abs/2301.08243.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning, 2025. URL https://arxiv.org/abs/2506.09985.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and Scalable Self-Supervised Learning Without the Heuristics, 2025. URL https://arxiv.org/abs/2511.08544.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting Feature Prediction for Learning Visual Representations from Video, 2024. URL https://arxiv.org/abs/2404.08471.

Sylee Beltiukov, Satyandra Guthula, Wenbo Guo, Walter Willinger, and Arpit Gupta. Demystifying Network Foundation Models, 2025. URL https://arxiv.org/abs/2509.23089.

Sylee Beltiukov, Satyandra Guthula, Haarika Manda, Jaber Daneshamooz, Wenbo Guo, Walter Willinger, Arpit Gupta, and Inder Monga. netFound: Principled Design for Network Foundation Models, 2026. URL https://arxiv.org/abs/2310.17025v5. Version 5, revised May 12, 2026.

CAIDA. Anonymized Two-Way Traffic Packet Header Traces 100G (5 sec) Sampler, 2024. URL https://catalog.caida.org/dataset/passive_100g_sampler. Dataset; traces from November 2024. Accessed January 21, 2025.

Canadian Institute for Cybersecurity. CICFlowMeter: Network Traffic Flow Generator and Analyser, n.d. URL https://www.unb.ca/cic/research/applications.html. Software, University of New Brunswick. Accessed September 26, 2026.

Jiayi Chen, Nihal Sharma, Debajit Chakraborty, Saurabh Agarwal, Jeffrey Zhou, Aditya Akella, and Sanjay Shakkottai. UNUM: A New Framework for Network Control. 23rd USENIX Symposium on Networked Systems Design and Implementation (NSDI 26), pp. 1–18, 2026. URL https://www.usenix.org/conference/nsdi26/presentation/chen-jiayi.

Kenjiro Cho, Koushirou Mitsuya, and Akira Kato. Traffic Data Repository at the WIDE Project. USENIX Annual Technical Conference, FREENIX Track, pp. 263–270, 2000. URL https://www.usenix.org/publications/library/proceedings/usenix2000/freenix/full_papers/cho/cho.pdf.

Tianyu Cui, Xinjie Lin, Sijia Li, Miao Chen, Qilei Yin, Qi Li, and Ke Xu. TrafficLLM: Enhancing Large Language Models for Network Traffic Analysis with Generic Traffic Representation, 2025. URL https://arxiv.org/abs/2504.04222.

Jaber Daneshamooz, Jessica Nguyen, William Chen, Sanjay Chandrasekaran, Satyandra Guthula, Ankit Gupta, Arpit Gupta, and Walter Willinger. Addressing the ML Domain Adaptation Problem for Networking: Realistic and Controllable Training Data Generation with NetReplica, 2025. URL https://arxiv.org/abs/2507.13476v1. Version 1, July 17, 2025.

Gerard Draper-Gil, Arash Habibi Lashkari, Mohammad Saiful Islam Mamun, and Ali A. Ghorbani. Characterization of Encrypted and VPN Traffic using Time-related Features. 2nd International Conference on Information Systems Security and Privacy (ICISSP), pp. 407–414, 2016. URL https://www.scitepress.org/papers/2016/57407/57407.pdf.

Satyandra Guthula, Roman Beltiukov, Navya Battula, Wenbo Guo, Arpit Gupta, and Inder Monga. netFound: Foundation Model for Network Security, 2023. URL https://arxiv.org/abs/2310.17025v4. First submitted in 2023; cited version 4, revised January 29, 2025.

Jordan Holland, Paul Schmitt, Nick Feamster, and Prateek Mittal. New directions in automated traffic analysis. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3366–3383. Association for Computing Machinery, 2021. doi: 10.1145/3460120.3484758. URL https://doi.org/10.1145/3460120.3484758.

Arthur S. Jacobs, Roman Beltiukov, Walter Willinger, Ronaldo A. Ferreira, Arpit Gupta, and Lisandro Z. Granville. AI/ML for Network Security: The Emperor has no Clothes. ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 1537–1551, 2022. URL https://doi.org/10.1145/3548606.3560609.

Xi Jiang, Shinan Liu, Aaron Gember-Jacobson, Arjun Nitin Bhagoji, Paul Schmitt, Francesco Bronzino, and Nick Feamster. NetDiffusion: Network Data Augmentation Through Protocol-Constrained Traffic Generation, 2023. URL https://arxiv.org/abs/2310.08543.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network Representations Revisited, 2019. URL https://arxiv.org/abs/1905.00414.

Xiaochang Li, Chen Qian, Qineng Wang, Jiangtao Kong, Yuchen Wang, Ziyu Yao, Bo Ji, Long Cheng, Gang Zhou, and Huajie Shao. Lens: A Knowledge-Guided Foundation Model for Network Traffic, 2024. URL https://arxiv.org/abs/2402.03646.

Xinjie Lin, Gang Xiong, Gaopeng Gou, Zhen Li, Junzheng Shi, and Jing Yu. ET-BERT: A Contextualized Datagram Representation with Pre-training Transformers for Encrypted Traffic Classification, 2022. URL https://arxiv.org/abs/2202.06335.

Lisa Liu, Gints Engelen, Timothy Lynar, Daryl Essam, and Wouter Joosen. Error Prevalence in NIDS datasets: A Case Study on CIC-IDS-2017 and CSE-CIC-IDS-2018. IEEE Conference on Communications and Network Security (CNS), pp. 254–262, 2022. URL https://intrusion-detection.distrinet-research.be/CNS2022/index.html.

Runwei Lu, Yanran Deng, Ruixuan Li, Jinting Liu, Yuejie Wang, Xinyu Li, Deming Xu, Han Tian, Kai Chen, and Guyue Liu. CascadeNet: Generating Network Traffic with High-Fidelity Temporal Patterns. 23rd USENIX Symposium on Networked Systems Design and Implementation (NSDI 26), pp. 1613–1633, 2026. URL https://www.usenix.org/conference/nsdi26/presentation/lu-runwei.

Xuying Meng, Chungang Lin, Yequan Wang, and Yujun Zhang. NetGPT: Generative Pretrained Transformer for Network Traffic, 2023. URL https://arxiv.org/abs/2304.09513.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, 2019. URL https://arxiv.org/abs/1912.01703.

Chen Qian, Xiaochang Li, Qineng Wang, Gang Zhou, and Huajie Shao. NetBench: A Large-Scale and Comprehensive Network Traffic Benchmark Dataset for Foundation Models, 2024. URL https://arxiv.org/abs/2403.10319.

Jian Qu, Xiaobo Ma, and Jianfeng Li. TrafficGPT: Breaking the Token Barrier for Efficient Long Traffic Analysis and Generation, 2024. URL https://arxiv.org/abs/2403.05822.

Jingjing Ren, Daniel J. Dubois, and David Choffnes. An International View of Privacy Risks for Mobile Apps, n.d. URL https://recon.meddle.mobi/papers/cross-market.pdf. Undated technical report, Northeastern University.

Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. 4th International Conference on Information Systems Security and Privacy (ICISSP), 2018. URL https://www.scitepress.org/papers/2018/66398/66398.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, 2017. URL https://arxiv.org/abs/1706.03762.

Tongze Wang, Xiaohui Xie, Wenduo Wang, Chuyi Wang, Youjian Zhao, and Yong Cui. NetMamba: Efficient Network Traffic Classification via Pre-training Unidirectional Mamba, 2024. URL https://arxiv.org/abs/2405.11449.

Duo Wu, Xianda Wang, Yaqi Qiao, Zhi Wang, Junchen Jiang, Shuguang Cui, and Fangxin Wang. NetLLM: Adapting Large Language Models for Networking, 2024. URL https://arxiv.org/abs/2402.02338.

Ruijie Zhao, Mingwei Zhan, Xianwen Deng, Yanhao Wang, Yijun Wang, Guan Gui, and Zhi Xue. Yet Another Traffic Classifier: A Masked Autoencoder Based Traffic Transformer with Multi-Level Flow Representation. Proceedings of the AAAI Conference on Artificial Intelligence, 37(4):5420–5427, 2023. URL https://ojs.aaai.org/index.php/AAAI/article/view/25674.

Yuqi Zhao, Giovanni Dettori, Matteo Boffa, Luca Vassio, and Marco Mellia. The Sweet Danger of Sugar: Debunking Representation Learning for Encrypted Traffic Classification, 2025. URL https://arxiv.org/abs/2507.16438.

A Implementation Details

A.1 Training Environments

All experiments are conducted on Linux servers with CUDA-enabled Nvidia GPUs. The implementation is based on PyTorch (Paszke et al., 2019) 2.6.0 with CUDA 12.4 and cuDNN 9. The software environment includes standard compilation, packet-processing, and deep-learning dependencies, including `gcc`, `g++`, `cmake`, `make`, `python3-dev`, `libpcap-dev`, PcapPlusPlus v24.09, and `flash-attn==2.7.4.post1`. We also use `graphviz` and GNU `parallel` for auxiliary visualization and experiment-management scripts.

We train a single compact 48M-parameter model. The NetCast-48M model is trained on a Linux server equipped with four Nvidia A10 GPUs. This configuration evaluates intrinsic representation quality within a compact parameter budget.

A.2 Data Preprocessing

The corpus consists of unsampled passive packet traces collected at a campus border gateway router under approval from the university IRB and Committee on IT Policies. Only the first 96 bytes of each packet are retained, and campus IP addresses are anonymized using prefix-preserving randomization. Campus IT collects traces in 15-minute intervals totaling approximately one hour per collection period. After filtering flows with fewer than six packets and bursts containing at most two packets, Campus 1 contains approximately 2.6 million flows; its timestamp-based 70% training split is used for pretraining. NetCast uses burst sequences for self-supervised latent future prediction rather than masked-token reconstruction. Public downstream benchmarks support evaluation, while independent reproduction of pretraining requires access to the authorized corpus.

We pretrain NetCast on unlabeled packet-level flow data. Each raw packet trace is parsed into flows, and each flow is decomposed into a temporally ordered sequence of traffic bursts. Each burst contains packet-derived tokens and aligned traffic metadata, such as direction, byte count, packet count, inter-arrival time, flow duration, and transport protocol.

Each flow and burst is truncated or padded to a fixed maximum length. A burst-level `CLS` token is prepended to each burst and used as the burst summary representation. When payload stripping is enabled, payload tokens are removed while protocol-header tokens and traffic metadata are retained, encouraging the model to focus on protocol and traffic dynamics rather than raw payload content.

A.3 Compact Model Configuration

We evaluate a single 48M-parameter instantiation, NetCast-48M. It follows the proposed architecture, consisting of a burst encoder, a latent projector, a causal predictor, and the state-distribution regularizer. The evaluation centers on alignment with measured network statistics and intrinsic embedding geometry. The compact configuration provides context for these comparisons, with downstream transfer used as an auxiliary check of practical utility.

A.4 Evaluation Metrics

Intrinsic representation analysis is the primary evaluation. We compute Centered Kernel Alignment between model embeddings and CICFlowMeter traffic features, reporting the equally weighted mean over Crossmarkets, CIC-IDS2017, CAIDA, and MAWI alongside the dataset-level values. We compute mean pairwise cosine similarity to measure global angular concentration and Mean Cosine Contribution to measure single-dimension dominance. Lower positive mean cosine similarity indicates reduced angular concentration; MCC describes dimensional contributions to cosine similarity. These geometric diagnostics are interpreted jointly with alignment to traffic statistics.

For auxiliary downstream evaluation, Crossmarkets reports Acc@10, while ISCXVPN-2016 and CICIDS2017 report F1 score. Aggregate CKA and downstream comparisons use the available point estimates; we do not infer statistical significance from their rankings.

B SAFEGUARDS

Network traffic data may contain sensitive information about user behavior, application usage, network infrastructure, and operational conditions. Although our method does not require raw-packet reconstruction, the underlying pretraining data and learned representations may still encode traffic patterns that should be handled carefully. Therefore, we adopt several safeguards for responsible release and evaluation.

First, we do not release raw packet payloads, personally identifiable traffic content, or non-anonymized packet traces. When public datasets are used, we follow their corresponding licenses, access conditions, and anonymization policies. When controlled or internally collected traces are used, we restrict release to aggregate results, processed representations, or anonymized artifacts that do not expose user-identifiable content.

Second, released artifacts are intended to focus on model code, preprocessing scripts, evaluation scripts, and public benchmark configurations. If pretrained checkpoints are released, they should be accompanied by responsible-use guidance that restricts misuse for unauthorized traffic surveillance, user fingerprinting, or offensive network reconnaissance. The model is designed for research on network representation learning, traffic understanding, and security evaluation, not for deployment in settings that violate user privacy or network policy.

Third, downstream evaluations are conducted on public or controlled benchmark datasets, and reported results are aggregated at the dataset level. We do not report or expose individual user traces, raw flow payloads, or personally identifying metadata. These safeguards are intended to reduce privacy risk while preserving the scientific value of the proposed predictive network representation learning framework.

C COMPARISON WITH NETFOUND-LARGE

Table 5 reports the 640M-parameter netFound-large comparison separately. NetCast has higher average CKA (0.104 versus 0.094) and lower average cosine similarity (0.55 versus 0.66), extending the aggregate intrinsic comparisons to this larger reference. In the auxiliary tasks, NetCast obtains higher VPN F1 and the same CIC-IDS2017 F1, while netFound-large obtains higher Crossmarkets Acc@10.

Table 5: netFound-large results. Downstream scores are percentages; CKA and cosine values are reported on their original scales.

Evaluation	Dataset	netFound-large
Parameters		640M
Acc@10	Crossmarkets	66.35
F1	ISCXVPN-2016	91.02
F1	CIC-IDS2017	99.99
Mean cosine	Crossmarkets	0.64
	CIC-IDS2017	0.55
	CAIDA	0.68
	MAWI	0.76
	Avg.	0.66
CKA	Crossmarkets	0.155
	CIC-IDS2017	0.144
	CAIDA	0.046
	MAWI	0.033
	Avg.	0.094

D REPRESENTATION SIMILARITY TO FLOW-LEVEL FEATURES

To better understand what information is captured by the learned network representations, we measure their similarity to conventional flow-level statistics extracted by CICFlowMeter. Specifically, we use centered kernel alignment (CKA) to compare model embeddings with individual CICFlowMeter features. For each model, Table 6 reports the five features with the highest average CKA similarity, together with the standard deviation across samples or evaluation splits.

Table 6: Average CKA similarity between model embeddings and CICFlowMeter features. For each model, we report the five CICFlowMeter features with the highest similarity.

Model	Feature name	Mean CKA	CKA Std
YaTC	Fwd Seg Size Min	0.423	0.185
	FWD Init Win Bytes	0.339	0.241
	Packet Length Min	0.286	0.085
	SYN Flag Count	0.282	0.230
	Bwd Packet Length Min	0.272	0.209
ET-BERT	SYN Flag Count	0.122	0.080
	FWD Init Win Bytes	0.081	0.068
	Packet Length Std	0.078	0.094
	FIN Flag Count	0.074	0.074
	Packet Length Max	0.068	0.097
NetMamba	Fwd Segment Size Avg	0.225	0.244
	Fwd Packet Length Mean	0.225	0.244
	Average Packet Size	0.221	0.258
	Packet Length Min	0.215	0.273
	Packet Length Mean	0.210	0.253
netFound-640M	FWD Init Win Bytes	0.385	0.352
	Packet Length Max	0.372	0.302
	Fwd Seg Size Min	0.359	0.338
	Fwd IAT Total	0.331	0.173
	Flow Duration	0.331	0.170
NETCAST-48M	Bwd Packet Length Min	0.513	0.260
	Fwd Seg Size Min	0.405	0.369
	Bwd Packet Length Max	0.383	0.098
	Packet Length Max	0.373	0.192
	FWD Init Win Bytes	0.345	0.331

The resulting feature profiles differ substantially across representation models. ET-BERT exhibits relatively weak correspondence to individual CICFlowMeter statistics, with its largest mean CKA being 0.122 for SYN Flag Count. NetMamba shows stronger similarity to packet-size statistics, with its highest-ranked features dominated by forward and aggregate packet lengths. YaTC exhibits still stronger alignment, particularly with minimum forward segment size and the initial TCP receive-window field.

The representations learned by netFound-640M and NETCAST-48M show the strongest correspondence to several conventional flow descriptors. For netFound-640M, highly aligned features include the initial forward window, maximum packet length, inter-arrival time, and flow duration. NETCAST-48M reaches the largest individual CKA value among the models shown, with a mean CKA of 0.513 for Bwd Packet Length Min. Its top-ranked features span both forward and backward packet-size statistics as well as the initial forward window size.

These results suggest that the learned NETCAST representation retains substantial information associated with classical packet- and flow-level statistics despite operating with a compact 48M-parameter model. Importantly, CKA measures representational similarity rather than direct feature reconstruction or causal dependence. We therefore use this analysis as evidence of correspondence between the learned embedding geometry and established traffic descriptors, rather than as evidence that any individual CICFlowMeter feature is explicitly encoded or solely responsible for downstream predictions.