Proceedings of Machine Learning Research vol 284:1–34, 2026 20th Conference on Neurosymbolic Learning and Reasoning

Generate in the Chart, Not on the Boundary: Function-Symbol Grounding for Hard Constraints in LTN-GANs

Nijesh Upreti N.UPRETI@ED.AC.UK
Vaishak Belle VBELLE@ED.AC.UK
The University of Edinburgh, 10 Crichton Street, Edinburgh, EH8 9AB, UK

Editors: Alessandra Mileo, Andrea Passerini and Cogan Shimizu

Abstract

Logic Tensor Network-Enhanced Generative Adversarial Networks (LTN-GANs) inject background knowledge by grounding each logical axiom as a predicate and training the generator to raise its satisfaction, a fuzzy truth value in $[0, 1]$. Previous LTN-GAN work grounded every constraint this way, at the predicate level, and improved constraint satisfaction. A predicate, however, only scores a sample, so it cannot embed hard structural constraints, rules such as orderings, positivity, and definitional identities that must hold in every generated sample. In this work, we investigate grounding each axiom as a function symbol inside the LTN framework. We compare against the state-of-the-art alternative, a constraint layer that clamps each violating sample onto the feasible boundary and so produces outputs that are always valid. Our investigation shows that a valid sample is not always a realistic one. An inequality is not merely satisfied or violated. It holds by a margin, and a faithful generator should also reproduce the margin's real distribution. We find that the resolution ratio R, the data's scale over the margin's spread, is a diagnostic, computable before training, of which constraints a chosen grounding can learn. When R is large, the predicate receives no learning signal, the clamp pushes every sample onto the boundary, and the margin distribution is lost while every standard metric still looks fine. A function symbol avoids both failures, computing the constrained variable rather than scoring it. Together the function symbols form a chart, a coordinate system inside the feasible region, where every sample is valid by construction and the margin is learned like any other quantity. This recovers the margin distribution on four high-resolution datasets, with Kolmogorov–Smirnov distance up to 25× smaller than the constraint layer's, and a hybrid of charting and clamping matches or exceeds the constraint layer on its own benchmark.

1. Introduction

Deep generative models are a standard tool for producing synthetic tabular and scientific data, used to augment scarce records, share sensitive data, and propose candidate designs. Adversarial and variational generators such as CTGAN and TVAE (Xu et al., 2019), score-based and diffusion models (Kim et al., 2023; Kotelnikov et al., 2023), and relational-structure models (Liu et al., 2023) learn increasingly accurate approximations of the distributions of data such as trip records, flight records, and molecular property profiles.

However, approximating the data distribution well is not enough, because such data obey known rules that an approximation can still violate. For example, a synthetic trip record must have its drop-off after its pick-up, a flight must report its departure as schedule plus delay, and a molecule's internal energy must rise from 0 K (U_0) to room temperature (U)

and remain below its enthalpy (H), so that $U_0 < U < H$. *Constrained generation* therefore asks for samples that are both realistic and provably valid under such rules, which in tabular and scientific data are typically linear orderings between properties, positivity requirements, and definitional identities. Two main families of methods inject this knowledge into a generator. The first adds a penalty to the training loss whenever a sample violates a constraint. Such a constraint is called *soft*, since the generator can trade it against the rest of the loss, and satisfaction is encouraged but never guaranteed. The Logic Tensor Network-Enhanced Generative Adversarial Network (LTN-GAN) (Upreti and Belle, 2026) *grounds* each constraint as a *predicate*, a differentiable function that scores how well a sample satisfies it, and trains the generator to raise this score alongside its adversarial objective (Badreddine et al., 2022). The second family builds the constraints into the model, so that a violating sample cannot be produced. Such a constraint is called *hard*, since no output can break it. The differentiable *constraint layer* of Stoian et al. (2024), which turns a generator into a Constrained Deep Generative Model (C-DGM) and is the state of the art for tabular data, moves each violating sample into the feasible region by clamping and guarantees 100% validity for any conjunction of linear inequalities.

Our starting observation is that validity alone does not make constrained data realistic. A valid sample satisfies each inequality by some amount, and validity ignores how these amounts are distributed. In a trip record, the time from pick-up to drop-off is the trip's duration. A generator can place every drop-off after its pick-up yet produce durations unlike those of real trips. The energy step $U - U_0$ can fail the same way. We call each such amount (drop-off minus pick-up, or $U - U_0$) the constraint's *margin*, and we call its distribution over the real data the *distribution of the constraint margin*, which a faithful generator must reproduce. Whether a generator can reproduce this distribution depends on the *resolution ratio* R, the data's scale over the margin's spread, computed before training. When R is small, the margin is visible at the data's scale, so an unconstrained generator places it, the satisfaction score has usable gradient, and a clamp rarely fires. When R is large, the margin is invisible at the data's scale. For thermochemical energies the step $U - U_0$ is five orders of magnitude smaller than the energies themselves. In this regime the discriminator can no longer resolve the margin, the satisfaction score has vanishing gradient, and the generator violates the constraint on a constant fraction of samples. Trained in the loop, the constraint layer moves nearly every sample onto the boundary, collapsing the margin distribution to a point mass while validity reads 100% and every per-feature statistic still matches the real data (margin Kolmogorov–Smirnov (KS) distance near 1; Figure 2). High-R constraints are common in scientific data, yet no standard metric detects the collapse.

We show that the LTN framework already contains the mechanism to avoid this collapse. A constraint can also be grounded through a *function symbol*, a term that computes the constrained variable directly. For an ordering $b > a$, the generator emits a free value for a and assembles b from a by adding a positive, smoothly parameterised increment. Every sample satisfies the constraint *by construction*, and the margin becomes a coordinate that the discriminator can resolve and shape at unit scale. The function symbols form a *chart* of the feasible region, a coordinate system in which the generator produces samples inside the region rather than on its boundary. The chart's per-variable admissible intervals are exactly those the constraint layer computes by Fourier-Motzkin reduction, and the constraint layer becomes the special case that always clamps. We call the result the *FSG-LTN-GAN*, for

function-symbol grounding (FSG) in LTN-GANs. Since charting helps on high-R continuous constraints while low-R and discrete margins are better clamped, we use a short pre-run to decide per constrained variable, yielding a **hybrid** model. We show that R acts as a condition number for the constrained density-estimation problem (Section 4).

Contributions. (i) We show that hard-constrained generation can distort the *distribution of the constraint margin* while passing every standard metric, and that the resolution ratio R acts as a condition number that predicts this failure before training. (ii) We develop *function-symbol grounding*, a change of coordinates that generates inside the feasible region with exact validity and subsumes the constraint layer as its clamp-everything special case, and extend it to a per-constraint *hybrid*. (iii) We demonstrate that function-symbol grounding cuts the margin KS by up to 25× at equal validity on four real high-resolution datasets, that the hybrid matches or exceeds the constraint layer on the benchmark of Stoian et al. (2024), that the R-based prediction holds out of sample (nycflights13), and that the method transfers unchanged to CTGAN and TVAE backbones.

2. Background: Logic Tensor Networks and LTN-GANs

Real Logic and grounding. Logic Tensor Networks (LTN) interpret a first-order language, Real Logic, in real-valued tensors (Badreddine et al., 2022). A *grounding* $\mathcal{G}$ assigns meaning to symbols, and every term becomes a tensor. Each *predicate* P becomes a map $\mathcal{G}(P)$ into the truth interval $[0,1]$ that *scores* its arguments. Each *function symbol* f of arity k becomes a real map $\mathcal{G}(f) : \mathbb{R}^{Dk} \to \mathbb{R}^D$, fixed or learnable, that *computes* a term. A predicate acts on learning only through the truth values it contributes, while a function symbol shapes the object being scored. Logical connectives become fuzzy operators (a t-norm for $\wedge$, its dual for $\vee$, a fuzzy implication), and quantifiers become aggregations ($\forall$ as a generalized mean over errors). The truth value of a closed formula under $\mathcal{G}$ is its *satisfaction* $\mathrm{Sat} \in [0,1]$, with 1 fully true. An *axiom* is a closed formula asserted to hold of the domain, a knowledge base KB is a finite set of axioms, and learning maximizes their aggregated satisfaction Sat(KB).

The LTN-GAN objective. A GAN couples a generator $G_\theta : \mathcal{Z} \to \mathbb{R}^D$, mapping latent noise $\zeta \sim p_\zeta$ (a standard Gaussian) to a sample, and a discriminator $D_\psi : \mathbb{R}^D \to [0,1]$, trained to score real samples near 1 and generated ones near 0, while G_θ is trained to fool it. An LTN-GAN treats the generated sample as the grounding of the constrained variables and trains the generator to also satisfy a knowledge base, using the objective $\mathcal{L}_G = \mathcal{L}_{\mathrm{adv}}(G_\theta, D_\psi) + \lambda\,(1 - \mathrm{Sat}(\mathrm{KB}))$, whose axioms encode the constraints.

Predicate grounding of a constraint. The standard generator-side LTN-GAN (G-LTN-GAN) (Upreti and Belle, 2026) grounds an ordering axiom $b > a$ as a predicate, $\mathcal{G}(P_>)(a,b) = \sigma\big((b-a)/s\big)$ with σ the logistic, at a band s (a width hyperparameter), and the term $\lambda(1 - \mathrm{Sat})$ in the objective pushes samples toward $b > a$. This grounding is *soft*. It encourages, but does not guarantee, satisfaction, and its gradient is informative only where the predicate is unsaturated. Section 4 shows this band covers a fraction $\Theta(1/R)$ of samples, so the usable gradient vanishes as R grows (RQ4). Our method keeps the LTN-GAN objective but regrounds each structural axiom through a function symbol. The constrained variable becomes a term that $\mathcal{G}(f)$ computes rather than a free output that

a predicate scores, so Sat = 1 holds by construction. *Soft* thus refers to the satisfaction signal, since function-symbol grounding also uses smooth links yet satisfies the axiom for every output.

3. Problem Statement

Let p_X be an unknown distribution over $X \in \mathbb{R}^D$ and $\mathcal{D}$ a dataset of N i.i.d. samples. A *sample* is a vector $x = (x_1, \ldots, x_D)$ whose scalar components $x_k \in \mathbb{R}$ are its *features*. A generative model with parameters θ (for us the generator G_θ) induces the distribution p_θ of its output $G_\theta(\zeta)$, $\zeta \sim p_\zeta$, and learning chooses θ so that $p_\theta \approx p_X$. The background knowledge is a finite set Π of linear-inequality axioms over the features $\{x_1, \ldots, x_D\}$, each of the form $\sum_k w_k x_k + b \trianglerighteq 0$ with $\trianglerighteq \in \{\geq, >\}$, real coefficients w_k, and offset b, following the formulation of Stoian et al. (2024). A sample $\tilde{x}$ *satisfies* $\phi \in \Pi$ if $\sum_k w_k \tilde{x}_k + b \trianglerighteq 0$. A generator is *compliant* (valid) if all its samples satisfy all of Π. Orderings ($x_i > x_j$), positivity ($x_i > 0$), and definitional identities ($x_i = \sum_j w_j x_j$, encoded as two inequalities) are the *structural* fragment we study.

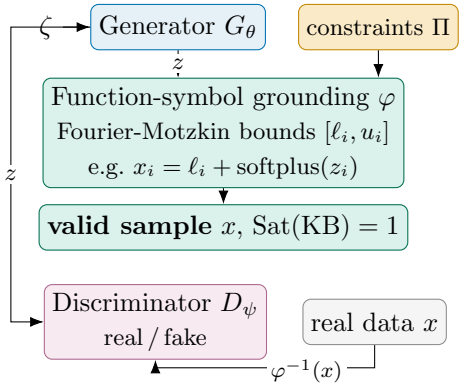

Figure 1: Overview of the FSG-LTN-GAN. The grounding map φ assembles valid samples from the generator's free coordinates z within the Fourier-Motzkin bounds of Π, and the discriminator operates in the chart coordinates. The hybrid instead clamps low-R and discrete variables in the loop (Appendix B).

The margin and its distribution. For $\phi : \sum_k w_k x_k + b \geq 0$, its *margin* on a sample is $m_\phi(x) = \sum_k w_k x_k + b$, with $m_\phi \geq 0$ exactly when ϕ holds. The right target is the *distribution of the constraint margin*, the conditional distribution of the margin under the data, $p_X(m_\phi \mid m_\phi \geq 0)$. We quantify this by the Kolmogorov–Smirnov (KS) distance between the generated and real constraint margins on a fixed reference sample of the real data.

The resolution ratio. The *spread* $\sigma_m = \text{std}_\mathcal{D}(m_\phi)$ is the standard deviation of ϕ's margin over the dataset. The *scale* $\sigma_s = \max_{k:\, w_k \neq 0} \text{std}_\mathcal{D}(x_k)$ is the largest standard deviation among the features ϕ relates: for $\phi : U - U_0 \geq 0$ it is $\text{std}(U)$. Both are computed on the raw, unstandardized data, before training. The *resolution ratio* of ϕ is $R_\phi = \sigma_s / \sigma_m$. When R_ϕ is large the margin is a tiny difference of large near-equal quantities. In standardized coordinates it occupies a band of relative width about $1/R_\phi$, sub-resolution to a Lipschitz discriminator, so a *free* generator (one trained with no constraint mechanism) that has matched the marginals still violates ϕ on a constant fraction of samples (RQ4). Section 4 casts R_ϕ as a scaling condition number for recovering the distribution of the constraint margin.

The constraint layer (clamping). C-DGM (Stoian et al., 2024) appends a differentiable *constraint layer* (CL): given a variable ordering, it computes for each variable an admissible interval $[\ell_i, u_i]$ (piecewise-linear in the already-set variables, by Fourier-Motzkin reduction)

and *clamps* the generated value into it, $\mathrm{CL}(\tilde{x})_i = \min(\max(\tilde{x}_i, \ell_i), u_i)$, guaranteeing validity. A clamp moves a violating sample to the nearest boundary, where the margin is near zero. Section 4.1 shows this is exactly where the distribution of the constraint margin is lost, at a rate governed by R.

4. Function-Symbol Grounding as a Change of Coordinates

Change of coordinates and *chart* carry their differential-geometric meaning throughout. Inside the affine subspace its identities define, the feasible set $\mathcal{M} = \{x : \bigwedge_\phi \phi(x)\}$ of a satisfiable linear system is a relatively open convex polytope, diffeomorphic to $\mathbb{R}^d$ and covered by a single global chart. The map φ below is such a chart, with one coordinate per free variable.

We ground each structural axiom through a function symbol (Figure 1). The generator emits free terms $z \in \mathbb{R}^d$ (one per variable not fixed by an identity), and a grounded map φ assembles the constrained sample by processing the variables in the Fourier-Motzkin order. With Π_i^+ (Π_i^-) the reduced constraints in which x_i has positive (negative) coefficient (Stoian et al., 2024), the admissible interval of x_i is piecewise-linear in the already-assembled $x_{<i}$,

$$\ell_i(x_{<i}) = \max_{\phi \in \Pi_i^+} \varepsilon_i^\phi(x_{<i}), \quad u_i(x_{<i}) = \min_{\phi \in \Pi_i^-} \varepsilon_i^\phi(x_{<i}), \quad \varepsilon_i^\phi = -\Big(\textstyle\sum_{k<i} w_k x_k + b\Big)/w_i, \tag{1}$$

the bounds the constraint layer clamps into (Section 3). Per variable, φ applies the grounding that fits the interval:

$$\text{one-sided: } x_i = \ell_i + \mathrm{softplus}(z_i) \text{ or } u_i - \mathrm{softplus}(z_i), \qquad \text{box: } x_i = \ell_i + (u_i - \ell_i)\,\sigma(z_i),$$
$$\text{free: } x_i = z_i, \qquad \text{identity: } x_i = \textstyle\sum_{j<i} w_{ij}\, x_j + w_{i0}.$$

(For Alchemy's $U_0 < U < H$: $U_0 = z_1$, $U = U_0 + \mathrm{softplus}(z_2)$, $H = U + \mathrm{softplus}(z_3)$.) Each bounded variable is a smooth, monotone function of a unit-scale coordinate z_i that stays within its admissible interval, and each identity derives its dependent variable, so the axiom holds by construction ($\mathrm{Sat}(\mathrm{KB}) = 1$, hence the logical term in the objective vanishes for these axioms). The decoder $\varphi : z \mapsto x$ is the *chart* of $\mathcal{M}$ defined above, turning unconstrained coordinates z into feasible samples x. It inverts in closed form (for a one-sided variable, $z_i = \mathrm{softplus}^{-1}(x_i - \ell_i)$), which gives the encoder φ^{-1} applied to real data. The discriminator operates on the chart coordinates z, where every margin is unit scale, receiving z for generated samples and $\varphi^{-1}(x)$ for real ones, so a standard discriminator can learn the distribution of the constraint margin directly. For heavy-tailed margins we ground through exp rather than softplus (a multiplicative increment), set by a fixed dynamic-range rule (Appendix G). Monotone softplus links enforcing order and positivity go back to Dugas et al. (2009). Function-symbol grounding deploys them as the groundings of structural axioms inside adversarial training.

Proposition 1 (Validity by construction) *For any satisfiable finite set Π of linear inequalities and any generator output z, the assembled sample $\varphi(z)$ satisfies Π.*

The proof (Appendix C) is the soundness of Fourier-Motzkin elimination. In the elimination order each variable's admissible interval is non-empty given the finalized earlier variables,

and softplus and σ map $\mathbb{R}$ into its interior, so every axiom is met. Validity is exact, as for the constraint layer. The two differ only in *where in the interval* the sample lands, and that is what the distribution of the constraint margin captures.

4.1. Why the chart can preserve the distribution of the constraint margin and the clamp cannot

For $\phi : b > a$, the clamp sends every violator to $b = a$, while function-symbol grounding emits the margin as a unit-scale coordinate, $m_\phi = \mathrm{softplus}(z)$. Remark 2 casts R as a condition number, and Proposition 5 quantifies both mechanisms.

Remark 2 (R as a condition number) *Recovering the distribution of the constraint margin of ϕ requires resolving a margin of scale $\sigma_m = \mathrm{std}(m_\phi)$ inside data of scale $\sigma_s = R\,\sigma_m$. A Lipschitz discriminator on the ambient coordinates must resolve a relative magnitude $1/R$. We describe this as a condition number $\Theta(R)$ in the scaling sense, versus $\Theta(1)$ in the chart, where the margin is standardized.*

Corollary 3 (Definitional identities are measure-zero) *Let $\mathcal{M}_= = \{x \in \mathbb{R}^D : c(x) = 0\}$ be the zero set of the identities, with $c : \mathbb{R}^D \to \mathbb{R}^p$ a C^1 map whose Jacobian has full rank p on $\mathcal{M}_=$. Then $\mathcal{M}_=$ is Lebesgue-null, so any generated distribution ν that is absolutely continuous has $\Pr_{x\sim\nu}[c(x) = 0] = 0$, and no satisfaction loss can raise exact satisfaction above probability 0. Function-symbol grounding derives the dependent variables from their parents, so $c \equiv 0$ holds with probability 1.*

Corollary 4 (Predicate-grounded ordering: $\Theta(1/R)$ gradient) *Ground an ordering $\phi : b > a$ as the predicate $P_s = \sigma\big((b-a)/s\big)$ with band $s = \Theta(\sigma_m)$. If the generator has matched the marginals of a and b but not their dependence (margin correlation bounded away from 1), its margin has spread $\Theta(R\,\sigma_m)$ and density $\Theta(1/\sigma_s)$ near $b = a$ (no anomalous concentration). The predicate gradient $P_s(1-P_s)/s$ is $\Theta(1/s)$ on the band $|b-a| \lesssim s$ and exponentially small outside it, so the fraction of generated samples with usable gradient is $\Theta(1/R)$, vanishing as $R \to \infty$.*

Proposition 5 (Margin laws) *Let the real margin of $\phi : b > a$ have CDF F supported on $(0,\infty)$, and clamp a free generator with violation rate v post hoc, with offset ε below the real support. (i) The clamped margin distribution is $v\,\delta_\varepsilon + (1-v)\,\nu^+$, with ν^+ the generator's satisfying-margin distribution. Under the hypotheses of Corollary 4, its KS distance to F is at least $\max(v, 1-v) - \Theta(1/R) \geq \frac{1}{2} - \Theta(1/R)$, tending to 1 as $v \to 1$ (observed in-loop; Section 5). (ii) The chart φ is a bijection from $\mathbb{R}^d$ onto the relative interior of $\mathcal{M}$ with closed-form inverse, both smooth off the measure-zero set where the active Fourier-Motzkin bound switches, places no probability on the boundary, and realizes every margin distribution on $(0,\infty)$ exactly (proofs in Appendix D).*

4.2. Which axioms to ground through function symbols: a hybrid

Function-symbol grounding is most effective where clamping is most costly, on high-R constraints the generator cannot meet on its own. But a continuous chart cannot represent a margin with a discrete point mass (an integer count, a two-valued category). The softplus

or σ increment smears it, whereas the in-loop clamp, which the generator learns to anticipate, matches it empirically (RQ3). We therefore decide per bounded variable, over the conjunction of the reduced constraints that bound it. A short pre-run of a free generator measures the fraction s_i of its samples that already lie in variable x_i's admissible interval, and the largest single-value frequency of x_i's binding margin in $\mathcal{D}$ measures the discreteness d_i. We chart x_i if $s_i < 0.9$ and $d_i \leq 0.2$ (violated and continuous). Otherwise we clamp it in the loop, with the discriminator seeing the clamped value so the generator adapts to it. The resulting generator charts the high-R continuous variables and clamps the rest. Validity remains exact (Proposition 1) since every bounded variable is either charted or clamped. Algorithm 1 (Appendix B) summarizes the procedure.

5. Experimental Analysis

We answer four questions. **RQ1:** does guaranteeing validity guarantee a realistic constrained quantity? **RQ2:** does the effect persist across generator architectures? **RQ3:** does the hybrid generalize to the constraint-layer benchmark of Stoian et al. (2024)? **RQ4:** can predicate grounding (the standard LTN-GAN satisfaction loss) reach high-R constraints instead?

Datasets. We use four real high-resolution datasets, each carrying an ordering whose margin is a small difference of large quantities: **Alchemy** (Chen et al., 2019) ($U_0 < U < H$), **tmQM** (Balcells and Skjelstad, 2020), **Transition1x** (Schreiner et al., 2022) (reaction barriers), and **Taxi** (New York City Taxi and Limousine Commission, 2024) (trip duration). Constraints and per-margin R (3.6 to $7{\times}10^6$) are in Table 4, Appendix F. For RQ3 we use the six-dataset benchmark of Stoian et al. (2024). **Baselines.** The *constraint layer* (C-DGM) on the same MLP-GAN as FSG-LTN-GAN; *CTGAN* and *TVAE* (Xu et al., 2019) with and without it; and post-hoc *projection*. **Metrics.** Validity, the *margin KS* against a fixed reference sample of the real data (Appendix E), and per-property moment error, as means $\pm$ std over n=10 seeds with paired Wilcoxon p-values. Full tables and the constraint-layer corrections are in Appendices E, H, I, and J.

5.1. RQ1: validity does not imply a realistic constrained quantity

Throughout, the constraint layer runs from the code released by Stoian et al. (2024), with corrections that only help it (including one for a latent bug in its Fourier-Motzkin reduction; Appendix E). Table 1 shows that on all four datasets the constraint layer and FSG-LTN-GAN are both 100% valid, yet the constraint layer's margin KS is 0.57 to 1.00 while FSG-LTN-GAN's is 0.04 to 0.10 ($p < 0.01$, all ten seeds). Inspecting the raw margins (Figure 2) explains the numbers. The clamp places every sample's margin at the boundary (about 10^{-6} at every quantile), a point mass disjoint from the real distribution (the collapse of Proposition 5(i)). FSG-LTN-GAN's margin quantiles track the real ones. The failure is invisible to the standard metrics we report. Per-property moment error is comparable across methods (the constraint layer is even *better* on tmQM and Transition1x), so one can match every feature's marginal while losing the distribution of the constrained quantity. The clamp's distortion tracks R per margin (KS near 1 on the high-R margins, e.g. Alchemy $U - U_0$ at $R \approx 10^5$). Taxi's lower table value averages in a low-R margin the clamp handles well, so the effect tracks R rather than the domain. A final, pre-registered test on a fifth

Table 1: **Validity does not capture the distribution of the constraint margin (RQ1).** Four real high-resolution datasets, n=10, mean ± std. Both methods are 100% valid. Only FSG-LTN-GAN recovers the distribution of the constraint margin, while per-property moment error (lower better) is comparable, so the gap is invisible to it. †paired Wilcoxon $p < 0.01$.

		margin KS ↓		per-prop. moment ↓	
Dataset	R	C-DGM	**FSG-LTN-GAN**	C-DGM	FSG-LTN-GAN
Alchemy	3×10^4 to 7×10^6	1.000 ± 0.000	$\mathbf{0.040 \pm 0.007}^\dagger$	0.472 ± 0.012	0.327 ± 0.058
tmQM	2.3×10^4	1.000 ± 0.000	$\mathbf{0.045 \pm 0.010}^\dagger$	0.183 ± 0.014	0.192 ± 0.025
Transition1x	10^3	0.999 ± 0.000	$\mathbf{0.065 \pm 0.010}^\dagger$	0.115 ± 0.036	0.196 ± 0.038
Taxi	3.6 to 1.1×10^3	0.574 ± 0.005	$\mathbf{0.099 \pm 0.010}^\dagger$	0.492 ± 0.024	0.282 ± 0.032

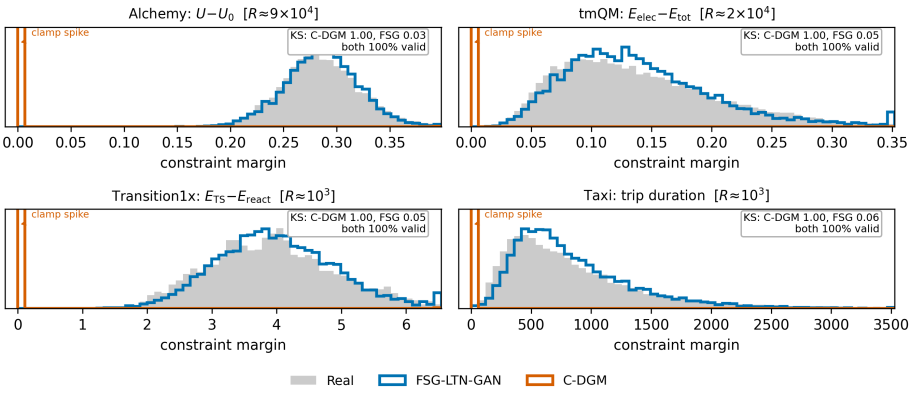

Figure 2: **The distribution of the constraint margin on all four datasets.** Real margin (grey) against C-DGM (orange, the constraint layer) and FSG-LTN-GAN (blue). All are 100% valid. The clamp collapses every margin to a boundary point mass. FSG-LTN-GAN reproduces the real distribution. The annotated KS is for the displayed high-R margin. Table 1 averages each dataset's constraints, so its Taxi value also counts the low-R total > fare margin, which the clamp handles well.

dataset outside both papers' suites (nycflights13) confirmed every prediction (Appendix J, Table 11).

Overall sample quality is preserved. On the constraint layer's own density and coverage metrics (Naeem et al., 2020), function-symbol grounding is close on the chemistry sets (density modestly favours the constraint layer) and substantially exceeds it on Transition1x and Taxi, where clamping collapses the joint distribution (coverage 0.20 and 0.70 against 0.05; Appendix J).

Table 2: **Architecture generality (RQ2).** Margin KS ↓ (n=10, mean ± std) for CTGAN+CL, TVAE+CL, the constraint layer on our MLP generator (C-DGM), and FSG-LTN-GAN. All are 100% valid. The constraint layer misses the distribution of the constraint margin on all three generator families.

Dataset	CTGAN+CL	TVAE+CL	C-DGM (MLP)	**FSG-LTN-GAN**
Alchemy	0.674 ± 0.036	0.651 ± 0.015	1.000 ± 0.000	$\mathbf{0.040 \pm 0.007}$
tmQM	0.579 ± 0.052	0.517 ± 0.011	1.000 ± 0.000	$\mathbf{0.045 \pm 0.010}$
Transition1x	0.592 ± 0.040	0.510 ± 0.053	0.999 ± 0.000	$\mathbf{0.065 \pm 0.010}$
Taxi	0.419 ± 0.024	0.348 ± 0.029	0.574 ± 0.005	$\mathbf{0.099 \pm 0.010}$

Table 3: **The constraint-layer benchmark of Stoian et al. (2024) (RQ3).** Margin KS ↓, n=10, mean ± std, all methods 100% valid. $R_{\max}$ is the dataset's largest per-constraint R. The hybrid outperforms the constraint layer (paired Wilcoxon) on faults (p=0.002, the high-R case), url (p=0.03), and wids (p=0.004), ties on heloc, lcld, news, and never underperforms it. FSG-all is worse than the constraint layer everywhere but faults, showing the selective in-loop clamp is necessary.

Dataset	$R_{\max}$	C-DGM	FSG-all	**hybrid**
faults	1.8×10^3	0.745 ± 0.014	0.117 ± 0.008	$\mathbf{0.117 \pm 0.008}$
heloc	2	0.156 ± 0.012	0.208 ± 0.005	0.158 ± 0.014
lcld	1	0.131 ± 0.008	0.657 ± 0.004	0.133 ± 0.007
url	1	0.159 ± 0.010	0.196 ± 0.018	$\mathbf{0.144 \pm 0.021}$
news	2	0.294 ± 0.008	0.318 ± 0.007	0.297 ± 0.007
wids	4	0.140 ± 0.007	0.319 ± 0.004	$\mathbf{0.133 \pm 0.008}$

5.2. RQ2: the effect persists across the architectures we test

Table 2 applies the constraint layer to the two most-cited tabular generators, CTGAN and TVAE. Both reach 100% validity but leave margin KS at 0.35 to 0.67, against 0.04 to 0.10 for FSG-LTN-GAN. The in-loop constraint layer on our MLP generator is worse still (0.57 to 1.00), consistent with the in-loop clamp removing the generator's incentive to place the sub-resolution margin. The method transfers across architectures the same way the failure does. Because the chart is a change of coordinates on the data, an *unmodified* CTGAN or TVAE trained in chart coordinates and decoded through φ is exactly valid and recovers the margin (Appendix J, Table 10).

5.3. RQ3: the hybrid matches or exceeds the constraint layer on the Stoian et al. (2024) benchmark

Does the method help on the constraint layer's own benchmark? Yes, wherever R is high. Table 3 reports all six datasets of Stoian et al. (2024), predominantly low-R tabular-ML tasks where clamping is already adequate. On the one high-R dataset, faults (bounding-box orderings on large sensor coordinates, $R_{\max} \approx 1.8 \times 10^3$), the hybrid charts every bounded

variable, coinciding with FSG-all, and improves (p=0.002, all ten seeds). On the low-R remainder the hybrid clamps and matches the constraint layer, for three wins, three ties, and zero losses across the benchmark, all at 100% validity. The FSG-all ablation, by contrast, is worse than the constraint layer everywhere but faults, since charting a point-mass or sub-resolution margin smears it. The hybrid's per-constraint selection (Section 4.2) is therefore necessary, and R predicts which datasets it helps before training. The selection is robust to its two thresholds. Only extreme grid corners that chart most of wids' constraints erode that dataset's win (Appendix J, Table 15).

5.4. RQ4: predicate grounding does not reach high-R constraints

Corollary 4 bounds a predicate grounding's usable-gradient fraction at $\Theta(1/R)$. We test three placements of the same well-scaled predicate (the generator loss, i.e. G-LTN-GAN; discriminator re-weighting; discriminator-feature augmentation). We score the per-ordering satisfaction fraction, averaged over the dataset's orderings (n=10, mean $\pm$ std): 0.58 ± 0.02, 0.51 ± 0.03, 0.14 ± 0.08 on Alchemy, where a free generator with no mechanism scores 0.49 ± 0.03, and 0.59 ± 0.03, 0.55 ± 0.08, 0.29 ± 0.07 on tmQM (free generator 0.53 ± 0.08), against 1.000 ± 0.000 for function-symbol grounding. No placement helps by more than 0.09. The failure tracks the coordinates, not the placement (Table 16).

6. Related Work

Neuro-symbolic generation and LTNs. Logic Tensor Networks train by maximizing grounded satisfaction (Badreddine et al., 2022), and related work realises logic as a differentiable loss (Xu et al., 2018; Fischer et al., 2019). All are *predicate-style* soft constraints (RQ4). Hard-constraint output layers guarantee specific fragments (Ahmed et al., 2022; Hoernle et al., 2022; Giunchiglia and Lukasiewicz, 2020). **Constrained tabular generation.** The constraint layer of Stoian et al. (2024), our primary baseline, guarantees linear inequalities by clamping. GOGGLE (Liu et al., 2023) injects only simple correlations, and CTGAN and TVAE (Xu et al., 2019) give no guarantees. Closest are constrained adversarial networks (Di Liello et al., 2020) and a non-convex constraint layer (Stoian and Giunchiglia, 2025); Appendix A surveys the rest and our function symbols' antecedents (Dugas et al., 2009; Carpenter et al., 2017).

7. Discussion and Conclusions

In an LTN-GAN the choice of grounding is decisive. Predicate grounding cannot reach high-resolution constraints, and clamping is valid but collapses the margin distribution, invisibly to standard metrics and predictably from R. Function-symbol grounding recovers that distribution with exact validity, outperforming the constraint layer on four high-resolution datasets, and the hybrid at least matches it on the benchmark of Stoian et al. (2024), with the same advantage in conditional inverse design (Appendix J).

Scope and limitations. Our charts cover the linear fragment, where Fourier-Motzkin gives the admissible intervals. Non-convex feasible sets from nonlinear or disjunctive constraints are left to future work, and discrete margins are boundary point masses the hybrid correctly clamps.

References

Kareem Ahmed, Stefano Teso, Kai-Wei Chang, Guy Van den Broeck, and Antonio Vergari. Semantic probabilistic layers for neuro-symbolic learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/c182ec594f38926b7fcb827635b9a8f4-Abstract-Conference.html.

Kareem Ahmed, Kai-Wei Chang, and Guy Van den Broeck. A pseudo-semantic loss for autoregressive models with logical constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/3accfe8332366a6f740d8740cd4cd653-Abstract-Conference.html.

Samy Badreddine, Artur d'Avila Garcez, Luciano Serafini, and Michael Spranger. Logic Tensor Networks. *Artificial Intelligence*, 303:103649, 2022. doi: 10.1016/j.artint.2021.103649. URL https://doi.org/10.1016/j.artint.2021.103649.

David Balcells and Bastian Bjerkem Skjelstad. tmQM dataset—quantum geometries and properties of 86k transition metal complexes. *Journal of Chemical Information and Modeling*, 60(12):6135–6146, 2020. doi: 10.1021/acs.jcim.0c01041. URL https://doi.org/10.1021/acs.jcim.0c01041.

Bob Carpenter, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1):1–32, 2017. doi: 10.18637/jss.v076.i01.

Xiaopeng Chao, Jiangzhong Cao, Yuqin Lu, Qingyun Dai, and Shangsong Liang. Constrained generative adversarial networks. *IEEE Access*, 9:19208–19218, 2021. doi: 10.1109/ACCESS.2021.3054822. URL https://doi.org/10.1109/ACCESS.2021.3054822.

Guangyong Chen, Pengfei Chen, Chang-Yu Hsieh, Chee-Kong Lee, Benben Liao, Renjie Liao, Weiwen Liu, Jiezhong Qiu, Qiming Sun, Jie Tang, Richard Zemel, and Shengyu Zhang. Alchemy: A quantum chemistry dataset for benchmarking AI models. *arXiv preprint arXiv:1906.09427*, 2019. doi: 10.48550/arXiv.1906.09427. URL https://arxiv.org/abs/1906.09427.

Luca Di Liello, Pierfrancesco Ardino, Jacopo Gobbi, Paolo Morettin, Stefano Teso, and Andrea Passerini. Efficient generation of structured objects with constrained adversarial networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/a87c11b9100c608b7f8e98cfa316ff7b-Abstract.html.

Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating functional knowledge in neural networks. *Journal of Machine Learning Research*, 10(42):1239–1262, 2009.

Aaron M. Ferber, Arman Zharmagambetov, Taoan Huang, Bistra Dilkina, and Yuandong Tian. GenCO: Generating diverse designs with combinatorial constraints. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 13445–13459, 2024. URL https://proceedings.mlr.press/v235/ferber24a.html.

Marc Fischer, Mislav Balunović, Dana Drachsler-Cohen, Timon Gehr, Ce Zhang, and Martin Vechev. DL2: Training and querying neural networks with logic. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 1931–1941, 2019. URL https://proceedings.mlr.press/v97/fischer19a.html.

Eleonora Giunchiglia and Thomas Lukasiewicz. Coherent hierarchical multi-label classification networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6dd4e10e3296fa63738371ec0d5df818-Abstract.html.

Eric Heim. Constrained generative adversarial networks for interactive image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10753–10761, 2019. doi: 10.1109/CVPR.2019.01101. URL https://doi.org/10.1109/CVPR.2019.01101.

Philipp Hess, Markus Drüke, Stefan Petri, Felix M. Strnad, and Niklas Boers. Physically constrained generative adversarial networks for improving precipitation fields from Earth system models. *Nature Machine Intelligence*, 4(10):828–839, 2022. doi: 10.1038/s42256-022-00540-1. URL https://doi.org/10.1038/s42256-022-00540-1.

Nick Hoernle, Rafael-Michael Karampatsis, Vaishak Belle, and Kobi Gal. MultiplexNet: Towards fully satisfied logical constraints in neural networks. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, pages 5700–5709, 2022. doi: 10.1609/aaai.v36i5.20512. URL https://doi.org/10.1609/aaai.v36i5.20512.

Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, Lianhui Qin, Xiaodan Liang, Haoye Dong, and Eric P. Xing. Deep generative models with learnable knowledge constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pages 10522–10533, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/d7e77c835af3d2a803c1cf28d60575bc-Abstract.html.

Jayoung Kim, Chaejeong Lee, and Noseong Park. STaSy: Score-based tabular data synthesis. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=1mNssCWt_v.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. TabDDPM: Modelling tabular data with diffusion models. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 17564–17579, 2023. URL https://proceedings.mlr.press/v202/kotelnikov23a.html.

Isaac E. Lagaris, Aristidis Likas, and Dimitrios I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998. doi: 10.1109/72.712178. URL https://doi.org/10.1109/72.712178.

Wanxin Li. Supporting database constraints in synthetic data generation based on generative adversarial networks. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 2875–2877, 2020. doi: 10.1145/3318464.3384414. URL https://doi.org/10.1145/3318464.3384414.

Zenan Li, Yunpeng Huang, Zhaoyu Li, Yuan Yao, Jingwei Xu, Taolue Chen, Xiaoxing Ma, and Jian Lu. Neuro-symbolic learning yielding logical constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/4459c3c143db74ee52afebdf56836375-Abstract-Conference.html.

Tennison Liu, Zhaozhi Qian, Jeroen Berrevoets, and Mihaela van der Schaar. GOGGLE: Generative modelling for tabular data by learning relational structure. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=fPVRcJqspu.

Xianggen Liu, Qiang Liu, Sen Song, and Jian Peng. A chance-constrained generative framework for sequence optimization. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 6271–6281, 2020. URL http://proceedings.mlr.press/v119/liu20i.html.

Lu Lu, Raphaël Pestourie, Wenjie Yao, Zhicheng Wang, Francesc Verdugo, and Steven G. Johnson. Physics-informed neural networks with hard constraints for inverse design. *SIAM Journal on Scientific Computing*, 43(6):B1105–B1132, 2021. doi: 10.1137/21M1397908. URL https://doi.org/10.1137/21M1397908.

Miguel Ángel Méndez-Lucero, Enrique Bojorquez Gallardo, and Vaishak Belle. Semantic objective functions: A distribution-aware method for adding logical constraints in deep learning. In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART)*, pages 909–917, 2025. doi: 10.5220/0013229200003890. URL https://doi.org/10.5220/0013229200003890.

Eleonora Misino, Giuseppe Marra, and Emanuele Sansone. VAEL: Bridging variational autoencoders and probabilistic logic programming. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/1e38b2a0b77541b14a3315c99697b835-Abstract-Conference.html.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 7176–7185, 2020. URL https://proceedings.mlr.press/v119/naeem20a.html.

New York City Taxi and Limousine Commission. TLC trip record data. https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page, 2024. Accessed: 2026-06-16.

Wamiq Reyaz Para, Shariq Farooq Bhat, Paul Guerrero, Tom Kelly, Niloy J. Mitra, Leonidas J. Guibas, and Peter Wonka. SketchGen: Generating constrained CAD sketches. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 5077–5088, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/28891cb4ab421830acc36b1f5fd6c91e-Abstract.html.

Yifei Peng, Zijie Zha, Yu Jin, Zhexu Luo, Wang-Zhou Dai, Zhong Ren, Yao-Xiang Ding, and Kun Zhou. Generating by understanding: Neural visual generation with logical symbol groundings. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2291–2302, 2025. doi: 10.1145/3711896.3736978. URL https://doi.org/10.1145/3711896.3736978.

Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Big data meets quantum chemistry approximations: The δ-machine learning approach. *Journal of Chemical Theory and Computation*, 11(5):2087–2096, 2015. doi: 10.1021/acs.jctc.5b00099.

Mathias Schreiner, Arghya Bhowmik, Tejs Vegge, Jonas Busk, and Ole Winther. Transition1x—a dataset for building generalizable reactive machine learning potentials. *Scientific Data*, 9:779, 2022. doi: 10.1038/s41597-022-01870-w. URL https://doi.org/10.1038/s41597-022-01870-w.

Ari Seff, Wenda Zhou, Nick Richardson, and Ryan P. Adams. Vitruvion: A generative model of parametric CAD sketches. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=Ow1C7s3UcY.

Luciano Serafini and Artur d'Avila Garcez. Learning and reasoning with Logic Tensor Networks. In *AI*IA 2016: Advances in Artificial Intelligence*, volume 10037 of *Lecture Notes in Computer Science*, pages 334–348. Springer, 2016. URL https://doi.org/10.1007/978-3-319-49130-1_25.

Mihaela Cătălina Stoian and Eleonora Giunchiglia. Beyond the convexity assumption: Realistic tabular data generation under quantifier-free real linear constraints. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=rx0TCew0Lj.

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. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=tBR0YsEz9G.

Nijesh Upreti and Vaishak Belle. Logic Tensor Network-Enhanced Generative Adversarial Network. *Electronic Proceedings in Theoretical Computer Science*, 439:89–113, 2026. doi: 10.4204/EPTCS.439.8. URL https://doi.org/10.4204/EPTCS.439.8.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. A semantic loss function for deep learning with symbolic knowledge. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pages 5502–5511, 2018. URL https://proceedings.mlr.press/v80/xu18h.html.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional GAN. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pages 7333–7343, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/254ed7d2de3b23ab10936522dd547b78-Abstract.html.

Yexiang Xue and Willem-Jan van Hoeve. Embedding decision diagrams into generative adversarial networks. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*, volume 11494 of *Lecture Notes in Computer Science*, pages 616–632. Springer, 2019. URL https://doi.org/10.1007/978-3-030-19212-9_41.

Zhun Yang, Joohyung Lee, and Chiyoun Park. Injecting logical constraints into neural networks via straight-through estimators. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 25096–25122, 2022. URL https://proceedings.mlr.press/v162/yang22h.html.

Halley Young, Maxwell Du, and Osbert Bastani. Neurosymbolic deep generative models for sequence data with relational constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/f13ceb1b94145aad0e54186373cc86d7-Abstract-Conference.html.

Yang Zeng, Jin-Long Wu, and Heng Xiao. Enforcing imprecise constraints on generative adversarial networks for emulating physical systems. *Communications in Computational Physics*, 30(3):635–665, 2021. doi: 10.4208/cicp.OA-2020-0106. URL https://doi.org/10.4208/cicp.OA-2020-0106.

Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y. Chen. CTAB-GAN: Effective table data synthesizing. In *Proceedings of the 13th Asian Conference on Machine Learning (ACML)*, volume 157 of *Proceedings of Machine Learning Research*, pages 97–112, 2021. URL https://proceedings.mlr.press/v157/zhao21a.html.

Appendix A. Extended Related Work

This appendix expands Section 6. The body cites only the most directly related work.

Logic as a soft training signal. Beyond Logic Tensor Networks (Badreddine et al., 2022; Serafini and d'Avila Garcez, 2016), many methods inject logic as a differentiable loss. The semantic loss (Xu et al., 2018) and its pseudo-semantic extension for autoregressive models (Ahmed et al., 2023) penalise probability mass on violating assignments. DL2 (Fischer et al., 2019), straight-through estimators (Yang et al., 2022), distribution-aware objectives (Méndez-Lucero et al., 2025), and bilevel formulations (Li et al., 2023) optimise logical losses directly, and VAEL (Misino et al., 2022) couples a variational autoencoder with probabilistic logic. These all inject logic as a *soft* satisfaction signal rather than a hard guarantee. We do not evaluate each of them on high-resolution structural constraints, because the obstruction we identify is a property of soft satisfaction itself. In the ambient coordinates, a structural margin predicate has usable gradient on a $\Theta(1/R)$ fraction of samples (Corollary 4), and RQ4 confirms this for the three predicate placements we test.

Hard constraints by construction. A second family of methods guarantees satisfaction, each for a specific class of constraints. Semantic Probabilistic Layers (Ahmed et al., 2022), MultiplexNet (Hoernle et al., 2022), and coherent hierarchical classifiers (Giunchiglia and Lukasiewicz, 2020) cover classes of logical constraints, and physics-informed networks enforce differential constraints (Lagaris et al., 1998; Lu et al., 2021). For linear constraints on tabular data, the constraint layer of Stoian et al. (2024) clamps onto the feasible polytope, recently extended to quantifier-free non-convex constraints (Stoian and Giunchiglia, 2025). We share the exact-validity goal but, rather than clamping, ground the constraint as a change of coordinates that can also preserve the distribution of the constraint margin.

Constrained generative models. Many domains have built constraints into generators. Constrained adversarial networks enforce logical requirements during training (Di Liello et al., 2020; Hu et al., 2018; Chao et al., 2021), and related mechanisms appear in interactive image editing (Heim, 2019), physical fields (Hess et al., 2022; Zeng et al., 2021), decision diagrams (Xue and van Hoeve, 2019), parametric CAD sketches (Seff et al., 2022; Para et al., 2021), combinatorial design (Ferber et al., 2024), sequence optimisation (Liu et al., 2020; Young et al., 2022), and visual scene generation (Peng et al., 2025). Almost all of them enforce the constraints by penalty, rejection, or post-hoc projection. Function-symbol grounding instead reparameterises the output space, so the constraints hold by construction. Monotone link functions that enforce order and positivity in regression networks go back to Dugas et al. (2009), and probabilistic-programming samplers routinely transform positive, interval, and ordered parameters to unconstrained space before sampling (Carpenter et al., 2017). We use the same links, but in a new role. They ground logical axioms, they are assembled along the Fourier-Motzkin order so that any satisfiable linear system can be charted, and the discriminator is trained in the transformed coordinates. Modelling a difference rather than the levels that form it also echoes Δ-machine learning for chemical energies (Ramakrishnan et al., 2015), a supervised antecedent of the margin-form baseline of Appendix J.

Rejection sampling and chance-constrained formulations. Two further strategies deserve direct comparison. *Rejection sampling* draws from an unconstrained generator and

keeps only the valid samples. It recovers the correct conditional distribution in principle, but its acceptance rate is the unconstrained generator's own validity, which is exactly what collapses in the high-R regime (that validity is 0.05 on Alchemy, Table 5, and decreases as R grows), and for definitional identities it is zero outright. By Corollary 3 an absolutely continuous generator satisfies an identity with probability 0, so no rejection budget suffices. *Chance-constrained* formulations (Liu et al., 2020) require constraints to hold with a prescribed probability rather than on every sample. They operate in the same soft-satisfaction regime as predicate grounding and inherit its high-R gradient obstruction (Corollary 4). Function-symbol grounding avoids both: validity is guaranteed, no rejection loop is needed, and the margin is learned at unit scale.

Tabular data generation. For tabular data, adversarial and variational generators (CTGAN and TVAE (Xu et al., 2019), CTAB-GAN (Zhao et al., 2021)), score-based and diffusion models (STaSy (Kim et al., 2023), TabDDPM (Kotelnikov et al., 2023)), and relational-structure models (GOGGLE (Liu et al., 2023)) improve marginal and dependency fidelity but provide no validity guarantees. Database-constraint enforcement has also been added to GAN-based synthesis (Li, 2020). The constraint layer (Stoian et al., 2024) and our function-symbol grounding both add hard guarantees, but only function-symbol grounding recovers the distribution of the constraint margin for high-resolution constraints.

Appendix B. The Hybrid Procedure

Algorithm 1: Hybrid function-symbol grounding in an LTN-GAN.

Input : linear constraints Π; data $\mathcal{D}$; thresholds τ_s=0.9 (satisfaction), τ_d=0.2 (discreteness)

Output: an LTN-GAN generator with exact validity on Π

▷ Phase 1: reduce Π to per-variable interval bounds

Fix a variable order; by Fourier-Motzkin elimination, write each x_i's bounds ℓ_i, u_i as maxima and minima of linear functions of the strictly earlier variables

▷ Phase 2: choose a grounding per bounded variable (one short pre-run)

Train a free generator briefly; measure each bounded variable x_i's interval satisfaction s_i on the generator's samples and its binding-margin discreteness d_i on $\mathcal{D}$

for *each bounded variable* x_i **do**
  if $s_i < \tau_s$ **and** $d_i \leq \tau_d$ **then**
    $\text{mode}_i \leftarrow$ CHART ▷ high-R, continuous
  else
    $\text{mode}_i \leftarrow$ CLAMP ▷ low-R or discrete
  end
end

▷ Phase 3: train; charted variables' axioms hold by construction (no satisfaction loss)

for *each training step* **do**
  draw noise ζ and set the free coordinates $z \leftarrow G_\theta(\zeta)$
  for *each variable i in Fourier-Motzkin order* **do**
    if x_i *is fixed by an identity* **then**
      $x_i \leftarrow \sum_{j<i} w_{ij} x_j + w_{i0}$
    else if x_i *is free (highest-order in no axiom)* **then**
      $x_i \leftarrow z_i$
    else if $\text{mode}_i =$ CHART **then**
      $x_i \leftarrow$ the link fitting $[\ell_i, u_i]$ (softplus or σ, Section 4; exp, Appendix G) ▷ Sat=1
    else
      $x_i \leftarrow \min(\max(z_i, \ell_i), u_i)$ ▷ in-loop clamp
    end
  end
  update G_θ, D_ψ adversarially on the chart features: each charted variable enters as its coordinate z_i and each clamped variable as its standardized clamped value, with real samples encoded the same way through φ^{-1}
end

Appendix C. Proof of Proposition 1

Fourier-Motzkin elimination over Π in the variable order gives, for each x_i, its lower and upper bounds ℓ_i, u_i as maxima and minima of linear functions of the strictly-earlier variables. Assembling in this order, x_i's bounds depend only on finalized variables; if Π is satisfiable the interval $[\ell_i, u_i]$ is non-empty (Fourier-Motzkin completeness). The interior placement also requires the *open* interval to be non-empty, $\ell_i < u_i$: if Π pinches $\ell_i = u_i$ (an equality implied by the inequalities though not declared as an identity), the variable is routed to the dependent branch below and set to its single admissible value. For a bounded or one-sided variable with a non-degenerate interval, softplus and σ map $\mathbb{R}$ into the interior of its admissible interval, so each strict inequality in which x_i is highest-order holds with positive margin; a dependent variable is set by its identity to the single admissible value $\ell_i = u_i$, meeting its two non-strict encoding inequalities with margin 0; and a free variable is highest-order in no constraint, so it imposes nothing. Induction over the order gives $\varphi(z) \models \Pi$. $\square$

Appendix D. The Resolution Ratio as Condition Number

Figure 3 gives the intuition behind this appendix. The two regimes differ not in the constraint but in the size of its margin relative to the data scale, and that ratio is what makes the clamp and the predicate succeed or fail. The statements below make this precise.

A worked example. Two of our orderings sit at the two ends of the range. Alchemy's $U - U_0$ has $R \approx 9 \times 10^4$. The margin's spread σ_m is about 10^5 times smaller than the scale σ_s, so after standardization the margin occupies a band of relative width $\sim 1/R \approx 10^{-5}$, far below what a Lipschitz discriminator can resolve. A free generator that matches every feature's marginal therefore still violates the orderings on a constant fraction of samples (about half on average; RQ4). Trained in the loop, the constraint layer moves every sample onto $U{=}U_0$, and the distribution of the constraint margin collapses to a point mass at 0. Taxi's total $-$ fare has $R \approx 3.6$. The margin is about a third of the scale, the discriminator resolves it, the free generator already satisfies the constraint on most samples, the clamp rarely acts, and clamping leaves the distribution largely intact. The hybrid charts the former (high R) and clamps the latter (low R).

We now prove Corollaries 3 and 4 and Proposition 5.

Proof of Corollary 3. Full rank of ∇c on $\mathcal{M}_=$ makes 0 a regular value, so by the regular-value theorem $\mathcal{M}_= = c^{-1}(0)$ is an embedded C^1 submanifold of dimension $D - p$; a submanifold of dimension below the ambient D has Lebesgue measure zero. If $\nu \ll \text{Leb}$ with density f, then $\nu(\mathcal{M}_=) = \int_{\mathcal{M}_=} f \, \mathrm{d}\,\text{Leb} = 0$. Grounding sets the dependent coordinate to the exact value its identity prescribes, so every generated x lies in $\mathcal{M}_=$ by construction; the generated distribution is then carried by the null set $\mathcal{M}_=$, so $\nu \not\ll \text{Leb}$, escaping the hypothesis, and $c(x) = 0$ for every sample. □

Proof of Corollary 4. Differentiating, $\partial_{(b-a)}\sigma\big((b-a)/s\big) = \sigma'\big((b-a)/s\big)/s = P_s(1-P_s)/s$. The logistic derivative $\sigma'(t) = \sigma(t)\,(1-\sigma(t))$ is $\Theta(1)$ on any fixed band $|t| \le c$ (peak $\sigma'(0) = \frac{1}{4}$) and decays as $e^{-|t|}$ for $|t| \gg 1$, so the gradient is $\Theta(1/s)$ exactly for $|b-a| \lesssim s$ and exponentially suppressed otherwise. With $b-a$ at scale $\sigma_s = R\,\sigma_m$ and density $\Theta(1/\sigma_s)$ near 0, the probability mass on the width-$\Theta(s) = \Theta(\sigma_m)$ active band is $\Theta(\sigma_m/\sigma_s) = \Theta(1/R)$. □

Proof of Proposition 5. *(i)* On the ordering ϕ, the clamp leaves a satisfying sample's margin unchanged and moves each violator to the boundary value ε, giving the mixture CDF $F_{\text{CL}}(t) = v\,\mathbf{1}[t \ge \varepsilon] + (1-v)\,F^+(t)$, with F^+ the CDF of ν^+. Two evaluations bound $\text{KS} = \sup_t |F_{\text{CL}}(t) - F(t)|$. At $t = \varepsilon$, below the real support, $F_{\text{CL}}(\varepsilon) - F(\varepsilon) \ge v$. At $t = q_\alpha$, the real margin's α-quantile (so $q_\alpha = \Theta(\sigma_m)$), $F(q_\alpha) - F_{\text{CL}}(q_\alpha) \ge \alpha - v - (1-v)\,F^+(q_\alpha)$, and under Corollary 4's density hypothesis the generator places mass $\Theta\big(q_\alpha/(R\,\sigma_m)\big) = \Theta(1/R)$ in $(0, q_\alpha]$, so $(1-v)\,F^+(q_\alpha) = \Theta(1/R)$. Taking $\alpha \to 1$, $\text{KS} \ge \max\big(v,\, 1 - v - \Theta(1/R)\big) \ge \max(v, 1-v) - \Theta(1/R) \ge \frac{1}{2} - \Theta(1/R)$, and the first evaluation alone gives $\text{KS} \to 1$ as $v \to 1$. With several constraints the statement reads per ordering through its own clamp step; empirically the atom sits at the boundary on every dataset (Figure 2). *(ii)* Each link is a smooth bijection onto its admissible interval with smooth inverse: softplus maps $\mathbb{R}$ onto $(0, \infty)$ with inverse $\text{softplus}^{-1}(y) = \log(e^y - 1)$; the affine logistic map $\ell + (u-\ell)\,\sigma(\cdot)$ maps $\mathbb{R}$ onto (ℓ, u) with inverse the scaled logit $z = \log \frac{t}{1-t}$ at $t = (x-\ell)/(u-\ell)$; the heavy-tailed exp link maps $\mathbb{R}$ onto (ℓ, ∞) with inverse $\log(x-\ell)$; a free coordinate is the identity. Assembling in the Fourier-Motzkin order, the bounds of x_i read only $x_{<i}$ (Appendix C),

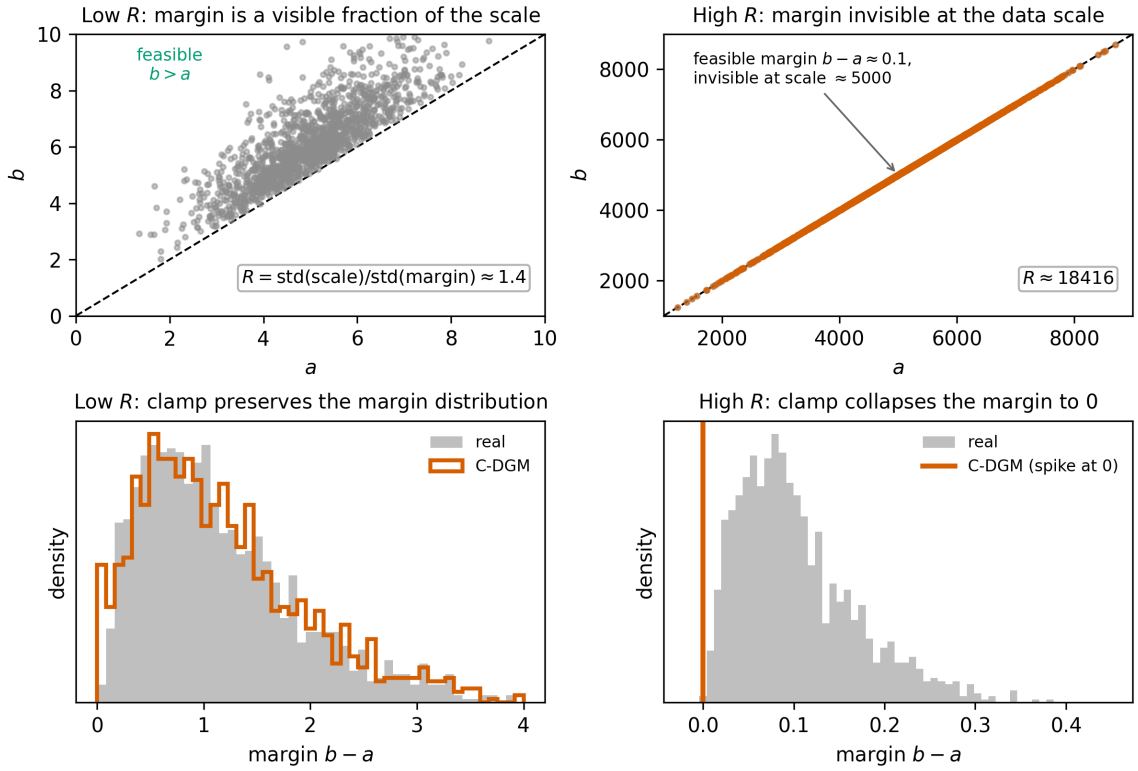

Figure 3: **High versus low resolution ratio R (schematic).** The same ordering $b > a$ in two regimes. *Top:* the data (a, b) and the boundary $b = a$. At low R (left) the feasible margin $b - a$ is a visible fraction of the scale, so a free generator resolves it and the constraint layer's clamp rarely acts. At high R (right) the margin is tiny relative to the scale and invisible at the data's magnitude, so every sample lies close to the boundary. *Bottom:* the margin distribution. At low R the clamp (orange) preserves the distribution of the real constraint margin (grey); at high R the free generator cannot place the sub-resolution margin, so the clamp collapses every sample onto the boundary, a point mass at 0 disjoint from the real distribution. Function-symbol grounding repairs the high-R case by making the margin a unit-scale coordinate (Figure 4).

so the inverse is computed coordinate-wise. Each z_i is the link inverse of x_i at the bounds set by $x_{<i}$, defined and smooth exactly when every constrained x_i is strictly interior, i.e. on the relative interior of $\mathcal{M}$ (dependent coordinates are recovered by their identities and contribute no coordinate). Induction along the order gives $\varphi \circ \varphi^{-1} = \mathrm{id}$ on $\operatorname{relint} \mathcal{M}$ and $\varphi^{-1} \circ \varphi = \mathrm{id}$ on $\mathbb{R}^d$, and interior placement (Appendix C) means φ places no probability on the boundary. Realizability: given any margin distribution μ on $(0, \infty)$, the coordinate distribution $\mathrm{softplus}^{-1}_{\#}\mu$ pushes forward under softplus to exactly μ. When several reduced constraints bound the same variable, ℓ_i and u_i are maxima and minima of linear forms,

hence piecewise-linear. φ and φ^{-1} are then smooth off the measure-zero set where the active bound switches and remain continuous bijections, so the validity, boundary, and realizability claims are unaffected. □

Figure 4 shows the mechanism of Remark 2 in pictures. The change of coordinates that gives $\Theta(1)$ conditioning turns the sub-resolution margin of Figure 3 into a unit-scale coordinate the GAN can learn, then decodes it back to valid samples with the distribution of the constraint margin recovered.

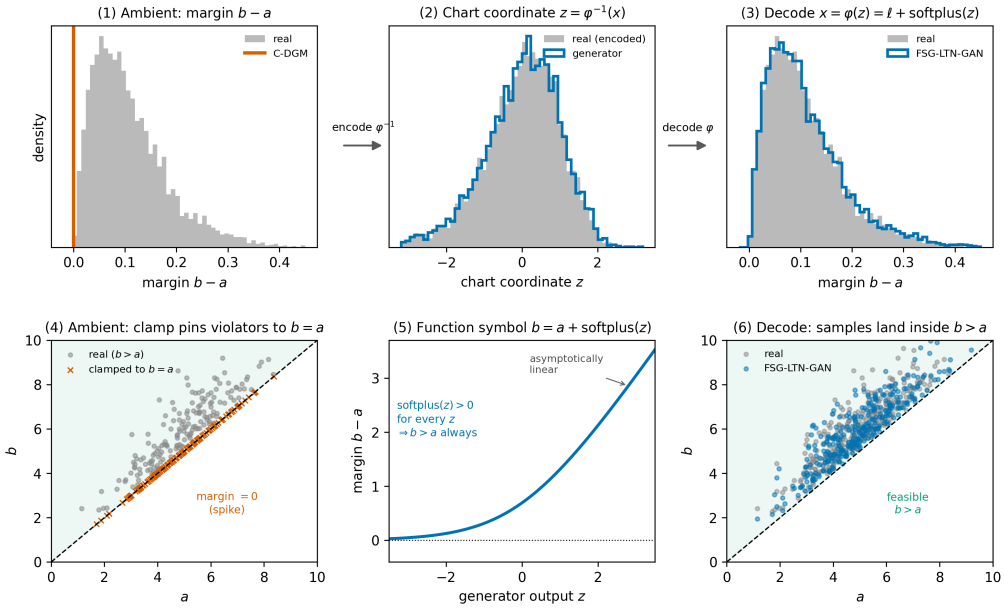

Figure 4: **How function-symbol grounding repairs the high-R collapse (schematic).** The same high-R ordering margin as in Figure 3; top row shows the margin *distribution* at each stage, bottom row the corresponding (a, b) *geometry*. *(1)* In ambient space the margin $b - a$ is sub-resolution, so the constraint layer clamps it to a point mass at 0. *(2)* The chart coordinate $z = \varphi^{-1}(x)$ standardizes the margin to unit scale, where the discriminator resolves it and a standard GAN learns the distribution of the real constraint margin (generator in blue matches real in grey). *(3)* Decoding $x = \varphi(z) = \ell + \mathrm{softplus}(z)$ carries that distribution back to the ambient margin. *(4)* Geometrically the clamp moves every violating sample onto the boundary $b = a$ (margin 0, the point mass of panel 1). *(5)* The function symbol $b = a + \mathrm{softplus}(z)$ is positive for every generator output z, so $b > a$ holds by construction (Sat(KB) = 1); it is smooth and monotone, so matching the distribution of z matches the margin's distribution, and asymptotically linear, so generator tails do not diverge. *(6)* Our decoded samples therefore land inside the feasible region $b > a$, matching the real distribution with exact validity.

Appendix E. Constraint-Layer Code Corrections and Evaluation Protocol

We run the constraint layer from the code released by Stoian et al. (2024). For genuine 100% validity (a fair margin comparison) we correct a latent bug in its Fourier-Motzkin reduction, which keeps a stale term for a variable whose coefficients cancel (triggered by equalities encoded as two inequalities), and run the layer in double precision with the strict-inequality ε set above one unit in the last place (its default $\varepsilon = 10^{-12}$ underflows here, falsely reporting 0% valid). Our decoder runs in double precision on Taxi, nycflights13, and the RQ3 benchmark, and in single precision on the chemistry sets, whose reference data are single precision. These changes only help the constraint layer, and the distortion it induces is unchanged. All generators are sampled in `eval` mode. The metrics (KS and satisfaction fractions) cannot be improved by shrinking magnitudes. Margins are scored against a fixed reference sample of the real data. On the four high-resolution datasets and nycflights13, which ship as single files with no canonical split, this is a 20,000-row sample drawn once with a fixed seed from the same data the generators train on. On the RQ3 benchmark it is the held-out validation and test splits. Every method in a comparison is scored against the same reference. The grounding link (softplus or exp) is set by the fixed dynamic-range rule of Appendix G. A definitional identity is scored satisfied when its residual is within a per-dataset tolerance in raw units (0.05 on Alchemy, 10^{-3} on tmQM, 0.5 on nycflights13). Corollary 3 concerns exact satisfaction. The tolerance is why unconstrained validity is small but nonzero on the chemistry sets, while the wider-scaled nycflights13 identity is almost never met and its validity is 0.000. Per-dataset R, sizes, constraints, and full per-method results with standard deviations are given in Appendices F to J.

Appendix F. Datasets, Constraints, and Resolution Ratios

Table 4 summarizes the four real high-resolution datasets used in RQ1, RQ2, and RQ4. Each property is standardized to zero mean and unit variance before training. The resolution ratio $R = \sigma_s/\sigma_m$ (Section 3) is computed on the raw data before training. The constraint-layer benchmark of Stoian et al. (2024) used in RQ3 is described in Appendix I.

Table 4: The four real high-resolution datasets. N rows, D properties; R is the resolution ratio of each evaluated ordering margin.

Dataset	N	D	Evaluated margin(s)	R
Alchemy	202,579	12	$U - U_0$; $H - U$; $H - G$	$9{\times}10^4$; $7{\times}10^6$; $3{\times}10^4$
tmQM	108,541	8	$E_{\text{elec}} - E_{\text{tot}}$	$2.3{\times}10^4$
Transition1x	10,073	3	$E_{\text{TS}} - E_{\text{react}}$; $E_{\text{TS}} - E_{\text{prod}}$	$1.0{\times}10^3$; $8.1{\times}10^2$
Taxi	50,000	6	duration; total − fare	$1.1{\times}10^3$; 3.6

The full structural constraint set per dataset is as follows. **Alchemy** (Chen et al., 2019), 12 molecular properties (mu, alpha, homo, lumo, gap, r2, zpve, U0, U, H, G, Cv, following QM9 naming): positivity of mu (dipole moment), gap, and zpve (zero-point vibrational energy); the algebraic identity lumo = homo + gap over the frontier-orbital energies; the thermochemical orderings $U_0 < U < H$ and $G < H$, where U_0 and U are the internal energies at 0 K and 298 K, H the enthalpy, and G the free energy. **tmQM** (Balcells and Skjelstad, 2020), 8 properties of transition-metal complexes: the high-R dispersion ordering $E_{\text{tot}} < E_{\text{elec}}$; positivity of dipole, gap, and polarizability; the low-R identity LUMO = HOMO + gap. **Transition1x** (Schreiner et al., 2022), reaction energy profiles: the transition state is the maximum, $E_{\text{TS}} > E_{\text{react}}$ and $E_{\text{TS}} > E_{\text{prod}}$. (0.08% of raw rows violate $E_{\text{TS}} > E_{\text{prod}}$, near-barrierless reverse reactions, and are retained, so the real reference margin carries that small negative mass.) **Taxi** (New York City Taxi and Limousine Commission, 2024), NYC yellow-cab records (2024-01): the duration ordering dropoff > pickup on absolute second-scale timestamps (high R); total > fare (low R); positivity of distance and fare.

Appendix G. Architecture and Training Protocol

All methods share one generator architecture so that only the constraint mechanism differs. The generator is a multilayer perceptron with latent dimension 64 and three hidden layers of width 256 with LeakyReLU(0.2) and dropout 0.1, and batch normalization on the two inner layers, followed by a linear map to the property dimension. Weights use Kaiming-uniform initialization. The discriminator is a multilayer perceptron $D \to 256 \to 128 \to 1$ with LeakyReLU(0.2), dropout 0.1, and a sigmoid output. Both train with Adam (learning rate $2{\times}10^{-4}$, $\beta = (0.5, 0.999)$), batch size 256, binary cross-entropy adversarial loss with label smoothing (targets 0.9 real, 0.1 generated), for 1000 steps. Two numerical safeguards apply throughout. The decoder floors each charted margin at 10^{-6} in raw units, and the encoder clips the real chart coordinates at their 0.5 and 99.5 percentiles before standardization. Both act only in a thin boundary layer, and neither affects validity. The analysis of Section 4 concerns the exact map. All experiments use ten seeds (0 to 9, except the Alchemy and Transition1x inverse-design runs, which use 10 to 19). For function-symbol grounding the same generator emits the free coordinates z and the grounding map φ assembles the sample (Section 4). No satisfaction loss is added because charted axioms hold by construction. The constraint layer runs on the identical architecture (Appendix E). The grounding link is softplus by default and exp (a multiplicative increment) for heavy-tailed margins. This is the method's only added hyperparameter beyond the hybrid's two thresholds, and it was set once per dataset by a fixed criterion. The criterion measures each charted margin's dynamic range as the ratio q_{99}/q_{50} of its positive part on the training data, and selects the exp link when any charted margin reaches $q_{99}/q_{50} \geq 10$. Across the eleven datasets in this paper it selects exp exactly once, on faults, whose bounding-box margins have $q_{99}/q_{50} = 18.0$ and 13.0. It selects softplus everywhere else (every charted margin outside faults has $q_{99}/q_{50} \leq 5.1$). The released code records the resulting per-dataset choice.

Appendix H. Full Per-Method Results (RQ1 and RQ2)

Table 5 extends Table 1 with all baselines and their validities; Table 6 extends Table 2 with the raw unconstrained architectures, whose validities are given in its caption. The paired Wilcoxon test for FSG-LTN-GAN versus C-DGM returns $p = 0.00195$ on every dataset, the minimum attainable at $n = 10$. In Table 5, the projection row coincides with the unconstrained row on tmQM for the reason given in Table 6's caption. Moving violators to the boundary leaves the KS supremum unchanged when the reference margin lies above it.

Table 5: Full RQ1 results, $n = 10$ seeds, mean $\pm$ std. Validity is the fraction satisfying all constraints. Margin KS is the two-sample KS between the generated margins and a fixed real reference sample (lower is better). C-DGM and FSG-LTN-GAN are both exactly valid. Only FSG-LTN-GAN recovers the distribution of the constraint margin.

Dataset	Method	Validity	Margin KS ↓
Alchemy	unconstrained GAN	0.053 ± 0.021	0.539 ± 0.014
	projection	1.000 ± 0.000	0.664 ± 0.026
	C-DGM (clamp)	1.000 ± 0.000	1.000 ± 0.000
	FSG-LTN-GAN	1.000 ± 0.000	$\mathbf{0.040 \pm 0.007}$
tmQM	unconstrained GAN	0.229 ± 0.046	0.575 ± 0.037
	projection	1.000 ± 0.000	0.575 ± 0.037
	C-DGM (clamp)	1.000 ± 0.000	1.000 ± 0.000
	FSG-LTN-GAN	1.000 ± 0.000	$\mathbf{0.045 \pm 0.010}$
Transition1x	unconstrained GAN	0.293 ± 0.384	0.833 ± 0.140
	projection	1.000 ± 0.000	0.881 ± 0.093
	C-DGM (clamp)	1.000 ± 0.000	0.999 ± 0.000
	FSG-LTN-GAN	1.000 ± 0.000	$\mathbf{0.065 \pm 0.010}$
Taxi	unconstrained GAN	0.253 ± 0.087	0.346 ± 0.042
	projection	0.483 ± 0.112	0.345 ± 0.042
	C-DGM (clamp)	1.000 ± 0.000	0.574 ± 0.005
	FSG-LTN-GAN	1.000 ± 0.000	$\mathbf{0.099 \pm 0.010}$

Table 6: Full RQ2 (margin KS by architecture, $n = 10$, mean $\pm$ std). Raw CTGAN/TVAE validity is below 3% on the chemistry sets and 30 to 46% on Transition1x and Taxi. The +CL rows, C-DGM, and FSG-LTN-GAN all reach 100% validity. Only FSG-LTN-GAN recovers the distribution of the constraint margin across the architectures we test. On tmQM and Taxi the raw and +CL margin-KS entries coincide exactly. This is forced rather than copied, because the clamp maps the violating mass to the boundary point mass at ε while the reference margin lies above ε, so the KS supremum, attained at ε, has the same value before and after. Validity and moment error do move. On Alchemy and Transition1x the chained orderings share endpoints, so clamping shifts the evaluated margins and the KS.

Method	Alchemy	tmQM	Transition1x	Taxi
CTGAN (raw)	0.596 ± 0.029	0.579 ± 0.052	0.569 ± 0.057	0.419 ± 0.024
CTGAN + CL	0.674 ± 0.036	0.579 ± 0.052	0.592 ± 0.040	0.419 ± 0.024
TVAE (raw)	0.528 ± 0.012	0.517 ± 0.011	0.478 ± 0.047	0.348 ± 0.029
TVAE + CL	0.651 ± 0.015	0.517 ± 0.011	0.510 ± 0.053	0.348 ± 0.029
C-DGM (CL, MLP)	1.000 ± 0.000	1.000 ± 0.000	0.999 ± 0.000	0.574 ± 0.005
FSG-LTN-GAN	$\mathbf{0.040 \pm 0.007}$	$\mathbf{0.045 \pm 0.010}$	$\mathbf{0.065 \pm 0.010}$	$\mathbf{0.099 \pm 0.010}$

Appendix I. The Constraint-Layer Benchmark, Full (RQ3)

Table 7 extends Table 3 with each dataset's constraint count and charted count. KS is averaged over each dataset's constraints. C-DGM, FSG-all, and the hybrid all reach 100% validity; unconstrained validity ranges from 0.005 (wids) to 0.736 (url). The hybrid charts the constraints its pre-run selects (free-generator satisfaction below 0.9, margin not discrete) and clamps the rest. On heloc and lcld it charts nothing and reduces to the constraint layer. On url, news, and wids it charts one or two variables, with smaller but significant gains on url and wids ($p = 0.03$ and 0.004) and a statistical tie on news (heloc and lcld are ties as well). On faults (whose high-R margin is the bounding-box ordering $Y_{\max} > Y_{\min}$) it charts every variable and improves substantially (paired Wilcoxon $p = 0.002$, all ten seeds). The FSG-all ablation, which charts every variable, is worse than the constraint layer on every dataset but faults.

Table 7: Full RQ3 results on the benchmark of Stoian et al. (2024), $n = 10$ seeds, mean $\pm$ std, margin KS. "charted" is the number of variables the hybrid charts as function symbols; **bold** marks where the hybrid significantly outperforms the constraint layer (paired Wilcoxon $p < 0.05$).

Dataset	$\lvert\Pi\rvert$	charted	$R_{\max}$	C-DGM	FSG-all	**hybrid**
faults	4	4	$1.8{\times}10^3$	0.745 ± 0.014	0.117 ± 0.008	$\mathbf{0.117 \pm 0.008}$
heloc	7	0	2	0.156 ± 0.012	0.208 ± 0.005	0.158 ± 0.014
lcld	4	0	1	0.131 ± 0.008	0.657 ± 0.004	0.133 ± 0.007
url	8	1	1	0.159 ± 0.010	0.196 ± 0.018	$\mathbf{0.144 \pm 0.021}$
news	5	2	2	0.294 ± 0.008	0.318 ± 0.007	0.297 ± 0.007
wids	31	1	4	0.140 ± 0.007	0.319 ± 0.004	$\mathbf{0.133 \pm 0.008}$

Appendix J. Additional Analyses

Density and coverage. Table 8 reports density and coverage (Naeem et al., 2020), the constraint layer's own realism metrics, computed on standardized features and shown alongside margin KS.

Table 8: Density and coverage (Naeem et al., 2020) ($n = 10$, mean $\pm$ std), with margin KS for reference. Higher is better for density and coverage, lower for KS. The better method per column is in bold.

Dataset	Method	Density	Coverage	Margin KS ↓
Alchemy	C-DGM	$\mathbf{0.923 \pm 0.031}$	0.619 ± 0.014	1.000 ± 0.000
	FSG-LTN-GAN	0.780 ± 0.034	$\mathbf{0.636 \pm 0.017}$	$\mathbf{0.040 \pm 0.007}$
tmQM	C-DGM	$\mathbf{0.839 \pm 0.015}$	$\mathbf{0.752 \pm 0.009}$	1.000 ± 0.000
	FSG-LTN-GAN	0.801 ± 0.022	0.735 ± 0.011	$\mathbf{0.045 \pm 0.010}$
Transition1x	C-DGM	0.200 ± 0.039	0.050 ± 0.005	0.999 ± 0.000
	FSG-LTN-GAN	$\mathbf{0.248 \pm 0.030}$	$\mathbf{0.203 \pm 0.009}$	$\mathbf{0.065 \pm 0.010}$
Taxi	C-DGM	0.066 ± 0.008	0.052 ± 0.004	0.574 ± 0.005
	FSG-LTN-GAN	$\mathbf{0.870 \pm 0.013}$	$\mathbf{0.704 \pm 0.015}$	$\mathbf{0.099 \pm 0.010}$

The failure is not a variable-ordering artifact. A constraint layer must fix a Fourier-Motzkin elimination order. Table 9 runs the constraint layer on Transition1x under every dependency-valid ordering: all stay near 1.0 margin KS, so this failure is intrinsic to clamping. Function-symbol grounding, which fixes one ordering, is at 0.065. All configurations are 100% valid. (The two apex-first orders reduce to the same triangular program and give identical runs.)

Table 9: Constraint-layer robustness to variable ordering (Transition1x, $n = 10$, mean $\pm$ std, margin KS). Bracketed lists are the Fourier-Motzkin variable elimination orders.

Configuration	Margin KS ↓
C-DGM, natural order $[0, 1, 2]$	0.999 ± 0.000
C-DGM, apex-first $[1, 0, 2]$	0.999 ± 0.001
C-DGM, apex-first $[1, 2, 0]$	0.999 ± 0.001
C-DGM, apex-last $[0, 2, 1]$	0.903 ± 0.049
FSG-LTN-GAN	$\mathbf{0.065 \pm 0.010}$

Other backbones: CTGAN and TVAE in the chart. Table 2 showed the *failure* on CTGAN and TVAE. The chart supplies the *fix* for them as well. Because function-symbol grounding is a change of coordinates on the data, any tabular generator can be trained in the chart: encode the training data by φ^{-1}, fit the unmodified backbone with its package-default hyperparameters at 100 epochs (the RQ2 protocol), and decode its samples through φ. Table 10 applies this recipe to CTGAN and TVAE with no architectural change:

both become exactly valid and recover the margin, with margin KS 3.6 to 8.6× below their constraint-layer counterparts (paired Wilcoxon p=0.002 on every dataset–backbone pair). One backbone-specific effect persists. TVAE's latent-variance shrinkage, applied in chart coordinates, propagates through the free base variable and inflates ambient per-property moment error on tmQM and Transition1x (CTGAN in the chart does not show this, and margin KS is unaffected either way).

Table 10: **The method transfers to other generator families** (margin KS ↓, n=10, mean ± std). "in chart" trains the unmodified backbone on φ^{-1}-encoded data and decodes through φ. Both chart columns are exactly 100% valid. "+CL" columns from Table 2.

Dataset	CTGAN+CL	**CTGAN in chart**	TVAE+CL	**TVAE in chart**
Alchemy	0.674 ± 0.036	$\mathbf{0.105 \pm 0.041}$	0.651 ± 0.015	$\mathbf{0.127 \pm 0.005}$
tmQM	0.579 ± 0.052	$\mathbf{0.109 \pm 0.061}$	0.517 ± 0.011	$\mathbf{0.060 \pm 0.014}$
Transition1x	0.592 ± 0.040	$\mathbf{0.089 \pm 0.019}$	0.510 ± 0.053	$\mathbf{0.141 \pm 0.037}$
Taxi	0.419 ± 0.024	$\mathbf{0.086 \pm 0.016}$	0.348 ± 0.029	$\mathbf{0.064 \pm 0.014}$

Flight records (nycflights13). The four main datasets and the RQ3 benchmark were each chosen by the authors of one of the two papers being compared. We therefore ran a fifth dataset that neither paper had used, the nycflights13 flight records (327,346 rows after cleaning, subsampled to 50,000; six properties), with the ordering arrival > departure on absolute timestamps spanning a year, the exact identity dep = sched + delay, and positivity of air time and distance. *Before training* we computed $R = 3.0 \times 10^3$ for the ordering from the raw data and recorded the predicted outcome (constraint-layer margin KS near 1, FSG-LTN-GAN below 0.15, both exactly valid; unconstrained validity near 0), together with explicit falsification criteria, in a file included in the code release. The sweep then ran once, with ten seeds. Every prediction held (Table 11). The clamp collapses the flight-duration margin to the boundary exactly as on Taxi and the chemistry sets, and the chart recovers it, at equal exact validity.

Table 11: **Flight records (nycflights13)** (n=10, mean ± std). R and the predicted outcome were recorded before training.

Method	Validity	Margin KS ↓
unconstrained GAN	0.000 ± 0.000	0.581 ± 0.044
projection	0.530 ± 0.068	0.541 ± 0.041
C-DGM (clamp)	1.000 ± 0.000	1.000 ± 0.000
FSG-LTN-GAN	1.000 ± 0.000	$\mathbf{0.086 \pm 0.011}$

Where the margin-KS gain comes from. FSG-LTN-GAN differs from the constraint layer in two coupled ways: samples are valid by construction, and the discriminator operates on the chart coordinates z, where every margin is unit scale, while the constraint layer's discriminator sees the original, ill-scaled coordinates. To separate the two contributions we

train *FSG-ambient-D*: the generator and the chart φ are unchanged, so validity remains exact, but the discriminator receives the decoded, standardized ambient sample instead of z. Table 12 shows margin KS degrades on every dataset (paired Wilcoxon p=0.002 each), and the degradation tracks R (largest on the chemistry sets), yet remains well below the clamp's. Both mechanisms therefore contribute. Generating *inside* the feasible region rather than clamping onto its boundary already improves the margin, and on three of the four datasets the larger share of the recovery comes from the discriminator operating at unit scale. On Transition1x the shares reverse, with generating inside contributing most of the difference. The two are parts of the same grounding. The chart supplies the coordinates, and the discriminator is best run in them. (Validity is 1.000 in every run except a single tmQM seed at 0.999, a float32 identity-reconstruction round-off in the ambient arm's training-time decode.)

Table 12: **Validity by construction versus discriminating in the chart.** Margin KS (n=10, mean ± std). FSG-ambient-D keeps the chart generator (validity exact) but shows the discriminator the decoded ambient sample; C-DGM from Table 5 for reference.

Dataset	**FSG (D in chart)**	FSG-ambient-D	C-DGM (clamp)
Alchemy	$\mathbf{0.040 \pm 0.007}$	0.838 ± 0.154	1.000 ± 0.000
tmQM	$\mathbf{0.045 \pm 0.010}$	0.737 ± 0.339	1.000 ± 0.000
Transition1x	$\mathbf{0.065 \pm 0.010}$	0.267 ± 0.106	0.999 ± 0.000
Taxi	$\mathbf{0.099 \pm 0.010}$	0.340 ± 0.126	0.574 ± 0.005

Margin-form preprocessing is a manual chart. For a single ordering one can instead transform the *data*: on Taxi, replace dropoff with duration = dropoff − pickup, standardize, train the constraint layer with duration > 0, and decode afterwards. Because the margin becomes its own column, its resolution ratio drops to R=1 (from 1.1×10^3), and the clamp then largely preserves the margin distribution. Margin KS falls $0.574 \rightarrow 0.134 \pm 0.020$ (duration KS $1.000 \rightarrow 0.118 \pm 0.033$) at exact validity (Table 13). This is further evidence for the coordinate explanation. The preprocessing *is* function-symbol grounding applied to the data by hand, for one constraint. Three observations make the chart its general form. First, the transform alone confers no validity. The same preprocessing with an unconstrained GAN is $52.1\% \pm 5.5$ valid (duration can still be generated negative, and the untouched constraints are violated), so a clamp or a chart is still required. Second, rewriting a *conjunction* into consistent margin form (Alchemy's $U_0 < U < H$ chain together with its lumo identity) must proceed variable by variable in a triangular order, which is exactly the Fourier-Motzkin substitution the chart automates. Third, the chart also standardizes what the discriminator sees for *all* constraints at once (previous paragraph), which manual rewriting achieves only for the rewritten margins. FSG-LTN-GAN accordingly remains best (0.099 ± 0.010; paired Wilcoxon p=0.002 against the preprocessed constraint layer, all ten seeds).

Conditional inverse design. The margin KS matters when a consumer of the samples needs the constrained quantity to be realistic. In *conditional inverse design*, we condition

Table 13: **Margin-form preprocessing on Taxi** (n=10, mean ± std). Preprocessing the data into margin form is a manual, single-constraint chart: it recovers the clamp's margin, though not validity (without a constraint mechanism) and not the untransformed margins. Duration KS is the KS of the high-R duration margin alone.

Method	Validity	Margin KS ↓	Duration KS ↓
C-DGM, ambient (Table 5)	1.000 ± 0.000	0.574 ± 0.005	1.000 ± 0.000
margin-form + C-DGM	1.000 ± 0.000	0.134 ± 0.020	0.118 ± 0.033
margin-form + unconstrained GAN	0.521 ± 0.055	0.121 ± 0.015	0.111 ± 0.024
FSG-LTN-GAN	1.000 ± 0.000	$\mathbf{0.099 \pm 0.010}$	$\mathbf{0.081 \pm 0.024}$

the generator on a target value t for a designable property (the HOMO–LUMO gap on Alchemy, the reactant energy on Transition1x, the electronic energy on tmQM) and ask for complete property profiles that are (i) valid under the dataset's constraints, (ii) on target, and (iii) realistic in their margins. All arms share the conditional architecture (the discriminator sees the sample–target pair). They differ only in the constraint mechanism: none, a predicate-style satisfaction penalty, post-hoc projection, or the chart. In Table 14, function-symbol grounding is the only configuration that delivers all three at once: exact validity (the tmQM chart value is 0.9999 before rounding, a float32 identity round-off), target error at the unconstrained level, and margin KS an order of magnitude below every alternative. The penalty arm shows the cost of soft satisfaction at high R. Pushed toward satisfaction, it misses the target (target error 2 to 5× worse) yet still fails validity. Projection attains validity but inherits the unconstrained margins.

Table 14: **Conditional inverse design** (n=10 seeds, mean ± std, 1500 steps). Target error is the mean absolute deviation of the designable property from its conditioning target (standardized units). Margin KS is as in the main text. Only the chart is simultaneously valid, on target, and realistic in its margins.

Dataset	Mechanism	Validity	Target err. ↓	Margin KS ↓
Alchemy (target: gap)	unconstrained	0.089 ± 0.050	0.067 ± 0.003	0.629 ± 0.072
	penalty	0.019 ± 0.009	0.349 ± 0.091	0.624 ± 0.035
	projection	0.999 ± 0.001	0.067 ± 0.003	0.652 ± 0.050
	chart (FSG)	$\mathbf{1.000 \pm 0.000}$	0.073 ± 0.004	$\mathbf{0.049 \pm 0.008}$
Transition1x (target: E_{react})	unconstrained	0.662 ± 0.235	0.053 ± 0.004	0.477 ± 0.170
	penalty	0.797 ± 0.043	0.124 ± 0.009	0.851 ± 0.031
	projection	1.000 ± 0.000	0.053 ± 0.004	0.470 ± 0.174
	chart (FSG)	$\mathbf{1.000 \pm 0.000}$	0.060 ± 0.006	$\mathbf{0.049 \pm 0.010}$
tmQM (target: E_{elec})	unconstrained	0.269 ± 0.061	0.065 ± 0.004	0.548 ± 0.042
	penalty	0.036 ± 0.016	0.253 ± 0.020	0.608 ± 0.051
	projection	1.000 ± 0.000	0.065 ± 0.004	0.548 ± 0.042
	chart (FSG)	$\mathbf{1.000 \pm 0.000}$	0.073 ± 0.008	$\mathbf{0.038 \pm 0.006}$

Sensitivity of the hybrid's thresholds. The hybrid's two thresholds are fixed once (τ_s=0.9, τ_d=0.2) and shared by every dataset in the paper. Sweeping the grid $\tau_s \in \{0.70, 0.80, 0.85, 0.90, 0.95\} \times \tau_d \in \{0.05, 0.10, 0.20, 0.30, 0.50\}$ changes the charted set on 5 to 19 of the 25 cells, depending on the dataset, and retraining every distinct alternative set the grid produces (ten seeds each, Table 15) changes no conclusion: every configuration keeps exact validity, alternates move margin KS by at most ≈0.03 on the low-R datasets, and every cell near the paper's setting keeps every win and tie. The one qualification is wids, where single extreme cells that chart seven or more constraints move to KS 0.143 to 0.152, at or slightly above the constraint layer's 0.140. The one large change shows where the sensitivity lies. On faults, the extreme τ_d=0.05 row stops charting the two bounding-box margins (including the high-R one), and the main win shrinks (KS 0.117 → 0.548, still below the constraint layer's 0.745). The outcome is sensitive not to the threshold values but to whether the high-R margins are charted, which is what R predicts before training.

Table 15: **Threshold sensitivity on the RQ3 benchmark** (margin KS, n=10, mean ± std). "cells" is how many of the 25 grid cells select each charted set. The paper's cell is (τ_s, τ_d)=(0.9, 0.2).

Dataset	Paper's charted set (cells)	KS	Alternative sets (cells)	KS
faults	4 charted (20)	0.117 ± 0.008	2 charted (5)	0.548 ± 0.006
heloc	none (15)	0.158 ± 0.014	1 charted (10)	0.132 ± 0.008
lcld	none (20)	0.133 ± 0.007	1 charted (5)	0.134 ± 0.007
url	1 charted (9)	0.144 ± 0.021	none (16)	0.150 ± 0.011
news	2 charted (6)	0.297 ± 0.007	none or 1 charted (19)	0.292 to 0.296
wids	1 charted (6)	0.133 ± 0.008	11 distinct sets (19)	0.128 to 0.152

RQ4 in table form. Table 16 restates the RQ4 numbers of Section 5.4.

Table 16: **RQ4: predicate placements on the high-R orderings** (ordering satisfaction, the per-ordering satisfaction fraction averaged over the dataset's orderings, n=10, mean ± std). "chance" is a free generator with no constraint mechanism. "G-LTN-GAN" places the predicate in the generator loss. "D re-weight" re-weights each sample's discriminator loss by its predicate satisfaction. "D feature aug." appends the predicate value to the discriminator's input. FSG-LTN-GAN is function-symbol grounding. No placement of the predicate moves the ordering more than 0.09 above chance, and the discriminator-feature placement falls well below it. Function-symbol grounding satisfies the ordering for every sample.

Dataset	chance	G-LTN-GAN	D re-weight	D feature aug.	**FSG-LTN-GAN**
Alchemy	0.49 ± 0.03	0.58 ± 0.02	0.51 ± 0.03	0.14 ± 0.08	$\mathbf{1.000 \pm 0.000}$
tmQM	0.53 ± 0.08	0.59 ± 0.03	0.55 ± 0.08	0.29 ± 0.07	$\mathbf{1.000 \pm 0.000}$

Appendix K. Code and Data Availability

All datasets are publicly available (Appendix F); code, experiment scripts, and the prediction record of Appendix J are available at FSG-LTN-GAN.