逻辑张量网络增强型生成对抗网络

NijeshUpreti Vaishak Belle爱丁堡大学,
Crichton 街 10 号，爱丁堡 EH89AB，英国
n.upreti@ed.ac.uk vbelle@ed.ac.uk

本文介绍了一种新颖框架——逻辑张量网络增强型生成对抗网络（LTN-GAN）。该框架将逻辑张量网络（LTN）融入生成对抗网络（GAN），以便在样本生成过程中施加领域特定逻辑约束。尽管 GAN 在生成真实数据方面取得了显著成功，但它们通常缺少融入先验知识或确保逻辑一致性的机制，因此在要求遵循规则的领域中应用受限。LTN 提供了一种有原则的方法，可将一阶逻辑与神经网络相结合，使模型能够对逻辑约束进行推理并满足这些约束。通过结合 GAN 在真实数据合成方面的优势与 LTN 在逻辑推理方面的优势，我们得以深入了解逻辑约束如何影响生成过程，同时提升生成样本的多样性和逻辑一致性。我们在多个数据集上评估了 LTN-GAN，包括合成数据集（高斯、网格、环）和 MNIST 数据集。结果表明，在保持生成样本质量和多样性的同时，我们的模型在遵循预定义逻辑约束方面显著优于传统 GAN。本文突显了神经符号方法在知识密集型领域增强生成建模的潜力。

1 引言

生成对抗网络（GANs）显著推动了生成模型领域的发展，使得创建高度逼真的合成数据成为可能。典型的生成对抗网络由两个神经网络组成：生成器负责生成合成数据，判别器负责评估生成数据是真实的还是伪造的 [6]。通过这一对抗训练过程，GANs 在图像合成和数据增强等应用中取得了显著成功 [6, 29]。然而，传统 GANs 面临一个关键限制——它们缺乏将领域特定知识和逻辑约束融入生成过程的能力。因此，它们可能生成在视觉上看似合理、却违反关键领域规则或逻辑约束的样本 [29]。

为了解决这一限制，研究人员提出了多种将逻辑约束嵌入生成对抗网络架构的方法。其中一种方法是受约束对抗网络（CANs），该方法引入可微语义损失，根据符号约束对无效样本的生成进行惩罚[17]。这使得能够进行强制执行逻辑规则的端到端训练，从而生成符合预定义约束的结构化对象，例如分子图。类似地，约束深度生成模型（C-DGMs）将约束层直接融入 WGANs 和 TableGAN 等生成对抗网络架构中，以强制执行以线性不等式表示的用户定义约束 [30]。这确保生成样本符合这些约束，同时保持生成质量。

其他研究也探索了特定应用约束强制执行。例如， [36]将物理定律作为软约束纳入 GAN 训练，确保生成数据符合

M. Gebser、D. Inclezan、F. Ricca、M. Carro 和 M. Truszczynski：《第 41 届国际逻辑编程会议》（ICLP 2025），EPTCS 439，2026 年，第 89–113 页，doi:10.4204/EPTCS.439.8

© N. Upreti 和 V. Belle 本作品采用知识共享署名许可协议授权。

守恒定律等关键物理属性。另一个例子是，CONGAN 通过将语义相似性约束映射到结构化空间，引入用户交互式约束，使用户能够引导生成过程，并确保输出符合高层规范 [7]。这些方法共同展现了将约束直接嵌入生成对抗网络架构的不同策略，以平衡样本质量与规则遵从度。

受这些进展的启发，本文探索了一个颇具前景的新方向：将逻辑张量网络（LTN）与生成对抗网络相结合，以增强其纳入逻辑约束的能力。LTN 是一种神经符号框架，将深度学习与一阶逻辑相结合，使神经网络既能从数据中学习，也能依据逻辑规则进行推理 [28]。LTN 已成功应用于语义图像理解和知识图谱补全等任务，展现了将逻辑规则直接嵌入学习过程的能力 [2, 28]。

基于这些基础，我们提出了逻辑张量网络增强型生成对抗网络（LTN-GAN）这一新型框架，将 LTN 与生成对抗网络相结合，把逻辑约束直接嵌入数据生成过程。借助 LTN 的推理能力，LTN-GAN 引导生成器生成既逼真又符合预定义逻辑规则的数据，从而解决传统生成对抗网络的一项关键局限。这使生成的样本不仅具有多样性，而且在逻辑上保持连贯，这对于逻辑一致性至关重要的领域尤为重要。我们在合成数据集和 MNIST 数据集上进行了大量实验，结果表明，LTN-GAN 在样本多样性和逻辑遵循度方面均优于基线生成对抗网络。通过将逻辑推理直接融入生成过程，LTN-GAN 提供了一种灵活且可扩展的方式，可将领域特定知识注入生成对抗网络，从而提升生成数据的可靠性和语义质量。

2 相关工作

近期神经符号生成建模的发展主要聚焦于将深度学习的灵活性与符号逻辑的结构性和可解释性相结合。这些混合方法在生成输出必须遵循明确规则的领域尤其有价值，例如物理约束、组合结构或逻辑关系 [15, 16]。在此背景下，将符号约束整合到生成模型，尤其是生成对抗网络（GANs）中，已成为提升输出有效性与可控性的一个有前景的方向。

已有多项研究探索了受约束生成对抗网络框架，将约束或规则直接融入生成过程 [3, 23, 14]。受约束对抗网络（CANs） [17] 将约束求解器引入GAN训练，通过修改生成器的目标函数来惩罚无效样本。GenCO [5] 将组合优化过程嵌入生成模型，确保每个生成输出都满足可行性约束。析取细化层（DRL） [31] 通过动态纠正输出，使其符合析取逻辑约束。尽管这些方法颇为有效，但它们通常针对结构化或离散领域而设计，其形式化方法也往往依赖于具体问题。

除了通用型受约束生成对抗网络，面向特定领域的应用也展现了规则引导生成的优势。BFGAN [18] 通过结合前向和后向生成器，在自然语言生成中融入词汇约束，以保持流畅性和连贯性。在科学建模中，物理约束生成对抗网络 [8] 施加物理定律，以提高地球系统模型输出的合理性。同样，RCFL-GAN [26] 在联邦学习环境中应用资源约束，确保模型不超出通信和计算限制。这些

方法展现了约束感知生成的灵活性，但它们在很大程度上依赖领域特定公式化。

除了对抗训练之外，符号推理技术还通过其他机制融入生成模型。后验正则化方法 [10]通过在训练过程中添加惩罚项来实施结构化约束，从而促使模型生成符合逻辑的样本。其他策略则利用决策图或基于逻辑的编码，将符号知识编码到潜在空间中，以引导样本生成 [33]。对于序列数据，诸如 [35] 的方法通过程序合成学习关系约束，从而确保诗歌或音乐等输出具有结构。尽管这些方法展示了符号约束的价值，但它们通常并非针对对抗训练架构而设计。

新兴的神经符号生成领域更明确地架起了神经学习与符号推理之间的桥梁。生成式神经符号机器（GNMs） [11]将潜在神经表示与符号层相结合，从而能够在生成过程中实施逻辑规则。在视觉领域，AbdGen[25] 运用溯因推理，通过使潜在语义因子与符号抽象相对齐来改进图像生成。这些技术增强了控制能力和可解释性，但它们通常面向通用生成模型，而非生成对抗网络等对抗框架。

神经符号推理的最新进展进一步凸显了基于逻辑的控制在生成建模中的潜力。例如，约束引导推理方法 [4] 会在测试时动态应用符号修正，以提升泛化能力和对逻辑规则的遵循度。同样，逻辑注入技术 [34] 将符号约束直接整合到神经网络架构中，为引导学习和输出生成提供了新途径。

尽管相关研究日益丰富，将逻辑张量网络（逻辑张量网络）融入生成对抗网络仍未得到充分探索。逻辑张量网络提供了一种原理严谨且可微的框架，用于施加一阶逻辑约束，但其在对抗训练中的应用仍然有限。

3 提出的方法：LTN-GAN

LTN-GAN 是一种混合生成框架，旨在通过将领域特定逻辑约束整合到训练过程中，增强传统生成对抗网络（生成对抗网络）。该框架通过引入逻辑张量网络来实现这一目标，使生成器不仅能够生成与真实样本相似的数据，还能满足一组预定义的、用于编码领域知识的逻辑规则。本节将介绍 LTN-GAN 的架构组件、逻辑集成机制和自适应训练策略。

3.1 模型架构

LTN-GAN 遵循标准的生成对抗网络框架，由生成器 G 和判别器 D组成，其中 G 学习生成与目标分布相似的数据样本，而 D 学习区分真实样本和生成样本。生成器将来自 $\mathbb{R}^{d_z}$ 的潜向量 $z \sim \mathcal{N}(0, I$映射为一个样本

$x \in \mathcal{X}$，而 th 判别器输出概率 $D(x) \in (0,1)$，用于指示该样本是否为真实样本 .[1]

网络类型和激活函数：所有实验均使用多层感知器（多层感知机），即全连接前馈神经网络。这类网络由一系列线性层构成，层间交替使用非线性激活函数，所有隐藏层均采用 LeakyReLU。判别器中使用 Dropout 进行正则化。对于二维数据集（高斯、网格、环），G 和D 均为处理二维向量的浅层多层感知器。G 的最终激活函数为线性激活函数（即不使用 tanh 等边界激活函数），以避免引入几何偏差。对于 MNIST，生成器输出一张 28×28 图像，并使用 Sigmoid 激活函数将像素值限制在 $[0,1]$, 范围内；判别器则包含 Sigmoid 输出层和辅助分类器头。

生成器架构：在二维数据集（高斯、网格、环）中，G 是一个具有 128 个隐藏单元的三层多层感知机，采用 LeakyReLU 激活函数和线性输出层。高斯和网格数据不应用最终激活函数。对于**环**，输出表示一个极坐标对（$r\ \theta$），并将其转换为笛卡尔形式（$x,\ y$）=（$r\cos\ \theta,\ r\ \sin\ \theta$），从而便于生成环状结构。在**MNIST**中，G通过将数字标签与潜向量进行独热拼接来实现条件化。输入经过三个带有批归一化和 LeakyReLU 的全连接层，然后重塑为一张 28×28 灰度图像。最后使用 Sigmoid 将像素值限制在 $[0,1]$ 范围内。

判别器架构： 判别器D 采用与多层感知机相似的结构，使用 LeakyReLU 激活函数，并通过最终的 Sigmoid 进行真假预测。在 MNIST 设置中，还附加了一个辅助分类头（对各类别应用 Softmax），以帮助生成器生成可辨识的数字，并支持基于逻辑的数字约束。

3.2 逻辑张量网络（LTN）组件

逻辑张量网络（LTN）[28, 2] 是一种神经符号框架，将一阶逻辑的表征能力与神经网络的学习能力相结合。在 LTN 中，领域知识表示为一组加权逻辑公式，这些公式由谓词、连接词和量词构成，且全部以可微形式实现，从而使真值度能够与神经网络参数共同优化。

在经典LTN设置中，每个谓词$P_k(\cdot;\ \phi_k)$：$\mathcal{X} \to [0,1]$ 都是一种可训练神经函数，将基项（例如数据样本）映射为 $[0,1]$ 中的实值真值度。逻辑公式通过模糊逻辑连接词由这些谓词构造而成（即 $\wedge$、$\vee$、$\neg$、$\Rightarrow$ 的可微推广），并使用模糊全称（$\forall$）或存在（$\exists$）算子对可用数据进行量化。随后，知识库的满足程度相对于谓词参数和模型参数是可微的，从而允许逻辑约束指导学习。

适配 GAN 训练。 在 LTN–GAN 中，LTN 模块 $\mathscr{C}$ 将领域知识编码为一组加权公式 $\mathscr{K} = \{(F_i,\ w_i)\}_{i=1}^{m}$。谓词通过直接作用于生成样本 $x_G = G(z)$ 的小型神经网络实现（例如，“位于网格上”“是数字 3”），并在 $[0,1]$ 中产生真值度。当生成器生成的样本违反这些规则时，会受到惩罚：将 LTN 导出的逻辑损失 加到对抗损失中。

[1]对抗损失。 判别器使用二元交叉熵损失进行训练：

$$L_D = L_D^{\text{real}} + L_D^{\text{fake}},\quad L_D^{\text{real}} = \mathbb{E}_{x\sim p_{\text{data}}}\ell_{\text{BCE}}(D(x), y_{\text{real}}),\quad L_D^{\text{fake}} = \mathbb{E}_z\ell_{\text{BCE}}(D(G(z)), y_{\text{fake}}),$$

$$L_G^{\text{adv}} = \mathbb{E}_z\ell_{\text{BCE}}(D(G(z)), y_{\text{real}}) \equiv -\mathbb{E}_z \log D(G(z)).$$

我们采用生成器损失的非饱和版本，以实现稳定训练。部分数据集会应用标签平滑（例如，将真实标签设为 0.9），二元交叉熵定义为 $\ell_{\text{BCE}}(p,\ y)\ = -y\log p\ -(1-y)\log(1-p)$。

公式常量　　构造。公式F_i 由谓词通过以下模糊联结词构成　　：

$$\begin{aligned}
&\text{Conjunction:} && a \wedge b = a \cdot b && \text{(product } t\text{-norm)},\\
&\text{Disjunction:} && a \vee b = \max(a,b) \text{ or } a+b-ab && \text{(max or probabilistic sum)},\\
&\text{Negation:} && \neg a = 1-a, &&\\
&\text{Implication:} && a \Rightarrow b = \max(1-a,b) && \text{(Reichenbach implication)}.
\end{aligned}$$

析取类型按约束选择：对于严格的“至少一个”语义，使用最大值；对于应累积部分满足度的情况，则使用概率和。

量化。 为了评估逻辑公式 F_i 在一批生成样本上的满足度，我们使用基于幂平均的可微聚合方案：

$$\mathrm{Sat}_{F_i}(B) = \mathrm{Agg}_{p_i}\left(\{F_i(x) : x \in B\}\right),$$

其中 Agg_p 表示指数为 $p \in \mathbb{R}$的幂平均 （或广义平均）。这使我们能够对一阶逻辑中的量词进行软近似。

对于全称量化（$\forall x$），我们使用指数较小且为正的幂平均（通常$p = 2$）。这样，聚合结果会对较低的真值更加敏感，确保即使只有少数违反情况，也会显著降低整体满足度。

对于存在量化（$\exists x$），我们采用一种基于补集的变体，以奖励至少存在一个满足度很高的样本。具体而言，我们计算：

$$\exists x\, F(x) \quad \approx \quad 1 - \mathrm{Agg}_p(1 - F(x)),$$

该方法通过反转并聚合低值，软性近似集合上的逻辑最大值。这会促使生成器至少生成一个能够很好地满足约束的样本，即使其他样本不满足约束也如此。

在对抗学习中的作用。 通过将L^{logic} 纳入L_G，生成器经过优化后，不仅能够欺骗判别器，还能生成高度满足所有领域规则的输出。这种紧密集成使 $\mathscr{K}$ 中的符号知识能够直接影响生成器的梯度更新，引导其朝着输出空间中逻辑有效的区域前进，而无需为规则本身提供配对训练数据。

3.3 复合生成器目标

生成器最小化一个复合目标

$$L_G \;=\; \alpha\, L_G^{\text{adv}} \;+\; \lambda_e\, L_{\text{logic}} \;+\; \beta\, L_{\text{aux}}.$$

这里，L_G^{adv}是非饱和对抗损失，L^{logic} 用于惩罚违反知识库的行为，而L_{aux}是特定于任务的（例如，MNIST 上的类别交叉熵），(α,β)则根据实验进行选择。[2]

[2]数据集特定细节。 **高斯：** 两个个体级谓词（范围和高斯形状），采用 $\forall$聚合；在前$K=80$ 个训练轮次中， $\lambda_e\uparrow$从 0.05 线性调度至 0.30；标签平滑（y_{real},y_{fake}）=（0.9,0.1）。**网格：** 在相同的G/D下，采用“位于网格上”和覆盖率（$\exists$）约束；所有情况均使用二元交叉熵；基线和 LTN 实验采用相同的架构与训练超参数；在G 中固定使用较小的噪声尺度，以形成紧密的簇。**环：** 采用层次化约束（内环/外环的存在性、互斥性、死区规避、空间一致性、平衡、精度细化），并使用自适应规则权重；单独的约束权重 λ_e 从 2 逐步提升至 10。**MNIST：** L_{aux} 为类别交叉熵；G 还包含少量模板混合（ 10–20% 的逐样本权重，由 $[z;y]$学习），以对笔画进行弱引导；LTN 项采用平衡混合 $\alpha=0.6,\ \beta=0.3$ 和$\lambda_e=0.1$。 MNIST 中的标签不进行平滑，而二维高斯和网格实验中的标签则进行平滑。

3.4 逻辑权重调度与自适应规则权重

我们采用一个简单的单调调度方案来设定全局逻辑权重：

$$\lambda_e = \begin{cases} \lambda_{\text{start}} + \frac{e}{K}(\lambda_{\text{end}} - \lambda_{\text{start}}), & e \le K, \\ \lambda_{\text{end}}, & e > K, \end{cases}$$

随着L^{logic}的学习逐渐强调G数据流形。在环设置中，我们还会根据满足度反馈调整每条规则的权重。令$s_{i,t} = \text{Sat}_{F_i}(B_t)$表示当前满足度，$w_{i,t}$表示当前权重。我们更新

$$\tilde{w}_{i,t} = \text{clip}\left(w_{i,t} \cdot \begin{cases} 1+\eta, & s_{i,t} < 0.3, \\ 1-\frac{1}{2}\eta, & s_{i,t} > 0.8, \\ 1+\eta\,(0.6 - s_{i,t}), & \text{otherwise}, \end{cases} ; w_{\min}, w_{\max} \right), \qquad w_{i,t+1} = m\, w_{i,t} + (1-m)\,\tilde{w}_{i,t},$$

采用动量m和学习率 η。这里，$\text{clip}(x; a,b) = \min(\max(x,a),b)$用于将值限制在固定范围内：如果$x < a$，则将其设为$a$；如果$x > b$，则将其设为$b$。这可以防止每条规则的权重消失或无限增长，确保其保持在稳定范围$[w_{\min}, w_{\max}]$内。这样可以提高满足度较低的规则的权重，并放宽始终得到满足的规则，从而稳定训练，同时在训练后期提高精确性。有关算法的更多细节，请参阅（附录 1）。

3.5 特定于数据集的谓词集

尽管整体的对抗式—逻辑框架在各项实验中保持一致，但具体的逻辑谓词集合及其聚合策略会根据每个领域进行调整。每个谓词$P(\cdot)$都实现为一个小型神经网络，其输出范围为 $[0,1]$ 的真值度。约束被表述为一阶模糊逻辑句子，由全称（$\forall$）和存在（$\exists$）量词、合取（$\wedge$）、析取（$\vee$）以及蕴含（$\Rightarrow$）构成。我们通过软幂平均实现量词，使用乘积t-范数表示合取，并对析取应用最大值或概率和。否定实现为 $\neg a = 1 - a$。

为了支持这些谓词的有效学习——尤其是在数据几何结构至关重要的合成领域中——我们在各项任务中采用温和的几何初始化策略。例如，我们为 GRID 数据集使用以网格为中心的初始化，在 RING 中采用径向先验，并在 MNIST 中使用模板混合。这些归纳偏置并不强制规定最终结构，而是充当脚手架，使符号谓词能够在训练早期发挥有意义的作用。

高斯（二维）。该任务涉及从以原点为中心的类二维高斯分布中生成样本（附录 B.1）。我们定义了两个在每个生成点上进行评估的个体级谓词。第一个谓词 $\text{InRange}(x) = \sigma(3 - \|x\|_\infty)$ 会对落在有界正方形之外的点进行软惩罚，其中 $\|x\|_\infty$ 是最大坐标幅值，σ 表示 Sigmoid 函数。第二个谓词 $\text{GaussianShape}(x) = \exp\left(-\frac{1}{2}(\|x\|/2.5)^2\right)$ 通过为靠近高斯众数的点赋予更高分数，促使样本集中在原点附近。两个谓词均通过以下全称公式进行约束：

$$\forall x \left(\text{InRange}(x) \wedge \text{GaussianShape}(x) \right),$$

这引导生成器生成空间范围受限且集中于中心的点。满足度通过批次中谓词真值度的均值计算。

二维网格。此任务旨在生成按结构化 2×2 网格布局排列的样本，其中包含四个间距均匀的网格中心（附录 B.2）。谓词 $\mathrm{OnGrid}(x) = f^{\theta_{\text{grid}}}(x)$ 通过可训练多层感知机实现，用于评估点 x 是否靠近任一网格中心，并通过全称约束 $\forall x \mathrm{OnGrid}(x)$ 加以约束。为确保网格覆盖率，我们为每个单元 $i = 1, \ldots, 4$ 定义四个单元特定谓词 $\mathrm{InCell}_i(x)$，并施加存在约束 $\exists x \mathrm{InCell}_i(x)$，以确保每个单元至少包含一个生成的点。完整的逻辑规范如下：

$$\Big(\forall x\, \mathrm{OnGrid}(x)\Big) \wedge \bigwedge_{i=1}^{4} \Big(\exists x\, \mathrm{InCell}_i(x)\Big).$$

这促使生成器生成几何形状规则且在网格布局中均匀分布的样本。

环（二维）。本实验针对一个由两个同心圆区域组成的合成数据集：一个内环和一个外环，两者之间隔着一片死区（附录 B.3）。目标是生成沿两个环分布、避开中间区域且满足六项逻辑约束层级的样本。所用谓词包括：$\mathrm{InnerRing}(x)$ 和 $\mathrm{OuterRing}(x)$，用于识别点 x 是否位于相应的环带内；$\mathrm{DeadZone}(x)$，用于指示 x 是否落在两个环之间的不理想区域；以及 $\mathrm{NearInnerCenter}(x)$ 和 $\mathrm{NearOuterCenter}(x)$，用于促使样本靠近各环的理想半径。完整的约束集为：(i) *存在性*：样本集中必须同时包含两个环中的样本，通过 $\exists x \mathrm{InnerRing}(x)$ 和 $\exists x \mathrm{OuterRing}(x)$ 实现；(ii) *互斥性*：任何点都不应同时属于两个环，通过 $\forall x\ \neg(\mathrm{InnerRing}(x) \wedge \mathrm{OuterRing}(x))$ 约束；(iii) *死区规避*：点不得落入分隔区域，通过 $\forall x\ \neg \mathrm{DeadZone}(x)$ 实现；(iv) *空间一致性*：环的归属必须符合真实的几何位置；(v) *平衡*：两个环中的样本比例应大致相同，并通过对样本数量设置软等式约束来建模；(vi) *精确性细化*：每个环上的点还应靠近其理想半径，通过类似 $\forall x\, (\mathrm{InnerRing}(x) \Rightarrow \mathrm{NearInnerCenter}(x))$ 的蕴含关系进行形式化，外环亦然。训练过程中逐步收紧内环和外环的容差，以随着训练推进提高精确性。

MNIST。在 MNIST 实验中，[13], 知识库结合了类别特定谓词和结构谓词，以确保生成数字在语义和视觉上的完整性（附录 B.4）。对于分类，我们为每个数字类别 $k \in \{0, \ldots, 9\}$ 定义一个谓词 $\mathrm{IsDigit}_k(x)$，每个谓词都由一个神经分类器实现，用于预测图像 x 是否对应数字 k。为评估视觉质量，我们引入一组通用有效性谓词：$\mathrm{ValidPixels}(x)$ 鼓励像素强度保持在有意义的动态范围内；$\mathrm{IsConnected}(x)$ 通过检查是否存在单个连通前景区域来促进拓扑连贯性；$\mathrm{IsComplete}(x)$ 确保数字形状闭合或形态完整；$\mathrm{HasProperIntensity}(x)$ 则对过于暗淡或饱和的输出施加惩罚，从而促进对比度平衡。我们通过 $\forall x \bigvee_k \mathrm{IsDigit}_k(x)$ 强制类别互斥性，并通过 $\forall x \bigwedge_{k \neq m} \neg(\mathrm{IsDigit}_k(x) \wedge \mathrm{IsDigit}_m(x))$ 强制互斥性。其他蕴含规则，例如 $\forall x\ \big(\mathrm{ValidPixels}(x) \Rightarrow (\mathrm{IsConnected}(x) \wedge \mathrm{IsComplete}(x))\big)$，引导生成器生成不仅可分类，而且在视觉上连贯、结构上有效的数字。

3.6 训练流程

在所有实验中，每个小批量的训练都交替进行一次判别器更新和一次生成器更新。设 $B \subset \mathscr{D}$ 表示训练轮次 e 中的一批真实样本。

判别器步骤。 我们采样真实样本 $x_R \sim B$ 和潜在向量 $z \sim \mathcal{N}(0,I)$，生成虚假样本 $x_G = G(z)$，并对 x_G 分离梯度，以防判别器更新期间生成器的梯度反向传播。判别器使用真实样本和虚假样本的二元交叉熵（BCE）损失进行训练：

$$L_D^{\text{real}} = \ell_{\text{BCE}}\big(D(x_R), y_{\text{real}}\big), \quad L_D^{\text{fake}} = \ell_{\text{BCE}}\big(D(x_G), y_{\text{fake}}\big),$$

$$L_D = L_D^{\text{real}} + L_D^{\text{fake}}.$$

此处的 $\ell_{\text{BCE}}(p, y) = -y\log p - (1-y)\log(1-p)$使用 Sigmoid 输出后的概率进行计算。在二维数据集（高斯、网格、环）中应用标签平滑，设置为$(y_{\text{real}}, y_{\text{fake}}) = (0。9, 0。1)$，而在 MNIST 中则省略（目标值为 $(1, 0)$）。计算出L_D后，我们进行反向传播并更新D的参数。

逻辑评估。 将同一批生成样本x_G 传入 LTN $\mathscr{C}$，由其评估每条逻辑规则$F_i \in \mathscr{K}$，并生成满足度分数 $\text{Sat}_{F_i}(B) \in [0,1]$。每条规则的满足度根据其谓词真值度，通过模糊联结词计算，并使用指定的量词（∀或 ∃）进行聚合。这些结果合并为全局满足度

$$S_{\text{logic}}(B) = \frac{\sum_i w_i \, \text{Sat}_{F_i}(B)}{\sum_i w_i},$$

其中w_i 是固定或自适应规则权重，具体取决于数据集。逻辑损失为 s

$$L_{\text{logic}}(B) = 1 - S_{\text{logic}}(B).$$

在环形数据集中，w_i 会根据满足度趋势进行自适应更新；在其他数据集中则保持固定。

生成器步骤。 我们重新采样 $z \sim \mathcal{N}(0,I)$，重新生成 $x_G = G(z)$，并计算非饱和对抗损失：

$$L_G^{\text{adv}} = \ell_{\text{BCE}}\big(D(x_G), y_{\text{real}}\big) \equiv -\mathbb{E}_z \log D(G(z)).$$

对于 MNIST，我们还会根据判别器分类头计算辅助分类损失 L_{aux}，将预测标签与生成器使用的条件标签进行比较。生成器总损失为

$$L_G = \alpha L_G^{\text{adv}} + \lambda(e) L_{\text{logic}} + \beta L_{\text{aux}},$$

其中，α 和 β 是数据集特定系数（例如，MNIST 中的 $\alpha = 0.6$、$\beta = 0.3$），而 $\lambda(e)$遵循第 3.4 节中针对相应数据集设定的逻辑权重计划。我们对 L_G 进行反向传播，并更新 G （以及 $\mathscr{C}$ 中任何可学习的谓词参数，如有）。

日志记录。 训练期间，我们记录 L_G、其对抗分量L_G^{adv}和 L_D；判别器准确率（真实样本和生成样本准确率的平均值）；全局逻辑满足度$S^{\text{logic}}(B)$；以及当前逻辑权重 $\lambda(e)$。在环形数据集中，我们还会记录每条规则的满足度及其自适应权重；在 MNIST 中，我们还会跟踪辅助分类准确率。

训练一致性。 基线生成对抗网络与 LTN–GAN 变体的优化器设置完全相同：使用 Adam，参数为 $(\beta_1, \beta_2) = (0.5, 0.999)$，采用特定于数据集的学习率和固定批次大小。更新计划始终是先进行一次 判别器 更新，再进行一次 生成器 更新。唯一的差异在于谓词集合、是否包含 L^{logic} 和 L_{aux}，以及 $\lambda(e)$ 计划。

3.7 实验设置

我们将 LTN–GAN 与基线生成对抗网络进行比较；两者的生成器/判别器架构、优化器设置、标签平滑和更新计划均相同，唯一差异是是否包含逻辑组件。实验涵盖三个合成二维数据集——高斯（有界范围和高斯形状谓词）、网格（2×2 对齐和单元格覆盖约束）和 环（存在性、互斥性、精确性、平衡和死区规避）——以及 MNIST（28×28 个数字类别，并采用类别一致性和视觉有效性规则）。二维生成器是使用 LeakyReLU、Dropout 和线性 $\mathbb{R}^2$ 输出的多层感知机；MNIST 生成器则通过 Sigmoid 输出，将潜在编码和类别嵌入映射到图像空间。判别器是使用 LeakyReLU 和 Dropout 的多层感知机；MNIST 判别器还包含一个辅助分类头。训练周期遵循代码设置：100 个训练轮次（高斯）、120 个（网格）、150 个（环），以及一个包含 5 步的演示循环（MNIST）；每个批次进行一次 判别器 更新和一次 生成器 更新。所有运行均使用 Adam（$\beta_1 = 0.5$，$\beta_2 = 0.999$），二维任务采用带标签平滑的二元交叉熵损失（0.9，0.1），MNIST 采用（1，0），并在 MNIST 中使用辅助分类损失。逻辑权重 $\lambda(e)$ 按数据集采用不同计划：线性递增（高斯）、固定常数（网格）、带有逐规则重新加权的自适应分阶段方案（环），以及固定的小常数（MNIST）。除 LTN 损失外，基线与 LTN–GAN 变体的所有超参数、随机种子和训练方案均相同。在不同数据集中，谓词要么是固定的解析函数（例如高斯任务中的范围内和高斯形状），要么是通过学习获得的神经网络（例如位于网格上、位于单元格内$_i$和 MNIST 数字分类器）。学习得到的谓词与生成器和判别器联合训练，并输出软真值 $[0,1]$,，从而能够对逻辑公式进行基于梯度的优化。

3.8 评估指标

我们使用训练期间计算的各项数值来评估模型：全局逻辑满足度分数$S^{\text{logic}}(B)$，它通过模糊量词聚合逐规则真值度 Sat$_{F_i}(B)$，并结合固定或自适应权重w_i得到；对抗损失，包括生成器总损失L_G、其对抗分量L_G^{adv}以及判别器损失L_D；还有判别器准确率，以真实样本和伪造样本分类准确率的平均值衡量。每个训练轮次都会记录逻辑权重 $\lambda(e)$的值，以直观展示数据集特定调度策略的效果；在环形数据集中，还会跟踪逐规则权重轨迹。对于所有任务，我们都会绘制L_G、L_G^{adv}、L_D、S^{logic}和 $\lambda(e)$随训练轮次变化的曲线；对于二维数据集，我们将生成点的散点图与目标几何形状进行比较；对于 MNIST，则检查生成的图像网格，以评估视觉质量和数字可辨识度。

4 结果

我们通过在四个数据集上将 LTN-GAN 与基线生成对抗网络进行比较，评估其有效性：**高斯**、**网格**、**环**和**MNIST** [3]。比较重点关注三个关键方面：对抗生成器损失、样本质量和逻辑规则满足度。这些指标既反映了传统的生成性能，也体现了模型融入符号结构的能力。基线生成对抗网络完全依赖数据驱动学习，而 LTN-GAN 则通过逻辑监督融入特定领域的约束。借助这一设置，我们可以评估逻辑感知训练对生成结果的几何结构、样本多样性和规则一致性的影响。

[3]代码可在以下网址获取：github.com/nuuoe/LTN-GAN/

跨数据集比较

表1展示了 LTN-GAN 与基线生成对抗网络在全部四个数据集上的性能比较，报告了生成器损失（对抗损失和总损失）、判别器损失、样本质量和逻辑满足度。在所有情况下，LTN-GAN 都获得了更高的质量评分，并实现了显著的逻辑规则满足度，而基线模型不具备这一点。在**高斯**数据集上，LTN-GAN 生成的样本具有更好的统计结构，质量评分从 0.183 提高到 0.470，逻辑满足度达到 0.916。由于逻辑惩罚，总生成器损失略有增加（0.671，而对抗损失为 0.635），但训练仍保持稳定。对于**网格** 数据集，空间一致性和网格对齐显著改善，质量评分从 0.387 升至 0.775，逻辑满足度达到 0.823。生成器损失同样适度增加（总损失为 1.031，而对抗损失为 0.441），同时判别器损失保持平衡。在更复杂的**环** 数据集上，LTN-GAN 在内环和外环上生成了径向分布均衡的样本，质量评分较高（0.964，而基线为 0.562），逻辑满足度达到 0.817，但由于存在多个空间约束，生成器损失也更高。最后，在**MNIST**上，逻辑感知监督改善了数字结构和识别效果，使质量评分从 0.360 升至 0.395，逻辑满足度接近完美（0.978）；尽管由于分类要求更强，判别器损失有所增加。这些结果表明，LTN-GAN 能够有效平衡对抗与符号目标，提升其在不同领域的生成性能。

数据集	模型	G损失（对抗/总计）	判别器损失	质量评分	逻辑满足度
高斯	基线生成对抗网络	0.830	1.390	0.183	N/A
高斯	LTN-GAN	0.635/0.671	1.488	**0.470**	**0.916**
Grid	基线生成对抗网络	0.700	0.692	0.387	N/A
Grid	LTN-GAN	0.441/1.031	0.693	**0.775**	**0.823**
Ring	基线生成对抗网络	0.708	1.334	0.562	N/A
Ring	LTN-GAN	3.566/3.566	1.364	**0.964**	**0.817**
MNIST	基线生成对抗网络	1.717	2.319	0.360	N/A
MNIST	LTN-GAN	0.727/0.279	2.789	**0.395**	**0.978**

表1：LTN-GAN 与基线生成对抗网络在各数据集上的性能比较。

消融研究

我们针对特定数据集开展消融研究，以评估不同逻辑配置对样本生成和结构保真度的影响。每种变体均测试约束强度、调度动态或辅助监督组件的有无。有关用于量化生成质量、结构对齐和逻辑满足度的特定领域指标的详细说明，请参阅附录6。

高斯数据集：表2显示，在逻辑约束下，统计和几何遵循度均显著提升。ltn_high_约束变体取得了最高的综合质量（0.598），逻辑满足度为0.925，证实更严格的约束能够增强结构。相比之下，基线和无约束变体在结构和误差指标方面均表现较差。即使没有显式逻辑公式，ltn_- no_constraints变体仍能从对软谓词的提升推理中获益。

网格数据集： 从表 3 可以看出，逻辑驱动变体表现出更高的聚类精度和完整的网格覆盖率。尽管 fast_scheduling 由于约束暴露不足而导致覆盖率较低，但其仍保持了较高的样本质量。

变体	高斯符合度	统计质量	综合质量	均值误差	标准差误差	逻辑满足度
基线_生成对抗网络	0.200	0.165	0.183	0.641	0.175	0.000
完整_LTN_生成对抗网络	0.476	0.465	0.470	0.431	0.719	0.916
ltn_no_约束	0.402	0.501	0.451	0.327	0.670	0.916
LTN_高_约束	0.608	0.587	**0.598**	**0.030**	0.673	0.925
LTN_快速_逻辑调度	0.339	0.637	0.488	0.043	0.527	0.903
LTN_慢速_逻辑调度	0.393	0.408	0.400	0.946	0.508	0.849

表2：高斯　　数据集消融：逻辑调度和逻辑约束能够改善统计结构　　e.

基线无法将样本对齐到目标区域，导致 1000 个样本中仅有 607 个位于定义的网格边界内。相比之下，所有逻辑变体都通过存在量化网格单元约束确保了完全对齐。

变体	网格簇	覆盖率	质量评分	总体性能	目标内	逻辑满足度
基线_生成对抗网络	0.607	0.745	0.387	0.573	607	0.000
完整_LTN_生成对抗网络	1.000	0.638	0.775	**0.882**	1000	0.823
ltn_no_约束	0.057	0.718	0.205	0.288	57	0.500
LTN_高_约束	1.000	0.591	0.793	0.877	1000	0.809
LTN_快速_调度	1.000	0.179	0.778	0.790	1000	0.802
LTN_慢速_调度	1.000	0.445	0.721	0.830	1000	0.806

表 3：网格数据集消融：逻辑约束与聚类遵循度。

环形数据集： 如表 4 所示，LTN 变体在环形遵循度、平衡和死区规避方面显著优于基线。移除渐进式训练阶段或层次化规则权重会使性能略有下降，但仍能保持较高的逻辑满足度。值得注意的是，简单_-约束配置在平衡和规则遵从度方面与更复杂的设置不相上下，甚至表现更优，这表明即使是扁平化约束结构，也能从符号归纳偏置中受益。

变体	环形遵循度	内部	外部	平衡	死区规避	逻辑满足度
基线_生成对抗网络	0.524	231	0	0.000	0.462	0.000
完整_LTN_生成对抗网络	0.970	291	188	0.785	0.958	0.817
no_约束	0.462	76	156	0.655	0.868	0.702
无_层次化_权重	0.970	265	217	0.900	0.964	0.826
无_渐进式_阶段	0.922	258	209	0.895	0.934	**0.856**
简单_约束	0.960	258	224	**0.929**	0.964	0.824

表 4：环形数据集消融：对称性、平衡和逻辑一致性。

MNIST 数据集： 表 5 显示，基于逻辑的模型在数字识别和结构方面表现更佳，尤其是在包含模板先验时。移除模板会导致识别效果和逻辑满足度下降，这表明模板对于早期引导至关重要。有趣的是，即使是较弱的_- ltn 配置，在数字分类方面也能达到或超过其他变体，这凸显了符号监督与数据层面的多样性相结合时的稳健性。

变体	质量评分	覆盖率	数字识别	模板依赖	逻辑满足度
基线_生成对抗网络	0.360	0.069	0.717	0.700	0.000
完整_LTN_生成对抗网络	0.395	0.070	0.725	0.700	**0.978**
no_ltn_约束	0.392	0.069	0.720	0.700	0.000
无_模板	0.367	0.057	0.685	0.000	0.977
weak_ltn	0.391	0.069	**0.731**	0.700	0.977
强_LTN	**0.398**	0.069	0.727	0.700	0.978

表 5：MNIST 消融研究：逻辑提升了识别能力和语义准确性。

4.1 理解输出变异性与逻辑设计的影响

我们在所有数据集上进行的消融研究揭示了影响 LTN-GAN 性能波动的若干重要因素。尽管集成逻辑显然能够改善结构、语义和泛化能力，但提升幅度很大程度上取决于逻辑组件的设计与实现方式。

变异性的一个关键来源在于逻辑约束的制定方式。例如，在环形数据集中，我们发现，过于简单的约束会导致几何结构较差，且各样本类别之间的平衡度降低。另一方面，层次丰富且语义充实的逻辑能够提升环形遵循度和逻辑满足度。这表明，有意义的约束设计对于生成结构化样本至关重要，尤其是在几何结构或符号关系起重要作用的任务中。

另一个重要因素是训练过程中应用逻辑的方式。采用渐进式调度的变体——即逐步增强逻辑影响力——始终优于那些以统一或突然的方式应用约束的变体。这种分阶段的方法使模型能够先探索数据空间，随后再依据逻辑结构优化输出。这种课程式训练的优势在 Ring 和 Grid 数据集中尤为明显。

约束加权也起着重要作用。当所有约束都被同等对待，或其重要性固定不变时，模型往往会忽略较弱的约束，或对较强的约束过拟合。我们的实验表明，动态层次加权——根据约束当前的满足程度来调整其权重——有助于在学习一般模式与保持逻辑结构之间取得平衡。这一策略能够同时提升质量和逻辑遵循度。

此外，我们观察到不同数据集对相同的架构变更会产生不同的响应。例如，MNIST 在大多数消融实验中表现稳定，这可能得益于其明确的视觉结构。相比之下，需要显式空间关系的合成数据集对约束设计和调度更加敏感。这凸显了在应用基于 LTN 的生成模型时，针对特定数据集进行逻辑设计的重要性。

最后，我们的研究结果证实了样本质量与逻辑满足度之间存在权衡。提高逻辑约束力度通常会导致样本变异性降低，尤其是在约束较强的变体中。这表明，必须在逻辑约束与生成灵活性之间进行谨慎平衡，以避免模式崩塌或输出过于僵化。尽管 LTN-GAN 相较于传统生成对抗网络具有显著优势，但要实现最高性能，仍需仔细调节约束结构、逻辑权重调度策略，并考虑数据集的特定因素。我们的结果表明，相较于固定配置，这些模型更受益于自适应策略，尤其是在结构准确性和语义有效性至关重要的领域。

5 讨论

我们的研究表明，LTN-GAN 能够有效地将符号推理整合到生成过程中，使生成器产生的不仅是类似真实数据的输出，还能符合领域特定逻辑约束。在各项实验中，该模型在逻辑满足度和结构质量方面始终优于无约束基线。然而，这种整合也带来了一个众所周知的矛盾：强制维持逻辑一致性往往会限制样本多样性，从而削弱生成器探索完整数据流形的能力。在消融研究中，这种权衡尤为明显：更高的约束权重提高了逻辑遵循度，但有时会降低视觉变化性或覆盖率。生成器的目标函数使用可调权重 α、λ（e）和 β，将对抗损失与逻辑损失结合起来。这些超参数在平衡质量、多样性和规则满足度方面发挥着关键作用。尽管我们的调度策略和针对架构的调整实现了合理的控制，但仍需仔细调参，这表明 LTN-GAN 目前尚缺乏能够自主管理这一权衡的自适应机制。

为了将 LTN-GAN 置于更广泛的神经符号领域中，我们注意到，一些模型将符号结构嵌入潜在空间，以间接实施规则。例如，VAEL [21] 将逻辑编程融入变分自编码器，而 MultiplexNet [9] 对分层类别变量进行建模，以引导生成。这些方法与 LTN-GAN 一样，旨在施加结构约束，但它们依赖概率嵌入，而不是在训练目标中直接进行逻辑推理。

其他策略使用可微替代项近似符号约束，使其与基于梯度的学习兼容。语义损失 [32] 是最早将逻辑规则表示为可微惩罚项的方法之一。这启发了伪语义损失 [1], 等方法，该方法将约束实施局部化于自回归模型中；以及语义目标函数 [20], ，该方法借助信息几何工具将逻辑嵌入损失函数。这些方法功能强大，但通常与对抗动态结合得不够紧密，而不稳定梯度和非平稳目标函数会使约束实施变得复杂。

通过投影或细化实现事后约束满足，是另一个颇有前景的方向。例如，SketchGen [24]和 Vitruvion [27]可确保 CAD 草图中的几何一致性，而基于朗之万动力学的细化 [12] 则通过迭代调整生成结果，使其趋近可行区域。这些方法无需影响核心生成过程，即可提高约束满足度，但会牺牲训练时的推理能力，还可能导致效率低下或输出脆弱。

我们的研究结果还揭示了硬约束与软约束实施方式的局限性。硬约束虽然能确保合规，却可能限制多样性并加剧模式崩塌。相比之下，软惩罚更具灵活性，但往往无法提供保证。为弥合这一差距，研究者提出了混合机制，例如用于离散约束的直通估计器 [34], ，以及在可行性与探索之间取得平衡的对偶优化方法 [5, 19]。LTN-GAN采用软逻辑损失形式，但可能仍需进一步细化，以避免陷入局部极小值或未能充分利用规则。

当前神经符号生成对抗网络的核心缺陷之一，是符号推理与对抗学习之间缺乏深度融合。大多数模型通过潜在空间先验或事后修正间接注入逻辑，而非直接影响生成器更新。例如，像 [25] 这样的研究依赖规则注入表示，却无法动态调整规则。LTN-GAN 通过使逻辑满足度可微且可联合训练，向前迈进了一步，但仍依赖预定义的规则集。未来的扩展或可受益于动态规则归纳，即从数据中学习约束，或根据上下文推断约束。元溯因 [25]和潜在符号对齐 [4] 为这一方向提供了有力基础，而结构化离散数据的目标导向生成方面的进展[22] 则展示了通过学习得到的奖励优化直接强制满足属性的互补策略

。

可扩展性仍然是一项尚待解决的挑战。LTN-GAN 在 MNIST 和合成几何体等结构化数据集上表现出色，但扩展到更高维的任务（例如自然图像或分子图）并非易事。一种可行的方向是采用分层或几何求解器 [27], ，从而使逻辑能够在多个抽象层次上运行。此外，集成能够在训练期间评估规则违反情况的约束感知型判别器，可以在不牺牲多样性的前提下加强逻辑约束。

总体而言，我们的工作推动了约束感知生成建模这一新兴领域的发展，与将逻辑和学习相结合的研究方向一致 [17, 30, 33, 10]。近期提出的框架，例如生成式神经符号机器 [11] 以及溯因细化方法 [25]，进一步彰显了符号推理增强生成控制的潜力。然而，在对抗训练中稳健地集成逻辑仍有待深入研究。要充分发挥神经符号生成对抗网络的潜力，关键在于解决约束优化、对抗稳定性和动态逻辑学习方面的挑战。

5.1 生成式推理的更广泛影响

尽管本研究在标准生成对抗框架下评估了 LTN-GAN，但值得指出的是，逻辑张量网络的潜力并不局限于提升此类场景中的样本质量或多样性。生成对抗网络受其约束性和对抗性特征所限，往往难以充分发挥符号逻辑集成的表达能力。具体而言，生成器仍主要针对对抗目标进行优化，而这可能与更深层次的结构化推理相冲突。

我们认为，逻辑张量网络的真正优势在于能够支持逻辑引导的生成式推理。这对于场景生成、组合式图像合成或多模态符号落地等任务尤其重要，因为这些领域的生成过程涉及满足复杂关系、空间配置和基于规则的依赖关系。在此类情境中，逻辑张量网络不仅可以充当正则化器，还能推动结构化生成，在生成过程中施加高层语义规则。未来的研究应探索这一方向，将逻辑张量网络集成到生成对抗网络之外的架构中，例如扩散模型、自回归解码器或神经符号场景模拟器，使符号推理能够直接塑造内容生成，而非仅在事后对其加以约束。

6 结论

LTN-GAN 是神经符号生成建模领域的一项进展，表明逻辑约束能够有效地引导对抗学习。然而，多样性与遵循度之间的权衡、手动规则指定以及可扩展性方面的限制等挑战，凸显了未来需要重点改进的领域。自适应规则学习、约束感知对抗训练和可扩展的逻辑执行技术方面的研究，对于推动神经符号生成对抗网络发展为更稳健、灵活的框架至关重要。解决这些挑战，将使生成对抗网络能够在不同领域生成逻辑结构清晰、质量高的输出，同时保持生成多样性和灵活性。

7 致谢

Vaishak B 该项目由思科资助，并获得皇家学会大学研究奖学金支持 p.

参考文献

[1] *Kareem Ahmed*、*Kai-Wei Chang & Guy Van den Broeck*（*2023*）：具有逻辑约束的自回归模型的伪语义损失。载于：《神经信息处理系统进展》第*36*卷：*2023*年神经信息处理系统年度会议，*NeurIPS 2023*，美国路易斯安那州新奥尔良，*2023*年*12*月*10*日至*16*日。*URL http://papers.nips.cc/paper_files/paper/2023/hash/3accfe8332366a6f740d8740cd4cd653-Abstract-Conference.html*。[2] *Samy Badreddine*、*Artur S. d'Avila Garcez*、*Luciano Serafini & Michael Spranger*（*2022*）：逻辑张量网络。《人工智能》*303*，*103649*页。*URL https://doi.org/10.1016/j.artint.2021.103649*。[3] *Xiaopeng Chao*、*Jiangzhong Cao*、*Yuqin Lu*、*Qingyun Dai & Shangsong Liang*（*2021*）：受约束生成对抗网络。《*IEEE Access*》*9*，*19208–19218*页。*URL https://doi.org/10.1109/ACCESS. 2021.3054822*。[4] *Cristina Cornelio*、*Jan Stuehmer*、*Shell Xu Hu & Timothy M. Hospedales*（*2023*）：在神经符号推理中学习何时以及在哪里进行推理。载于：第十一届国际学习表征会议，*ICLR 2023*，卢旺达基加利，*2023*年*5*月*1*日至*5*日，*OpenReview.net*。*URL https://openreview.net/forum? id=en9V5F8PR-*。[5] *Aaron M. Ferber*、*Arman Zharmagambetov*、*Taoan Huang*、*Bistra Dilkina & Yuandong Tian*（*2024*）：*GenCO*：利用组合约束生成多样化设计。载于：第四十一届国际机器学习会议，*ICML 2024*，奥地利维也纳，*2024*年*7*月*21*日至*27*日。*URL https://openreview.net/ forum?id=DiyE6OOGBa*。[6] *Ian J. Goodfellow*、*Jean Pouget-Abadie*、*Mehdi Mirza*、*Bing Xu*、*David Warde-Farley*、*Sherjil Ozair*、*Aaron C. Courville & Yoshua Bengio*（*2014*）：生成对抗网络。载于：《神经信息处理系统进展》第*27*卷：*2014*年神经信息处理系统年度会议，*2014*年*12*月*8*日至*13*日，加拿大魁北克省蒙特利尔，*2672–2680*页。*URL https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html*。[7] *Eric Heim*（*2019*）：用于交互式图像生成的受约束生成对抗网络。载于：*IEEE*计算机视觉与模式识别会议，*CVPR 2019*，美国加利福尼亚州长滩，*2019*年*6*月*16*日至*20*日，计算机视觉基金会 / *IEEE*，*10753–10761*页。*URL https://doi.org/10.1109/CVPR. 2019.01101*。[8] *Philipp Hess*、*Markus Drüke*、*Stefan Petri*、*Felix M. Strnad & Niklas Boers*（*2022*）：用于改进地球系统模型降水场的物理约束生成对抗网络。《自然机器智能》*4(10)*，*828–839*页。*URL https://doi.org/10.1038/s42256-022-00540-1*。[9] *Nick Hoernle*、*Rafael-Michael Karampatsis*、*Vaishak Belle & Kobi Gal*（*2022*）：*MultiplexNet*：迈向神经网络中完全满足的逻辑约束。载于：第三十六届*AAAI*人工智能会议，*AAAI 2022*；第三十四届人工智能创新应用会议，*IAAI 2022*；第十二届人工智能教育进展研讨会，*EAAI 2022*，线上会议，*2022*年*2*月*22*日至*3*月*1*日，*5700–5709*页。*URL https://doi.org/10.1609/aaai.v36i5. 20512*。[10] *Zhiting Hu*、*Zichao Yang*、*Ruslan Salakhutdinov*、*Lianhui Qin*、*Xiaodan Liang*、*Haoye Dong & Eric P. Xing*（*2018*）：具有可学习知识约束的深度生成模型。载于：《神经信息处理系统进展》第*31*卷：*2018*年神经信息处理系统年度会议，*NeurIPS 2018*，*2018*年*12*月*3*日至*8*日，加拿大蒙特利尔，*10522–10533*页。*URL https://proceedings.neurips.cc/paper/2018/hash/d7e77c835af3d2a803c1cf28d60575bc-Abstract.html*。[11] *Jindong Jiang & Sungjin Ahn*（*2020*）：生成式神经符号机器。载于：《神经信息处理系统进展》第*33*卷：*2020*年神经信息处理系统年度会议，*NeurIPS 2020*，*2020*年*12*月*6*日至*12*日，线上会议。*URL https://proceedings.neurips.cc/paper/2020/hash/94c28dcfc97557df0df6d1f7222fc384-Abstract.html*。

[12] *Sachin Kumar*、*Biswajit Paria* 和 *Yulia Tsvetkov*（*2022*）：基于梯度的语言模型约束采样。载于：*2022*年自然语言处理经验方法会议论文集，第*2251–2277*页。网址 *https://doi.org/10.48550/arXiv.2205.12558*。[13] *Yann LeCun*、*Léon Bottou*、*Yoshua Bengio* 和 *Patrick Haffner*（*1998*）：应用于文档识别的基于梯度的学习。《*IEEE*论文集》*86(11)*，第*2278–2324*页。网址 *https://doi.org/10.1109/5.726791*。[14] *Wanxin Li*（*2020*）：基于生成对抗网络的合成数据生成中的数据库约束支持。载于：*2020*年国际数据管理会议论文集，*SIGMOD 2020*会议，线上会议 *[*美国俄勒冈州波特兰*],2020*年*6*月*14*日至*19*日，第*2875–2877*页。网址 *https://doi.org/10.1145/3318464.3384414*。[15] *Zenan Li*、*Yunpeng Huang*、*Zhaoyu Li*、*Yuan Yao*、*Jingwei Xu*、*Taolue Chen*、*Xiaoxing Ma* 和 *Jian Lu*（*2023*）：产生逻辑约束的神经符号学习。载于：《神经信息处理系统进展》*36*：*2023*年神经信息处理系统年度会议，*NeurIPS 2023*，美国路易斯安那州新奥尔良，*2023*年*12*月*10*日至*16*日。网址 *http://papers.nips.cc/paper_files/paper/2023/hash/ 4459c3c143db74ee52afebdf56836375-Abstract-Conference.html*。[16] *Zenan Li*、*Zehua Liu*、*Yuan Yao*、*Jingwei Xu*、*Taolue Chen*、*Xiaoxing Ma* 和 *Jian Lü*（*2023*）：在有逻辑约束、但无捷径满足的情况下学习。载于：第十一届国际学习表征会议，*ICLR 2023*，卢旺达基加利，*2023*年*5*月*1*日至*5*日。网址 *https://openreview.net/forum? id=M2unceRvqhh*。[17] *Luca Di Liello*、*Pierfrancesco Ardino*、*Jacopo Gobbi*、*Paolo Morettin*、*Stefano Teso* 和 *Andrea Passerini*（*2020*）：利用约束对抗网络高效生成结构化对象。载于：《神经信息处理系统进展》*33*：*2020*年神经信息处理系统年度会议，*NeurIPS 2020*，*2020*年*12*月*6*日至*12*日，线上会议。网址 *https://proceedings.neurips.cc/paper/2020/hash/a87c11b9100c608b7f8e98cfa316ff7b-Abstract.html*。[18] *Dayiheng Liu*、*Jie Fu*、*Qian Qu* 和 *Jiancheng Lv*（*2019*）：*BFGAN*：用于词汇约束句子生成的后向和前向生成对抗网络。《*IEEE/ACM*音频、语音与语言处理汇刊》*27(12)*，第*2350–2361*页。网址 *https://doi.org/10.1109/TASLP.2019.2943018*。[19] *Xianggen Liu*、*Qiang Liu*、*Sen Song* 和 *Jian Peng*（*2020*）：用于序列优化的机会约束生成框架。载于：第*37*届国际机器学习会议论文集，*ICML 2020*，*2020*年*7*月*13*日至*18*日，线上会议，第*6271–6281*页。网址 *http://proceedings.mlr.press/v119/ liu20i.html*。[20] *Miguel Ángel Méndez Lucero*、*Enrique Bojorquez Gallardo* 和 *Vaishak Belle*（*2025*）：语义目标函数：一种在深度学习中添加逻辑约束的分布感知方法。载于：第*17*届智能体与人工智能国际会议论文集，*ICAART 2025*——第*3*卷，葡萄牙波尔图，*2025*年*2*月*23*日至*25*日，第*909–917*页。网址 *https://doi.org/10.5220/0013229200003890*。[21] *Eleonora Misino*、*Giuseppe Marra* 和 *Emanuele Sansone*（*2022*）：*VAEL*：连接变分自编码器与概率逻辑编程。载于：《神经信息处理系统进展》*35*：*2022*年神经信息处理系统年度会议，*NeurIPS 2022*，美国路易斯安那州新奥尔良，*2022*年*11*月*28*日至*12*月*9*日。网址 *http://papers.nips.cc/paper_files/paper/2022/hash/ 1e38b2a0b77541b14a3315c99697b835-Abstract-Conference.html*。[22] *Amina Mollaysa*、*Brooks Paige* 和 *Alexandros Kalousis*（*2020*）：利用条件生成模型目标导向地生成离散结构。载于：《神经信息处理系统进展》*33*：*2020*年神经信息处理系统年度会议，*NeurIPS 2020*，*2020*年*12*月*6*日至*12*日，线上会议。网址 *https://proceedings.neurips.cc/paper/2020/hash/f9b9f0fef2274a6b7009b5d52f44a3b6-Abstract.html*。[23] *Paolo Morettin*、*Andrea Passerini* 和 *Stefano Teso*（*2021*）：利用约束对抗网络共同创作平台跳跃游戏关卡。载于：与第*26*届*ACM*智能用户界面会议（*ACM IUI 2021*）同期举行的*ACM IUI 2021*研讨会联合论文集，美国大学城，*2021*年*4*月*13*日至*17*日。网址 *https://ceur-ws.org/Vol-2903/IUI21WS-HAIGEN-9.pdf*。

[24] Wamiq Reyaz Para、Shariq Farooq Bhat、Paul Guerrero、Tom Kelly、Niloy J. Mitra、Leonidas J. Guibas 和 Peter Wonka（2021）：SketchGen：生成受约束的CAD草图。载于：《神经信息处理系统进展》34：2021年神经信息处理系统年度会议，NeurIPS 2021，2021年12月6日至14日，线上会议，第5077–5088页。网址 https://proceedings.neurips.cc/paper/2021/hash/28891cb4ab421830acc36b1f5fd6c91e-Abstract.html。[25] Yifei Peng、Yu Jin、Zhexu Luo、Yao-Xiang Ding、Wang-Zhou Dai、Zhong Ren 和 Kun Zhou（2023）：通过理解来生成：具有逻辑符号落地的神经视觉生成。CoRR abs/2310.17451。网址 https://doi.org/10.48550/arXiv.2310.17451。[26] Yuyan Quan、Songtao Guo 和 Dewen Qiao（2024）：RCFL-GAN：利用生成对抗网络实现资源受限的联邦学习。载于：第27届计算机支持的设计协同工作国际会议，CSCWD 2024，中国天津，2024年5月8日至10日，IEEE，第513–518页。网址 https://doi.org/10.1109/CSCWD61410.2024.10580839。[27] Ari Seff、Wenda Zhou、Nick Richardson 和 Ryan P. Adams（2022）：Vitruvion：参数化CAD草图生成模型。载于：第十届国际学习表征会议，ICLR 2022，线上会议，2022年4月25日至29日，OpenReview.net。网址 https://openreview.net/forum?id=Ow1C7s3UcY。[28] Luciano Serafini 和 Artur S. d'Avila Garcez（2016）：利用逻辑张量网络进行学习与推理。载于：AI*IA 2016：人工智能进展——第十五届意大利人工智能协会国际会议，意大利热那亚，2016年11月29日至12月1日，论文集，《计算机科学讲义》10037卷，Springer，第334–348页。网址 https://doi.org/10.1007/ 978-3-319-49130-1_25。[29] Preeti Sharma、Manoj Kumar、Hitesh Kumar Sharma 和 Soly Mathew Biju（2024）：生成对抗网络（GAN）：简介、分类法、变体、局限性与应用。《多媒体工具与应用》83(41)，第88811–88858页。网址 https://doi.org/10.1007/s11042-024-18767-y。[30] Mihaela C. Stoian、Salijona Dyrmishi、Maxime Cordy、Thomas Lukasiewicz 和 Eleonora Giunchiglia（2024）：你的合成数据有多逼真？对表格数据的深度生成模型施加约束。载于：第十二届国际学习表征会议，ICLR 2024，奥地利维也纳，2024年5月7日至11日，OpenReview.net。网址 https://openreview.net/forum?id=tBROYsEz9G。[31] Mihaela C. Stoian 和 Eleonora Giunchiglia（2025）：超越凸性假设：在无量词实线性约束下生成逼真的表格数据。载于：第十三届国际学习表征会议，ICLR 2025，新加坡，2025年4月24日至28日，OpenReview.net。网址 https:// openreview.net/forum?id=rx0TCew0Lj。[32] Jingyi Xu、Zilu Zhang、Tal Friedman、Yitao Liang 和 Guy Van den Broeck（2018）：用于结合符号知识进行深度学习的语义损失函数。载于：第35届国际机器学习会议论文集，ICML 2018，斯德哥尔摩国际展览中心，瑞典斯德哥尔摩，2018年7月10日至15日，第5498–5507页。网址 http://proceedings.mlr.press/v80/xu18h.html。[33] Yexiang Xue 和 Willem-Jan van Hoeve（2019）：将决策图嵌入生成对抗网络。载于：约束编程、人工智能与运筹学融合——第16届国际会议，CPAIOR 2019，希腊塞萨洛尼基，2019年6月4日至7日，论文集，《计算机科学讲义》11494卷，Springer，第616–632页。网址 https://doi.org/10.1007/ 978-3-030-19212-9_41。[34] Zhun Yang、Joohyung Lee 和 Chiyoun Park（2022）：通过直通估计器将逻辑约束注入神经网络。载于：国际机器学习会议，ICML 2022，美国马里兰州巴尔的摩，2022年7月17日至23日，第25096–25122页。网址 https://proceedings.mlr.press/v162/ yang22h.html。[35] Halley Young、Maxwell Du 和 Osbert Bastani（2022）：具有关系约束的序列神经符号深度生成模型。载于：《神经信息处理系统进展》35：2022年神经信息处理系统年度会议，NeurIPS 2022，美国路易斯安那州新奥尔良，

美国，*2022*年*11*月*28*日至*12*月*9*日。网址：http://papers.nips.cc/paper_files/paper/2022/hash/f13ceb1b94145aad0e54186373cc86d7-Abstract-Conference.html。[36] Yang Zeng、Jin-Long Wu 和 Heng Xiao（2021）：通过不精确约束实现生成对抗网络对物理系统的模拟。《计算物理通讯》 30(3)，第 635–665 页。网址：https://doi.org/10.48550/arXiv.1911.06671。

A 扩展方法

A.1 详细实验设置

我们将 LTN-GAN 与基线生成对抗网络进行评估比较。两者采用相同的生成器和判别器架构、优化设置及训练计划，但基线网络不采用任何基于逻辑的监督。这种并行设置确保观察到的性能差异源于逻辑约束的整合，而非架构或学习动态上的偶然差异。

实验在三个合成数据集——高斯、网格和环——以及基于图像的*MNIST* 数据集上进行。在所有情况下，生成器均为全连接多层感知机（MLP），采用 LeakyReLU 激活函数，隐藏层大小为 128。对于合成数据集，生成器接收来自标准正态分布的二维潜向量，并生成不使用输出激活函数的二维样本，从而使 $\mathbb{R}^2$中的输出不受限。MNIST 生成器以与 100 维高斯噪声拼接的独热数字标签作为条件，输出 28×28 灰度图像。判别器的结构与生成器相匹配，输出单个概率分数；对于 MNIST，还添加了一个用于预测数字的辅助分类头。

训练使用 Adam 优化器，学习率为 0.001，$\beta_1 = 0.5$和 $\beta_2 = 0.999$，共进行 100 个训练轮次。合成数据使用 32 的批次大小，MNIST 使用 64。判别器采用标签平滑：真实样本的标签设为 0.9，生成样本的标签设为 0.1。在所有 LTN-GAN 设置中，生成器损失由对抗组件和逻辑组件构成：$L_G = \alpha L^{\text{adv}} + \lambda(e) L^{\text{logic}} + \beta L^{\text{aux}}$，其中 $\alpha = 1$、$\beta = 0.2$（如适用）和 $\lambda(e)$由前K 个训练轮次内的线性递增调度控制。对于高斯数据集，λ 在 80 个训练轮次内从 0.05 增至 0.3；对于环和网格数据集，在 50 个训练轮次内从 0.1 增至 0.5；对于 MNIST，在 60 个训练轮次内从 0.05 增至 0.25。

逻辑增强训练中采用了其他稳定机制。对于环和 MNIST，如果检测到规则满足度突然下降，基于检查点的回溯机制会使生成器回退到之前保存的 20 个状态之一。MNIST 还采用逻辑感知课程：训练初期混入数字模板先验以引导结构，随后逐渐淡化这些先验，使生成器能够自主学习符号约束。在所有合成数据集上，都会向生成器的潜在空间添加高斯噪声（标准差为 0.1），以促进探索多样性。每 10 个训练轮次进行一次评估：从生成器采样 1,000 个点，并计算两个主要指标：(i) 使用模糊逻辑聚合对约束集进行评估，得到逻辑规则满足度；(ii) 几何多样性，通过成对欧几里得距离衡量（MNIST 则通过像素方差衡量）。此外，MNIST 还通过学习得到的谓词网络跟踪分类准确率以及亮度和居中性统计数据。所有实验均使用 PyTorch 实现，并在 MacBook Pro （ 2.4 GHz 8 核 Intel Core i9 、 64 GB 内存、 AMD Radeon Pro 5500M 8 GB GPU）上运行。

为便于进行符号监督，每个数据集都采用一组针对其几何或语义结构定制的逻辑谓词。对于合成领域，这些谓词通过坐标空间上的固定数学函数或简单的基于阈值的启发式方法实现。在网格和环形数据集中，存在约束和全称约束确保几何区域的空间覆盖与分离，并通过模糊逻辑聚合计算满足度。相比之下，MNIST 谓词由小型多层感知机学习，并与生成器同步训练，用于评估数字类别、视觉清晰度和空间对齐等属性。量词通过可微幂平均实现——全称规则使用类似最小值的运算，存在规则使用类似最大值的运算——从而使梯度能够通过逻辑公式传播。系统会逐条追踪规则的逻辑满足度，并可选择应用自适应权重，将学习重点放在表现欠佳的约束上。这个模块化逻辑评估框架使 LTN-GAN 能够灵活地纳入符号规则，同时不干扰对抗学习循环。

算法 1LTN–GAN 训练算法[1][2]

需要： 训练数据 $\mathscr{D}$、生成器 G、判别器 D、包含规则 R 的约束系统 $\mathscr{C}$、配置 θ（包括 $\lambda(e)$），可选辅助损失 $\mathscr{A}$

确保： 训练后的模型 G^* ，D^*

初始化 G、D 以及可选的 $\mathscr{C}$ 的参数
创建优化器：opt_G、opt_D（如适用，还包括 $\text{opt}_{\mathscr{C}}$）
设置标签平滑值：$y_{\text{real}} \leftarrow 0.9$、$y_{\text{fake}} \leftarrow 0.1$
定义逻辑损失权重计划 $\lambda(e)$
对于 $e = 1$ **到** N_{epochs} **执行**
  $\lambda \leftarrow \lambda(e)$
  对于批次 $B \subset \mathscr{D}$ **执行**
    采样真实数据 $x_R \sim B$，潜在噪声 $z \sim \mathcal{N}(0, I)$
    生成生成数据 $x_G \leftarrow G(z)$
    ▷ — 判别器更新 —
    opt_D.zero_grad()
    $L_D^{\text{real}} \leftarrow \ell_{\text{BCE}}(\text{判别器}(x_R), y_{\text{real}})$
    $L_D^{\text{fake}} \leftarrow \ell_{\text{BCE}}(D(\text{分离梯度}(x_G)), y_{\text{fake}})$
    $L_D \leftarrow L_D^{\text{real}} + L_D^{\text{fake}}$
    反向传播 L_D，opt_D.step()
    ▷— 逻辑约束评估 —
    约束结果 $\leftarrow \mathscr{C}$.评估(x_G, R)
    $(S^{\text{logic}}, L_{\text{constraint}}) \leftarrow \mathscr{C}$.聚合(约束结果)
    $\mathscr{C}$.更新_权重_或_阶段（约束结果）
    ▷ — 生成器更新 —
    opt_G.清零_梯度()
    如果 $\mathscr{C}$ 具有可学习参数**则**
      $\text{opt}_{\mathscr{C}}$.清零_梯度()
    结束如果
    $L_G^{\text{adv}} \leftarrow \ell_{\text{BCE}}$（判别器（$x_G$），$y_{\text{real}}$）
    $L_{\text{aux}} \leftarrow \mathscr{A}(x_G)$ ▷可选，若定义了 $\mathscr{A}$
    $L_G \leftarrow \alpha L_G^{\text{adv}} + \lambda L_{\text{constraint}} + \beta L_{\text{aux}}$
    反向传播 L_G，opt_G.step()
    如果 $\mathscr{C}$ 具有可学习参数**则**
      $\text{opt}_{\mathscr{C}}$.step()
    结束 if
    日志指标：L_G, L_G^{adv}, L_D, S^{logic}, λ, $L_{\text{constraint}}$, 判别器准确率
  结束 for
结束 for
返回 G^* ，D^*

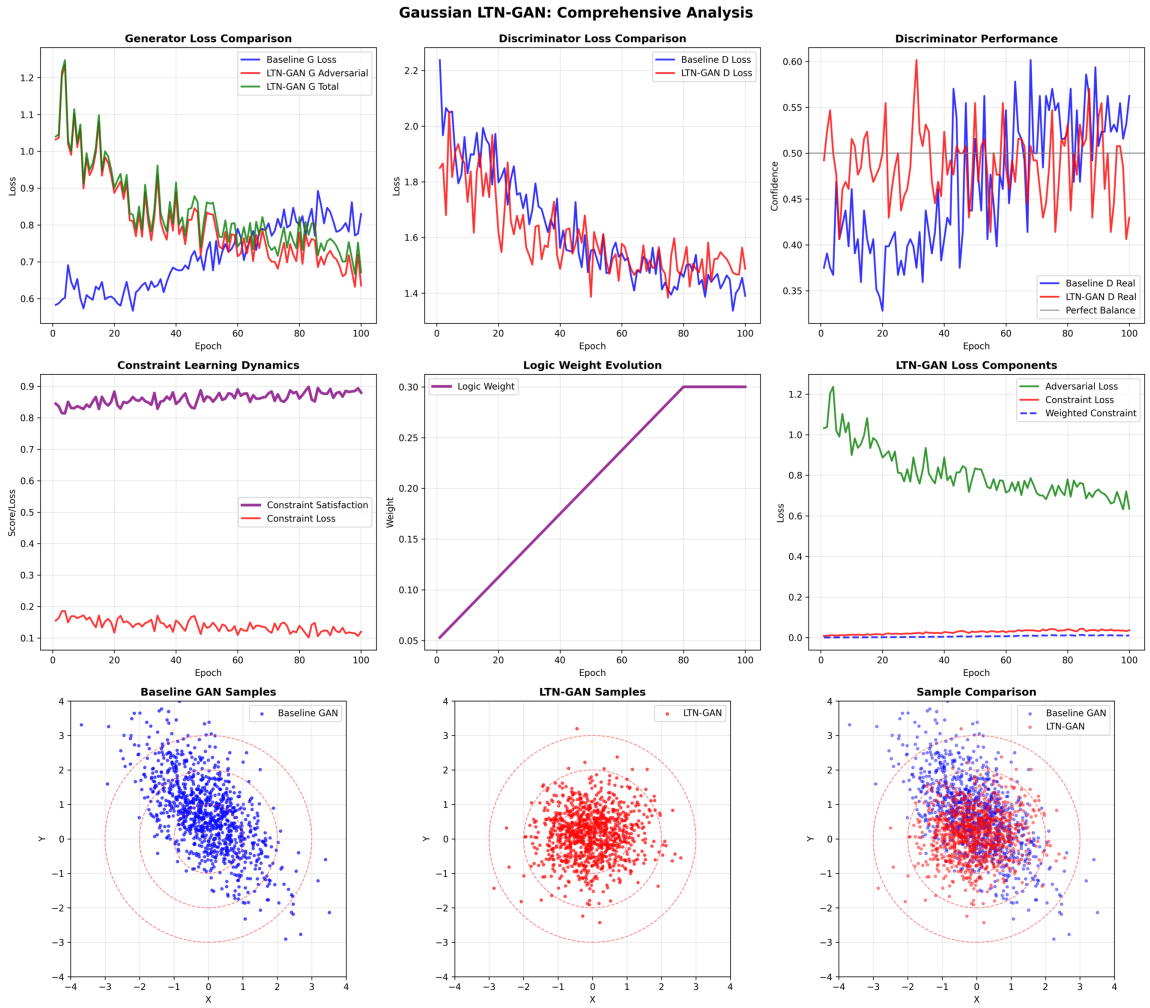
[1]损失定义。令 ℓ_{BCE} 表示二元交叉熵损失。判别器损失：$\mathrm{L}_D^{\text{real}} = \mathbb{E}_{x_R}[\ell_{\text{BCE}}(\mathrm{D}(\mathrm{x}_R), \mathrm{y}_{\text{real}})]$, $\mathrm{L}_D^{\text{fake}} = \mathbb{E}_z[\ell_{\text{BCE}}(\mathrm{D}(\mathrm{G}(\mathrm{z})), \mathrm{y}_{\text{fake}})]$,，总损失为 $\mathrm{L}_D = \mathrm{L}_D^{\text{real}} + \mathrm{L}_D^{\text{fake}}$。生成器损失包括：$\mathrm{L}_G^{\text{adv}} = \mathbb{E}_z[\ell_{\text{BCE}}(\mathrm{D}(\mathrm{G}(\mathrm{z})), \mathrm{y}_{\text{real}})]$, $\mathrm{L}_{\text{constraint}} = 1 - \mathrm{S}^{\text{logic}}$（或自适应加权和），以及可选的 L_{aux}，用于分类、模板匹配或其他任务。生成器总损失为 $\mathrm{L}_G = \alpha \mathrm{L}_G^{\text{adv}} + \lambda(\mathrm{e})_{\text{constraint}} + \beta \mathrm{L}_{\text{aux}}$，其中 α、β 和 $\lambda(\mathrm{e})$ 用于调节各损失分量。逻辑权重调度。 $\lambda(\mathrm{e})$ 在前 K 个训练轮次内线性递增：

$$\lambda(e) = \begin{cases} \lambda_{\text{start}} + \dfrac{e-1}{K-1}(\lambda_{\text{end}} - \lambda_{\text{start}}), & e \le K \\ \lambda_{\text{end}}, & \text{otherwise} \end{cases}$$

[2]**数据集说明：** 高斯和环使用线性 λ 渐变。网格同时记录 $L_{\text{constraint}}$ 和 $\lambda L_{\text{constraint}}$。环采用分层约束和自适应约束。MNIST 包含用于分类和形状模板的辅助损失 L_{aux}。不使用早停。所有实验均将 L_D 作为普通求和计算（不进行 12 倍缩放）。

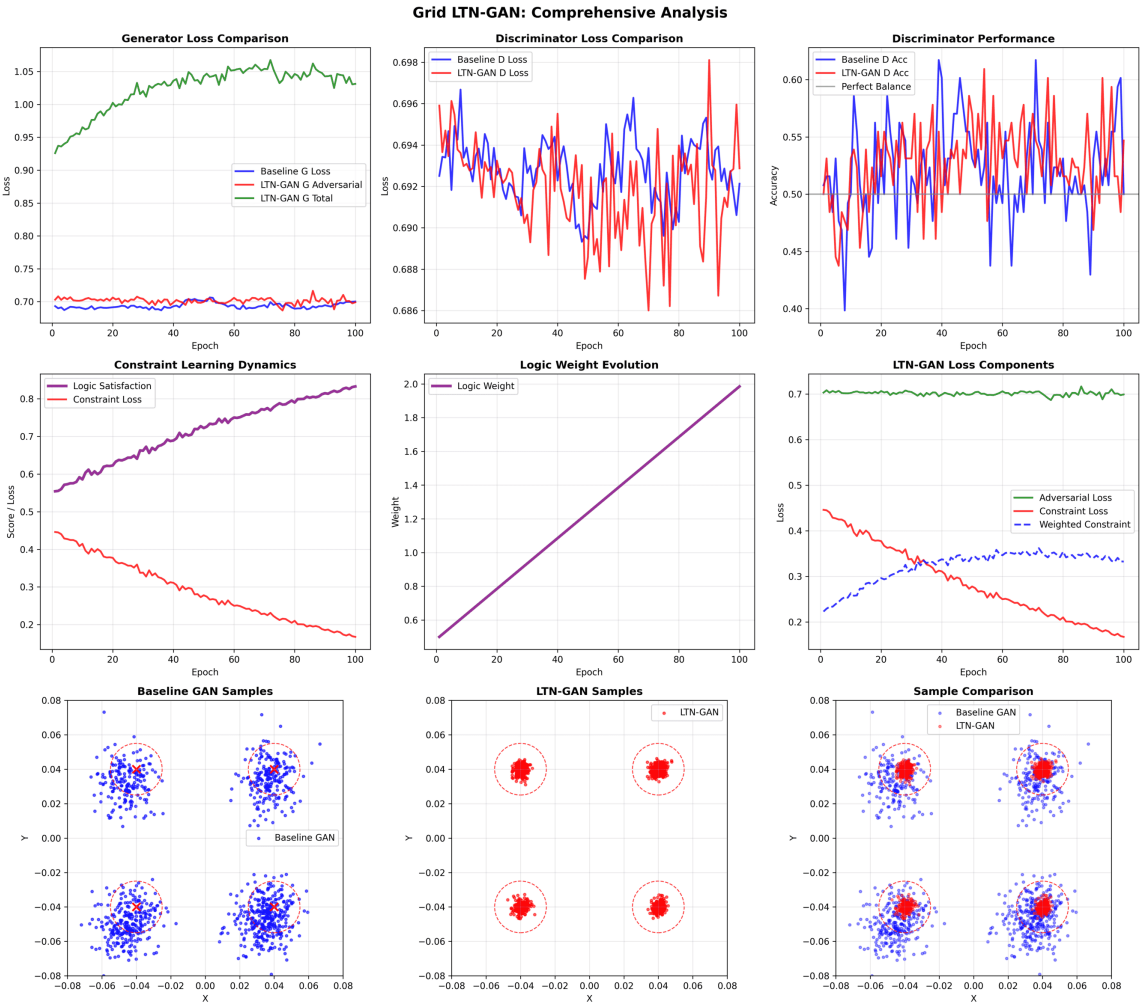
B 实验结果与可视化

B.1 高斯数据集



对于高斯实验，两个谓词——范围内(x)和高斯形状(x)——均实现为固定的数学函数，而非通过神经网络学习得到，因此无需额外的学习开销即可进行符号监督。生成器是一个全连接多层感知机，采用 LeakyReLU 激活函数且不使用输出激活函数，因此能够在整个 $\mathbb{R}^2$ 平面上生成样本。为稳定训练，采用标签平滑（真实样本标记为 0.9，伪样本标记为 0.1），逻辑损失权重$\lambda(e)$则在前 80 个训练轮次中从 0.05 线性增至 0.3。公式满足度通过计算一个批次中谓词合取的平均真值度得出。由于所有约束都是几何约束且有明确的定义，高斯任务可作为一个受控测试平台，用于隔离符号逻辑对 GAN 训练的影响，而不涉及谓词学习。

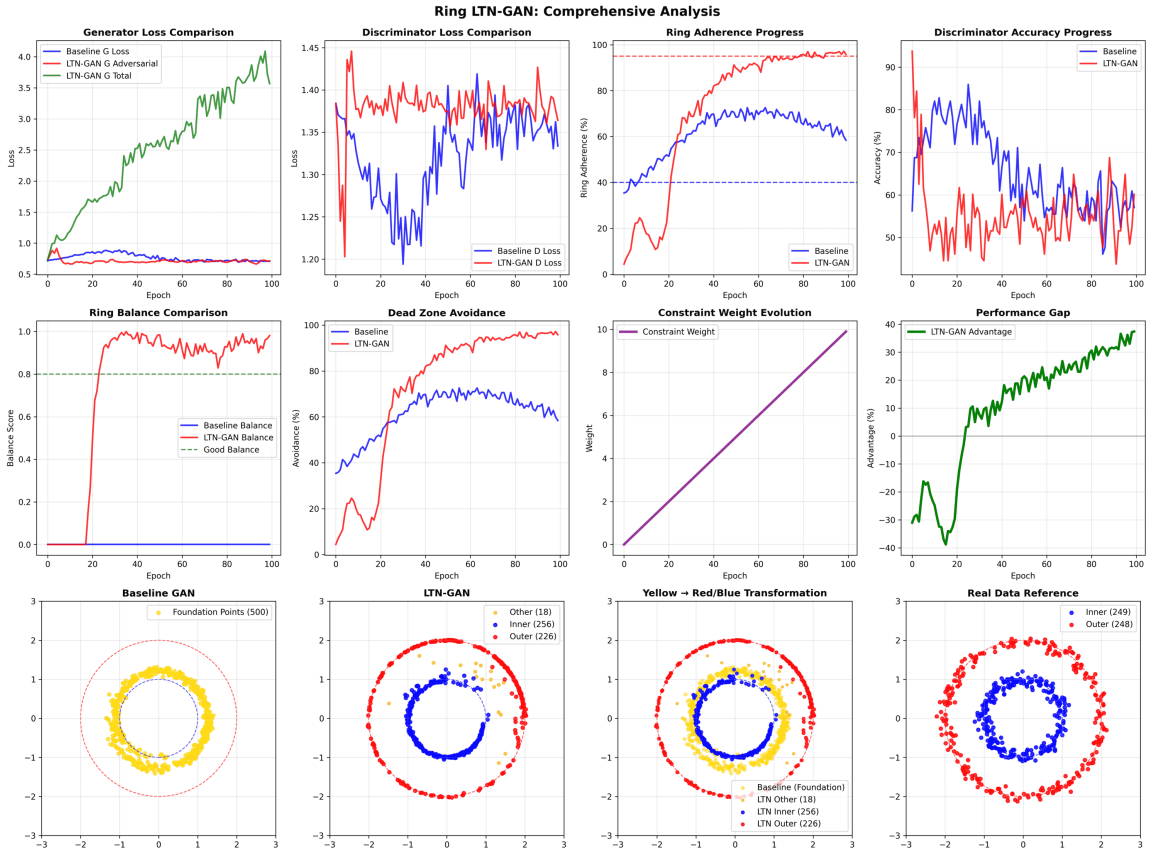
B.2 网格数据集



在网格实验中，生成器是一个采用 LeakyReLU 激活函数且不使用输出激活函数的全连接多层感知机，因此能够生成不受约束的二维输出。类别选择网络将每个潜向量分配到四个网格单元之一，并在该网格单元的中心附近施加扰动，从而得到最终坐标。这种归纳偏置有助于初步形成结构。

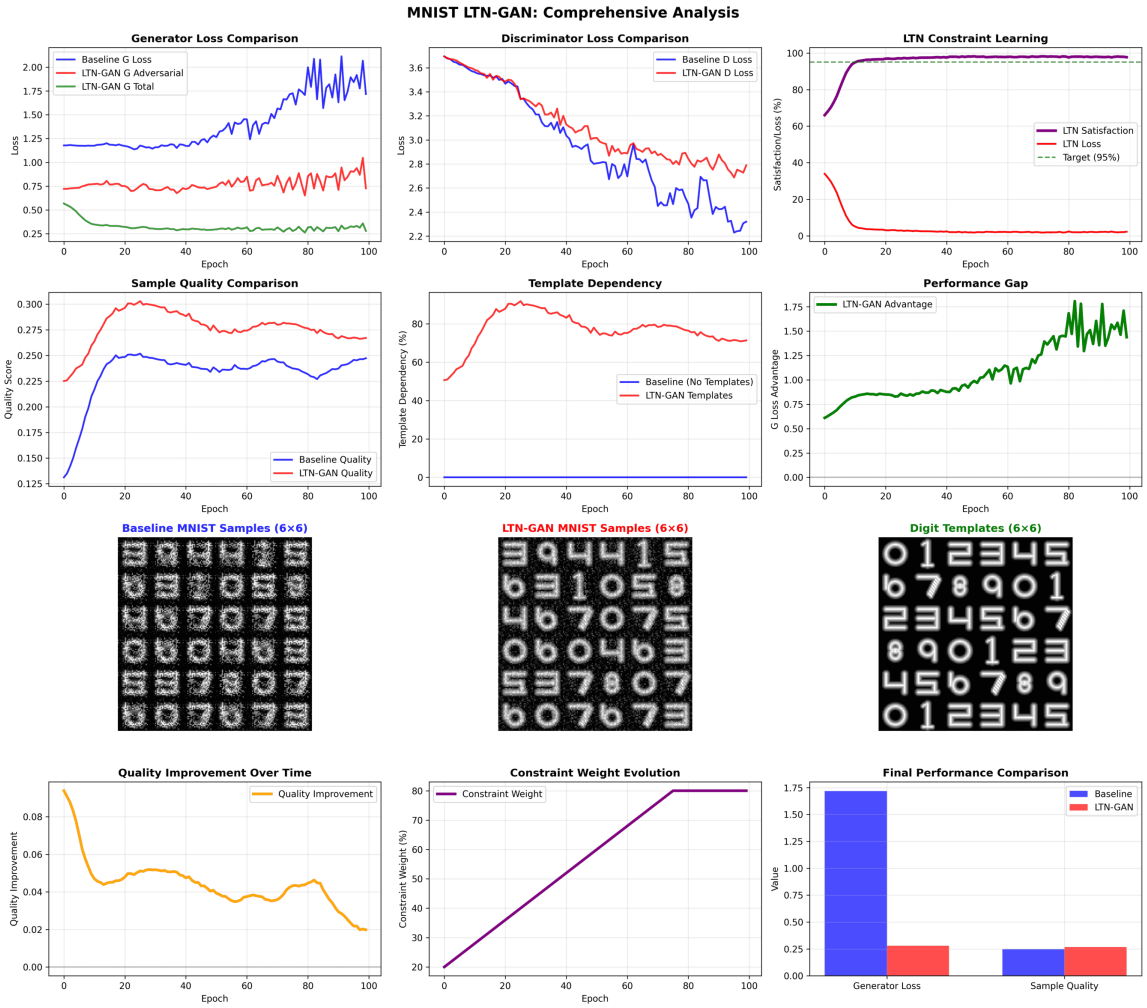
LTN 谓词包括一个共享的 OnGrid(x) 网络，用于评估与任意网格中心的接近程度；以及四个 InCell$_i$(x) 网络，用于检测单元特定归属。这些网络通过基于距离的特征进行联合训练。通过对 OnGrid(x) 应用全称约束，并对每个 InCell$_i$(x) 应用存在约束，实施逻辑监督，从而确保对齐和覆盖率。逻辑损失权重 $\lambda(e)$ 在 100 个训练轮次内从 0.5 线性增加至 2.0。真实样本是对网格中心进行合成扰动得到的（噪声 $\sigma = 0.008$），并使用标签平滑（0.9/0.1）来稳定训练。

B.3 环形数据集



在环形实验中，数据以极坐标采样后转换为笛卡尔空间，从而能够控制径向分布。硬编码的几何先验引导生成器倾向于生成对应内环和外环的环形流形，而两者之间的死区则不进行采样。生成器采用多层感知机，使用 LeakyReLU 激活函数和 Tanh 输出，以约束输出空间。与高斯设置不同，大多数谓词（包括 InnerRing、OuterRing、DeadZone 和 NearCenter）均由可微神经分类器实现，并通过联合训练提供灵活的模糊隶属度。动态规则重新加权策略根据每个批次的满足率更新子句权重，使模型能够自适应地优先处理满足程度较低的约束。逻辑损失调度从 $\lambda = 0.01$ 开始，逐渐增加至 $\lambda = 0.2$；预热轮次结束后收紧谓词容差（例如环半径区间），以逐步提高精确性。

B.4 MNIST 数据集



在 MNIST 实验中，生成器是一个采用转置卷积和批归一化的卷积神经网络，经过训练后能够合成 28×28 灰度数字图像。类别谓词 $\text{IsDigit}_k(x)$ for $k \in \{0, \ldots, 9\}$ 由经过真实 MNIST 数字训练的冻结多层感知机实现，从而对数字身份进行符号监督。此外，四个结构质量谓词——$\text{ValidPixels}(x)$、$\text{IsConnected}(x)$、$\text{IsComplete}(x)$ 和 $\text{HasProperIntensity}(x)$——采用简单的图像处理启发式方法和浅层神经网络定义。通过手工构建的二值数字模板数据集，将几何先验融入训练初期，在符号约束生效之前引导样本的初始形成。逻辑损失权重 $\lambda(e)$ 在前 40 个训练轮次中从 0.1 逐渐增加到 0.5。所有谓词均在生成批次上进行评估，其逻辑满足度分数通过加权模糊逻辑聚合进行合并。该设置使 LTN-GAN 能够生成结构有效且类别一致的图像，同时保持数字间的多样性。MNIST 谓词网络是端到端训练的浅层卷积神经网络，用于捕捉居中性和形状等数字属性。训练初期会融入模板先验，并逐步淡出，从而将监督转向符号规则。

B.5 特定领域指标

数据集	指标	描述／计算
高斯	高斯符合度	综合评分 基于 严格的检验： 正态性（安德森-达林），双变量形状（马氏距离），矩（均值 ≈ 0，标准差 ≈ 1）、径向分布（瑞利拟合度）、独立性（皮尔逊相关系数）和四分位数。若逻辑满足度较低，则予以惩罚。
	统计质量	目标均值和标准差总偏差的倒数： $1/(1+\|\mu\|+\|\sigma-1\|)$
	综合质量	遵循度和统计质量的算术平均值。
	均值误差	样本均值与 0 之差的 L2 范数。
	标准差误差	样本标准差与 1 之差的 L2 范数。
	逻辑满足度	LTN约束的平均真值度（范围和高斯性）。
Grid	网格簇	属于有效网格簇中心的样本比例（例如，对中心匹配进行K均值+阈值处理）。
	覆盖率	至少被覆盖一次的网格位置占比。
	质量	用于评估落入以下范围的点数的精确率类评分预期簇的有效容差内。
	总体性能	覆盖率和质量的加权平均。
	目标内	有效网格区域内的点数（四舍五入为整数或空间掩码）。
	逻辑满足度	结构化 LTN 约束的平均满足度。
Ring	环形遵循度	位于环形区域附近的样本比例，通过径向距离评分和圆形对称性进行测量。
	内侧／外侧计数	被分类为位于环形边界内侧或外侧的样本数量。界。
	平衡分数	内部与外部之间对称性的归一化度量区域（例如， $1-\|n_{in}-n_{out}\|/(n_{in}+n_{out})$）.
	死区规避	未落入不希望出现的中心死区的样本比例区。
	逻辑满足度	基于 LTN逻辑对空间环的均值约束满足度。
MNIST	质量	标准生成对抗网络图像保真度指标（可包括 Inception-类代理指标或视觉多样性）。
	覆盖率	生成的目标数字比例（使用分类器或模板匹配）。
	数字识别	训练好的分类器对生成样本的准确率。
	模板依赖	用于衡量与预定义模板块。
	逻辑满足度	对数字身份和位置约束的满足程度，这些约束定义于LTN逻辑中。

表6：特定数据集消融研究中使用的评估指标概览。每项指标都旨在兼顾生成质量和逻辑结构保真度。