MMT-VAE: Controllable Multi-Modal VAE for Malicious Encrypted Traffic Generation in 5G Networks

Yongjun Huang[1], Pengfei Du[2], Siting Xu[2], Zehong He[3], and Ruifan Li[1]*
huangyj2022@bupt.edu.cn; 002344@sdupsl.edu.cn; xvshiting@live.com; wouterhe0121@gmail.com;
rfli@bupt.edu.cn
[1]School of Artificial Intelligence,
Beijing University of Posts and Telecommunications, Beijing, China, 100876
[2]Shandong University of Political Science and Law, Jinan, China, 250014
[3]School of Cyberspace Security,
Beijing University of Posts and Telecommunications, Beijing China, 100876

***Abstract*—Encrypted traffic dominates modern 5G mobile core and edge networks and increasingly carries sophisticated attacks hidden behind TLS tunnels and protocol obfuscation. Training robust encrypted-traffic IDS models is hindered by the scarcity of labeled malicious traces and by privacy constraints that limit data sharing. We propose MMT-VAE, a controllable multi-modal generative framework that models what defenders can reliably observe under modern TLS: record/packet sizes and timing plus coarse protocol metadata, rather than assuming ciphertext byte values are informative. Specifically, MMT-VAE encodes (i) a window-level TLS record/packet length histogram (ciphertext lengths), (ii) an inter-packet timing graph, and (iii) protocol metadata into modality-specific latent codes. A condition-dependent hyper-prior aligns these latents under requested conditions (attack type, protocol family, and deployment context), and a cross-attention fusion block enables condition-aware generation. We further introduce a continuous obfuscation-strength control at sampling time to generate progressively harder variants for "what-if" evaluation. On a mixed real and controlled 5G TLS corpus, augmenting downstream detectors with our generated windows improves macro-F1 by up to 6.3 under cross-scenario evaluation.**

Index Terms—Encrypted traffic, multi-modal generative modeling, variational autoencoders, data augmentation, intrusion detection, 5G traffic.

I. Introduction

Encrypted protocols, for example TLS, QUIC, and VPN tunnels dominate Internet traffic. This is especially true in 5G networks. Diverse services and network slicing increase heterogeneity across edge, core, and radio segments [1]–[4]. Encryption protects confidentiality, but it also reduces the usefulness of payload-based inspection. It can therefore conceal behaviors such as C2 beaconing and data exfiltration. As a result, 5G intrusion detection systems often rely on side-channel signals (sizes, timings, directions) and coarse protocol metadata [5], [6].

In addition, real 5G deployments face rapid distribution shift. Protocol stacks evolve, applications update frequently, and obfuscation frameworks intentionally perturb observable statistics. These effects weaken the generalization of supervised detectors trained on limited traces. Data sharing is also constrained by privacy and operational policy. This makes labeled malicious samples scarce, and it limits coverage of attack variants.

We consider an IDS operator who observes only defender-visible features and has no decryption keys. Our goal is *offline* data augmentation and stress testing. We generate trace *skeletons* at the feature level (sizes, timings, and metadata). We do not claim that generated traces are protocol-valid or replayable TLS sessions (Section IV-D).

Existing traffic generators often either ignore long-range timing structure or provide limited controllability. In this work, we propose MMT-VAE, a controllable multi-modal VAE. It jointly models window-level size/length statistics, inter-packet timing graphs, and protocol metadata in a shared conditional latent space. The model supports conditioning on attack type, protocol family, and deployment context. At sampling time, we apply a simple latent-space warping to control obfuscation strength continuously.

There are three main contributions of this work.

- We formulate controllable encrypted-traffic generation for 5G IDS augmentation using only defender-observable features. We focus on size/length statistics, timing structure, and coarse protocol metadata.
- We propose MMT-VAE, a multi-branch conditional VAE with a condition-dependent hyper-prior and cross-attention fusion. It generates samples under specified attack, protocol, and deployment conditions while modeling cross-modal dependencies.
- We introduce a continuous obfuscation-strength control in latent space via warping at sampling time. We show improved cross-scenario IDS generalization when training with generated windows.

II. Related Work

Encrypted-traffic detection has shifted from payload signatures to side-channel features (sizes, timings, and coarse metadata) under TLS/VPN settings. Both classical and deep

*Corresponding author.

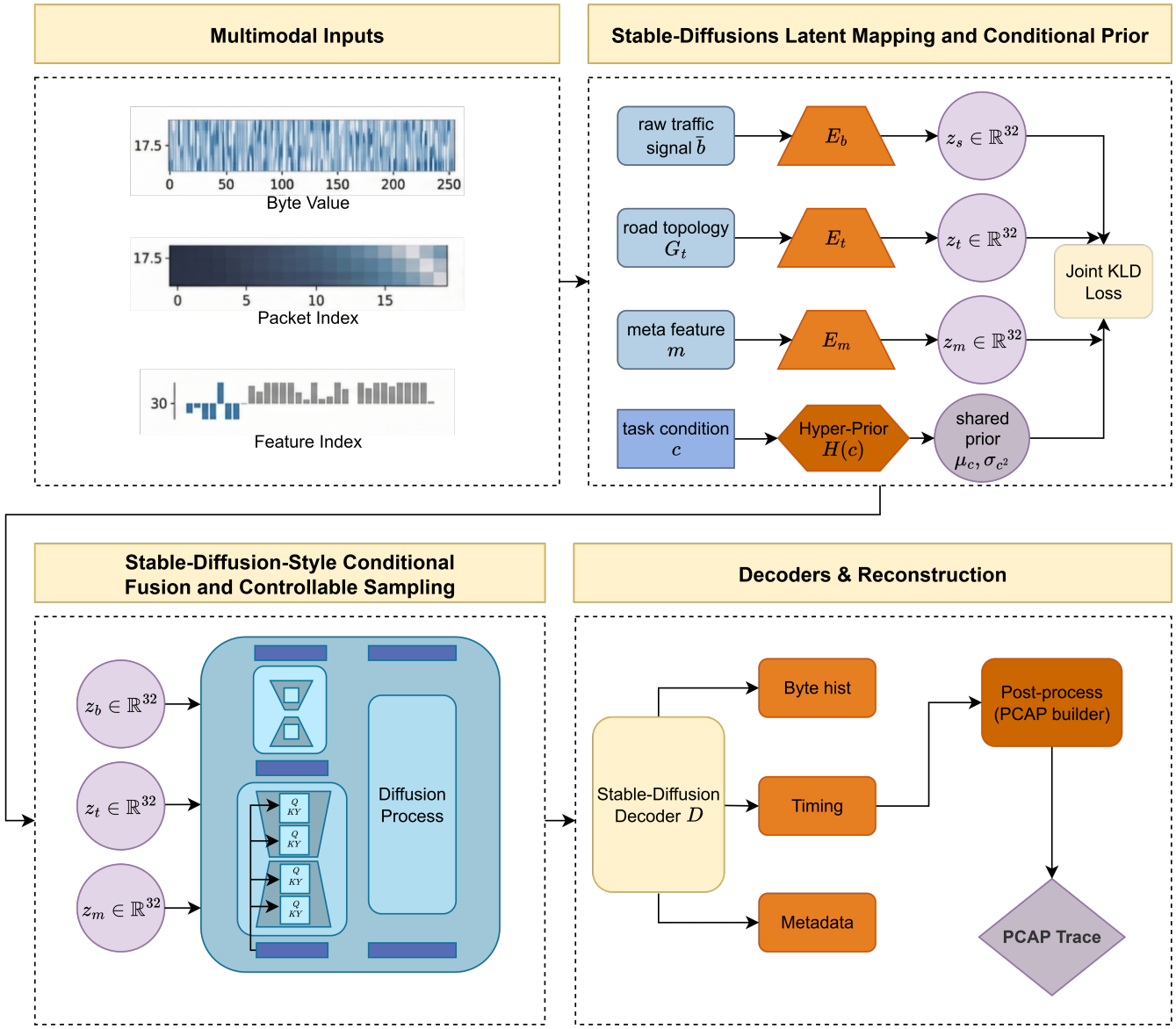

Fig. 1. MMT-VAE architecture of malicious encrypted traffic generation.

models have been studied [7]–[12]. Recent analyses caution against leakage and spurious shortcuts in encrypted-traffic ML. They motivate session-level splitting and avoiding ciphertext-byte assumptions [13]–[15].

On the generation side, prior work explores GAN/VAE/flow-based synthesis over packets or engineered features [16], [17]. Diffusion-based generators are also emerging [18]–[21]. Most existing generators provide limited control over deployment context and obfuscation. Our work targets *controllable* malicious 5G augmentation using only defender-observable size/timing/metadata.

III. Multi-Modal Traffic VAE

We now describe the proposed MMT-VAE. Our MMT-VAE follows a four-stage pipeline, i.e, encoding, alignment, conditional fusion, and decoding. As shown in Figure 1, the entire architecture is composed of four main components. We describe these details as follows.

A. Multi-Modal Inputs

Formally, each 10 s 5G encrypted-traffic window is represented as a multi-modal tuple $x = (\tilde{b}, G_t, m, c)$, where $\tilde{b} \in \mathbb{R}^{256}$ denotes an asinh-normalized length histogram over ciphertext *sizes* (packet lengths and/or TLS ApplicationData record lengths), $G_t = (V, E)$ is a timing graph over up to $N \leq 512$ packets with node features and edge weights

encoding inter-arrival gaps, $m \in \mathbb{R}^{42}$ is a protocol-metadata vector extracted from TLS, TCP, and DNS fields, and c is a condition vector describing the attack type, protocol family, 5G deployment context, and obfuscation level. These modalities capture complementary cues. $\tilde{b}$ summarizes size/length patterns, G_t captures temporal structure, and m encodes coarse protocol context. The condition c specifies what the generator should produce.

a) Ciphertext-length branch: For each 10 s window, we build a 256-dimensional histogram b over ciphertext *lengths* (e.g., packet lengths and/or TLS record lengths) rather than ciphertext byte values. This choice follows the standard assumption of modern TLS 1.3 with AEAD, where ciphertext bytes should be computationally indistinguishable from random, while sizes/timings remain observable and often informative for traffic analysis. To mitigate heavy-tailed distributions, we apply an inverse hyperbolic sine transform as follows,

$$\tilde{b} = \frac{1}{\alpha}\mathrm{asinh}(\alpha b), \tag{1}$$

which behaves logarithmically for large counts while remaining well-defined at zero. Here, the parameter α equals 10.

b) Inter-packet timing branch: We quantize packet arrival times to 1 ms resolution and construct a timing graph $G_t = (V, E)$ over up to 512 packets per window. Each node $v_i \in V$ represents a packet and edges carry weights $e_{i,j} = \Delta t_{i,j}$ encoding inter-packet gaps. We apply a 1-D edge-convolution GCN over G_t to obtain a 128-dimensional timing embedding h_t that summarizes both burstiness and long-range temporal patterns.

c) Protocol-metadata branch: From TLS ClientHello messages, TCP options, and DNS responses we extract a 42-dimensional feature vector m mixing categorical and numerical fields (e.g., cipher suites, ALPN, TLS versions, padding options). Categorical fields are mapped through learnable embeddings and concatenated with normalized numerical features, followed by a two-layer fully connected network that outputs a 64-dimensional embedding h_m.

d) 5G-oriented malicious scenarios: The multi-modal input $(\tilde{b}, G_t, m)$ provides complementary evidence about size patterns, temporal structure, and protocol usage; the condition vector c specifies the requested attack family, protocol context, and obfuscation level.

B. Latent Mapping and Conditional Prior (VAE-style)

a) Private encoders: Following a VAE-style latent autoencoder design (inspired by the autoencoder component used in latent diffusion models), each branch is equipped with a lightweight private encoder: E_b for $\tilde{b}$, E_t for G_t, and E_m for m. These encoders map raw traffic signals into a compact Gaussian latent space and output mean–log-variance pairs as follows,

$$\begin{cases} \{\mu_b, \log\sigma_b^2\} = E_b(\tilde{b}), \\ \{\mu_t, \log\sigma_t^2\} = E_t(G_t), \\ \{\mu_m, \log\sigma_m^2\} = E_m(m), \end{cases} \tag{2}$$

which parameterize Gaussian posteriors over 32-dimensional latent codes $z_b, z_t, z_m \in \mathbb{R}^{32}$ via the reparameterization trick, analogous to the VAE front-end in Stable Diffusion.

b) Shared hyper-prior: To align the three modalities under task conditions while preserving a shared latent-manifold view, we introduce a hyper-prior network H that maps the condition vector c to a shared Gaussian prior as follows,

$$\begin{cases} (\mu_c, \log\sigma_c^2) = H(c), \\ p(z_t \mid c) = \mathcal{N}(\mu_c, \sigma_c^2 I). \end{cases} \tag{3}$$

Intuitively, H plays the role of a condition-dependent latent prior, defining a traffic-aware manifold in latent space; different modalities are encouraged to inhabit this manifold when describing the same (b, G_t, m, c) configuration.

c) Joint KL regularization: As in the VAE component of Stable Diffusion, we regularize each modality-specific posterior toward the hyper-prior via a joint KL loss as follows,

$$\mathcal{L}_{\mathrm{KLD}} = \beta \sum_{t\in\{b,g,m\}} D_{\mathrm{KL}}\big(q(z_t \mid \cdot) \,\|\, p(z_t \mid c)\big), \tag{4}$$

where the parameter β is controlled by cyclical annealing. Specifically, across the first 20 epochs, we ramp β from 0 to 1 three times, which mitigates latent-collapse and encourages rich modality-specific codes before enforcing alignment, mirroring warm-up strategies used in Stable Diffusion training.

C. Condition-Aware Fusion and Controllable Sampling

a) Cross-attention fusion: We concatenate the three latent codes into a 96-dimensional vector $z^0 = [z_b; z_t; z_m]$. To inject conditional information and allow the model to emphasize different modalities under different scenarios, we apply a multi-head cross-attention block (analogous in structure to conditioning blocks used in latent diffusion systems), where c acts as the query and z^0 as key/value, i.e.,

$$z_c = \mathrm{MHA}\big(Q(c), K(z^0), V(z^0)\big), \tag{5}$$

followed by a residual MLP projection back to 32 dimensions. The resulting z_c is a condition-aware latent code that captures a fused view of ciphertext-length, timing, and metadata signals and serves as the input to the shared decoder/generator (our current design does not run iterative denoising steps).

b) Obfuscation-strength control: To support continuous control over obfuscation strength at sampling time, in the spirit of guidance-scale tuning in Stable Diffusion, we apply an affine warping to z_c as,

$$z_s = \gamma(s)\, z_c + \delta(s), \tag{6}$$

where the two functions $\gamma(s)$ and $\delta(s)$ are given as follows,

$$\begin{cases} \gamma(s) = 1 + \lambda s, \\ \delta(s) = \eta s\, \varepsilon, \end{cases} \tag{7}$$

with $\varepsilon \sim \mathcal{N}(0, I)$, $\lambda = 0.3$, and $\eta = 0.1$. Here $s \in [0, 1]$ encodes the desired level of obfuscation. Specifically, $s \approx 0$ keeps samples close to the empirical manifold, while larger s encourages more aggressive mixing and perturbation in latent space, yielding more challenging synthetic traces for detectors.

D. Decoders and Reconstruction

We employ decoders that largely reuse the Stable-Diffusion VAE decoder architecture while adapting output heads to traffic signals. Given z_s, a shared decoder D produces three outputs: reconstructed ciphertext-length histogram $\hat{b}$, timing graph $\hat{G}_t$, and metadata $\hat{m}$.

For the ciphertext-length branch, we apply a softmax over 256 bins and an inverse asinh transform to obtain a valid probability distribution that can be sampled into length bins (and then into packet/record sizes). For timing, the decoder outputs edge weights and node masks; we then apply a minimum-cost flow solver to map continuous edge weights back to discrete packet timestamps that satisfy basic causality constraints. The combination of $\hat{b}$, $\hat{G}_t$, and $\hat{m}$ is post-processed into PCAP-shaped trace skeletons for downstream detectors, thus completing a latent generative pipeline tailored to encrypted traffic.

E. Training Objective

The training objective combines reconstruction fidelity, latent-space regularization, and adversarial enhancement. The reconstruction loss aggregates errors over all three modalities as follows,

$$\mathcal{L}_{\text{rec}} = \|\hat{b} - \tilde{b}\|_1 + \|\hat{G}_t - G_t\|_F + \text{CE}(\hat{m}, m), \tag{8}$$

where $\|\cdot\|_F$ denotes the Frobenius norm over adjacency matrices and CE is cross-entropy over metadata categories plus an ℓ_2 term for numerical fields. The joint Kullback-Leibler (KL) divergence loss $\mathcal{L}_{\text{KL}}$ regularizes modality-specific posteriors toward the hyper-prior and is weighted by a factor λ_{KL}.

To increase coverage of the real latent manifold and make generated traces more challenging for detectors, we introduce a lightweight discriminator D_{lat} operating solely in latent space. D_{lat} receives either fused latents z_c sampled from the model or empirical latents z_{real} obtained by encoding real traffic, and is trained with a standard non-saturating game. The loss is given as follows,

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{z_{\text{real}}}[\log D_{\text{lat}}(z_{\text{real}})] + \mathbb{E}_{z_c}[\log(1 - D_{\text{lat}}(z_c))]. \tag{9}$$

The encoder–decoder pair is jointly optimized to minimize the expectation $\mathbb{E}_{z_c}[\log D_{\text{lat}}(z_c)]$, nudging generated latents toward regions favored by the discriminator.

The complete training loss is therefore give as follows,

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{KL}}\,\mathcal{L}_{\text{KL}} + \lambda_{\text{adv}}\,\mathcal{L}_{\text{adv}}, \tag{10}$$

where we set $\lambda_{\text{KL}} = 1$ and $\lambda_{\text{adv}} = 0.05$ in our experiments. The optimization proceeds with AdamW and cyclical annealing for β as described earlier.

IV. Experimental Settings

A. Dataset and Setup

We evaluate on a 5G-oriented TLS traffic corpus combining real traces captured at 5G core gateways and edge nodes with controlled malicious scenarios generated in a testbed. Benign flows consist of typical 5G user-plane traffic such as web browsing, video streaming, file synchronization, and enterprise VPN access across multiple slices, while malicious flows cover eight attack types including beaconing C2, slow data exfiltration, bulk exfiltration, tunneling, and protocol misuse targeting 5G user and control planes [6], [22].

To reduce leakage risks in windowed evaluation, we split the dataset at the connection/session granularity (5-tuple plus TLS session identifiers when available) and only then segment flows into 10 s windows. This ensures that windows derived from the same underlying connection do not appear in both train and test. We use a 70/15/15 split for training/validation/testing after session-level grouping. We retain windows with at least 20 packets and sample up to $50{,}000$ windows in total.

Feature extraction is performed using standard packet-analysis tooling: we compute per-window ciphertext-length histograms from packet/record sizes, build timing graphs from packet timestamps (1 ms quantization), and extract protocol metadata primarily from observable handshake/header fields (e.g., TLS versions/cipher suites/ALPN when present, TCP options, DNS features). We explicitly exclude any plaintext payload-dependent fields and do not require decryption keys.

a) 5G capture context and vantage points: Our goal is to model encrypted traffic as observed by operators in 5G deployments. We therefore treat packet sizes, timings, directions, and coarse protocol metadata as the primary observables; any payload-dependent signals are excluded by design. We capture traces at operationally relevant vantage points (core/edge) and tag each trace with a coarse deployment context label (core vs. edge and slice category) used for conditioning.

Importantly, 5G packet traces can be observed either before or after user-plane decapsulation (e.g., with or without GTP-U headers) and can exhibit scheduler/buffering effects that distort inter-arrival times. Our feature pipeline is designed to be robust to such differences: when encapsulation is present, we ignore tunnel payload bytes and extract only size/timing/direction plus coarse outer-header metadata; when decapsulated IP traffic is available, we compute the same features on the inner packets. We treat the capture vantage-point category as part of the deployment-context condition.

b) Controlled malicious scenario construction: Malicious traces are generated in a controlled testbed to obtain ground-truth labels (attack family and scenario parameters). For each attack type, we execute scripted scenarios (e.g., beaconing, tunneling, and exfiltration) and record the resulting encrypted flows. We report conditions used for generation as part of the conditioning vector c to support controllable synthesis and to enable cross-scenario evaluation.

Conditions c are derived from labels (attack type or benign), protocol family (TLS-over-HTTP, VPN, DoH, other), 5G deployment context (core vs. edge, slice category), and an obfuscation score estimated from header patterns and timing irregularity. We train MMT-VAE for 100 epochs with batch size 128, latent dimension 32, and β-cycles spanning 20 epochs each.

B. Baselines

We compare our method against three representative generative baselines. These baselines are briefly commented as follows.

- **LenGAN**: a GAN baseline that models window-level ciphertext-length histograms (rather than ciphertext bytes) with implicit timing modeled via simple Poisson processes.
- **PacketVAE**: a packet-level VAE that models packet sizes and inter-arrival times as sequences, without explicit protocol-metadata modeling.
- **CondFlow**: a conditional normalizing flow over hand-crafted traffic features (length distributions, burstiness,

metadata) trained per attack type.

In addition, all baselines are trained on the same windows and conditioned on attack type. Only our method additionally supports protocol family and obfuscation-strength conditioning. We discuss recent diffusion-based benign traffic generators (NetDiff/PacketDiff) in Related Work as the closest methodological neighbors. We do not include them as numeric baselines here because they target different outputs (benign service/flow traces rather than labeled malicious 5G attack windows with protocol-metadata conditioning) and because reproducing them faithfully would require their full preprocessing and labeling pipelines; instead, we align our claims by restricting to defender-observable features and by evaluating cross-scenario generalization under session-level splitting.

To ensure a fair comparison, we use the same feature extraction and windowing strategy for all methods. Each generator produces the same set of observables (length histogram, timing structure, and metadata fields when supported). When a baseline cannot generate a modality, we fill that modality using a simple empirical sampler fitted on the training split, which typically reduces controllability and downstream utility.

C. Evaluation Metrics

We evaluate these methods using three aspects. First, *statistical fidelity* measured via Wasserstein distances between real and generated feature distributions (ciphertext-length histograms, timing statistics, and metadata marginals). Second, *detector utility* measured as the improvement in macro-F1 and AUROC of downstream IDS models when trained with or without generated samples. Third, *controllability*, assessed by how well generated samples match requested attack types, protocol families, and obfuscation levels as judged by pre-trained classifiers.

For W_{byte}, we compute the Wasserstein distance between the 256-bin length histograms averaged over the test set. For W_{time}, we compute the distance on a compact vector of timing statistics (e.g., inter-arrival quantiles and burstiness descriptors) extracted from G_t. For controllability, we train lightweight classifiers on real windows and report the accuracy of predicting the requested condition labels on generated windows (Ctrl Acc).

D. PCAP-shaped traces and validity claims

We clarify what "PCAP-like" means in our setting to avoid misleading claims about replayability. Our post-processing produces PCAP-shaped trace skeletons that preserve packet sizes, directions, timestamps, and selected metadata fields required by common feature-extraction pipelines. These traces are intended for offline IDS training/evaluation and distributional analysis. We do not claim that the generated traces constitute protocol-valid, replayable TLS sessions: generating consistent TCP/TLS state machines, record semantics, retransmissions, congestion control, and ACK/SEQ dynamics is out of scope and is studied in complementary work focused on online constraints (e.g., Mirage). [21]

In addition, for detector utility, we train a gradient-boosted tree and a lightweight 1D-CNN over flow-level features, evaluating under cross-scenario conditions where attack families or obfuscation schemes in the test set are partially unseen during training. Both detectors are trained on the same feature pipeline used for generation. For augmentation, we add synthetic windows with a 1:1 ratio per class and re-train the detector from scratch. We report macro-F1 on the held-out test split.

TABLE I
COMPARISON OF ENCRYPTED-TRAFFIC GENERATORS ON STATISTICAL FIDELITY (W_{BYTE}, W_{TIME}) AND DETECTOR UTILITY (ΔF1) WITH A 1:1 SYNTHETIC-TO-REAL RATIO.

Generator	W_{byte}	W_{time}	ΔF1 (**GBT**)	ΔF1 (**CNN**)
LenGAN	0.124	0.137	$+0.3$	-0.2
PacketVAE	0.109	0.121	$+0.8$	$+0.5$
CondFlow	0.096	0.104	$+2.1$	$+1.7$
MMT-VAE (ours)	**0.071**	**0.079**	**+4.1**	**+6.3**

V. Results and Analysis

A. Main Results

Across all 5G scenarios considered, including core-network enterprise slices, consumer broadband slices, and mixed IoT deployments, MMT-VAE consistently achieves the lowest Wasserstein distances on both ciphertext-length and timing distributions, indicating superior statistical fidelity of the generated traffic compared with the baselines. In particular, the gap in W_{time} suggests that jointly modeling timing graphs and metadata helps MMT-VAE better capture burstiness and scheduling-induced variability that are characteristic of 5G user-plane traffic.

When augmenting detectors with generated traffic at a 1:1 synthetic-to-real ratio, our framework improves macro-F1 by 4.1–6.3 points under cross-scenario evaluation, whereas LenGAN and PacketVAE often provide negligible gains or even hurt performance due to distributional artifacts that bias the decision boundary toward unrealistic samples. CondFlow offers moderate improvements but lacks fine-grained control over obfuscation strength and protocol context, which limits its usefulness for targeted stress-testing of specific slices or attack families.

Table I summarizes the overall trend: stronger statistical fidelity generally correlates with better detector utility, but high fidelity alone is not sufficient. Methods that ignore protocol context or cross-modal dependencies can match marginal distributions yet still introduce unrealistic joint patterns (e.g., a size burst with implausible timing), which may degrade generalization. In contrast, conditioning on deployment context and protocol family helps us generate harder yet plausible windows for cross-scenario evaluation.

B. Runtime and Limitations

Sampling a window requires a single encoder–fusion–decoder forward pass (no iterative diffusion sampling); the dominant cost is graph encoding/decoding over up to $N \leq 512$ packets. Our PCAP-shaped outputs are intended for offline IDS pipelines and do not guarantee replayable TLS sessions (Section IV-D). The runtime scales approximately linearly with the number of packets per window and the number of edges

TABLE II
ABLATION RESULTS ON THE 5G TLS CORPUS (1:1 SYNTHETIC-TO-REAL). LOWER $W_{\text{BYTE}}/W_{\text{TIME}}$ IS BETTER; ΔF1 AND CTRL ACC ARE HIGHER IS BETTER.

Variant	W_{byte}	W_{time}	ΔF1 (**GBT**)	ΔF1 (**CNN**)	**Ctrl Acc**
Full model	**0.071**	**0.079**	**+4.1**	**+6.3**	**0.92**
NoHyperPrior	0.083	0.095	+3.0	+4.6	0.90
NoCrossAttn	0.091	0.106	+2.3	+3.4	0.82
NoStrengthCtrl	0.076	0.084	+3.0	+4.2	0.75
NoLatentAdv	0.087	0.101	+2.6	+3.7	0.88

used in the timing graph. In practice, our model is suitable for offline generation of large synthetic corpora, but it is not designed as an online packet-by-packet simulator. Another limitation is that we generate window-level skeletons rather than end-to-end connections. This design matches common encrypted-traffic IDS pipelines, but it cannot enforce long-horizon session semantics beyond the window boundary.

We also observe that tuning the obfuscation-strength parameter s enables progressively more challenging samples: small s stays close to real traffic distributions, while larger s increases detector error rates without obviously corrupting traces, supporting systematic "what-if" stress testing in 5G deployments.

C. Ablation Studies

We conduct ablations to quantify the contribution of each design element. We keep the dataset, optimization hyperparameters, and evaluation protocol unchanged. Each variant removes exactly one component from the full model. **NoHyperPrior** replaces the condition-aware hyper-prior $H(c)$ with a standard Gaussian. **NoCrossAttn** removes cross-attention and uses a linear projection of $[z_b; z_t; z_m]$. **NoStrengthCtrl** fixes the obfuscation strength to $s = 0$ and removes the warping in Eq. (7). **NoLatentAdv** drops the latent discriminator D_{lat}.

Table II indicates that each component contributes to a different aspect of the objective. The hyper-prior improves both W_{byte} and W_{time}, which suggests that condition-aware alignment helps the three modality encoders agree on a shared manifold. Removing cross-attention hurts all metrics most, and it reduces detector gains substantially. This is consistent with the need to model cross-modal dependencies under protocol and slice context.

The strength controller has a small effect on Wasserstein distances but a large effect on Ctrl Acc. This confirms that the warping mechanism is the primary handle for continuous control at sampling time. Finally, latent adversarial regularization improves fidelity and detector utility. It likely discourages over-smoothed generations by pulling sampled latents toward the empirical latent support.

VI. CONCLUSION AND FUTURE WORK

We presented a controllable multi-modal VAE framework for malicious encrypted traffic generation rooted in a Multi-Modal Traffic VAE. By jointly modeling ciphertext-length statistics, inter-packet timing graphs, and protocol metadata in a shared conditional latent space and introducing controllable warping and latent adversarial regularization, our approach generates realistic yet diverse encrypted trace skeletons tailored to specific attack and protocol conditions. Experiments show that detectors trained with our generated traffic generalize better to unseen obfuscation strategies and attack scenarios.

Future work includes integrating full latent diffusion dynamics (iterative denoising in latent space) on top of MMT-VAE and extending the framework to multi-hop flow graphs in large-scale enterprise networks.

ACKNOWLEDGMENT

This work was supported in part by National Key Research and Development Program of China (No. 2023YFC3305902), in part by Special Project for Industrial Foundation Reconstruction and High-Quality Development of Manufacturing under Grant Agreement No. ZC25T320057/100, in part by Key Research and Development and Achievement Transformation Program of Inner Mongolia Autonomous Region (No. 2026SYFHH0570), in part by Industry-University-Research Innovation Fund for Chinese Universities No. 2024MZ028, and in part by BUPT Kunpeng and Ascend Center of Cultivation The authors would like to thank the anonymous editors and reviewers for their valuable comments on improving the final version of this paper.

REFERENCES

[1] J. G. Andrews *et al.*, "What will 5g be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.
[2] F. Boccardi *et al.*, "Five disruptive technology directions for 5g," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, 2014.
[3] G. Arfaoui *et al.*, "Security for 5g and beyond," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 454–481, 2020.
[4] S. Ayoubi *et al.*, "Machine learning for 5g mobile networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2793–2834, 2019.
[5] T. Nguyen *et al.*, "A survey of encrypted traffic classification: Approaches and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2658–2679, 2019.
[6] L. Zhang *et al.*, "Deep learning based 5g network traffic analysis and security: A survey," *IEEE Access*, vol. 7, pp. 129 868–129 881, 2019.
[7] M. Roesch, "Snort: Lightweight intrusion detection for networks," in *Proceedings of LISA'99*, 1999, pp. 229–238.
[8] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proceedings of ICDM 2008*, 2008, pp. 413–422.
[9] B. Schölkopf *et al.*, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.
[10] H. Kim, J. Park, and S. Lee, "Encrypted traffic classification using deep neural networks," in *Proceedings of IEEE CNS 2015*, 2015, pp. 1–9.
[11] G. Aceto *et al.*, "Mobile encrypted traffic classification using deep learning," *Computer Networks*, vol. 165, p. 106944, 2019.
[12] Y. Mirsky *et al.*, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in *Proceedings of NDSS 2018*, 2018.
[13] Y. Huang, P. Du, R. Li, X. Li, and L. Li, "Met-llm: Enhancing large language models for malicious encrypted traffic detection," *Expert Systems with Applications*, vol. 303, p. 130621, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417425042368
[14] N. Wickramasinghe, A. Shaghaghi, G. Tsudik, and S. Jha, "Sok: Decoding the enigma of encrypted network traffic classifiers," in *2025 IEEE Symposium on Security and Privacy (SP)*, 2025, pp. 1825–1843.
[15] Y. Zhao, G. Dettori, M. Boffa, L. Vassio, and M. Mellia, "The sweet danger of sugar: Debunking representation learning for encrypted traffic classification," in *Proceedings of the ACM SIGCOMM 2025 Conference*, ser. SIGCOMM '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 296–310. [Online]. Available: https://doi.org/10.1145/3718958.3750498
[16] M. Shafiq and J. Zhang, "Flowgan: Synthetic network traffic generation with generative adversarial networks," in *Proceedings of IEEE ICC 2018*, 2018, pp. 1–6.

[17] Z. Gao, Y. Li, and X. Chen, "Netvae: A deep generative model for network traffic," in *Proceedings of IEEE INFOCOM WKSHPS 2020*, 2020, pp. 1–6.
[18] S. Zhang, T. Li, D. Jin, and Y. Li, "Netdiff: a service-guided hierarchical diffusion model for network flow trace generation," *Proceedings of the ACM on Networking*, vol. 2, no. CoNEXT3, pp. 1–21, 2024.
[19] S. Zhang, H. Chai, Y. Li, B. Qiu, L. Yue, and R. Pan, "Packetdiff: A flow guided diffusion model for network packet trace generation," *IEEE Internet of Things Journal*, 2025.
[20] A. Bin Jasni, A. Manada, and K. Watabe, "Diffupac: Contextual mimicry in adversarial packets generation via diffusion model," *Advances in Neural Information Processing Systems*, vol. 37, pp. 133 846–133 878, 2024.
[21] W. Li, J. Chen, Z. Li, S. Wang, H. Jin, and X.-Y. Zhang, "Mirage: Evading online network traffic analyzers via deep reinforcement learning," SSRN, 2024, available at SSRN: 5090063. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5090063
[22] W. Li *et al.*, "Encrypted traffic analysis and intrusion detection in 5g networks," *Computer Communications*, vol. 176, pp. 90–103, 2021.