Latent Semantic Regularization: Enhancing Semantic Integrity in Tabular Data Synthesis

Saba Amiri[1,†] **Carlijn Nijhuis**[1] **Eric Nalisnick**[2] **Adam Belloum**[1] **Sander Klous**[1] **Leon Gommans**[3]

[1]Informatics Institute, University of Amsterdam
[2]Department of Computer Science, Johns Hopkins University
[3]Air France–KLM
[†]Correspondence to s.amiri@uva.nl

Abstract

Generative models trained on finite data could potentially assign non-negligible probability mass to regions that are statistically plausible but semantically invalid as a structural consequence of distribution estimation from samples. We propose Latent Semantic Regularization (LSR), which learns implicit semantic constraints directly from data and uses them to regularize a generative model, requiring no explicit constraint definitions, feasibility oracles, or negative examples. We instantiate LSR with a Conditional VAE using a tail-adaptive normalizing flow prior as the validator and a WGAN-GP as the synthesizer. We also introduce Exploration Factor metric measuring exploration/exploitation, and a controlled semantic integrity protocol based on injected zero-probability constraints with known ground truth. Across five benchmark datasets and against GAN- based, diffusion-based, and LLM-based baselines, LSR reduces semantic violation rates consistently while remaining competitive on statistical fidelity and downstream utility.

1. Introduction

While training machine learning models on huge troves of data has many benefits (e.g. see (Kaplan et al., 2020)), it also comes with challenges. Availability is one of these issues: physical constraints limit access to distributed big data. Privacy is another: data holders often are unable to share their private datasets. One way to mitigate these problems is by utilizing synthetic data (Rubin, 1993), where a proxy dataset is generated for downstream tasks. Modern data synthesis is often done by a model, which can be viewed as a compressed representation of the data. This solves the data access problem since the model can generate synthetic data from the learned compact representation. Privacy is preserved to some extent by not using real data directly.

We consider the task of synthesizing high-fidelity tabular data, focusing specifically on the critical challenge of ensuring the synthetic data is *semantically correct*. Semantic violations manifest as records containing feature combinations that contradict domain knowledge or implicit real-world constraints. Such violations, which can be considered as *hallucinations*, represent implausible scenarios. Preserving semantic integrity is paramount in high-stakes domains like healthcare (for example, generating a record for a male patient with pregnancy history is unacceptable.)

To address this challenge, we propose **Latent Semantic Regularization (LSR)**, a novel framework that formulates semantic integrity enhancement as an unsupervised, regularized optimization problem. LSR unsupervisedly learns implicit semantic rules from the training data and leverages them to

guide the synthesis process. LSR features a probabilistic latent-space *validator*, which models the data's semantic structure, and a *synthesizer* which generates diverse samples. The validator imposes the learned soft semantic bounds on the synthesizer's latent space, steering it towards semantic consistency.

We formalize LSR as a modular framework and instantiate it with a Variational Autoencoder (VAE) as the validator and a Generative Adversarial Network (GAN) as the synthesizer. This VAE/GAN pairing provides a well-understood probabilistic setting in which to study the central mechanism of LSR: a frozen validator learns semantic structure from real data, while a separate generator is regularized by the validator's signal during synthesis. The GAN objective encourages exploration and sample diversity, whereas the VAE-based validator penalizes departures from learned semantic structure, yielding a unified objective that balances exploration and exploitation. Importantly, LSR is not tied to this specific architecture. The VAE/GAN instantiation serves as a proof of concept, while the loosely coupled design allows different latent variable models to be used for either component. This separation of roles is the main contribution: it makes LSR modular, clarifies how semantic learning and generative synthesis interact, and allows the validator to be interpreted as a reusable semantic scoring component.

Our contributions are:

- **LSR**, a modular framework that treats semantic integrity as a regularized optimization problem, learning implicit semantic constraints directly from data without requiring explicit constraint definitions, feasibility oracles, or curated negative examples.
- **A concrete instantiation** of LSR using a CVAE with a tail-adaptive normalizing flow prior as the validator and a WGAN-GP as the synthesizer, together with a full derivation showing how the general penalty reduces to a tractable, differentiable objective.
- **New evaluation metric and protocol**: an Exploration Factor quantifying generative diversity over dependency-relevant feature combinations, and a controlled semantic integrity protocol based on injected zero-probability constraints with known ground truth, filling a gap in existing tabular synthesis benchmarks.
- **Comprehensive experiments** on five benchmark datasets against GAN-based, diffusion-based, and LLM-based baselines, showing that LSR reduces semantic violation rates consistently while remaining competitive on statistical fidelity and downstream utility, without dataset-specific tuning of the penalty weight.

2. The Challenge of Semantic Integrity in Generative Models

Generative models learn a distribution $p_{\text{model}}(\mathbf{x})$ approximating $p_{\text{data}}(\mathbf{x})$, typically via a generator $G_\theta(\mathbf{z})$ mapping latent variables $\mathbf{z} \sim p(\mathbf{z})$ to samples $\mathbf{x}'$, aiming for statistical fidelity ($p_{\text{model}} \approx p_{\text{data}}$). However, synthesized data must also exhibit *semantic integrity*, defined as consistency with domain knowledge and constraints (e.g., physical laws, logical rules), ensuring contextual coherence beyond statistical plausibility. Standard objectives may fail to capture such constraints, especially implicit ones not directly encoded in $p_{\text{data}}(\mathbf{x})$, leading to semantic violations. This deficiency stems from factors including model limitations in capturing complex data manifolds, model biases, and incompleteness in $p_{\text{data}}(\mathbf{x})$.

We frame this as a constrained optimization problem. Let $C(\mathbf{x})$ denote the unknown semantic constraints. The objective is to learn generator parameters θ maximizing data fidelity, $\mathcal{L}_{\text{fidelity}}(\theta; p_{\text{data}})$, such that generated samples $\mathbf{x}' = G_\theta(\mathbf{z})$ satisfy these constraints:

$$\begin{aligned} \underset{\theta}{\text{maximize}} \quad & \mathcal{L}_{\text{fidelity}}(\theta; p_{\text{data}}) \\ \text{subject to} \quad & C(G_\theta(\mathbf{z})) \text{ holds}, \ \forall \mathbf{z} \sim p(\mathbf{z}). \end{aligned} \tag{1}$$

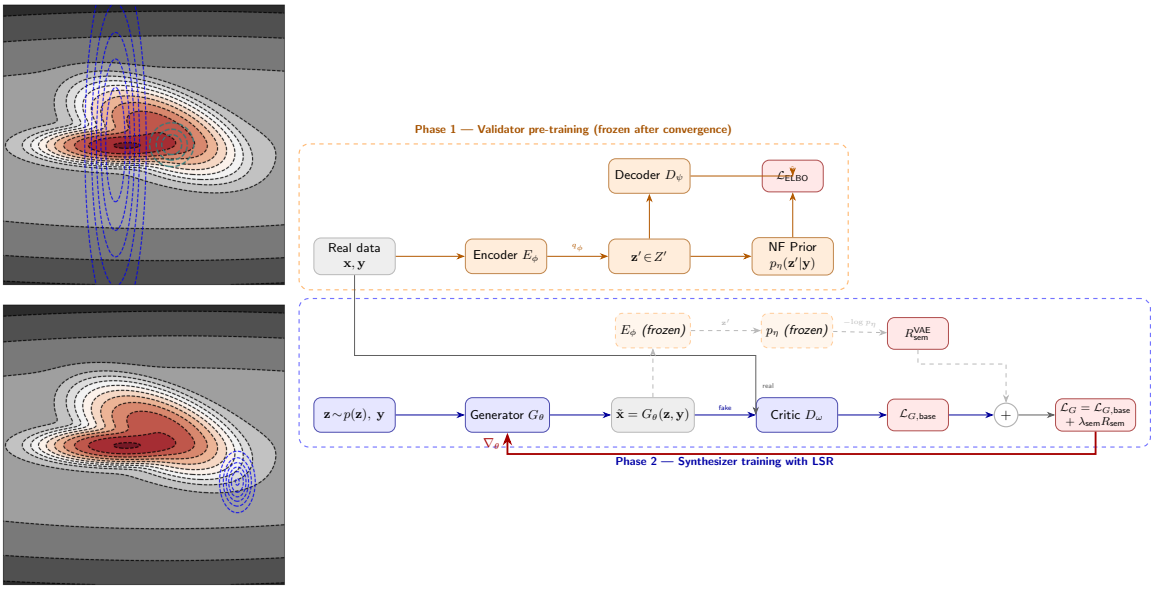

Figure 1.
Left: The Explore/Exploit Dilemma in Semantically Constrained Synthesis. *Top-left*: *Green*: high exploitation (strong semantic integrity) concentrates mass, missing tails and diversity. *Blue*: high exploration (diversity) may venture into semantically invalid regions (hallucinations). *Bottom-left*: example empirical posterior density of generated samples mapped into the validator's latent space, showing how LSR structures this space.

Right: LSR architecture. Phase 1 (orange): the validator (CVAE with tail-adaptive NF prior) is pre-trained on real data to learn a structured semantic latent space Z'; all validator parameters are frozen afterwards. **Phase 2** (blue): the synthesizer (WGAN-GP) is trained with the combined objective $\mathcal{L}_G$. Generated samples $\tilde{\mathbf{x}}$ are encoded by the frozen E_ϕ and scored by the frozen p_η, yielding $R_{\text{sem}}^{\text{VAE}}$ (dashed paths); gradients (red) flow back to the generator only.

Equation 1 highlights the core tension between optimizing fidelity and enforcing constraint satisfaction. This inherently requires balancing exploration (diversity of $\mathbf{x}'$) and exploitation (semantic integrity of $\mathbf{x}'$), illustrated in Figure 1.

Assuming no access to closed-form constraints, the existing literature takes different approaches to solve this problem (see Appendix A). The simplest approach is to assume access to a constraint verification ((Naderiparizi et al., 2023)), providing noisy observations of an underlying latent feasibility function and informing the posterior over the feasible region of the data space which allows for iterative refinement of the variational approximation to the true posterior. Other methods rely on an existing mechanisms to generate semantically non-conforming samples (e.g., (Giannone et al., 2023)), helping us refine our posterior distribution over the latent variables governing the generation process.

We address the challenging **unsupervised** setting where $C(\mathbf{x})$ is implicit, and external tools like explicit constraint definitions and feasibility oracles are unavailable. This makes unsupervised learning and enforcement of semantic integrity face significant hurdles; $C(\mathbf{x})$ may resist direct quantification, the manifold of valid data can have intricate boundaries or complex topological features (e.g., heavy tails corresponding to rare valid examples), and the curse of dimensionality exacerbates these challenges.

To our knowledge, existing literature often focus on specific domains (e.g., images (Xu et al., 2024)) or implicitly address integrity without formulating it as the primary objective. Our work explicitly targets the unsupervised learning and enforcement of implicit semantic constraints within a generalizable framework, aiming to robustly balance diversity and semantic validity.

3. Latent Semantic Regularization Framework

We introduce **Latent Semantic Regularization (LSR)**, a general framework to explicitly enhance the semantic integrity of synthetic data produced by generative models. LSR achieves this goal by unsupervisedly learning implicit constraints from real data and using them to regularize a generative model. The framework comprises two loosely-coupled components:

The Synthesizer (G_θ) is a generative model that learns $p_{\text{data}}(\mathbf{x})$ and produces samples $\mathbf{x}' = G_\theta(\mathbf{z})$, $\mathbf{z} \sim p(\mathbf{z})$, optimizing a fidelity-and-diversity objective $\mathcal{L}_{\text{syn}}(\theta)$.

The Validator (V_ϕ) is a latent scoring model parameterized by (ϕ, η) that maps inputs through an encoder $q_\phi(\mathbf{z}' \mid \mathbf{x})$ into a semantic latent space $\mathcal{Z}'$ and maintains a learned density $p_\eta(\mathbf{z}')$ over it.

The validator provides a *semantic regularization signal* that quantifies how well a generated sample $\mathbf{x}'$ conforms to the learned semantic structure. Adding this signal to the synthesizer's objective penalizes implausible samples while preserving diversity, directly addressing the exploration–exploitation tension from Eq. 1. The signal is formalized in Eqs. 2–3.

3.1. Formalization

Conditioning variables $\mathbf{y}$ are omitted below for readability. All terms extend naturally to the conditional setting.

Semantic feasibility region. Let $\mathcal{F} \subseteq \mathcal{X}$ denote the unknown set of semantically feasible records. We assume p_{data} concentrates on $\mathcal{F}$, i.e. $\mathbb{P}_{\mathbf{x}\sim p_{\text{data}}}[\mathbf{x} \in \mathcal{F}] \approx 1$, and learn a probabilistic surrogate for feasibility directly from data.

The LSR *validator* consists of an encoder $q_\phi(\mathbf{z}' \mid \mathbf{x})$ into a semantic latent space $\mathcal{Z}'$, and a learned latent density $p_\eta(\mathbf{z}')$ on $\mathcal{Z}'$, trained on real data so that encodings of real samples receive high density under p_η, after which (ϕ, η) are frozen.

LSR rests on the assumption that a validator trained on p_{data} tends to assign lower density to latent codes corresponding to semantically invalid configurations, which is empirically supported by Figure 3. This is an empirical assumption as degenerate ELBO optima can make the latent prior semantically uninformative. In our setting, the finite-capacity validator is therefore treated as a learned feasibility proxy whose usefulness is validated empirically.

Semantic regularization penalty. For a candidate sample $\mathbf{x}'$, the *semantic regularization penalty* is

$$R_{\text{sem}}(\mathbf{x}'; \phi, \eta) = \mathbb{E}_{\mathbf{z}'\sim q_\phi(\mathbf{z}'|\mathbf{x}')}\big[-\log p_\eta(\mathbf{z}')\big], \tag{2}$$

reducing to $-\log p_\eta(E_\phi(\mathbf{x}'))$ for a deterministic encoder. The penalty is large when $\mathbf{x}'$ encodes to an atypical region of p_η, signalling a likely semantic violation.

Regularized synthesizer training. The hard feasibility constraint of Eq. 1 is intractable because $\mathcal{F}$ is unknown, so we replace it with a differentiable penalty relaxation:

$$\underset{\theta}{\text{maximize}} \quad \mathcal{L}_{\text{syn}}(\theta) - \lambda\, \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\big[R_{\text{sem}}(G_\theta(\mathbf{z}); \phi, \eta)\big], \tag{3}$$

where $\lambda \geq 0$ controls the fidelity–semantics trade-off and the maximization is over θ alone, with (ϕ, η) fixed. One could instead regularize by the validator's marginal likelihood, $\log \int p_\psi(\mathbf{x}' \mid \mathbf{z}') p_\eta(\mathbf{z}')\, d\mathbf{z}'$. We use the latent prior score because semantic violations often concern dependency structure rather than gross input-space atypicality, and because this score is cheaper to compute with the frozen validator.

Two-phase training. The validator is first fit on real data and frozen. The synthesizer is then trained under Eq. 3. Freezing prevents collusion by stopping joint training from lowering R_{sem} by distorting the learned representation rather than improving semantic quality. Decoupling also allows each component to specialize—density modeling for the validator, fidelity and diversity for the synthesizer—and naturally supports extensions such as multiple or conditional validators within the same objective.

Interpretation and limitations. The penalty in Eq. 2 is large when $q_\phi(\mathbf{z}' \mid \mathbf{x}')$ places mass on low-density regions of p_η, so Eq. 3 discourages the synthesizer from generating samples whose validator encodings are atypical relative to those of real data. This mechanism is effective when semantic violations manifest as latent atypicality, but carries no formal guarantee. It degrades if the training data itself contains a non-negligible fraction of violations (which the validator may then assign high density), or if violations occupy latent regions that overlap substantially with valid samples.

3.2. A Concrete Realization of LSR

We instantiate the LSR framework using a Conditional Variational Autoencoder (CVAE) (Sohn et al., 2015) as the validator and a conditional Wasserstein GAN with Gradient Penalty (WGAN-GP) (Gulrajani et al., 2017) as the synthesizer. Full derivations, architectural details, and training procedures are given in Appendix B.

3.2.1. Validator: CVAE with Tail-Adaptive NF Prior

The CVAE encoder $q_\phi(\mathbf{z}' \mid \mathbf{x}, \mathbf{y})$ maps real samples into a semantic latent space $\mathcal{Z}'$. Training maximizes the conditional ELBO (Appendix B.1), aligning the approximate posterior $q_\phi(\mathbf{z}' \mid \mathbf{x}, \mathbf{y})$ with a learned conditional prior $p_\eta(\mathbf{z}' \mid \mathbf{y})$. After convergence the prior becomes a probabilistic model of semantic validity in $\mathcal{Z}'$. This instantiates the general penalty of Eq. 2 as

$$R_{\text{sem}}^{\text{VAE}}(\mathbf{x}'; \phi, \eta) = \mathbb{E}_{\mathbf{z}' \sim q_\phi(\mathbf{z}' \mid \mathbf{x}', \mathbf{y}')}\big[-\log p_\eta(\mathbf{z}' \mid \mathbf{y}')\big], \tag{4}$$

which is large when the synthetic sample $\mathbf{x}'$ encodes to a low-density region of the learned prior, signalling a likely semantic violation.

Tail-Adaptive Normalizing Flow Prior. A fixed isotropic Gaussian prior lacks the capacity to model the complex and potentially heavy-tailed structure of $p_\eta(\mathbf{z}' \mid \mathbf{y})$ that arises from real semantic boundaries. Rare but valid data configurations inhabit the tails of $\mathcal{Z}'$. A Gaussian prior cannot distinguish these from genuine violations. We therefore parameterize p_η via a tail-adaptive normalizing flow (Amiri et al., 2024), letting the prior adapt its tail behavior during training. The resulting expressive prior assigns appropriate density to valid tail configurations while keeping semantically invalid encodings in genuinely low-density regions. The modified ELBO used to train the flow jointly with (ϕ, ψ) is given in Appendix B.1.

3.2.2. Synthesizer: WGAN-GP with LSR

The synthesizer generator $G_\theta : \mathcal{Z} \times \mathcal{Y} \to \mathcal{X}$ maps latent noise $\mathbf{z} \sim p(\mathbf{z})$ and conditioning $\mathbf{y}$ to synthetic samples $\tilde{\mathbf{x}} = G_\theta(\mathbf{z}, \mathbf{y})$. Its base objective combines the standard WGAN generator loss with an auxiliary classifier term (Odena et al., 2017; Gong et al., 2019) and a feature-matching information loss (Park et al., 2018):

$$\mathcal{L}_{G,\text{base}}(\theta) = \mathcal{L}_{G,\text{WGAN}}(\theta) + \lambda_{ac,g}\, \mathcal{L}_{G,\text{AC}}(\theta) + \lambda_{\text{info}}\, \mathcal{L}_i(\theta). \tag{5}$$

Incorporating the frozen validator's semantic penalty (Eq. 4) via the general LSR objective (Eq. 3) gives the full synthesizer objective:

$$\mathcal{L}_G(\theta) = \underbrace{\mathcal{L}_{G,\text{base}}(\theta)}_{\mathcal{L}_{\text{syn}}(\theta) \text{ in Eq. 3}} + \lambda_{\text{sem}}\, \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}),\, \mathbf{y}\sim p(\mathbf{y})}\Big[R^{\text{VAE}}_{\text{sem}}\big(G_\theta(\mathbf{z},\mathbf{y}),\mathbf{y};\phi,\eta\big)\Big], \tag{6}$$

where $\lambda_{\text{sem}} \geq 0$ governs the fidelity–semantics trade-off and the validator parameters (ϕ, η) remain frozen throughout synthesizer training. The overall data flow of this particular instantiation of LSR is shown in Figure 1.

4. Experiments

We instantiate LSR using a CVAE with a Tail-Adaptive NF prior as the validator and a conditional WGAN-GP as the synthesizer, denoted **LSR (S+V)**. To isolate the validator's contribution, we include an ablation **LSR (S)** trained on the base objective alone (Eq. 5). GAN baselines are MedGAN (Choi et al., 2017), CTGAN (Xu et al., 2019), and TableGAN (Park et al., 2018). Non-adversarial baselines are TabDDPM (Kotelnikov et al., 2023) and GReaT (Borisov et al., 2022). All models use standard implementations on identical stratified 80/20 splits. Results are averaged over five runs. λ_{sem} is fixed to 1.0 across all datasets. A sensitivity analysis is reported in Appendix C.2. Full architecture and training details are in Appendix C.

4.1. Metrics

Statistical Fidelity Average KS statistic (numerical features) and average Log Chi-Squared statistic (categorical features) between real and synthetic distributions; lower is better for both.

Downstream Task Efficacy A classifier trained on synthetic data is evaluated on a held-out real test set (Xu et al., 2019; Park et al., 2018) and F1-score is reported.

Exploration Factor Since no established diversity metric targets dependency-preserving feature combinations, we define the Exploration Factor S_d as an aggregate nearest-neighbor dissimilarity score over dependency-relevant categorical combinations. All models are evaluated by generating exactly $N = |D_{\text{train}}|$ samples within each dataset, so S_d is directly comparable across models on the same dataset. For each generated sample $r \in D_g$, we find its nearest neighbor $\hat{r} \in D_o$ in the real training set and compute

$$S_d = \sum_{r \in D_g} d_H(r_M, \hat{r}_M), \tag{7}$$

where d_H is Hamming distance restricted to the set M of categorical attributes appearing in the top-10% strongest pairwise associations. Higher S_d indicates greater aggregate exploration of dependency-relevant feature combinations. Because S_d is an aggregate score, its magnitude also depends on dataset size and the size of M, so it is intended for comparison across models within a dataset rather than across datasets.

Semantic Integrity Quantification As no established framework exists for evaluating unsupervised semantic constraint adherence, we propose a controlled protocol. Binary synthetic variables V_i with known conditional distributions $V_i \mid X_{j,k} \sim \text{Binomial}(n_{j,k}, p_{i|j,k})$ are injected into the *Child* dataset, with $p_{i|j,k} = 0$ for selected subgroups defining ground-truth forbidden configurations. Violation frequency on these zero-probability rules serves as a quantitative proxy for semantic integrity. We use controlled injection because naturally occurring semantic constraints in unsupervised benchmarks lack ground-truth violation labels. The injected zero-probability rules give a reproducible semantic test beyond marginal fidelity metrics.

Table 1. Percentage of incorrectly labelled samples on the 2D Gaussian mixture dataset. Lower is better.

Model	Incorrect Labels ↓
MEDGAN	$51.4\% \pm 11.9\%$
TABLEGAN	$44.2\% \pm 3.5\%$
CTGAN	$10.9\% \pm 0.5\%$
TabDDPM	$9.1\% \pm 0.8\%$
GReaT	$12.3\% \pm 1.1\%$
LSR (S+V)	$\mathbf{6.4\% \pm 0.3\%}$
LSR (S)	$7.9\% \pm 0.3\%$
LSR (S+V, no $\mathcal{L}_i$)	$9.0\% \pm 0.7\%$

Table 2. Statistical Fidelity (Numerical Features): Average KS test statistic computed across numerical features by comparing real and synthetic marginal distributions feature-wise and averaging over features. Lower is better.

Model	Adult ↓	CovType ↓	Intrusion ↓	Census ↓	Child ↓
MEDGAN	0.92 ± 0.01	0.42 ± 0.01	0.18 ± 0.01	0.82 ± 0.01	N/A
TABLEGAN	0.40 ± 0.02	$\mathbf{0.11 \pm 0.01}$	0.03 ± 0.003	0.51 ± 0.007	N/A
CTGAN	0.21 ± 0.02	0.14 ± 0.02	0.034 ± 0.006	0.23 ± 0.02	N/A
TabDDPM	$\mathbf{0.16 \pm 0.01}$	0.12 ± 0.01	0.028 ± 0.004	0.17 ± 0.01	N/A
GReaT	0.22 ± 0.02	0.13 ± 0.02	0.036 ± 0.007	0.19 ± 0.02	N/A
LSR (S+V)	0.17 ± 0.01	0.13 ± 0.02	$\mathbf{0.026 \pm 0.005}$	$\mathbf{0.16 \pm 0.02}$	N/A
LSR (S)	0.18 ± 0.02	0.14 ± 0.02	0.031 ± 0.006	0.19 ± 0.03	N/A
LSR (S+V, no $\mathcal{L}_i$)	0.18 ± 0.02	0.14 ± 0.02	0.030 ± 0.006	0.18 ± 0.02	N/A

4.2. Motivating Experiment

To provide intuition for the role of LSR's validator in identifying and regularizing semantic violations, we first consider a controlled experiment on a low-dimensional synthetic dataset.

Dataset and Task Setup We construct a 2D dataset by sampling from a mixture of three isotropic Gaussian distributions. We apply K-Means clustering ($K = 3$) to the sampled points to obtain cluster assignments, which serve as ground-truth labels, and add them as a feature to the dataset. The task for the generative models is to capture the underlying 2D Gaussian mixture distribution and generate the correct labels. Generating samples with incorrect labels constitutes a semantic violation. This setup serves as a tractable proxy for more complex semantic constraints of real-world data.

Results We evaluate the semantic integrity of the generated data by measuring the percentage of samples with incorrect cluster labels, reported in Table 1. We observe that baseline method exhibit difficulty in adhering to the semantic constraint. TabDDPM achieves the best result among external baselines (9.1%), benefiting from its iterative refinement process, while GReaT (12.3%) performs comparably to CTGAN (10.9%). In contrast, the LSR (S+V) model achieves the lowest error rate (6.4%) followed by the ablation model, demonstrating clear improvement attributable to the validator's regularization signal.

4.3. Benchmark Experiments

We extend our evaluation to standard real-world benchmark datasets commonly used in tabular data synthesis literature. LSR (S+V, no $\mathcal{L}_i$) is the ablation model, explained in the following Section 4.3.

We utilize five benchmark datasets: *Adult Income*, *Covertype*, *Census*, *Intrusion*, obtained from the UCI Machine Learning Repository (Dua et al., 2017), and the *Child* dataset (Spiegelhalter, 1992). These datasets represent a variety of domains, feature types, and class balances, providing an extensive testbed for evaluating generative models.

Data Utility Evaluation We first assess the general utility of the models by measuring both statistical fidelity and downstream task performance.

Statistical Fidelity: Table 2 presents the average KS statistic for numerical features and Table 3 shows the average Log Chi-Squared statistic for categorical features. For numerical features (Table 2), our method is the strongest external baseline, leading on Intrusion and Census and staying near the best value on CovType and Adult. For categorical features (Table 3), LSR (S+V) attains the lowest error on Adult, Intrusion and Child and stays close to the leading models on Census, but its Covertype result is clearly weaker than TableGAN and TabDDPM. TabDDPM again outperforms CTGAN across all five datasets, making it the most credible external reference overall. Relative to

Table 3. Statistical Fidelity (Categorical Features): Average Log Chi-Squared test statistic across all categorical features. Lower is better.

Model	Adult ↓	CovType ↓	Intrusion ↓	Census ↓	Child ↓
MEDGAN	9.4 ± 7.8	10.8 ± 8.9	8.6 ± 6.5	9.6 ± 7.9	8.4 ± 6.8
TABLEGAN	8.7 ± 7.4	$\mathbf{5.4 \pm 4.4}$	3.1 ± 2.2	9.1 ± 8.0	6.0 ± 5.6
CTGAN	8.4 ± 5.9	9.7 ± 6.8	8.8 ± 6.1	8.7 ± 6.7	6.3 ± 5.1
TabDDPM	7.5 ± 4.7	6.6 ± 4.6	4.0 ± 2.8	8.3 ± 6.2	5.0 ± 3.7
GReaT	8.6 ± 6.1	8.0 ± 6.0	6.9 ± 5.0	$\mathbf{8.1 \pm 6.5}$	5.2 ± 4.0
LSR (S+V)	$\mathbf{7.3 \pm 4.5}$	9.4 ± 3.2	$\mathbf{2.9 \pm 1.2}$	8.2 ± 6.1	$\mathbf{4.8 \pm 3.0}$
LSR (S)	7.8 ± 4.9	10.0 ± 3.8	5.1 ± 2.9	8.8 ± 6.4	5.0 ± 3.1
LSR (S+V, no $\mathcal{L}_i$)	7.7 ± 5.0	9.7 ± 3.6	5.0 ± 2.8	8.4 ± 6.2	4.9 ± 3.1

Table 4. Downstream Task Efficacy (F1-Score): Performance of a classifier trained on synthetic data vs. real data. Higher is better.

Model	Adult ↑	CovType ↑	Intrusion ↑	Census ↑	Child ↑
Real Data	0.67 ± 0.01	0.82 ± 0.02	0.93 ± 0.03	0.51 ± 0.01	0.49 ± 0.01
MEDGAN	0.48 ± 0.03	0.11 ± 0.02	0.19 ± 0.08	0.03 ± 0.03	0.31 ± 0.08
TABLEGAN	0.24 ± 0.09	0.49 ± 0.02	0.79 ± 0.11	0.04 ± 0.01	0.46 ± 0.02
CTGAN	0.61 ± 0.02	0.46 ± 0.04	0.81 ± 0.04	$\mathbf{0.47 \pm 0.01}$	0.49 ± 0.01
TabDDPM	0.63 ± 0.01	0.56 ± 0.02	0.89 ± 0.03	0.44 ± 0.02	0.48 ± 0.01
GReaT	0.62 ± 0.02	0.54 ± 0.03	0.84 ± 0.04	0.43 ± 0.02	0.47 ± 0.02
LSR (S+V)	$\mathbf{0.64 \pm 0.03}$	$\mathbf{0.58 \pm 0.02}$	$\mathbf{0.92 \pm 0.03}$	0.42 ± 0.02	$\mathbf{0.50 \pm 0.01}$
LSR (S)	0.63 ± 0.03	0.56 ± 0.02	0.90 ± 0.03	0.40 ± 0.03	0.48 ± 0.02
LSR (S+V, no $\mathcal{L}_i$)	0.63 ± 0.02	0.55 ± 0.03	0.89 ± 0.04	0.39 ± 0.02	0.48 ± 0.02

Table 5. Exploration Factor (S_d): Aggregate adjusted Hamming distance to nearest neighbors in the real training set, summed over generated samples and computed over dependency-relevant categorical features M. Entries report mean ± standard error across five runs. Higher values indicate greater generative diversity. Scores are intended for within-dataset comparison across models.

Model	Adult ↑	CovType ↑	Intrusion ↑	Census ↑	Child ↑
MEDGAN	2240 ± 963	118 ± 118	2282 ± 1620	132774 ± 11950	1058 ± 444
TABLEGAN	6163 ± 1233	81 ± 35	1090 ± 294	240295 ± 26432	2381 ± 786
CTGAN	4296 ± 558	$\mathbf{721 \pm 209}$	7849 ± 2120	117534 ± 7052	1392 ± 181
TabDDPM	8760 ± 980	356 ± 62	10420 ± 1540	201880 ± 13840	1810 ± 190
GReaT	3980 ± 720	438 ± 96	10080 ± 1620	166430 ± 12150	1905 ± 230
LSR (S+V)	$\mathbf{11840 \pm 1260}$	640 ± 170	$\mathbf{12880 \pm 1090}$	$\mathbf{248620 \pm 18240}$	$\mathbf{2460 \pm 170}$
LSR (S)	10180 ± 880	452 ± 110	11940 ± 980	221540 ± 15420	1580 ± 140
LSR (S+V, no $\mathcal{L}_i$)	11260 ± 1180	518 ± 130	12360 ± 1050	236910 ± 17130	2240 ± 160

Table 6. Percentage of samples violating zero-probability constraints on Augmented *Child*. Models are grouped based on qualitative assessment (see Fig. 2). Lower percentages in Group 1 stem from mode collapse.

Model Grouping	V_1 Violations (%) ↓	V_2 Violations (%) ↓	V_3 Violations (%) ↓	V_4 Violations (%) ↓
Models exhibiting significant mode collapse				
MEDGAN	0.13 ± 0.04	0.04 ± 0.01	0.11 ± 0.01	0.07 ± 0.01
TABLEGAN	0.02 ± 0.01	0.37 ± 0.11	2.59 ± 0.91	0.36 ± 0.08
Models attempting nominal subgroup coverage				
CTGAN	1.43 ± 0.52	0.60 ± 0.18	0.88 ± 0.22	$\mathbf{0.41 \pm 0.04}$
TabDDPM	1.05 ± 0.28	0.67 ± 0.17	0.56 ± 0.10	0.49 ± 0.07
GReaT	1.18 ± 0.34	0.63 ± 0.19	0.71 ± 0.16	0.46 ± 0.06
LSR (S+V)	$\mathbf{0.90 \pm 0.12}$	$\mathbf{0.58 \pm 0.11}$	$\mathbf{0.54 \pm 0.09}$	0.50 ± 0.07
LSR (S)	1.04 ± 0.15	0.76 ± 0.14	0.67 ± 0.10	0.57 ± 0.08
LSR (S+V, no $\mathcal{L}_i$)	0.95 ± 0.13	0.63 ± 0.12	0.58 ± 0.09	0.53 ± 0.08

LSR (S), the full model still delivers modest gains on the datasets where semantic structure is most helpful.

Downstream Task Efficacy: Table 4 assesses the utility of synthetic data for training a linear SVM classifier and includes the classifier trained on real data as a reference. LSR (S+V) gives the strongest synthetic-data F1 on Adult, Covertype, Child and Intrusion and stays competitive on Census. Census remains the clearest exception, where CTGAN yields the best synthetic F1. In practice, Census appears more sensitive to class-conditional matching than to marginal fidelity summaries alone. Overall, LSR improves downstream utility on average without uniformly dominating every classification setting.

Having established the general data utility of LSR, we now turn to the core claim, which is its ability to improve semantic integrity while balancing exploration/exploitation.

Generative Diversity (Exploration) Table 5 reports the Exploration Factor (S_d) for each model, with higher S_d values indicating greater aggregate dissimilarity between generated samples and training data, signifying broader exploration of the potential data space.

The exploration scores remain highest for LSR (S+V) on Adult, Intrusion, Census, and Child while CTGAN retains the highest Covertype score. The comparison with LSR (S) and LSR (S+V, no $\mathcal{L}_i$) indicates that the validator increases diversity without the pronounced mode-collapse behavior visible in the weaker GAN baselines.

Controlled Semantic Constraint Setup We utilize the controlled experimental setup described in Section 4.1 and Appendix C for the *Child* dataset. We introduce (V_1, V_2) associated with a base feature X_1, and (V_3, V_4) associated with a base feature X_2. Their values are determined by conditional binomial distributions $V_i|X_{j,k} \sim \text{Binomial}(n_{j,k}, p_{i|j,k})$, setting $p_{i|j,k} = 0$ for at least one specific subgroup k of the corresponding base feature X_j. Violating these zero-probability rules constitutes a semantic error. Other $p_{i|j,k}$ parameters are set to induce under- and over-representation, allowing assessment of distributional fidelity within these semantic contexts.

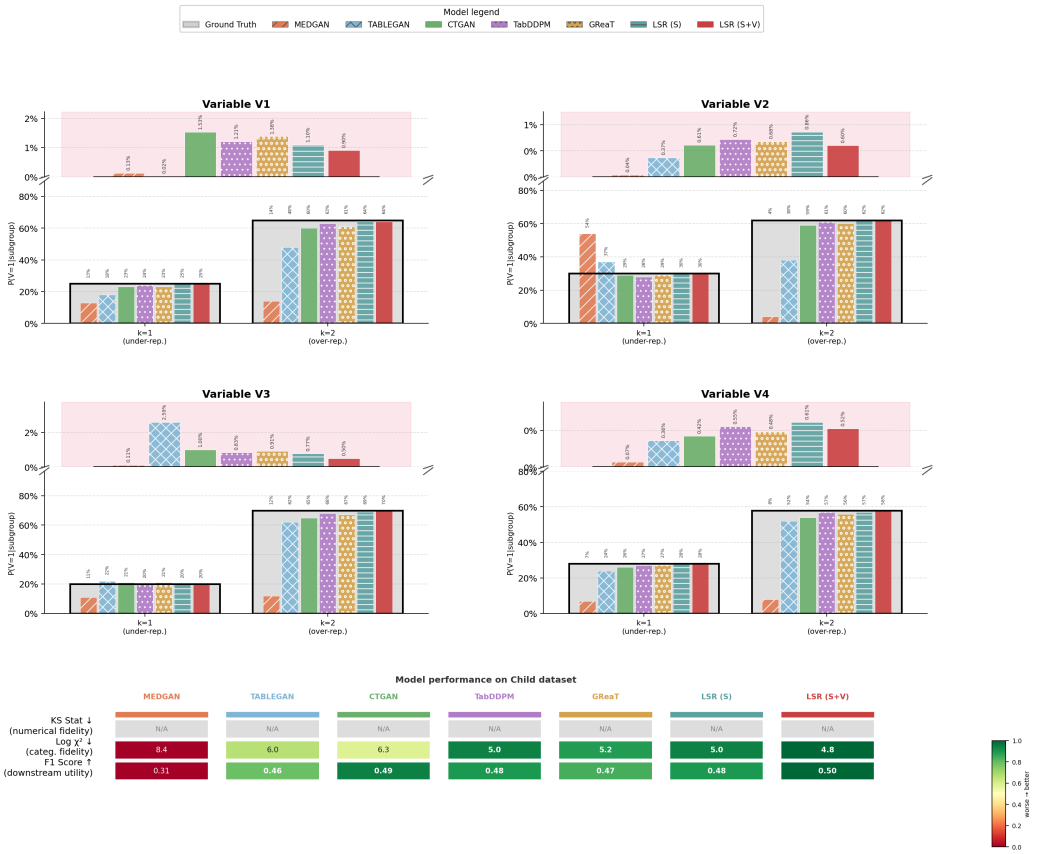

Figure 2. Distribution of synthetic variables (V_1–V_4) within subgroups for the augmented *Child* dataset. Black bars denote ground-truth frequencies; red-shaded columns mark zero-probability (forbidden) regions. *Bottom*: aggregate performance across Log χ^2 (categorical fidelity), F1-score (downstream utility), and KS statistic (numerical fidelity, N/A here). Colors are normalized per metric from red (worst) to green (best). Low violation rates in MedGAN and TableGAN are artifacts of mode collapse. CTGAN's marginal advantage on V_4 comes at the cost of lower aggregate fidelity and utility. LSR (S+V) preserves subgroup coverage while remaining near the top on aggregate utility.

Quantitative Analysis of Semantic Violations Table 6 reports the percentage of generated samples that violate the ground-truth zero-probability constraints. Lower percentages indicate better adherence to the engineered semantic rules. We report the *absolute percentage* of violations for all models. LSR (S+V) records the lowest rates on V_1, V_2, and V_3, and it remains close to the best value on V_4, where CTGAN is slightly lower. The gains over the ablated LSR (S) are modest but consistent across all four variables, indicating that the validator is responsible for most of the improvement in constraint adherence.

Examining the baselines, MedGAN and TableGAN still benefit from low-support generations on some variables, but they do so alongside poor overall utility and diversity (Tables 2-5). Among the stronger external baselines, TabDDPM is competitive on V_1 and especially V_3, and GReaT is close to LSR on V_2, so the semantic advantage is not uniform across every variable. Even so, LSR (S+V) is the only model that stays near the best result across all four variables while also preserving the stronger exploration and average utility seen earlier.

Relative to TabDDPM, LSR (S+V) remains competitive on fidelity, improves F1 on four of five datasets, has the lowest violation rate on three of four injected constraints, and achieves the highest Exploration Factor on four of five datasets.

Qualitative Analysis of Subgroup Distributions Figure 2 reveals whether low violation rates reflect genuine constraint adherence or merely a collapse of the generative support. MedGAN and TableGAN suppress entire subgroups—MedGAN nearly eliminates subgroup 1 for V_2, and TableGAN collapses coverage for V_3—so that the probability mass required to produce a violation is absent by construction. The aggregate panel confirms this, as both models rank low on data quality. CTGAN preserves broader subgroup coverage and achieves a marginal advantage on V_4, yet the aggregate panel shows it pays for this in fidelity and utility, consistent with its weaker scores in Tables 2–3. LSR (S+V) maintains close alignment with ground-truth subgroup frequencies across V_1–V_4, achieves the lowest violation rates on three of the four variables among models with nominal subgroup coverage (and stays close on V_4), and remains near the top of the aggregate utility metrics. The degradation of LSR (S) on both violation rates and subgroup fidelity isolates the validator-driven regularisation as the operative mechanism behind these gains.

Ablation: Removing the Information Loss To isolate the contribution of the information loss term $\mathcal{L}_i$ (Equation 5) from that of the semantic regularization, we train LSR (S+V) with $\lambda_{\text{info}} = 0$, keeping all other hyperparameters unchanged. The resulting variant, denoted LSR (S+V, no $\mathcal{L}_i$), appears in Tables 2–6. Removing the information loss leads to modest degradation in statistical fidelity and downstream efficacy across most datasets, reflecting the role of this term in stabilizing feature-level statistics during GAN training. Exploration decreases by roughly 5–20% relative to the full model. Semantic violation rates increase slightly but remain competitive with, and often below, the strongest external baselines, indicating that validator-driven regularization is the primary mechanism behind LSR's semantic gains. The information loss provides a complementary benefit at the feature-matching level, but it is not the source of the semantic constraint enforcement.

5. Discussion and Future Work

We introduced Latent Semantic Regularization, a principled unsupervised framework for mitigating semantic violations (*hallucinations*) in generative models. Instantiated with a VAE validator using a tail-adaptive NF prior and a GAN synthesizer, LSR reduces semantic errors while maintaining competitive data fidelity and strong generative diversity across standard benchmarks. The results highlight the value of LSR's decoupled architecture and flexible validator priors for capturing complex semantic boundaries.

Although demonstrated with a VAE/GAN pairing, LSR's modularity allows different latent models to serve as synthesizer and validator. Conditioning the validator on all categorical variables was a stringent test; in real applications, validation can be targeted to feature subsets. The penalty weight λ_{sem} directly controls the fidelity–semantics trade-off. Our sensitivity analysis (Section C.2) suggests that $\lambda_{\text{sem}} \in [0.5, 1.0]$ is a practical range for the datasets studied, with 1.0 a balanced default. LSR does not provide hard constraint-satisfaction guarantees, which would require explicit constraints, a feasibility oracle, or rejection from a verified feasible region, whereas our setting assumes implicit constraints learned only from p_{data}. Like other models learned from p_{data}, LSR depends on distribution quality: non-negligible violations in the data, or strong latent overlap between valid and invalid configurations, can limit validator separation. Finally, although synthetic data is partly motivated by data sharing, LSR provides no formal privacy guarantee; membership risk, attribute disclosure, and differentially private variants are left for future work.

A complementary future experiment is to apply the trained validator as a post-hoc filter to all synthesizers. This would test how much semantic improvement comes from rejection under the validator's learned feasibility region alone, rather than from steering the generator during training. Unlike LSR, such filtering would not change the learned generator and could affect diversity through the rejection rate. Future work could research whether the validator can do more than score whole samples. Validators with more structured latent spaces are a natural next step if this direction is pursued further.

Acknowledgements

This research has been performed as part of the *Enabling Personalized Intervention* (EPI) project. The EPI project is funded by the Dutch Science Foundation in the Commit2Data program, grant number 628.011.028.

References

Saba Amiri, Eric Nalisnick, Adam Belloum, Sander Klous, and Leon Gommans. Practical synthesis of mixed-tailed data with normalizing flows. *Transactions on Machine Learning Research*, 2024.

Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Cvae-gan: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE international conference on computer vision*, pages 2745–2754, 2017.

Jens Behrmann, Paul Vicol, Kuan-Chieh Wang, Roger Grosse, and Jörn-Henrik Jacobsen. Understanding and mitigating exploding inverses in invertible neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 1792–1800. PMLR, 2021.

Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. " best-of-many-samples" distribution matching. *arXiv preprint arXiv:1909.12598*, 2019a.

Apratim Bhattacharyya, Michael Hanselmann, Mario Fritz, Bernt Schiele, and Christoph-Nikolas Straehle. Conditional flow variational autoencoders for structured sequence prediction. *arXiv preprint arXiv:1908.09008*, 2019b.

Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators. *arXiv preprint arXiv:2210.06280*, 2022.

Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In *Machine learning for healthcare conference*, pages 286–305. PMLR, 2017.

Grigorios G Chrysos, Jean Kossaifi, and Stefanos Zafeiriou. Robust conditional generative adversarial networks. *arXiv preprint arXiv:1805.08657*, 2018.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

Dheeru Dua, Casey Graff, et al. Uci machine learning repository. 2017.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

Justin Engelmann and Stefan Lessmann. Conditional wasserstein gan-based oversampling of tabular data for imbalanced learning. *Expert Systems with Applications*, 174:114582, 2021.

Giorgio Giannone, Lyle Regenwetter, Akash Srivastava, Dan Gutfreund, and Faez Ahmed. Learning from invalid data: On constraint satisfaction in generative models. *arXiv preprint arXiv:2306.15166*, 2023.

Mingming Gong, Yanwu Xu, Chunyuan Li, Kun Zhang, and Kayhan Batmanghelich. Twin auxilary classifiers gan. *Advances in neural information processing systems*, 32, 2019.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

Priyank Jaini, Ivan Kobyzev, Yaoliang Yu, and Marcus Brubaker. Tails of lipschitz triangular flows. In *International Conference on Machine Learning*, pages 4673–4681. PMLR, 2020.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International conference on machine learning*, pages 17564–17579. PMLR, 2023.

Soochan Lee, Junsoo Ha, and Gunhee Kim. Harmonizing maximum likelihood with gans for multimodal conditional generation. *arXiv preprint arXiv:1902.09225*, 2019.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1429–1437, 2019.

Saeid Naderiparizi, Xiaoxuan Liang, Berend Zwartsenberg, and Frank Wood. Don't be so negative! score-based generative modeling with oracle-assisted guidance. *arXiv preprint arXiv:2307.16463*, 2023.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pages 2642–2651. PMLR, 2017.

Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *arXiv preprint arXiv:1806.03384*, 2018.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

Mihaela Rosca, Balaji Lakshminarayanan, and Shakir Mohamed. Distribution matching in variational inference. *arXiv preprint arXiv:1802.06847*, 2018.

Donald B Rubin. Statistical disclosure limitation. *Journal of official Statistics*, 9(2):461–468, 1993.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

David J Spiegelhalter. Learning in probabilistic expert systems. *Bayesian statistics*, 4:447–465, 1992.

Mihaela Cătălina Stoian, Salijona Dyrmishi, Maxime Cordy, Thomas Lukasiewicz, and Eleonora Giunchiglia. How realistic is your synthetic data? constraining deep generative models for tabular data. *arXiv preprint arXiv:2402.04823*, 2024.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

Jingyi Xu, Hieu Le, and Dimitris Samaras. Assessing sample quality via the latent space of generative models. *arXiv preprint arXiv:2407.15171*, 2024.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems*, 32, 2019.

Dingdong Yang, Seunghoon Hong, Yunseok Jang, Tianchen Zhao, and Honglak Lee. Diversity-sensitive conditional generative adversarial networks. *arXiv preprint arXiv:1901.09024*, 2019.

Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y Chen. Ctab-gan: Effective table data synthesizing. In *Asian Conference on Machine Learning*, pages 97–112. PMLR, 2021.

Zachary Ziegler and Alexander Rush. Latent normalizing flows for discrete sequences. In *International Conference on Machine Learning*, pages 7673–7682. PMLR, 2019.

A. Previous Works

Implicit Constraints In the literature the integration of generative models to address inherent limitations has been extensively explored. VAEs, for example, struggle with marginal distributions (Rosca et al., 2018) and have been paired with other generative models, e.g. GANS, to overcome this issue. (Dumoulin et al., 2016), (Donahue et al., 2016) combine GANs and VAEs to improve performance on discriminative tasks. These VAE-GAN hybrids aim to improve discriminative tasks (Dumoulin et al., 2016; Donahue et al., 2016), reframe the KL term using discriminators (Makhzani et al., 2015; Tolstikhin et al., 2017), address mode collapse (Bhattacharyya et al., 2019a), or enable finer generation control (Bao et al., 2017). None, however, explicitly target improving semantic integrity. Our dichotomous formulation for learning and imposing semantic boundaries offers distinct flexibility in model choice and validator influence.

Implicit Constraints for Improved Semantic Integrity While integration of explicit constraints has been previously studied (Stoian et al., 2024), we consider the task of implicit latent constraints, where the constraints are unknown and need to be both learned and applied to the generative model. Park et al. (2018) propose a tabular data generation method based on DCGAN add a classifier to the model, with its loss added to the generator objective function - alongside an information loss term - to increase semantic integrity of the generated records. However, they don't provide an examination of the semantic integrity gain of their method. In this work, we provide an examination of the semantic integrity of this method along a select group of widely cited and used methods for comparison. Both (Zhao et al., 2021) and (Engelmann and Lessmann, 2021) also use auxiliary classifier to train a GAN for tabular data generation. However, none of these methods explicitly mention using the auxiliary classifier to elevate the semantic integrity of the generated samples.

Flow-Based Hybrid Models with Implicit Constraints Previous research has combined normalizing flows with VAEs for addressing discrete sequence modeling (Ziegler and Rush, 2019) and structured sequence prediction (Bhattacharyya et al., 2019b). Our approach distinctively employs tail-adaptive flow to capture joint tail behavior of the validator prior, significantly impacting the synthesizer's semantic integrity.

Advances in Conditional GANs and Diversity Recent GAN advancements focus on diversity and conditional fidelity, primarily for images. DSCGAN (Yang et al., 2019) and MSGAN (Mao et al., 2019) encourage mode coverage and diversity. RCGAN (Chrysos et al., 2018) enhances robustness against noisy conditioning labels. HM-GAN (Lee et al., 2019) balances adversarial loss with maximum likelihood for multimodal conditional tasks. These differ fundamentally from our LSR approach, which specifically addresses *implicit semantic integrity* through a distinct, probabilistically-grounded validator learning semantic constraints unsupervisedly and imposing them as soft regularizing bounds (Eq. 3) on a separate synthesizer.

Non-Adversarial Tabular Synthesis Recent work has moved beyond GANs for tabular data generation. TabDDPM (Kotelnikov et al., 2023) adapts denoising diffusion probabilistic models to the tabular setting, achieving strong distributional fidelity through iterative denoising. GReaT (Borisov et al., 2022) leverages pre-trained large language models by fine-tuning them to generate tabular rows serialized as text, exploiting the models' capacity for capturing feature dependencies. While both methods produce high-fidelity samples, neither incorporates an explicit mechanism for learning or enforcing semantic constraints. Their improvements stem from better density estimation rather than targeted semantic regularization, leaving semantic integrity to be addressed only implicitly through distributional fit.

B. A Concrete Realization of LSR

B.1. Validator: CVAE with Tail-Adaptive NF Prior

The LSR validator V_ϕ is instantiated as a Conditional Variational Autoencoder (CVAE) (Sohn et al., 2015). The encoder $q_\phi(\mathbf{z}'|\mathbf{x},\mathbf{y})$ approximates the true posterior $p(\mathbf{z}'|\mathbf{x},\mathbf{y})$, and the learned prior $p_\eta(\mathbf{z}'|\mathbf{y})$ serves as an operational model of semantic validity in $\mathcal{Z}'$. Training maximizes the conditional ELBO:

$$\mathcal{L}_{\text{ELBO}}(\phi,\psi,\eta;\mathbf{x},\mathbf{y}) = \mathbb{E}_{q_\phi(\mathbf{z}'|\mathbf{x},\mathbf{y})}[\log p_\psi(\mathbf{x}|\mathbf{z}',\mathbf{y})] - KL[q_\phi(\mathbf{z}'|\mathbf{x},\mathbf{y}) \parallel p_\eta(\mathbf{z}'|\mathbf{y})] \leq \log p(\mathbf{x}|\mathbf{y}). \quad (8)$$

The KL term aligns the approximate posterior with the prior over the training data, so the prior concentrates mass on latent codes corresponding to real, semantically valid samples.

Deriving $R_{\text{sem}}^{\text{VAE}}$. The general penalty R_{sem} (Eq. 2) is instantiated using the stochastic encoder q_ϕ and the learned prior p_η:

$$R_{\text{sem}}^{\text{VAE}}(\mathbf{x}';\phi,\eta) = \mathbb{E}_{\mathbf{z}'\sim q_\phi(\mathbf{z}'|\mathbf{x}',\mathbf{y}')}[-\log p_\eta(\mathbf{z}'|\mathbf{y}')]. \quad (9)$$

Decomposing the ELBO's KL term reveals the relationship between this penalty and the encoder entropy:

$$KL[q_\phi(\mathbf{z}'|\mathbf{x}',\mathbf{y}') \parallel p_\eta(\mathbf{z}'|\mathbf{y}')] = R_{\text{sem}}^{\text{VAE}}(\mathbf{x}';\phi,\eta) - \mathbb{H}[q_\phi(\mathbf{z}'|\mathbf{x}',\mathbf{y}')], \quad (10)$$

where $\mathbb{H}[\cdot]$ is the differential entropy. Minimizing this KL during VAE training aligns $q_\phi(\mathbf{z}'|\mathbf{x},\mathbf{y})$ with $p_\eta(\mathbf{z}'|\mathbf{y})$, so generated samples $\mathbf{x}'$ whose encoded distributions deviate from this prior incur a large $R_{\text{sem}}^{\text{VAE}}$, imposing a differentiable semantic penalty via Eq. 3.

Tail-Adaptive Normalizing Flow Prior. A fixed isotropic Gaussian prior cannot capture the multi-modal, heavy-tailed structure that complex semantic latent spaces require. Rare but valid configurations occupy the tails of $\mathcal{Z}'$; a Gaussian cannot distinguish these from genuine violations, blunting $R_{\text{sem}}^{\text{VAE}}$.

We therefore parameterize $p_\eta(\mathbf{z}'|\mathbf{y})$ via a normalizing flow (Rezende and Mohamed, 2015): an invertible map $\mathbf{z}' = f_\eta(\mathbf{z}'_0)$ transforms a simple base distribution $p_0(\mathbf{z}'_0|\mathbf{y})$, yielding

$$\log p_\eta(\mathbf{z}'|\mathbf{y}) = \log p_0(f_\eta^{-1}(\mathbf{z}')|\mathbf{y}) + \log\left|\det\left(\frac{\partial f_\eta^{-1}}{\partial \mathbf{z}'}\right)\right|. \quad (11)$$

Using the change-of-variables $\mathbf{z}' = f_\eta(\mathbf{z}'_0)$, the log-prior under the flow (Eq. 11) can be substituted into the standard ELBO (Eq. 8), giving

$$\mathcal{L}_{\text{ELBO}}^{\text{NF}} = \mathbb{E}_{q_\phi(\mathbf{z}'|\mathbf{x},\mathbf{y})}[\log p_\psi(\mathbf{x}|\mathbf{z}',\mathbf{y})] - KL[q_\phi(\mathbf{z}'|\mathbf{x},\mathbf{y}) \parallel p_\eta(\mathbf{z}'|\mathbf{y})], \quad (12)$$

where $\log p_\eta(\mathbf{z}'|\mathbf{y})$ is evaluated via Eq. 11:

$$\log p_\eta(\mathbf{z}'|\mathbf{y}) = \log p_0\big(f_\eta^{-1}(\mathbf{z}')|\mathbf{y}\big) + \log\left|\det\left(\frac{\partial f_\eta^{-1}}{\partial \mathbf{z}'}\right)\right|. \quad (13)$$

Both expectations are taken with respect to $\mathbf{z}' \sim q_\phi(\mathbf{z}'|\mathbf{x},\mathbf{y})$, keeping all quantities in the same $\mathbf{z}'$ space. The objective is jointly optimized over (ϕ,ψ,η).

Standard normalizing flows still struggle with sharp semantic boundaries and heavy tails (Behrmann et al., 2021; Jaini et al., 2020). We therefore specifically use a *tail-adaptive normalizing flow* (Amiri et al., 2024), explicitly designed to model varying tail behaviors across latent dimensions. This allows the prior to assign appropriate density to rare but valid configurations in the tails, while keeping semantically invalid encodings in genuinely low-density regions, sharpening $R_{\text{sem}}^{\text{VAE}}$ without over-penalizing diversity.

B.2. Synthesizer: WGAN-GP with LSR

The synthesizer is a conditional WGAN-GP (Gulrajani et al., 2017) augmented with an AC-GAN auxiliary classifier (Odena et al., 2017; Gong et al., 2019) and an information loss (Park et al., 2018). Let $G_\theta : \mathcal{Z} \times \mathcal{Y} \to \mathcal{X}$ be the generator and $D_\omega : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be the critic.

Critic objective.

$$\mathcal{L}_D(\omega) = \underbrace{\mathbb{E}_{\tilde{\mathbf{x}}\sim\mathbb{P}_{G_\theta}}[D_\omega(\tilde{\mathbf{x}}|\mathbf{y})] - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{data}}}[D_\omega(\mathbf{x}|\mathbf{y})] + \lambda_{gp}\,\mathbb{E}_{\hat{\mathbf{x}}}\left[(\|\nabla_{\hat{\mathbf{x}}} D_\omega(\hat{\mathbf{x}}|\mathbf{y})\|_2 - 1)^2\right]}_{\mathcal{L}_{D,\text{WGAN}}(\omega)} + \lambda_{ac,d}\,\underbrace{\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{data}}}[\mathcal{L}_{CE}(A_\omega(\mathbf{x}|\mathbf{y}), \mathbf{y})]}_{\mathcal{L}_{D,\text{AC}}(\omega)}, \tag{14}$$

where $\hat{\mathbf{x}}$ is sampled uniformly along straight lines between real and generated points, and A_ω denotes the auxiliary classifier head of the critic.

Generator base objective. The generator minimizes

$$\mathcal{L}_{G,\text{base}}(\theta) = \underbrace{-\mathbb{E}_{\tilde{\mathbf{x}}}[D_\omega(\tilde{\mathbf{x}}|\mathbf{y})]}_{\mathcal{L}_{G,\text{WGAN}}} + \lambda_{ac,g}\,\underbrace{\mathbb{E}_{\tilde{\mathbf{x}}}[\mathcal{L}_{CE}(A_\omega(\tilde{\mathbf{x}}|\mathbf{y}), \mathbf{y})]}_{\mathcal{L}_{G,\text{AC}}} + \lambda_{\text{info}}\,\mathcal{L}_i(\theta), \tag{15}$$

where the information loss matches first and second moments of an intermediate critic layer between real and generated batches:

$$\mathcal{L}_i(\theta) = \|\mathbb{E}_{\mathbf{x}}[\mathbf{v}(\mathbf{x}|\mathbf{y})] - \mathbb{E}_{\tilde{\mathbf{x}}}[\mathbf{v}(\tilde{\mathbf{x}}|\mathbf{y})]\|_2 + \|\text{Std}_{\mathbf{x}}[\mathbf{v}(\mathbf{x}|\mathbf{y})] - \text{Std}_{\tilde{\mathbf{x}}}[\mathbf{v}(\tilde{\mathbf{x}}|\mathbf{y})]\|_2\,. \tag{16}$$

LSR-regularized objective. Adding the frozen validator's semantic penalty (Eq. 4) gives the full objective from the main paper (Eq. 6):

$$\mathcal{L}_G(\theta) = \mathcal{L}_{G,\text{base}}(\theta) + \lambda_{\text{sem}}\,\mathbb{E}_{\mathbf{z},\mathbf{y}}\Big[\mathbb{E}_{\mathbf{z}'\sim q_\phi(\mathbf{z}'|G_\theta(\mathbf{z},\mathbf{y}),\mathbf{y})}[-\log p_\eta(\mathbf{z}'|\mathbf{y})]\Big], \tag{17}$$

with validator parameters (ϕ, η) held fixed throughout.

C. Experimental Details

C.1. Architecture and Training Configuration

The validator encoder and decoder are MLPs with three hidden layers of 128 neurons each. The latent space dimensionality is $\dim(\mathcal{Z}') = 200$. The tail-adaptive NF prior is a Real NVP (Dinh et al., 2016) of depth 4 with a 20-component mixture of Generalized Gaussians as the base density; coupling layers are MLPs with a single hidden layer of 128 neurons. The synthesizer generator and critic are MLPs with two hidden layers of width 256.

Computational overhead. The validator is trained once and then frozen. During synthesizer training, LSR adds only a forward pass through the frozen encoder E_ϕ and prior p_η to compute $R_{\text{sem}}^{\text{VAE}}$ and therefore no validator parameters are updated. In our WGAN-GP setting, this overhead is small relative to the gradient-penalty computation, which requires a backward pass through the critic.

All GAN-based models (MedGAN, CTGAN, TableGAN, and LSR variants) are trained for 300 epochs with the Adam optimizer at a learning rate of 2×10^{-4}. Standard public implementations are used for all baselines. TabDDPM is run with its default diffusion schedule, MLP backbone, and recommended step-based optimizer configuration. GReaT fine-tunes DistilGPT-2 with AdamW, a language-model learning rate, linear warmup/decay scheduling, and early stopping on validation loss; both non-GAN baselines follow their native training schedules rather than the GAN epoch budget.

C.2. Sensitivity Analysis of λ_{sem}

The regularization weight λ_{sem} governs the trade-off between the synthesizer's base objective and semantic regularization. We study the sensitivity of LSR to this parameter by varying $\lambda_{\text{sem}} \in \{0, 0.01, 0.1, 0.5, 1.0, 2.0, 5.0\}$ on three datasets (Adult, Intrusion, Child) and measuring downstream F1-score, exploration factor and average semantic violation rate. The case $\lambda_{\text{sem}} = 0$ corresponds to the ablated synthesizer LSR (S). All other hyperparameters are held fixed. Results are reported in Table 7.

Table 7. Sensitivity of LSR to λ_{sem} on three datasets. We report mean $\pm$ standard error for downstream F1-score (↑), exploration factor S_d (↑), and average semantic violation rate (%) on the augmented Child dataset (↓). The setting $\lambda_{\text{sem}} = 0$ reduces to the ablated synthesizer. Minor numerical differences between these results and the main results are due to independent random seed draws across experimental runs.

	F1-Score ↑			Exploration (S_d) ↑			Avg. Violation (%) ↓
λ_{sem}	Adult	Intrusion	Child	Adult	Intrusion	Child	Child
0 (ablation)	0.63 ± 0.02	0.90 ± 0.02	0.48 ± 0.02	10180 ± 880	11940 ± 980	1580 ± 140	0.76 ± 0.08
0.01	0.63 ± 0.02	0.90 ± 0.02	0.48 ± 0.02	10320 ± 910	12010 ± 1030	1615 ± 148	0.75 ± 0.08
0.1	0.63 ± 0.02	0.91 ± 0.02	0.49 ± 0.02	10980 ± 960	12440 ± 1000	1870 ± 168	0.70 ± 0.07
0.5	0.64 ± 0.02	$\mathbf{0.92 \pm 0.02}$	0.49 ± 0.02	11690 ± 1170	12890 ± 1120	2320 ± 176	0.65 ± 0.07
1.0	$\mathbf{0.65 \pm 0.02}$	0.92 ± 0.013	$\mathbf{0.50 \pm 0.02}$	$\mathbf{12140 \pm 1240}$	$\mathbf{13110 \pm 1080}$	$\mathbf{2490 \pm 168}$	0.62 ± 0.07
2.0	0.63 ± 0.02	0.91 ± 0.02	0.49 ± 0.02	11380 ± 1210	12520 ± 1160	2260 ± 182	$\mathbf{0.60 \pm 0.06}$
5.0	0.61 ± 0.03	0.88 ± 0.03	0.47 ± 0.02	10220 ± 1300	11690 ± 1270	1940 ± 208	0.68 ± 0.08

At $\lambda_{\text{sem}} = 0.01$, the estimates overlap almost completely with the ablated setting, which is what we expect when the validator term is heavily down-weighted. Moving from 0.1 into the 0.5–1.0 range improves exploration and lowers the average violation rate, while Adult F1 changes only marginally and Intrusion reaches its peak plateau in this range. The pattern is therefore favorable but not strictly monotone. At $\lambda_{\text{sem}} = 2.0$, the average violation rate is lowest, but part of the diversity gain and a small amount of efficacy are given back relative to 1.0. At $\lambda_{\text{sem}} = 5.0$, both efficacy and exploration degrade, indicating over-regularization.

This non-monotonicity is expected since with very large λ_{sem}, the generator is pushed toward a narrow high-density core of the validator latent space, producing a form of support collapse analogous to the mode-collapse effects discussed in Fig. 2. Although this avoids some semantic-boundary regions, it also reduces coverage of low-frequency valid subgroups. Under the zero-probability injection protocol, such coverage loss can create a secondary violation pathway through incorrect conditional sampling.

Overall, $\lambda_{\text{sem}} \in [0.5, 1.0]$ provides the most practical operating region across the three datasets tested. The default value of 1.0 is not the unique optimum, but it offers the strongest exploration and Child efficacy with near-best semantic adherence, while avoiding the diversity loss observed at 2.0. This makes 1.0 a reasonable default rather than a universal optimum.

Exploration Factor validation. As a construct check, we compare S_d with two direct coverage statistics over the dependency-relevant categorical feature set M: the number of unique generated combinations and the empirical entropy of the joint marginal over those combinations. Across models, higher S_d tracks both larger support coverage and higher joint entropy, indicating that the metric captures meaningful exploration rather than arbitrary deviation from the training set.

C.3. Visualizing the Validator's Role: Impact of Prior Flexibility

To understand how the validator enables this improvement, we investigate the structure of its learned latent semantic space under different prior choices, using the same 2D dataset. We compare the validator component trained with a standard Gaussian prior versus one trained with our flexible Tail-Adaptive NF prior. We then visualize the latent representations $z' = E_\phi(x')$ for generated samples.

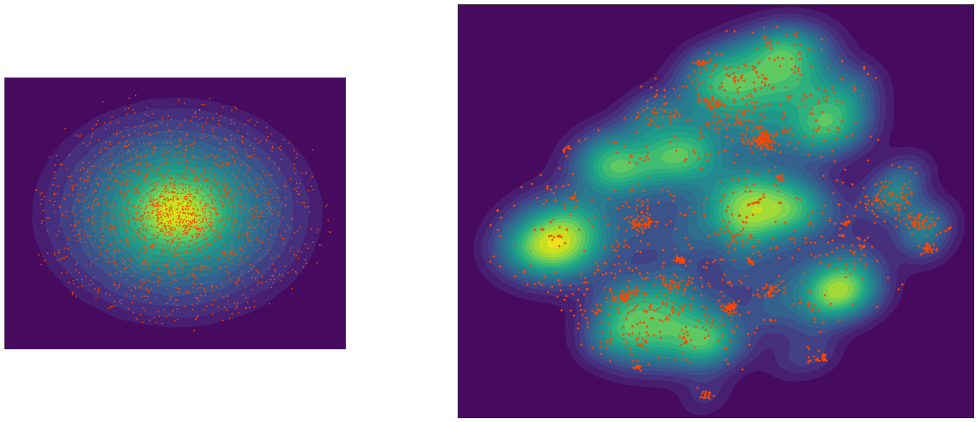

Figure 3. Comparative visualization of the validator's latent space for semantically incorrect samples (red dots) from the 2D experiment. **Left:** Validator with a standard Gaussian prior shows poor separation. **Right:** Validator with a Tail-Adaptive NF prior effectively clusters incorrect samples in low-density regions, enabling stronger semantic regularization.

Table 8. Effect of validator prior choice on LSR performance. "Gauss" denotes a standard Gaussian prior, "NF" the Tail-Adaptive NF prior used throughout this work. Entries report mean ± standard error across five runs. Avg. Violation is the mean violation percentage over the injected zero-probability semantic rules. All other hyperparameters are identical.

	Adult		Intrusion		Child	
Metric	Gauss	NF	Gauss	NF	Gauss	NF
F1-Score ↑	0.62 ± 0.02	$\mathbf{0.64 \pm 0.01}$	0.89 ± 0.02	$\mathbf{0.92 \pm 0.02}$	$\mathbf{0.51 \pm 0.01}$	0.50 ± 0.01
Exploration (S_d) ↑	$\mathbf{13820 \pm 960}$	13690 ± 1040	15480 ± 1040	$\mathbf{16520 \pm 720}$	2320 ± 150	$\mathbf{2570 \pm 132}$
Avg. Violation (%) ↓	0.83 ± 0.08	$\mathbf{0.65 \pm 0.07}$	0.11 ± 0.03	$\mathbf{0.08 \pm 0.02}$	0.88 ± 0.09	$\mathbf{0.67 \pm 0.07}$

Figure 3 presents this comparison. With a Gaussian prior, the latent codes of incorrectly labeled samples are scattered throughout the latent space mixed with that of correctly labeled samples. This lack of separation indicates that the validator, constrained by the simplistic prior, fails to learn a latent structure that effectively distinguishes semantically valid from invalid samples. In contrast, when using the Tail-Adaptive NF prior, the validator learns a more structured latent space. The incorrectly labeled samples are now predominantly clustered in lower-density regions of the learned prior, distinctly separated from the high-density regions corresponding to semantically correct samples. This improved separation implies that the flexible prior allows the validator to better capture the subtle distributional shifts associated with semantic violations, highlighting the importance of using an expressive, learnable prior within the validator component to accurately model semantic boundaries.

C.4. Quantitative Prior Comparison on Real Datasets

The qualitative advantage of the NF prior observed on toy data extends to the benchmark datasets. We retrain the validator with a standard Gaussian prior, keeping the synthesizer, architecture, and all other hyperparameters fixed, and compare the resulting LSR system against the default NF-prior variant. Table 8 reports downstream F1-score, exploration factor, and average semantic violation rate on Adult, Intrusion, and Child. Here, Avg. Violation is the mean percentage of generated samples that violate the injected zero-probability semantic rules for each dataset.

Across the three datasets, the NF prior improves semantic violation rates consistently and yields better downstream efficacy on Adult and Intrusion, while Child efficacy is effectively tied. The

exploration gains are also dataset-dependent: NF is clearly better on Intrusion and Child, whereas Adult shows overlapping estimates with a slight Gaussian edge. The strongest and most stable effect of the flexible prior is therefore on semantic adherence rather than on every metric in every setting, which still supports using the NF prior as the default validator component.