Mask-DDPM: Transformer-Conditioned Mixed-Type Diffusion for Semantically Valid ICS Telemetry Synthesis

Zhenglan Chen[1], Mingzhe Yang[1], Hongyu Yan[1], and Huan Yang[2]

[1] Aberdeen Institute of Data Science and Artificial Intelligence, South China Normal University, Guangzhou, Guangdong 510631, China
{20223803054,20223803063,20223803065}@m.scnu.edu.cn
[2] School of Artificial Intelligence, South China Normal University, Guangzhou, Guangdong 510631, China
huan.yang@m.scnu.edu.cn

Abstract. Industrial control systems (ICS) security research is increasingly constrained by the scarcity and limited shareability of realistic communication traces and process measurements, especially for attack scenarios. To mitigate this bottleneck, we study synthetic generation at the protocol-feature and process-signal level, where samples must simultaneously preserve temporal coherence, match continuous marginal distributions, and keep discrete supervisory variables strictly within valid vocabularies. We propose Mask-DDPM, a hybrid framework tailored to mixed-type, multi-scale ICS sequences. Mask-DDPM factorizes generation into (i) a causal Transformer trend module that rolls out a stable long-range temporal scaffold for continuous channels, (ii) a trend-conditioned residual DDPM that refines local stochastic structure and heavy-tailed fluctuations without degrading global dynamics, (iii) a masked (absorbing) diffusion branch for discrete variables that guarantees valid symbol generation by construction, and (iv) a type-aware decomposition/routing layer that aligns modeling mechanisms with heterogeneous ICS variable origins and enforces deterministic reconstruction where appropriate. Evaluated on fixed-length windows ($L = 96$) derived from the HAI Security Dataset, Mask-DDPM achieves stable fidelity across seeds with mean KS $= 0.3311 \pm 0.0079$ (continuous), mean JSD $= 0.0284 \pm 0.0073$ (discrete), and mean absolute lag-1 autocorrelation difference $= 0.2684 \pm 0.0027$, indicating faithful marginals, preserved short-horizon dynamics, and valid discrete semantics. The resulting generator provides a reproducible basis for data augmentation, benchmarking, and downstream ICS protocol reconstruction workflows.

Keywords: Machine Learning · Cyber Defense · ICS

1 Introduction

Industrial control systems (ICS) form the backbone of modern critical infrastructure, which includes power grids, water treatment, manufacturing, and transportation, among others. These systems monitor, regulate, and automate physical

processes through sensors, actuators, programmable logic controllers (PLCs), and monitoring software. Unlike conventional IT systems, ICS operate in real time, closely coupled with physical processes and safety-critical constraints, using heterogeneous and legacy communication protocols such as Modbus/TCP and DNP3 that were not originally designed with robust security in mind. This architectural complexity and operational criticality make ICS high-impact targets for cyber attacks, where disruptions can result in physical damage, environmental harm, and even loss of life. Recent reviews of ICS security highlight the expanding attack surface due to increased connectivity, vulnerabilities in legacy systems, and the inadequacy of traditional security controls in capturing the nuances of ICS networks and protocols [13, 24]

While machine learning (ML) techniques have shown promise for anomaly detection and automated cybersecurity within ICS, they rely heavily on labeled datasets that capture both benign operations and diverse attack patterns. In practice, real ICS traffic data, especially attack-triggered captures, are scarce due to confidentiality, safety, and legal restrictions, and available public ICS datasets are few, limited in scope, or fail to reflect current threat modalities. For instance, the HAI Security Dataset provides operational telemetry and anomaly flags from a realistic control system setup for research purposes, but must be carefully preprocessed to derive protocol-relevant features for ML tasks [33]. Data scarcity directly undermines model generalization, evaluation reproducibility, and the robustness of intrusion detection research, especially when training or testing ML models on realistic ICS behavior remains confined to small or outdated collections of examples [2].

Synthetic data generation offers a practical pathway to mitigate these challenges. By programmatically generating feature-level sequences that mimic the statistical and temporal structure of real ICS telemetry, researchers can augment scarce training sets, standardize benchmarking, and preserve operational confidentiality. Relative to raw packet captures, feature-level synthesis abstracts critical protocol semantics and statistical patterns without exposing sensitive fields, making it more compatible with safety constraints and compliance requirements in ICS environments. Modern generative modeling, including diffusion models, has advanced significantly in producing high-fidelity synthetic data across domains. Diffusion approaches, such as denoising diffusion probabilistic models, learn to transform noise into coherent structured samples and have been successfully applied to tabular or time series data synthesis with better stability and data coverage compared to adversarial methods [16, 27]

Despite these advances, most existing work either focuses on packet-level generation [12] or is limited to generic tabular data [16], rather than domain-specific control sequence synthesis tailored for ICS protocols where temporal coherence, multi-channel dependencies, and discrete protocol legality are jointly required. This gap motivates our focus on protocol feature-level generation for ICS, which involves synthesizing sequences of protocol-relevant fields conditioned on their temporal and cross-channel structure. In this work, we formulate a hybrid modeling pipeline that decouples long-horizon trends and local statistical detail while

preserving discrete semantics of protocol tokens. By combining causal Transformers with diffusion-based refiners, and enforcing deterministic validity constraints during sampling, our framework generates semantically coherent, temporally consistent, and distributionally faithful ICS feature sequences. We evaluate features derived from the HAI Security Dataset and demonstrate that our approach produces high-quality synthetic sequences suitable for downstream augmentation, benchmarking, and integration into packet-reconstruction workflows that respect realistic ICS constraints.

2 Related Work

Early generation of network data oriented towards "realism" mostly remained at the packet/flow header level, either through replay or statistical synthesis based on single-point observations. Swing, in a closed-loop, network-responsive manner, extracts user/application/network distributions from single-point observations to reproduce burstiness and correlation across multiple time scales [40, 41]. Subsequently, a series of works advanced header synthesis to learning-based generation: the WGAN-based method added explicit verification of protocol field consistency to NetFlow/IPFIX [28], NetShare reconstructed header modeling as flow-level time series and improved fidelity and scalability through domain encoding and parallel fine-tuning [45], and DoppelGANger preserved the long-range structure and downstream sorting consistency of networked time series by decoupling attributes from sequences [19]. However, in industrial control system (ICS) scenarios, the original PCAP is usually not shareable, and public testbeds (such as SWaT, WADI) mostly provide process/monitoring telemetry and protocol interactions for security assessment, but public datasets emphasize operational variables rather than packet-level traces [22, 1]. This makes "synthesis at the feature/telemetry level, aware of protocol and semantics" more feasible and necessary in practice: we are more concerned with reproducing high-level distributions and multi-scale temporal patterns according to operational semantics and physical constraints without relying on the original packets. From this perspective, the generation paradigm naturally shifts from "packet syntax reproduction" to "modeling of high-level spatio-temporal distributions and uncertainties", requiring stable training, strong distribution fitting, and interpretable uncertainty characterization.

Diffusion models exhibit good fit along this path: DDPM achieves high-quality sampling and stable optimization through efficient ϵ parameterization and weighted variational objectives [9], the SDE perspective unifies score-based and diffusion, providing likelihood evaluation and prediction-correction sampling strategies based on probability flow ODEs [35]. For time series, TimeGrad replaces the constrained output distribution with conditional denoising, capturing high-dimensional correlations at each step [27]; CSDI explicitly performs conditional diffusion and uses two-dimensional attention to simultaneously leverage temporal and cross-feature dependencies, suitable for conditioning and filling in missing values [38]; in a more general spatio-temporal structure, DiffSTG generalizes

diffusion to spatio-temporal graphs, combining TCN/GCN with denoising U-Net to improve CRPS and inference efficiency in a non-autoregressive manner [42], and PriSTI further enhances conditional features and geographical relationships, maintaining robustness under high missing rates and sensor failures [20]; in long sequences and continuous domains, DiffWave verifies that diffusion can also match the quality of strong vocoders under non-autoregressive fast synthesis [15]; studies on cellular communication traffic show that diffusion can recover spatio-temporal patterns and provide uncertainty characterization at the urban scale [21]. These results overall point to a conclusion: when the research focus is on "telemetry/high-level features" rather than raw messages, diffusion models provide stable and fine-grained distribution fitting and uncertainty quantification, which is exactly in line with the requirements of ICS telemetry synthesis. Meanwhile, directly entrusting all structures to a "monolithic diffusion" is not advisable: long-range temporal skeletons and fine-grained marginal distributions often have optimization tensions, requiring explicit decoupling in modeling.

Looking further into the mechanism complexity of ICS: its channel types are inherently mixed, containing both continuous process trajectories and discrete supervision/status variables, and discrete channels must be "legal" under operational constraints. The aforementioned progress in time series diffusion has mainly occurred in continuous spaces, but discrete diffusion has also developed systematic methods: D3PM improves sampling quality and likelihood through absorption/masking and structured transitions in discrete state spaces [4], subsequent masked diffusion provides stable reconstruction on categorical data in a more simplified form [19], multinomial diffusion directly defines diffusion on a finite vocabulary through mechanisms such as argmax flows [11], and Diffusion-LM demonstrates an effective path for controllable text generation by imposing gradient constraints in continuous latent spaces [17]. From the perspectives of protocols and finite-state machines, coverage-guided fuzz testing emphasizes the criticality of "sequence legality and state coverage" [23, 7, 30], echoing the concept of "legality by construction" in discrete diffusion: preferentially adopting absorption/masking diffusion on discrete channels, supplemented by type-aware conditioning and sampling constraints, to avoid semantic invalidity and marginal distortion caused by post hoc thresholding.

From the perspective of high-level synthesis, the temporal structure is equally indispensable: ICS control often involves delay effects, phased operating conditions, and cross-channel coupling, requiring models to be able to characterize low-frequency, long-range dependencies while also overlaying multi-facated fine-grained fluctuations on them. The Transformer series has provided sufficient evidence in long-sequence time series tasks: Transformer-XL breaks through the fixed-length context limitation through a reusable memory mechanism and significantly enhances long-range dependency expression [6]; Informer uses ProbSparse attention and efficient decoding to balance span and efficiency in long-sequence prediction [48]; Autoformer robustly models long-term seasonality and trends through autocorrelation and decomposition mechanisms [43]; FEDformer further improves long-period prediction performance in frequency domain enhancement

and decomposition [49]; PatchTST enhances the stability and generalization of long-sequence multivariate prediction through local patch-based representation and channel-independent modeling [26]. Combining our previous positioning of diffusion, this chain of evidence points to a natural division of labor: using attention-based sequence models to first extract stable low-frequency trends/conditions (long-range skeletons), and then allowing diffusion to focus on margins and details in the residual space; meanwhile, discrete masking/absorbing diffusion is applied to supervised/pattern variables to ensure vocabulary legality by construction. This design not only inherits the advantages of time series diffusion in distribution fitting and uncertainty characterization [27, 38, 42, 20, 15, 21], but also stabilizes the macroscopic temporal support through the long-range attention of Transformer, enabling the formation of an operational integrated generation pipeline under the mixed types and multi-scale dynamics of ICS.

3 Methodology

Industrial control system (ICS) telemetry is intrinsically mixed-type and mechanistically heterogeneous: continuous process trajectories (e.g., sensor and actuator signals) coexist with discrete supervisory states (e.g., modes, alarms, interlocks), and the underlying generating mechanisms range from physical inertia to program-driven step logic. This heterogeneity is not cosmetic: it directly affects what realistic synthesis means, because a generator must jointly satisfy (i) temporal coherence, (ii) distributional fidelity, and (iii) discrete semantic validity (i.e., every discrete output must belong to its legal vocabulary by construction). These properties are emphasized broadly in operational-technology security guidance and ICS engineering practice, where state logic and physical dynamics are tightly coupled [25].

We model each training instance as a fixed-length window of length L, comprising continuous channels $\boldsymbol{X} \in \mathbb{R}^{L \times d_c}$ and discrete channels $\boldsymbol{Y} = \{y_{1:L}^{(j)}\}_{j=1}^{d_d}$, where each discrete variable satisfies $y_t^{(j)} \in \mathcal{V}_j$ for a finite vocabulary $\mathcal{V}_j$. Our objective is to learn a generator that produces synthetic $(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}})$ that are simultaneously coherent and distributionally faithful, while also ensuring $\hat{y}_t^{(j)} \in \mathcal{V}_j$ for all j, t by construction (rather than via post-hoc rounding or thresholding).

A key empirical and methodological tension in ICS synthesis is that temporal realism and marginal/distributional realism can compete when optimized monolithically: sequence models trained primarily for regression often over-smooth heavy tails and intermittent bursts, while purely distribution-matching objectives can erode long-range structure. Diffusion models provide a principled route to rich distribution modeling through iterative denoising, but they do not, by themselves, resolve (i) the need for a stable low-frequency temporal scaffold, nor (ii) the discrete legality constraints for supervisory variables [10, 36]. Recent time-series diffusion work further suggests that separating coarse structure from stochastic refinement can be an effective inductive bias for long-horizon realism [14, 34]. Figure 1 summarizes how our framework maps these requirements into a staged generator for mixed-type ICS telemetry.

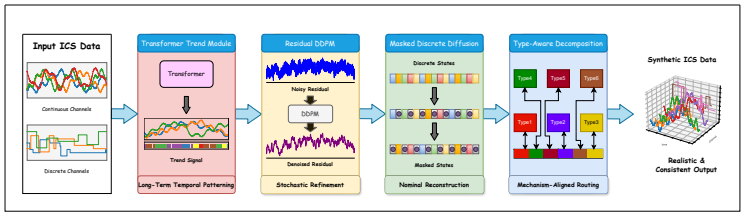

Fig. 1. Masked-DDPM: Unified Synthesis for ICS traffic

Motivated by these considerations, we propose Mask-DDPM, organized in the following order:

1. Transformer trend module: learns the dominant temporal backbone of continuous dynamics via attention-based sequence modeling [39].
2. Residual DDPM for continuous variables: models distributional detail as stochastic residual structure conditioned on the learned trend [10, 14].
3. Masked diffusion for discrete variables: generates discrete ICS states with an absorbing/masking corruption process and categorical reconstruction [3, 31].
4. Type-aware decomposition: a type-aware factorization and routing layer that assigns variables to the most appropriate modeling mechanism and enforces deterministic constraints where warranted.

This ordering is intentional. The trend module establishes a macro-temporal scaffold; residual diffusion then concentrates capacity on micro-structure and marginal fidelity; masked diffusion provides a native mechanism for discrete legality; and the type-aware layer operationalizes the observation that not all ICS variables should be modeled with the same stochastic mechanism. As shown in Figure 1, these components are arranged sequentially so that temporal scaffolding, residual refinement, and discrete legality are enforced in complementary rather than competing stages. Importantly, while diffusion-based generation for ICS telemetry has begun to emerge, existing approaches remain limited and typically emphasize continuous synthesis or augmentation; in contrast, our pipeline integrates (i) a Transformer-conditioned residual diffusion backbone, (ii) a discrete masked-diffusion branch, and (iii) explicit type-aware routing for heterogeneous variable mechanisms within a single coherent generator [47, 29].

3.1 Transformer trend module for continuous dynamics

We instantiate the temporal backbone as a causal Transformer trend extractor, leveraging self-attention's ability to represent long-range dependencies and cross-channel interactions without recurrence [39]. Compared with recurrent trend extractors (e.g., GRU-style backbones), a Transformer trend module offers a direct mechanism to model delayed effects and multivariate coupling common in ICS, where control actions may influence downstream sensors with nontrivial lags and regime-dependent propagation [39, 25]. Crucially, in our design the Transformer is

not asked to be the entire generator; instead, it serves a deliberately restricted role: providing a stable, temporally coherent conditioning signal that later stochastic components refine.

For continuous channels $\boldsymbol{X}$, we posit an additive decomposition:

$$\boldsymbol{X} = \boldsymbol{S} + \boldsymbol{R}, \tag{1}$$

where $\boldsymbol{S} \in \mathbb{R}^{L \times d_c}$ is a smooth trend capturing predictable temporal evolution, and $\boldsymbol{R} \in \mathbb{R}^{L \times d_c}$ is a residual capturing distributional detail (e.g., bursts, heavy tails, local fluctuations) that is difficult to represent robustly with a purely regression-based temporal objective. This separation reflects an explicit division of labor: the trend module prioritizes temporal coherence, while diffusion (introduced next) targets distributional realism at the residual level, a strategy aligned with predict-then-refine perspectives in time-series diffusion modeling [14, 34].

We parameterize the trend $\boldsymbol{S}$ using a causal Transformer f_ϕ. With teacher forcing, we train F_ϕ to predict the next-step trend from past observations:

$$\hat{\boldsymbol{S}}_{t+1} = f_\phi(\boldsymbol{X}_{1:t}), \quad t = 1, \ldots, L-1. \tag{2}$$

using the mean-squared error objective:

$$\mathcal{L}_{\text{trend}}(\phi) = \frac{1}{(L-1)d_c} \sum_{t=1}^{L-1} \left\| \hat{\boldsymbol{S}}_{t+1} - \boldsymbol{X}_{t+1} \right\|_2^2. \tag{3}$$

At inference, we roll out the Transformer autoregressively to obtain $\hat{\boldsymbol{S}}$, and then define the residual target for diffusion as $\boldsymbol{R} = \boldsymbol{X} - \hat{\boldsymbol{S}}$. This setup intentionally locks in a coherent low-frequency scaffold before any stochastic refinement is applied, thereby reducing the burden on downstream diffusion modules to simultaneously learn both long-range structure and marginal detail. In this sense, our use of Transformers is distinctive: it is a conditioning-first temporal backbone designed to stabilize mixed-type diffusion synthesis in ICS, rather than an end-to-end monolithic generator [39, 14, 47].

3.2 DDPM for continuous residual generation

We model the residual $\boldsymbol{R}$ with a denoising diffusion probabilistic model (DDPM) conditioned on the trend $\hat{\boldsymbol{S}}$ [10]. Diffusion models learn complex data distributions by inverting a tractable noising process through iterative denoising, and have proven effective at capturing multimodality and heavy-tailed structure that is often attenuated by purely regression-based sequence models [10, 36]. Conditioning the diffusion model on $\hat{\boldsymbol{S}}$ is central: it prevents the denoiser from re-learning the low-frequency scaffold and focuses capacity on residual micro-structure, mirroring the broader principle that diffusion excels as a distributional corrector when a reasonable coarse structure is available [14, 34].

Let $\boldsymbol{K}$ denote the number of diffusion steps, with a noise schedule $\{\beta_k\}_{k=1}^K$, $\alpha_k = 1 - \beta_k$, and $\bar{\alpha}_k = \prod_{i=1}^k \alpha_i$. The forward corruption process is:

$$q(\boldsymbol{r}_k \mid \boldsymbol{r}_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_k}\, \boldsymbol{r}_0,\ (1 - \bar{\alpha}_k)\mathbf{I}\big) \tag{4}$$

equivalently,

$$\boldsymbol{r}_k = \sqrt{\bar{\alpha}_k}\,\boldsymbol{r}_0 + \sqrt{1-\bar{\alpha}_k}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{5}$$

The learned reverse process is parameterized as:

$$p_\theta(\boldsymbol{r}_{k-1} \mid \boldsymbol{r}_k, \hat{\boldsymbol{S}}) = \mathcal{N}\big(\boldsymbol{\mu}_\theta(\boldsymbol{r}_k, k, \hat{\boldsymbol{S}}),\ \boldsymbol{\Sigma}(k)\big). \tag{6}$$

where μ_θ is implemented by a Transformer denoiser that consumes (i) the noised residual r_k, (ii) a timestep embedding for k, and (iii) conditioning features derived from $\hat{\boldsymbol{S}}$. This denoiser architecture is consistent with the growing use of attention-based denoisers for long-context time-series diffusion, while our key methodological emphasis is the trend-conditioned residual factorization as the object of diffusion learning [10, 34].

We train the denoiser using the standard DDPM ϵ-prediction objective:

$$\mathcal{L}_{\text{cont}}(\theta) = \mathbb{E}_{k,\boldsymbol{r}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{r}_k, k, \hat{\boldsymbol{S}})\right\|_2^2\right]. \tag{7}$$

Because diffusion optimization can exhibit timestep imbalance (i.e., some timesteps dominate gradients), we optionally apply an SNR-based reweighting consistent with Min-SNR training:

$$\mathcal{L}_{\text{cont}}^{\text{snr}}(\theta) = \mathbb{E}_{k,\boldsymbol{r}_0,\boldsymbol{\epsilon}}\left[w_k\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{r}_k, k, \hat{\boldsymbol{S}})\right\|_2^2\right], \tag{8}$$

where $\mathrm{SNR}_k = \bar{\alpha}_k/(1-\bar{\alpha}_k)$ and $\gamma > 0$ is a cap parameter [8].

After sampling $\hat{\boldsymbol{R}}$ by reverse diffusion, we reconstruct the continuous output as $\hat{\boldsymbol{X}} = \hat{\boldsymbol{S}} + \hat{\boldsymbol{R}}$. Overall, the DDPM component serves as a distributional corrector on top of a temporally coherent backbone, which is particularly suited to ICS where low-frequency dynamics are strong and persistent but fine-scale variability (including bursts and regime-conditioned noise) remains important for realism. Relative to prior ICS diffusion efforts that primarily focus on continuous augmentation, our formulation elevates trend-conditioned residual diffusion as a modular mechanism for disentangling temporal structure from distributional refinement [47, 29].

3.3 Masked diffusion for discrete ICS variables

Discrete ICS variables must remain categorical, making Gaussian diffusion inappropriate for supervisory states and mode-like channels. While one can attempt continuous relaxations or post-hoc discretization, such strategies risk producing semantically invalid intermediate states (e.g., in-between modes) and can distort the discrete marginal distribution. Discrete-state diffusion provides a principled alternative by defining a valid corruption process directly on categorical variables [3, 31]. In the ICS setting, this is not a secondary detail: supervisory tags often encode control logic boundaries (modes, alarms, interlocks) that must remain within a finite vocabulary to preserve semantic correctness [25].

We therefore adopt masked (absorbing) diffusion for discrete channels, where corruption replaces tokens with a special `[MASK]` symbol according to a schedule

[31]. For each variable j, define a masking schedule $\{m_k\}_{k=1}^{K}$ (with $m_k \in [0,1]$) increasing in k. The forward corruption process is:

$$q(y_k^{(j)} \mid y_0^{(j)}) = \begin{cases} y_0^{(j)}, & \text{with probability } 1 - m_k, \\ \texttt{[MASK]}, & \text{with probability } m_k, \end{cases} \tag{9}$$

applied independently across j and t. Let $\mathcal{M}$ denote the set of masked positions at step k. The denoiser h_ψ predicts a categorical distribution over $\mathcal{V}_j$ for each masked token, conditioned on (i) the corrupted discrete sequence, (ii) the diffusion step k, and (iii) continuous context. Concretely, we condition on $\hat{\boldsymbol{S}}$ and $\hat{\boldsymbol{X}}$ to couple supervisory reconstruction to the underlying continuous dynamics:

$$p_\psi\big(y_0^{(j)} \mid y_k, k, \hat{\boldsymbol{S}}, \hat{\boldsymbol{X}}\big) = h_\psi(y_k, k, \hat{\boldsymbol{S}}, \hat{\boldsymbol{X}}). \tag{10}$$

This conditioning choice is motivated by the fact that many discrete ICS states are not standalone, they are functions of regimes, thresholds, and procedural phases that manifest in continuous channels [25]. Training uses a categorical denoising objective:

$$\mathcal{L}_{\text{disc}}(\psi) = \mathbb{E}_k \left[\frac{1}{|\mathcal{M}|} \sum_{(j,t)\in\mathcal{M}} \text{CE}\big(h_\psi(y_k, k, \hat{\boldsymbol{S}}, \hat{\boldsymbol{X}})_{j,t},\ y_{0,t}^{(j)}\big) \right], \tag{11}$$

where $\text{CE}(\cdot,\cdot)$ is cross-entropy. At sampling time, we initialize all discrete tokens as `[MASK]` and iteratively unmask them using the learned conditionals, ensuring that every output token lies in its legal vocabulary by construction. This discrete branch is a key differentiator of our pipeline: unlike typical continuous-only diffusion augmentation in ICS, we integrate masked diffusion as a first-class mechanism for supervisory-variable legality within the same end-to-end synthesis workflow [31, 47].

3.4 Type-aware decomposition as factorization and routing layer

Even with a trend-conditioned residual DDPM and a discrete masked-diffusion branch, a single uniform modeling treatment can remain suboptimal because ICS variables are generated by qualitatively different mechanisms. For example, program-driven setpoints exhibit step-and-dwell dynamics; controller outputs follow control laws conditioned on process feedback; actuator positions may show saturation and dwell; and some derived tags are deterministic functions of other channels. Treating all channels as if they were exchangeable stochastic processes can misallocate model capacity and induce systematic error concentration on a small subset of mechanistically distinct variables [25].

We therefore introduce a type-aware decomposition that formalizes this heterogeneity as a routing and constraint layer. Let $\tau(i) \in 1, \ldots, 6$ assign each variable i to a type class. For expository convenience, the assignment can be viewed as a mapping $\tau(i) = \text{TypeAssign}(m_i, s_i, d_i)$, where m_i, s_i, and d_i denote metadata/engineering role, temporal signature, and dependency pattern,

respectively. The type assignment can be initialized from domain semantics (tag metadata, value domains, and engineering meaning), and subsequently refined via an error-attribution workflow described in the Benchmark section. Importantly, this refinement does not change the core diffusion backbone; it changes which mechanism is responsible for which variable, thereby aligning inductive bias with variable-generating mechanism while preserving overall coherence.

We use the following taxonomy:

1. Type 1 (program-driven / setpoint-like): externally commanded, step-and-dwell variables. These variables can be treated as exogenous drivers (conditioning signals) or routed to specialized change-point / dwell-time models, rather than being forced into a smooth denoiser that may over-regularize step structure.
2. Type 2 (controller outputs): continuous variables tightly coupled to feedback loops; these benefit from conditional modeling where the conditioning includes relevant process variables and commanded setpoints.
3. Type 3 (actuator states/positions): often exhibit saturation, dwell, and rate limits; these may require stateful dynamics beyond generic residual diffusion, motivating either specialized conditional modules or additional inductive constraints.
4. Type 4 (process variables): inertia-dominated continuous dynamics; these are the primary beneficiaries of the Transformer trend + residual DDPM pipeline.
5. Type 5 (derived/deterministic variables): algebraic or rule-based functions of other variables; we enforce deterministic reconstruction $\hat{x}^{(i)} = g_i(\hat{X}, \hat{Y})$ rather than learning a stochastic generator, improving logical consistency and sample efficiency.
6. Type 6 (auxiliary/low-impact variables): weakly coupled or sparse signals; we allow simplified modeling (e.g., calibrated marginals or lightweight temporal models) to avoid allocating diffusion capacity where it is not warranted.

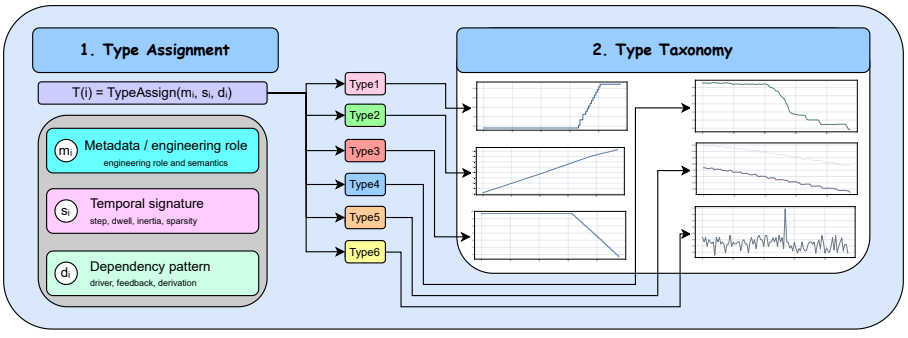

Fig. 2. Type assignment and six-type taxonomy.

Figure 2 visualizes the six-type taxonomy and the routing logic behind it. Type-aware decomposition improves synthesis quality through three mechanisms. First, it improves capacity allocation by preventing a small set of mechanistically atypical variables from dominating gradients and distorting the learned distribution for the majority class (typically Type 4). Second, it enables constraint enforcement by deterministically reconstructing Type 5 variables, preventing logically inconsistent samples that purely learned generators can produce. Third, it improves mechanism alignment by attaching inductive biases consistent with step/dwell or saturation behaviors where generic denoisers may implicitly favor smoothness.

From a novelty standpoint, this layer is not merely an engineering patch; it is an explicit methodological statement that ICS synthesis benefits from typed factorization, a principle that has analogues in mixed-type generative modeling more broadly, but that remains underexplored in diffusion-based ICS telemetry synthesis [32, 47, 25].

3.5 Joint optimization and end-to-end sampling

We train the model in a staged manner consistent with the above factorization, which improves optimization stability and encourages each component to specialize in its intended role. Specifically: (i) we train the trend Transformer f_ϕ to obtain $\hat{\boldsymbol{S}}$; (ii) we compute residual targets $\hat{\boldsymbol{R}} = \boldsymbol{X} - \hat{\boldsymbol{S}}$ for the continuous variables routed to residual diffusion; (iii) we train the residual DDPM $p_\theta(\boldsymbol{R} \mid \hat{\boldsymbol{S}})$ and masked diffusion model $p_\psi(\boldsymbol{Y} \mid \text{masked}(\boldsymbol{Y}), \hat{\boldsymbol{S}}, \hat{\boldsymbol{X}})$; and (iv) we apply type-aware routing and deterministic reconstruction during sampling. This staged strategy is aligned with the design goal of separating temporal scaffolding from distributional refinement, and it mirrors the broader intuition in time-series diffusion that decoupling coarse structure and stochastic detail can mitigate structure-vs.-realism conflicts [14, 34].

A simple combined objective is $\mathcal{L} = \lambda \mathcal{L}_{\text{cont}} + (1 - \lambda)\mathcal{L}_{\text{disc}}$ with $\lambda \in [0, 1]$ controlling the balance between continuous and discrete learning. Type-aware routing determines which channels contribute to which loss and which are excluded in favor of deterministic reconstruction. In practice, this routing acts as a principled guardrail against negative transfer across variable mechanisms: channels that are best handled deterministically (Type 5) or as exogenous / specialized state channels (e.g., driver-like or actuator-state variables) are prevented from forcing the diffusion models into statistically incoherent compromises.

At inference time, generation follows the same structured order: (i) trend $\hat{\boldsymbol{S}}$ via the Transformer, (ii) residual $\hat{\boldsymbol{R}}$ via DDPM, (iii) discrete $\hat{\boldsymbol{Y}}$ via masked diffusion, and (iv) type-aware assembly with deterministic reconstruction for routed variables. This pipeline produces $(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}})$ that are temporally coherent by construction (through $\hat{\boldsymbol{S}}$), distributionally expressive (through $\hat{\boldsymbol{R}}$ denoising), and discretely valid (through masked diffusion), while explicitly accounting for heterogeneous variable-generating mechanisms through type-aware routing. In combination, these choices constitute our central methodological contribution: a unified Transformer + mixed diffusion generator for ICS telemetry, augmented

by typed factorization to align model capacity with domain mechanism [10, 31, 47, 25].

4 Benchmark

A credible ICS generator must clear four progressively harder hurdles. It must first be *semantically legal*: any out-of-vocabulary supervisory token renders a sample unusable, no matter how good its marginals look. It must then match the heterogeneous statistics of mixed-type telemetry, including continuous process channels and discrete supervisory states. Third, it must preserve *mechanism-level realism*: switch-and-dwell behavior, bounded control motion, cross-tag coordination, and short-horizon persistence. Finally, these properties should matter downstream rather than only under offline similarity scores. We therefore organize the benchmark as a funnel rather than a flat metric list, moving from reproducibility and legality to diagnostic localization, extended realism, and ablation [5, 44, 37].

This organization is particularly important for ICS telemetry. A generator can look competitive on one-dimensional marginals while still failing on the aspects that make a trace operationally plausible: long plateaus in setpoint-like variables, concentrated occupancy in actuator states, tight controller–sensor coupling, or persistent support signals. Our goal is therefore not to maximize a single scalar, but to show which parts of realism have already been solved, which remain brittle, and which model components are responsible for each regime.

For continuous channels, we prioritize marginal agreement because ICS process signals often exhibit bounded support, long plateaus, saturation effects, and non-Gaussian tails that are poorly summarized by moment matching alone. We therefore use the Kolmogorov–Smirnov (KS) statistic per feature and average it over continuous variables: KS compares empirical cumulative distributions directly, requires no parametric assumption, and is sensitive to support shifts or local shape mismatches that are operationally meaningful in telemetry. For discrete channels, the object of interest is different: supervisory variables live on a finite vocabulary, so realism is primarily about whether the synthetic sampler places the right probability mass on the right states. We therefore compute Jensen–Shannon divergence (JSD) between per-feature categorical marginals and average across discrete variables [18, 46], since JSD is symmetric, bounded, and naturally suited to comparing categorical occupancy patterns. To assess short-horizon dynamics, we compare lag-1 autocorrelation feature-wise and report the mean absolute difference between real and synthetic lag-1 coefficients, which captures the short-memory persistence induced by actuator dwell, controller smoothing, and process inertia. We additionally track semantic legality by counting out-of-vocabulary discrete outputs, and we report a filtered KS that excludes near-constant channels whose variance is effectively zero so that trivially flat tags do not dominate the aggregate. These core measures are complemented with type-aware diagnostics, extended realism metrics, and ablations.

4.1 Core fidelity, legality, and reproducibility

Across three independent runs, Mask-DDPM achieves mean KS = 0.3311±0.0079, mean JSD = 0.0284±0.0073, and mean absolute lag-1 difference = 0.2684±0.0027, while maintaining a validity rate of **100%** across the modeled discrete channels. The small dispersion across runs suggests that the generator is reproducible at the level of global mixed-type fidelity rather than depending on a single favorable seed. This is the first major benchmark takeaway: semantic legality is already saturated by construction, so the remaining challenge is no longer whether the model can emit valid symbols, but whether it can place valid symbols and trajectories in the right temporal and cross-channel context.

A representative diagnostic slice provides the complementary localized view. As summarized in Table 1, the model attains mean KS = 0.4025, filtered mean KS = 0.3191, mean JSD = 0.0166, and mean absolute lag-1 difference = 0.2859 on that slice, again with zero invalid discrete tokens. Two patterns matter most. First, the discrete branch remains consistently reliable: low JSD together with perfect validity indicates that supervisory semantics are being learned rather than repaired after the fact. Second, the gap between overall KS and filtered KS suggests that continuous mismatch is concentrated in a limited subset of difficult channels instead of being spread uniformly across the telemetry space.

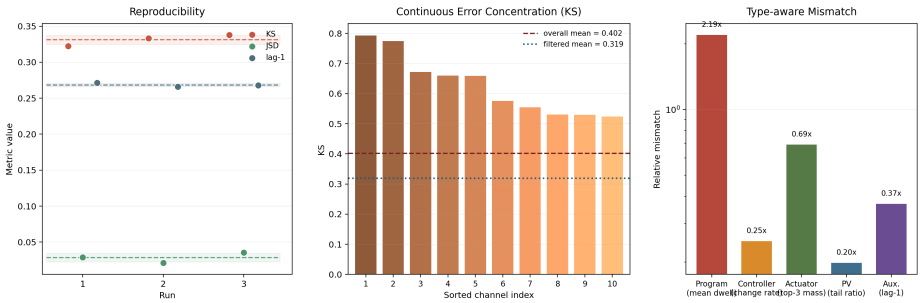

Fig. 3. Benchmark evidence chain.

Table 1. Core benchmark summary. Lower is better except for validity rate.

Metric	3-run mean ± std	Diagnostic slice
Mean KS (continuous)	0.3311 ± 0.0079	0.4025
Filtered mean KS	–	0.3191
Mean JSD (discrete)	0.0284 ± 0.0073	0.0166
Mean abs. Δ lag-1 autocorr	0.2684 ± 0.0027	0.2859
Validity rate (26 discrete tags) ↑	$100.0 \pm 0.0\%$	100.0%

Figure 3 turns the table into a structural diagnosis. The left panel visualizes seed-level stability across the three benchmark runs, showing that the reported KS, JSD, and lag-1 statistics are reproducible rather than the result of a single favorable seed. The middle panel ranks the most difficult continuous channels by KS and shows that the dominant continuous mismatch is concentrated in a relatively small subset of control-sensitive variables instead of indicating a global collapse of the generator. The right panel aggregates type-aware proxy mismatches and shows that the remaining realism gap is mechanism-specific, with program-like long-dwell behavior and actuator-state occupancy contributing more strongly than PV-like channels on this slice. In other words, the model has largely solved legality and a substantial portion of mixed-type marginal fidelity, but realism remains harder for behaviors governed by switching, long dwell, bounded operating regimes, and strong local persistence. This type-aware perspective is developed further in Section 4.3.

4.2 Extended realism and downstream utility

The next question is whether improvements under fidelity metrics correspond to broader structural realism and downstream usefulness. We therefore additionally evaluate two-sample distance, cross-variable coupling, spectral similarity, predictive consistency, memorization risk, and anomaly-detection utility on a representative diagnostic slice. Because this slice is intentionally small, we interpret the resulting numbers as diagnostic rather than definitive; their purpose is to show which aspects of realism respond to post-processing and which ones remain limited by mechanism-level dynamics.

Table 2. Extended realism and downstream utility. Lower is better except for AUPRC. For reference, the real-only predictor RMSE is 0.558 and the real-only anomaly AUPRC is 0.653.

Metric	**Raw generator**	**Post-processed**
Continuous MMD (RBF)	0.6499	0.2166
Discriminative accuracy (ideal 0.5)	1.0000	0.5000
Mean abs. corr. diff.	0.2134	0.1909
Mean abs. lag-1 corr. diff.	0.2132	0.1989
PSD L_1 distance	0.0195	0.0224
Memorization ratio	2.9515	1.6205
Predictive RMSE (synthetic-only)	0.9722	0.9641
Predictive RMSE (real + synthetic)	0.5433	0.5413
Anomaly AUPRC (synthetic-only)	0.5889	0.5894
Anomaly AUPRC (real + synthetic)	0.6449	0.6476

Table 2 reveals a useful asymmetry. Typed post-processing substantially improves distribution-level realism: continuous MMD drops from 0.6499 to 0.2166, discriminative accuracy moves from a trivially separable 1.0 to the chance-level

ideal of 0.5, both contemporaneous and lagged correlation errors decrease, and the memorization ratio contracts from 2.95 to 1.62. In other words, post-processing is very effective at pulling the generated windows closer to the real holdout manifold without collapsing into exact training-set copies. Yet predictive and downstream utility improve only modestly. Synthetic-only predictors remain clearly weaker than real-only ones, and real-plus-synthetic anomaly utility stays slightly below the real-only baseline. This is an important benchmark result: once legality and low-order marginals are largely under control, the remaining gap is driven less by superficial distribution mismatch and more by mechanism-level dynamics that post hoc distribution shaping cannot fully restore.

4.3 Type-aware diagnostics

Type-aware diagnostics make that mechanism gap explicit. Table 3 summarizes one representative statistic per variable family on the same diagnostic slice. These statistics are not redundant with the main benchmark table: they answer a different question, namely which operational behaviors remain hardest to match once legality and marginal alignment are largely in place. Because each family is evaluated with a different proxy, the absolute-error column should be interpreted within type, while the relative-error column is the more comparable cross-type indicator.

Table 3. Type-aware diagnostic summary. Lower values indicate better alignment.

Type	Proxy statistic	Mean abs. error	Mean rel. error
Program	mean dwell	318.70	2.19
Controller	change rate	0.104	0.25
Actuator	top-3 mass	0.0615	0.69
PV	tail ratio	1.614	0.20
Auxiliary	lag-1 autocorr	0.125	0.37

This typed view sharpens the story substantially. Program-like channels remain the hardest family by a wide margin: mean-dwell mismatch is still large in both absolute and relative terms, indicating that the generator does not yet sustain the long plateaus characteristic of schedule-driven or setpoint-like behavior. Actuator channels form the next clear difficulty, with a sizable top-3-mass gap showing that the sampler still spreads probability mass across operating states more broadly than the real system does. Auxiliary channels exhibit a moderate persistence mismatch under the lag-1 proxy, suggesting that support signals with short-memory structure are only partially captured. By contrast, PV channels are the most stable family under this diagnostic, and the controller proxy is comparatively closer on this slice. In short, legality is already solved, but the remaining realism gap is not uniform across types: it is dominated primarily by long-dwell program behavior and actuator-state occupancy.

4.4 Ablation study

A good ablation does more than show that removing components changes numbers; it should identify which failure mode each component is preventing. We therefore evaluate ten controlled variants under a shared pipeline and summarize six representative metrics: continuous fidelity (KS), discrete fidelity (JSD), short-horizon dynamics (lag-1), cross-variable coupling, predictive transfer, and downstream anomaly utility. Figure 4 visualizes signed changes relative to the full model, and Table 4 gives the underlying values.

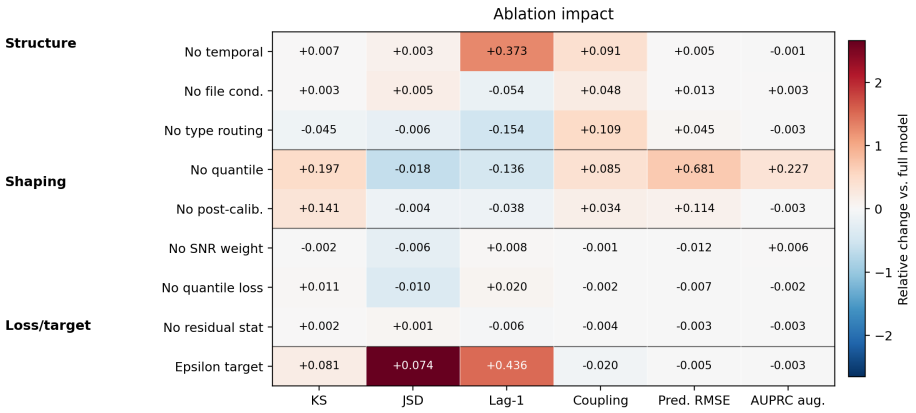

Fig. 4. Ablation impact.

The ablation results reveal three distinct roles. First, temporal staging is what makes the sequence look dynamical rather than merely plausible frame by frame: removing the temporal scaffold leaves KS nearly unchanged but more than doubles lag-1 error ($0.291 \rightarrow 0.664$) and substantially worsens coupling ($0.215 \rightarrow 0.306$). Second, quantile-based distribution shaping is what makes the continuous branch usable: without the quantile transform, KS degrades sharply ($0.402 \rightarrow 0.599$), synthetic-only predictive RMSE deteriorates dramatically ($0.972 \rightarrow 1.653$), and anomaly utility collapses ($0.644 \rightarrow 0.417$). This is not a cosmetic gain; it is one of the main contributors to usable process realism.

The routing ablation supplies the most instructive counterexample. Disabling type routing actually improves several one-dimensional metrics (for example KS and lag-1), yet it worsens coupling ($0.215 \rightarrow 0.324$) and predictive transfer ($0.972 \rightarrow 1.017$). This is exactly why the benchmark cannot stop at scalar per-feature scores: typed decomposition helps the generator coordinate variables and preserve mechanism-level consistency even when simpler metrics may look deceptively better without it. Finally, the target-parameterization ablation is the clearest failure case: replacing the current target with an epsilon target causes the largest degradation in JSD ($0.028 \rightarrow 0.102$) and lag-1 ($0.291 \rightarrow 0.728$), making it

Table 4. Ablation study. Lower is better except for anomaly AUPRC.

Variant	KS↓	JSD↓	Lag-1↓	Coupling↓	Pred. RMSE↓	AUPRC↑
Full model						
Full model	0.402	0.028	0.291	0.215	0.972	0.644
Structure and conditioning						
No temporal scaffold	0.408	0.031	0.664	0.306	0.977	0.645
No file-level context	0.405	0.033	0.237	0.262	0.986	0.640
No type routing	0.356	0.022	0.138	0.324	1.017	0.647
Distribution shaping						
No quantile transform	0.599	0.010	0.156	0.300	1.653	0.417
No post-calibration	0.543	0.024	0.253	0.249	1.086	0.647
Loss and target design						
No SNR weighting	0.400	0.022	0.299	0.214	0.961	0.637
No quantile loss	0.413	0.018	0.311	0.213	0.965	0.645
No residual-stat loss	0.404	0.029	0.285	0.210	0.970	0.647
Epsilon target	0.482	0.102	0.728	0.195	0.968	0.647

the most destructive ablation overall. By contrast, SNR weighting, quantile loss, and residual-stat regularization behave as second-order refinements whose effects are real but materially smaller.

Taken together, the benchmark now supports a sharper claim than a plain KS/JSD table could offer. Mask-DDPM already provides stable mixed-type fidelity, perfect discrete legality, and a meaningful amount of continuous realism. The remaining error is concentrated in a small subset of ICS-specific channels whose realism depends on rare switching, long dwell intervals, constrained occupancy, and persistent local dynamics. The ablation study clarifies why: temporal staging protects dynamical realism, quantile-based shaping protects continuous fidelity and downstream utility, and type-aware routing protects coordinated mechanism-level behavior even when simpler metrics do not fully reveal its value.

5 Conclusion and Future Work

This paper addresses the data scarcity and shareability barriers that limit machine-learning research for industrial control systems (ICS) security by proposing Mask-DDPM, a hybrid synthetic telemetry generator at the protocol-feature level. By combining a causal Transformer trend module, a trend-conditioned residual DDPM, a masked diffusion branch for discrete variables, and a type-aware routing layer, the framework preserves long-horizon temporal structure, improves local distributional fidelity, and guarantees discrete semantic legality. On windows derived from the HAI Security Dataset, the model achieves stable mixed-type fidelity across seeds, with mean KS = 0.3311 ± 0.0079 on continuous features, mean JSD = 0.0284 ± 0.0073 on discrete features, and mean absolute lag-1 autocorrelation difference = 0.2684 ± 0.0027.

Overall, Mask-DDPM provides a reproducible foundation for generating shareable and semantically valid ICS feature sequences for data augmentation, benchmarking, and downstream packet/trace reconstruction workflows. Future work will proceed in two complementary directions. Vertically, we will strengthen the theoretical foundation of the framework by introducing more explicit control-theoretic constraints, structured state-space or causal priors, and formal transition models for supervisory logic, so that legality, stability, and cross-channel coupling can be characterized more rigorously. Horizontally, we will extend the framework beyond the current setting to additional industrial control protocols such as Modbus/TCP, DNP3, IEC 104, and OPC UA, and investigate analogous adaptations to automotive communication protocols such as CAN/CAN FD and automotive Ethernet. A related extension is controllable attack or violation injection on top of legal base traces, enabling reproducible adversarial benchmarks for anomaly detection and intrusion-detection studies.

References

1. Ahmed, C.M., Palleti, V.R., Mathur, A.P.: Wadi: a water distribution testbed for research in the design of secure cyber physical systems. In: Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks. p. 25–28. CySWATER '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3055366.3055375, https://doi.org/10.1145/3055366.3055375
2. Ali, J., Ali, S., Al Balushi, T., Nadir, Z.: Intrusion detection in industrial control systems using transfer learning guided by reinforcement learning. Information **16**(10) (2025). https://doi.org/10.3390/info16100910, https://www.mdpi.com/2078-2489/16/10/910
3. Austin, J., Johnson, D.D., Ho, J., Tarlow, D., van den Berg, R.: Structured denoising diffusion models in discrete state-spaces. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 34, pp. 17981–17993 (2021), https://arxiv.org/abs/2107.03006
4. Austin, J., Johnson, D.D., Ho, J., Tarlow, D., van den Berg, R.: Structured denoising diffusion models in discrete state-spaces (2023), https://arxiv.org/abs/2107.03006
5. Coletta, A., Rossi, R., et al.: On the constrained time-series generation problem. In: Advances in Neural Information Processing Systems. vol. 36 (2023)
6. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-xl: Attentive language models beyond a fixed-length context (2019), https://arxiv.org/abs/1901.02860
7. Godefroid, P., Peleg, H., Singh, R.: Learn&fuzz: Machine learning for input fuzzing (2017), https://arxiv.org/abs/1701.07232
8. Hang, T., Gu, S., Li, C., Bao, J., Chen, D., Hu, H., Geng, X., Guo, B.: Efficient diffusion training via min-snr weighting strategy. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7407–7417 (2023). https://doi.org/10.1109/ICCV51070.2023.00702, https://arxiv.org/abs/2303.09556
9. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 6840–6851. Curran Associates,

Inc. (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf
10. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 33, pp. 6840–6851 (2020), https://arxiv.org/abs/2006.11239
11. Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., Welling, M.: Argmax flows and multinomial diffusion: Learning categorical distributions (2021), https://arxiv.org/abs/2102.05379
12. Jiang, X., Liu, S., Gember-Jacobson, A., Bhagoji, A.N., Schmitt, P., Bronzino, F., Feamster, N.: Netdiffusion: Network data augmentation through protocol-constrained traffic generation (2023), https://arxiv.org/abs/2310.08543
13. Koay, A.M.Y., Ko, R.K.L., Hettema, H., Radke, K.: Machine learning in industrial control system (ics) security: current landscape, opportunities and challenges. J. Intell. Inf. Syst. **60**(2), 377–405 (Oct 2022). https://doi.org/10.1007/s10844-022-00753-1, https://doi.org/10.1007/s10844-022-00753-1
14. Kollovieh, M., Ansari, A.F., Bohlke-Schneider, M., Fatir Ansari, A., Salinas, D.: Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 36 (2023), https://arxiv.org/abs/2307.11494
15. Kong, Z., Ping, W., Huang, J., Zhao, K., Catanzaro, B.: Diffwave: A versatile diffusion model for audio synthesis (2021), https://arxiv.org/abs/2009.09761
16. Kotelnikov, A., Baranchuk, D., Rubachev, I., Babenko, A.: TabDDPM: Modelling tabular data with diffusion models. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 17564–17579. PMLR (23–29 Jul 2023), https://proceedings.mlr.press/v202/kotelnikov23a.html
17. Li, X.L., Thickstun, J., Gulrajani, I., Liang, P., Hashimoto, T.B.: Diffusion-lm improves controllable text generation (2022), httpsarxiv.orgabs2205.14217
18. Lin, J.: Divergence measures based on the shannon entropy. IEEE Transactions on Information Theory **37**(1), 145–151 (1991)
19. Lin, Z., Jain, A., Wang, C., Fanti, G., Sekar, V.: Using gans for sharing networked time series data: Challenges, initial promise, and open questions. In: Proceedings of the ACM Internet Measurement Conference. p. 464–483. IMC '20, ACM (Oct 2020). https://doi.org/10.1145/3419394.3423643, http://dx.doi.org/10.1145/3419394.3423643
20. Liu, M., Huang, H., Feng, H., Sun, L., Du, B., Fu, Y.: Pristi: A conditional diffusion framework for spatiotemporal imputation (2023), https://arxiv.org/abs/2302.09746
21. Liu, X., Xu, X., Liu, Z., Li, Z., Wu, K.: Spatio-temporal diffusion model for cellular traffic generation. IEEE Transactions on Mobile Computing **25**(1), 257–271 (2026). https://doi.org/10.1109/TMC.2025.3591183
22. Mathur, A.P., Tippenhauer, N.O.: Swat: a water treatment testbed for research and training on ics security. In: 2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater). pp. 31–36 (2016). https://doi.org/10.1109/CySWater.2016.7469060
23. Meng, R., Pham, V.T., Böhme, M., Roychoudhury, A.: Aflnet five years later: On coverage-guided protocol fuzzing (2025), https://arxiv.org/abs/2412.20324
24. Nankya, M., Chataut, R., Akl, R.: Securing industrial control systems: Components, cyber threats, and machine learning-driven defense strategies. Sensors (Basel) **23**(21), 8840 (Oct 2023)

25. National Institute of Standards and Technology: Guide to operational technology (ot) security. Special Publication 800-82 Rev. 3, NIST (sep 2023). https://doi.org/10.6028/NIST.SP.800-82r3, https://csrc.nist.gov/pubs/sp/800/82/r3/final
26. Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers. In: International Conference on Learning Representations (ICLR) (2023), https://arxiv.org/abs/2211.14730
27. Rasul, K., Seward, C., Schuster, I., Vollgraf, R.: Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting (2021), https://arxiv.org/abs/2101.12072
28. Ring, M., Schlör, D., Landes, D., Hotho, A.: Flow-based network traffic generation using generative adversarial networks. Computers & Security **82**, 156–172 (May 2019). https://doi.org/10.1016/j.cose.2018.12.012, http://dx.doi.org/10.1016/j.cose.2018.12.012
29. Sha, Y., Yuan, Y., Wu, Y., Zhao, H.: Ddpm fusing mamba and adaptive attention: An augmentation method for industrial control systems anomaly data (jan 2026). https://doi.org/10.2139/ssrn.6055903, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=6055903, sSRN Electronic Journal
30. She, D., Pei, K., Epstein, D., Yang, J., Ray, B., Jana, S.: Neuzz: Efficient fuzzing with neural program smoothing (2019), https://arxiv.org/abs/1807.05620
31. Shi, J., Han, K., Wang, Z., Doucet, A., Titsias, M.K.: Simplified and generalized masked diffusion for discrete data. arXiv preprint (2024), https://arxiv.org/abs/2406.04329
32. Shi, J., Xu, M., Hua, H., Zhang, H., Ermon, S., Leskovec, J.: Tabdiff: A mixed-type diffusion model for tabular data generation. In: International Conference on Learning Representations (ICLR) (2025), https://arxiv.org/abs/2410.20626
33. Shin, H.K., Lee, W., Choi, S., Yun, J.H., Min, B.G., Kim, H.: Hai security dataset (2023). https://doi.org/10.34740/kaggle/dsv/5821622, https://www.kaggle.com/dsv/5821622
34. Sikder, M.F., Ramachandranpillai, R., Heintz, F.: Transfusion: Generating long, high fidelity time series using diffusion models with transformers. arXiv preprint (2023), https://arxiv.org/abs/2307.12667
35. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations (2021), https://arxiv.org/abs/2011.13456
36. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: International Conference on Learning Representations (ICLR) (2021), https://arxiv.org/abs/2011.13456
37. Stenger, M., Leppich, R., Foster, I.T., Kounev, S., Bauer, A.: Evaluation is key: a survey on evaluation measures for synthetic time series. Journal of Big Data **11**(1), 66 (2024)
38. Tashiro, Y., Song, J., Song, Y., Ermon, S.: Csdi conditional score-based diffusion models for probabilistic time series imputation (2021), httpsarxiv.orgabs2107.03502
39. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 30 (2017), https://arxiv.org/abs/1706.03762
40. Vishwanath, K.V., Vahdat, A.: Realistic and responsive network traffic generation. SIGCOMM Comput. Commun. Rev. **36**(4), 111–122 (Aug 2006).

https://doi.org/10.1145/1151659.1159928, https://doi.org/10.1145/1151659.1159928

41. Vishwanath, K.V., Vahdat, A.: Realistic and responsive network traffic generation. In: Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. p. 111–122. SIGCOMM '06, Association for Computing Machinery, New York, NY, USA (2006). https://doi.org/10.1145/1159913.1159928, https://doi.org/10.1145/1159913.1159928
42. Wen, H., Lin, Y., Xia, Y., Wan, H., Wen, Q., Zimmermann, R., Liang, Y.: Diffstg: Probabilistic spatio-temporal graph forecasting with denoising diffusion models (2024), https://arxiv.org/abs/2301.13629
43. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting (2022), https://arxiv.org/abs/2106.13008
44. Yang, S.H., Hsieh, M.C.: Automatic verification of safety interlock systems for industrial processes. Journal of Loss Prevention in the Process Industries **14**(6), 473–483 (2001)
45. Yin, Y., Lin, Z., Jin, M., Fanti, G., Sekar, V.: Practical gan-based synthetic ip header trace generation using netshare. In: Proceedings of the ACM SIGCOMM 2022 Conference. p. 458–472. SIGCOMM '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3544216.3544251, https://doi.org/10.1145/3544216.3544251
46. Yoon, J., Jarrett, D., van der Schaar, M.: Time-series generative adversarial networks. In: Advances in Neural Information Processing Systems. vol. 32 (2019)
47. Yuan, Y., Sha, Y., Zhao, W., Zhang, K.: Ctu-ddpm: Generating industrial control system time-series data with a cnn-transformer hybrid diffusion model. In: Proceedings of the 2025 International Symposium on Artificial Intelligence and Computational Social Sciences (ACM AICSS). pp. 123–132 (2025). https://doi.org/10.1145/3776759.3776845, https://dl.acm.org/doi/10.1145/3776759.3776845
48. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting (2021), https://arxiv.org/abs/2012.07436
49. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting (2022), https://arxiv.org/abs/2201.12740