TOPOBALANCE: DIVERSE LEGAL LAYOUT GENERATION VIA SUPPORT EXPANSION AND DISTRIBUTION BALANCING

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating large, diverse, and distributionally balanced libraries of legal layout patterns is important for design-for-manufacturability applications, yet remains challenging with limited legal training data. The combinatorial topology space leads to sparse coverage of complexity labels, while topology filtering and physical-feasibility constraints introduce acceptance-induced distribution shifts that concentrate the final legal library on a limited subset of labels. To address these challenges, we propose ***TopoBalance***, a distribution-aware layout-pattern generation framework that integrates training-stage support expansion and generation-stage distribution balancing. TopoBalance employs *Topology-Valid RegSLERP* (TV-RegSLERP) for training-stage support expansion and *Distribution-Balanced Adaptive Sampling and Generation* (DBAS-Gen) for closed-loop distribution balancing during generation. Experiments on ICCAD 2016 show that TopoBalance substantially improves complexity-label diversity and distributional balance while maintaining a high legality. It markedly reduces concentration on dominant complexity regions, yielding a broader and more balanced legal-layout library. The code and datasets are available at `https://anonymous.4open.science/r/RegSLERP-Hybrid-5BB1/`.

1 INTRODUCTION

Generative models have emerged as powerful tools for exploring combinatorial design spaces (Goodfellow et al., 2014; Ho et al., 2020; Rombach et al., 2022). However, their effectiveness in structured domains is often constrained by sparse valid observations and distribution mismatch among training data, generated samples, and feasible solutions (Ye et al., 2026; Zhang et al., 2026). Integrated circuit (IC) layout generation exemplifies this challenge. As semiconductor manufacturing advances toward nanometer and sub-nanometer nodes, complex design rules and manufacturing uncertainties strengthen the coupling between design and manufacturing, highlighting the importance of design technology co-optimization (DTCO) (Han & Hao, 2021). Design-for-manufacturability (DFM) applications (Yang et al., 2021), including lithography modeling (Feng et al., 2023), optical proximity correction (Kareem et al., 2020; Poonawala & Milanfar, 2007), and hotspot detection (Yang et al., 2017; 2019), require legal layout libraries with broad and balanced coverage of the layout space.

Despite advances in learning-based layout generation, constructing such libraries remains challenging due to the combinatorial topology space and limited legal training samples. Existing GAN-based (Goodfellow et al., 2014; Zhang et al., 2020), convolutional autoencoder-based (Yang et al., 2022; Masci et al., 2011), and diffusion-based methods (Ho et al., 2020; Austin et al., 2021; Chen et al., 2023; Rombach et al., 2022; Wang et al., 2023; 2025; Zhou et al., 2025) improve generation efficiency and diversity. However, more generated samples do not necessarily yield balanced legal libraries. Due to feasibility constraints, different complexity labels may have unequal acceptance probabilities, causing the final library to concentrate on a small subset of labels (Um et al., 2024).

Challenge 1: Limited support coverage in the training distribution. A training set of legal layout samples provides only sparse observations of the larger topology space. Under the topology-complexity representation used in prior layout-generation studies (Yang et al., 2022; Wang et al., 2023; 2025; Zhou et al., 2025), such sparsity appears as incomplete coverage of complexity la-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

bels and insufficient variation among layout samples within observed labels. Latent interpolation provides a natural approach to expand training coverage, but unconstrained interpolation may leave high-density regions of the latent manifold (Arvanitidis et al., 2018; Chung et al., 2025) and generate invalid topologies. Therefore, the first challenge is to enlarge the support of the training distribution across complexity labels and within observed labels while preserving topology validity.

Challenge 2: Acceptance-induced distribution shift during generation. Even with an expressive generator, the complexity-label distribution of the final legal library may deviate from that of the raw generated samples after topology filtering and physical-feasibility constraints are applied. Layout samples associated with complexity labels having higher generation and legalization acceptance probabilities accumulate faster, resulting in concentration of the complexity-label distribution and reduced representation of underrepresented labels. According to our statistical analysis in Section 5.2 (Table 1), existing methods exhibit substantial complexity-label distribution imbalance, characterized by limited distribution uniformity, large deviations from the uniform distribution, and strong concentration in a small number of complexity labels, with the Top-10 labels alone accounting for 44.21–72.75% of the generated legal layouts. Therefore, the second challenge is to establish a feedback-driven generation strategy that adaptively compensates for this acceptance-induced distribution shift while maintaining high legality.

To address these challenges, we propose ***TopoBalance***, a distribution-aware generative framework that performs closed-loop complexity-label distribution control throughout the generation lifecycle. During training, TopoBalance introduces *Topology-Valid RegSLERP* (TV-RegSLERP), which expands the support of the training distribution through controlled latent interpolation across complexity labels. Only topology-valid decoded layout samples are retained, enabling the diffusion model to learn from an augmented training set with broader complexity-label coverage and a more balanced complexity-label distribution. During generation, TopoBalance introduces *Distribution-Balanced Adaptive Sampling and Generation* (DBAS-Gen), which forms a closed-loop process of complexity-label sampling, layout generation and legalization, and feedback update. By continuously incorporating the evolving legal-layout distribution and historical feasibility outcomes, DBAS-Gen adaptively redirects generation toward underrepresented complexity regions while preserving exploration of the layout space.

Our main contributions are summarized as follows:

- We propose TopoBalance, a distribution-aware framework for legal layout-pattern generation that broadens complexity-label coverage and balances the final distribution.
- We introduce Topology-Valid RegSLERP (TV-RegSLERP), a training-time latent distribution expansion method that performs controlled cross-label interpolation with topology-valid sample selection to broaden complexity-label coverage and balance the diffusion training distribution.
- We introduce *Distribution-Balanced Adaptive Sampling and Generation* (DBAS-Gen), a closed-loop strategy that adaptively balances complexity-label distributions through sampling, generation, legalization, and feedback.
- Experiments on ICCAD 2016 demonstrate the effectiveness of TopoBalance in generating diverse, balanced legal-layout libraries efficiently.

2 Related Work

Traditional Layout Pattern Generation. Traditional layout generation relies on explicit geometric representations and manually designed transformations, perturbations, or combinations of existing patterns (Reddy et al., 2018). The Squish Pattern representation separates discrete topology from geometric dimensions, enabling compact and physically interpretable layout manipulation (Gennari & Lai, 2014; Yang et al., 2019). However, the exploration range of such methods is largely limited by predefined operators and available templates.

Learning-Based Layout Pattern Generation. Learning-based methods improve automation by learning layout distributions directly from data. DeePattern (Yang et al., 2022) uses a convolutional autoencoder to learn continuous layout representations. DiffPattern (Wang et al., 2023) introduces discrete diffusion for binary layout generation, while DiffPattern-Flex (Wang et al., 2025) further

improves diversity through data augmentation and flexible generation. PatternPaint (Zhou et al., 2025) adopts diffusion-based inpainting for layout generation from limited realistic samples.

Despite these advances, limited legal training data still provide sparse coverage of the legal topology space, while topology filtering and physical legalization can concentrate the final legal library on a subset of complexity labels. TopoBalance addresses these issues through training-time topology-valid cross-complexity augmentation and generation-time distribution balancing.

3 Problem Formulation

We present the essential formulation here and defer the full problem statement to Appendix A.

Layout Representation and Canonical Topology. A layout pattern is represented as $\mathcal{L} = (B, d^x, d^y)$, where $B \in \{0,1\}^{c_y \times c_x}$ denotes the canonical binary topology matrix, and $d^x \in \mathbb{R}_+^{c_x}$ and $d^y \in \mathbb{R}_+^{c_y}$ specify the physical dimensions of the column and row intervals, respectively. The canonical topology captures the discrete layout structure independently of its continuous geometry.

Complexity Label and Complexity-Label Distribution. Following prior work (Yang et al., 2022; Wang et al., 2023; 2025; Zhou et al., 2025), each canonical topology is assigned a complexity label

$$g = \kappa(B) = (c_x, c_y), \tag{1}$$

where c_x and c_y are the numbers of canonical column and row intervals, respectively.

For a layout library $\mathcal{D}$, its empirical complexity-label distribution and support are defined as

$$p_{\mathcal{D}}(g) = \frac{1}{|\mathcal{D}|} \sum_{\mathcal{L}_n \in \mathcal{D}} \mathbf{1}[\kappa(B_n) = g], \qquad \mathcal{G}(\mathcal{D}) = \{g \mid p_{\mathcal{D}}(g) > 0\}. \tag{2}$$

The corresponding complexity-label distribution entropy (Shannon, 1948) is

$$H_1(\mathcal{D}) = -\sum_{g \in \mathcal{G}(\mathcal{D})} p_{\mathcal{D}}(g) \log_2 p_{\mathcal{D}}(g). \tag{3}$$

A larger H_1 indicates greater complexity-label distributional diversity, jointly reflecting label coverage and distribution evenness.

Legal Layout Library Generation. Let $\Omega_{\mathcal{R}}$ denote the legal layout space under design-rule set $\mathcal{R}$. Given a limited legal training set $\mathcal{D}_{\text{train}} \subseteq \Omega_{\mathcal{R}}$, let $\mathcal{D}_{\text{out}}(\pi)$ denote the final legal-layout library produced by generation policy π. Our goal is to obtain

$$\mathcal{D}_{\text{out}}(\pi) = \{\mathcal{L}_m^{\text{out}}\}_{m=1}^{M} \subseteq \Omega_{\mathcal{R}} \tag{4}$$

with high complexity-label distributional diversity while maintaining high end-to-end legality.

Let $C(\pi)$ denote the total number of generated candidates required to obtain M legal layouts. The end-to-end legality rate is

$$R_{\text{legal}}(\pi) = \frac{M}{C(\pi)}. \tag{5}$$

We formulate legal layout-library generation as

$$\max_{\pi} \left(H_1(\mathcal{D}_{\text{out}}(\pi)), R_{\text{legal}}(\pi)\right), \qquad \text{s.t. } |\mathcal{D}_{\text{out}}(\pi)| = M,\ \mathcal{D}_{\text{out}}(\pi) \subseteq \Omega_{\mathcal{R}}. \tag{6}$$

4 TopoBalance Framework

4.1 Overview

As illustrated in Figure 1, the proposed **TopoBalance** contains two key components operating at complementary stages:

- *Topology-Valid RegSLERP* (TV-RegSLERP) expands the training-time complexity-label support through controlled latent interpolation across nearby complexity labels and retains only topology-valid decoded candidates.
- *Distribution-Balanced Adaptive Sampling and Generation* (DBAS-Gen) performs generation-time distribution balancing through a closed loop of mode-aware complexity-label sampling, layout generation and legalization, and feedback update.

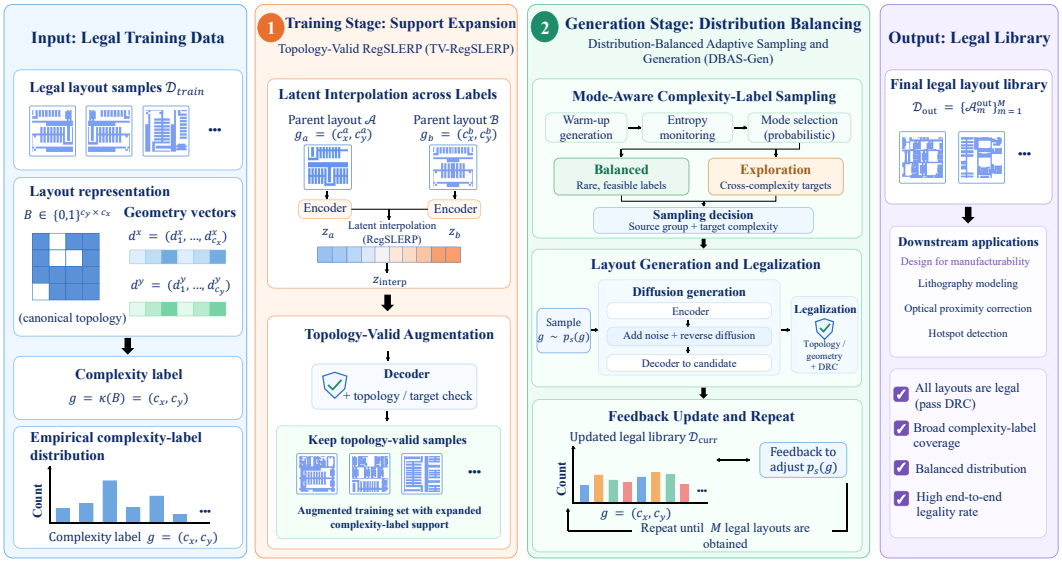


Figure 1: Overview of TopoBalance. TV-RegSLERP expands training-time structural coverage through topology-valid cross-complexity augmentation, while DBAS-Gen balances the evolving legal-layout distribution through closed-loop sampling, generation, and feedback.

4.2 Topology-Valid RegSLERP

To expand the sparse complexity-label support of the training set, we propose *Topology-Valid RegSLERP* (TV-RegSLERP), which performs controlled latent interpolation across nearby complexity labels and retains only topology-valid, target-consistent decoded samples.

4.2.1 Latent Interpolation across Labels

Let $\mathcal{G}_{\text{train}} = \mathcal{G}(\mathcal{D}_{\text{train}})$ denote the set of complexity labels observed in the original training set, where each label is $g = (c_x, c_y)$. Rather than forming arbitrary interpolation pairs, TV-RegSLERP first enumerates all candidate parent-label pairs

$$\mathcal{P} = \left\{ (g^a, g^b) \in \mathcal{G}_{\text{train}} \times \mathcal{G}_{\text{train}} \;\middle|\; \delta_1^{\min} \le \|g^a - g^b\|_1,\ \|g^a - g^b\|_\infty \le \delta_\infty^{\max} \right\}, \tag{7}$$

where $\delta_1^{\min}$ prevents the two parent labels from being overly similar, while $\delta_\infty^{\max}$ avoids excessively large cross-complexity transitions.

Given the predefined interpolation-position set $\Lambda = \{0.15, 0.20, \ldots, 0.85\}$, corresponding to $\lambda \in [0.15, 0.85]$ with a step size of 0.05, every tuple (g^a, g^b, λ) with $(g^a, g^b) \in \mathcal{P}$ and $\lambda \in \Lambda$ defines an expected target complexity

$$g^* = \text{round}\left[(1-\lambda)g^a + \lambda g^b\right]. \tag{8}$$

TV-RegSLERP then constructs a target-to-parent mapping

$$\mathcal{T}(g^*) = \left\{ (g^a, g^b, \lambda) \;\middle|\; (g^a, g^b) \in \mathcal{P},\ \lambda \in \Lambda,\ g^* = \text{round}\left[(1-\lambda)g^a + \lambda g^b\right] \right\}, \tag{9}$$

which collects all parent-label pairs and interpolation positions capable of reaching the same target complexity label. This mapping enables TV-RegSLERP to allocate augmentation attempts directly toward missing or underrepresented regions of the complexity-label space.

TV-RegSLERP prioritizes target labels g^* that are absent or underrepresented in the current training set. Once a target is selected, a compatible tuple $(g^a, g^b, \lambda) \in \mathcal{T}(g^*)$ is sampled, and one layout sample is drawn from each parent label. Their latent representations z^a and z^b are obtained using the pretrained VAE (Rombach et al., 2022) encoder, and RegSLERP is applied as

$$z_\lambda = \text{RegSLERP}(z^a, z^b; \lambda). \tag{10}$$

4.2.2 TOPOLOGY-VALID AUGMENTATION

The interpolated latent z_λ is then decoded back into the layout space to generate a candidate. The candidate is accepted into $\mathcal{D}_{\text{TV-RegSLERP}}$ only if its decoded topology is valid and its actual complexity label matches the intended target g^*.

This procedure is repeated under a predefined augmentation budget. After each attempt, the target priority is updated according to its current frequency and interpolation history, progressively reallocating augmentation effort toward missing or low-frequency labels that remain practically reachable. The accepted candidates collected in $\mathcal{D}_{\text{TV-RegSLERP}}$ are finally merged with the original training set:

$$\mathcal{D}^{+}_{\text{train}} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{TV-RegSLERP}}. \tag{11}$$

The resulting training set provides broader complexity-label support for subsequent diffusion training. The detailed settings for pair construction, target prioritization, RegSLERP interpolation, and topology screening are provided in Appendix B.1.

4.3 DISTRIBUTION-BALANCED ADAPTIVE SAMPLING AND GENERATION

To mitigate generation-time distribution imbalance, we propose *Distribution-Balanced Adaptive Sampling and Generation* (DBAS-Gen), which forms a closed-loop generation process using feedback from the evolving legal-layout library. DBAS-Gen consists of three stages: mode-aware complexity-label sampling, layout generation and legalization, and feedback update.

4.3.1 MODE-AWARE COMPLEXITY-LABEL SAMPLING

Because the legal-layout library is empty at initialization, DBAS-Gen first performs warm-up generation to obtain an initial library $\mathcal{D}_0$ containing N_0 accepted layouts. This initial library provides the complexity-label frequency and generation-feasibility estimates required for subsequent adaptive sampling.

At generation round t, let $\mathcal{D}_t$ denote the current legal-layout library, and let $f_t(g)$ denote the frequency of complexity label $g = (c_x, c_y)$ in $\mathcal{D}_t$. We define

$$p_t(g) = \frac{f_t(g)}{|\mathcal{D}_t|}, \qquad H_1(\mathcal{D}_t) = -\sum_{g \in \mathcal{G}(\mathcal{D}_t)} p_t(g) \log_2 p_t(g). \tag{12}$$

DBAS-Gen monitors the entropy gain over a generation window of length W:

$$\Delta H_t = H_1(\mathcal{D}_t) - H_1(\mathcal{D}_{t-W}), \qquad t \geq W. \tag{13}$$

A distributional plateau is detected when the entropy gain falls below ϵ_H:

$$\text{plateau}_t = \begin{cases} 1, & \Delta H_t < \epsilon_H, \\ 0, & \Delta H_t \geq \epsilon_H. \end{cases} \tag{14}$$

Based on plateau_t, DBAS-Gen adjusts the probabilities of the two sampling modes:

(i) *Balanced mode*: is selected with probability $p_{\text{balanced}} > 0.5$ when $\text{plateau}_t = 0$, and accordingly Exploration mode with probability $1 - p_{\text{balanced}}$.

(ii) *Exploration mode*: is selected with probability $p_{\text{exploration}} > 0.5$ when $\text{plateau}_t = 1$, and Balanced mode with probability $1 - p_{\text{exploration}}$.

Thus, Balanced mode dominates while the distribution continues to improve, whereas Exploration mode becomes dominant after saturation to reach insufficiently covered regions. This probabilistic transition preserves both modes in either state and avoids an abrupt hard switch.

Once the sampling mode is determined, DBAS-Gen selects a target complexity label g^* according to the corresponding mode-specific strategy, with the goal of generating a legal layout whose realized complexity matches the selected target.

Complexity-label sampling in Balanced mode. Balanced mode prioritizes underrepresented and feasible complexity labels in the current legal-layout library. Its sampling distribution is defined as

$$\begin{aligned} \mathcal{G}_t^+ &= \{g \mid f_t(g) > 0\}, \qquad \widetilde{f}_t = \frac{|\mathcal{D}_t|}{|\mathcal{G}_t^+|}, \\ s_t(g) &= \mathrm{clip}\left[\left(\frac{\widetilde{f}_t + \rho}{f_t(g) + \rho}\right)^{\gamma}, \frac{1}{w_{\max}}, w_{\max}\right], \\ p_{\mathrm{bal},t}(g) &= \frac{s_t(g) G_{\mathrm{leg},t}(g)}{\sum_{h \in \mathcal{G}_t^+} s_t(h) G_{\mathrm{leg},t}(h)}, \qquad g \in \mathcal{G}_t^+. \end{aligned} \tag{15}$$

Here, $\widetilde{f}_t$ is the mean frequency over the observed support; $\rho > 0$ stabilizes the frequency ratio, γ controls the balancing strength, and $w_{\max}$ bounds the scarcity weight $s_t(g)$. $G_{\mathrm{leg},t}(g)$ denotes the historical generation-and-legalization feasibility of label g.

Complexity-label sampling in Exploration mode. Exploration mode targets missing or weakly represented complexity labels. From the labels observed in the current legal-layout library $\mathcal{G}(\mathcal{D}_t)$, DBAS-Gen constructs all reachable targets g^* and their target-to-action mappings $\mathcal{T}_t^{\exp}(g^*)$, following the same complexity-space construction as g^* and $\mathcal{T}(g^*)$ in TV-RegSLERP. DBAS-Gen then ranks these reachable targets by current representation, historical exploration frequency, and generation feasibility, and samples one as the target complexity g^* for the current round.

4.3.2 Layout Generation and Legalization

Once the target complexity g^* is determined, DBAS-Gen selects a source layout and performs diffusion-based generation toward the selected target. In this work, we adopt Stable Diffusion 2 Inpainting (Rombach et al., 2022) as the diffusion backbone.

Source layout selection. For Balanced mode, the target complexity g^* is already covered by the current legal-layout library, and a source layout $\mathcal{L}_{\mathrm{src}}$ is sampled from layouts with $g_{\mathrm{src}} = g^*$. For Exploration mode, DBAS-Gen samples an exploration action $a = (g_{\mathrm{src}}, g_{\mathrm{ref}}, \lambda)$ from $\mathcal{T}_t^{\exp}(g^*)$ and samples $\mathcal{L}_{\mathrm{src}}$ from layouts with g_{src}. Here, g_{ref} only defines the cross-complexity direction toward g^* and does not provide an additional source layout or latent representation. Therefore, both modes generate from a single $\mathcal{L}_{\mathrm{src}}$.

Diffusion-based layout generation. The selected source layout $\mathcal{L}_{\mathrm{src}}$ is encoded into the clean latent representation z_{src}, followed by reverse diffusion and pre-trained VAE decoding to produce candidate layouts. Note that g_{ref} is used only to define the cross-complexity direction and target g^*. It does not provide a second reference layout for generation. Therefore, DBAS-Gen initializes diffusion only from z_{src} rather than an interpolated latent. Latent interpolation is used only by TV-RegSLERP for training-set augmentation.

Finally, each generated candidate undergoes topology verification, geometric legalization with its discrete topology fixed, and an independent design-rule check (DRC). Candidates that fail any stage are rejected. Only candidates that pass all checks are accepted as legal layouts and added to the current legal-layout library.

4.3.3 Feedback Update and Repeat

After the t-th generation round, accepted legal layouts are added to the current library. Moreover, the updated complexity-label distribution and generation-feasibility statistics are then fed back into the next round of mode and target-label sampling, forming a closed loop of sampling, generation, and feedback.

Finally, the sampling, generation, and feedback process is repeated until M legal layouts are collected, yielding

$$\mathcal{D}_T = \mathcal{D}_{\mathrm{out}}(\pi) = \left\{\mathcal{L}_m^{\mathrm{out}}\right\}_{m=1}^{M} \subseteq \Omega_{\mathcal{R}}, \qquad |\mathcal{D}_T| = M. \tag{16}$$

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Dataset. We use the publicly available ICCAD 2016 dataset, which contains 8,001 legal single-layer metal layout patterns. Each preprocessed layout is represented by a fixed-size $16 \times 32 \times 32$ discrete topology tensor, with the horizontal and vertical geometric dimensions retained separately. All methods are evaluated under the same $2{,}048\ \mathrm{nm} \times 2{,}048\ \mathrm{nm}$ physical window.

Baselines. We compare TopoBalance with four learning-based layout-pattern generation methods: DeePattern (Yang et al., 2022), which is based on a convolutional autoencoder; DiffPattern (Wang et al., 2023), which uses discrete diffusion for topology generation; DiffPattern-Flex (Wang et al., 2025), which extends DiffPattern with flexible generation and data augmentation; and PatternPaint (Zhou et al., 2025), which employs diffusion-based inpainting for layout-pattern synthesis.

Evaluation Metrics. We evaluate the generated legal-layout libraries from four aspects: *(i) complexity coverage and diversity*, using #Labels, H_1, and H_2; *(ii) distributional balance*, using $H_{1,\mathrm{norm}}$, distance to the uniform distribution, and Top-10 concentration; *(iii) legality*, using the end-to-end legality rate; and *(iv) generation efficiency*, using #Candidates and #Extra. Detailed definitions are provided in Appendix C.1.

Implementation Details. *(i) Experimental environment:* All models are implemented in PyTorch (Paszke et al., 2019) 2.4.1 with CUDA 12.1 and evaluated on a single NVIDIA GeForce RTX 4090 GPU. *(ii) Training and generation settings:* All diffusion models use the same training configuration with mixed precision and AdamW (Loshchilov & Hutter, 2019), a learning rate of 10^{-5}, and weight decay of 10^{-2}. Generation uses 50-step single-source diffusion sampling. All methods follow the same physical design rules, including a minimum metal width and internal spacing of $45\ \mathrm{nm}$ and a minimum interval size of $1\ \mathrm{nm}$. Unless otherwise specified, each method generates a library of 20,000 legal layouts. *(iii) TopoBalance hyperparameters:* For TV-RegSLERP, we set $\delta_1^{\min} = 2$ and $\delta_\infty^{\max} = 8$. For DBAS-Gen, we set $W = 2000$ and $\epsilon_H = 0.03$. Balanced mode is selected with probability $p_{\mathrm{balanced}} = 0.8$ before plateau detection, while Exploration mode is selected with probability $p_{\mathrm{exploration}} = 0.6$ after a plateau is detected.

5.2 RQ1: HOW DOES TOPOBALANCE COMPARE WITH EXISTING METHODS?

Motivation. We compare TopoBalance with existing methods in terms of complexity coverage and diversity, distributional balance, legality, and generation efficiency when constructing libraries of 20,000 legal layouts.

Table 1: Complexity-label distributional balance of the final 20,000 legal-layout library.

Method	$H_{1,\mathrm{norm}}$ ↑	Distance to Uniform↓	Top-10↓
DeePattern	0.7025	0.6619	72.75%
PatternPaint	0.6739	0.7435	44.21%
DiffPattern	0.7034	0.7006	64.17%
DiffPattern-Flex	0.7546	0.6165	53.10%
TopoBalance	**0.9702**	**0.1539**	**15.61%**

Table 2: Complexity coverage and diversity of the final 20,000 legal-layout library.

Method	H_1 ↑	H_2 ↑	#Labels
DeePattern	4.47	11.51	83
PatternPaint	**6.65**	13.92	939
DiffPattern	4.74	5.74	107
DiffPattern-Flex	5.34	5.35	136
TopoBalance	6.39	**14.28**	96

Results. As shown in Table 1, TopoBalance achieves the strongest complexity-label distributional balance, with the highest $H_{1,\mathrm{norm}}$ of 0.9702, the smallest distance to the uniform distribution of 0.1539, and the lowest Top-10 concentration of 15.61%. In contrast, the Top-10 labels account for 44.21% to 72.75% of the legal layouts generated by the baselines, indicating substantially stronger concentration on dominant complexity labels.

Table 3: Legality and generation efficiency for the final 20,000 legal-layout library construction.

Method	Legality↑	#Candidates↓	#Extra↓
DeePattern	58.69%	34,080	14,080
PatternPaint	63.80%	31,346	11,346
DiffPattern	95.24%	21,000	1,000
DiffPattern-Flex	**99.60%**	**20,080**	**80**
TopoBalance	98.66%	20,271	271

Table 2 shows that PatternPaint covers more complexity labels, while TopoBalance achieves the highest H_2 and the second-highest H_1. However, #Labels reflects only support breadth rather than distribution quality. Together with Table 1, the results show that TopoBalance achieves substantially more balanced representation across the observed complexity space.

As shown in Table 3, TopoBalance requires 20,271 candidates to obtain 20,000 legal layouts, corresponding to a legality rate of 98.66%. DiffPattern-Flex is slightly more efficient, requiring 20,080 candidates with 99.60% legality, but exhibits weaker distributional balance. Overall, TopoBalance achieves a favorable combination of diversity, distributional balance, and generation efficiency.

5.3 RQ2: Which Design Choices Drive the Performance of TopoBalance?

Motivation. We evaluate TopoBalance through ablations of TV-RegSLERP and DBAS-Gen, the constrained parent-pair selection in TV-RegSLERP, and the Balanced and Exploration modes in DBAS-Gen, as summarized in Table 4.

Table 4: Ablation study of TopoBalance and its key components.

Ablation	Variant	$H_{1,\mathrm{norm}}\uparrow$	Top-10↓	$H_1\uparrow$	$H_2\uparrow$	#Labels	Legality↑
	Full	**0.97**	**15.61%**	**6.39**	**14.28**	96	98.66%
Overall	w/o TV-RegSLERP	0.95	18.18%	6.20	14.28	92	98.30%
	w/o DBAS-Gen	0.87	47.55%	5.35	14.28	74	**99.37%**
	w/o Both	0.86	47.65%	5.32	14.28	70	98.74%
TV-RegSLERP	Random-RegSLERP	0.96	16.75%	6.07	14.28	80	98.41%
DBAS-Gen	w/o Balanced	0.94	21.77%	6.26	14.28	100	96.87%
	w/o Exploration	0.96	14.75%	6.07	14.28	80	97.28%
	w/o Both	0.85	49.55%	5.31	14.28	74	96.62%

Overall Ablation. Removing TV-RegSLERP reduces H_1 from 6.39 to 6.20 and the observed support from 96 to 92 labels. Removing DBAS-Gen causes a much larger loss in distributional balance, with $H_{1,\mathrm{norm}}$ dropping from 0.97 to 0.87 and Top-10 concentration rising from 15.61% to 47.55%. Removing both components further reduces the support to 70 labels. These results show that TV-RegSLERP mainly expands complexity support, while DBAS-Gen is critical for balancing the final distribution.

TV-RegSLERP Ablation. Random parent pairing reduces H_1 from 6.39 to 6.07 and the observed support from 96 to 80 labels, despite maintaining $H_{1,\mathrm{norm}} = 0.96$. This confirms the importance of constrained parent pairing for effective support expansion.

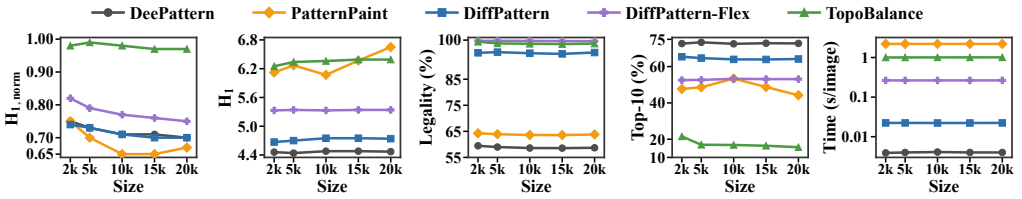
DBAS-Gen Ablation. The two modes are complementary. Without Balanced mode, Exploration reaches 100 labels but reduces $H_{1,\mathrm{norm}}$ to 0.94 and increases Top-10 concentration to 21.77%. Without Exploration mode, Balanced sampling achieves $H_{1,\mathrm{norm}} = 0.96$ and a low Top-10 concentration of 14.75%, but covers only 80 labels. Removing both modes yields the poorest distributional balance. Thus, Balanced mode improves distributional balance, while Exploration expands complexity support, and their combination achieves the highest overall H_1. Across all ablations, H_2 remains unchanged and Legality varies only moderately, indicating that the main effects lie in complexity-label coverage and distributional balance.

5.4 RQ3: How Does TopoBalance Scale with Increasing Legal-Library Size?

Motivation. We evaluate scalability by varying the final legal-library size from 2,000 to 20,000.

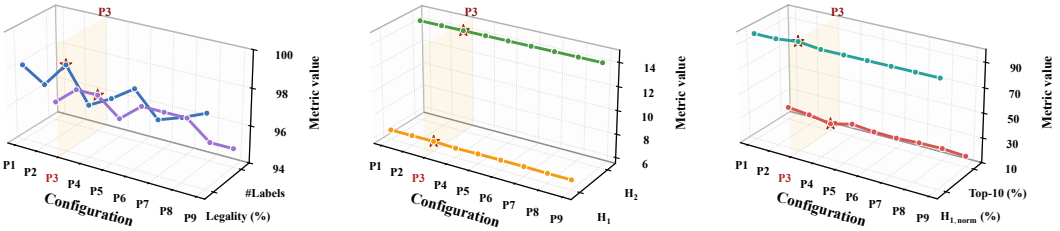
Results. As the legal-library size increases, the baseline methods show different scaling behaviors. DeePattern and DiffPattern remain relatively stable in H_1 but exhibit consistently low $H_{1,\mathrm{norm}}$ and high Top-10 concentration. DiffPattern-Flex maintains the highest legality, but its distributional balance gradually decreases as the library grows. PatternPaint achieves relatively high H_1 but shows larger fluctuations in $H_{1,\mathrm{norm}}$ and Top-10 concentration.

In contrast, TopoBalance maintains $H_{1,\mathrm{norm}}$ between 0.97 and 0.99 while increasing H_1 from 6.25 to 6.39. Its Top-10 concentration decreases from 21.55% to 15.61%, indicating increasingly balanced coverage with scale, while legality remains above 98.48%. Per-image generation time remains nearly constant for all methods across different library sizes, including TopoBalance at about 1.0 s/layout. These results show that TopoBalance preserves strong diversity, distributional balance, and legality as the legal-layout library grows.



(a) $H_{1,\text{norm}}$ (b) H_1 (c) Legality (d) Top-10 (e) Time cost

Figure 2: Scalability of different methods with increasing library size.



(a) Legality and #Labels (b) H_1 and H_2 (c) $H_{1,\text{norm}}$ and Top-10

Figure 3: Sensitivity of TopoBalance to Exploration probability configurations. Each configuration corresponds to a pair of Exploration probabilities before and after entropy plateau detection.

5.5 RQ4: Sensitivity to Mode-Selection Probabilities

Motivation. We investigate the sensitivity of TopoBalance to the DBAS-Gen mode-selection probabilities p_{balanced} and $p_{\text{exploration}}$. Specifically, we evaluate nine combinations of p_{balanced} in the non-plateau stage ($\text{plateau} * t = 0$) and $p * \text{exploration}$ in the plateau stage ($\text{plateau}_t = 1$), as summarized in Table 5.

Table 5: DBAS-Gen mode-selection probability configurations.

Cfg.	$\text{plateau}_t = 0$ $(P_{\text{bal}}, P_{\text{exp}})$	$\text{plateau}_t = 1$ $(P_{\text{bal}}, P_{\text{exp}})$	Cfg.	$\text{plateau}_t = 0$ $(P_{\text{bal}}, P_{\text{exp}})$	$\text{plateau}_t = 1$ $(P_{\text{bal}}, P_{\text{exp}})$	Cfg.	$\text{plateau}_t = 0$ $(P_{\text{bal}}, P_{\text{exp}})$	$\text{plateau}_t = 1$ $(P_{\text{bal}}, P_{\text{exp}})$
P1	(0.80, 0.20)	(0.20, 0.80)	P4	(0.70, 0.30)	(0.20, 0.80)	P7	(0.60, 0.40)	(0.20, 0.80)
P2	(0.80, 0.20)	(0.30, 0.70)	P5	(0.70, 0.30)	(0.30, 0.70)	P8	(0.60, 0.40)	(0.30, 0.70)
P3 (*)	(0.80, 0.20)	(0.40, 0.60)	P6	(0.70, 0.30)	(0.40, 0.60)	P9	(0.60, 0.40)	(0.40, 0.60)

Results. The performance of TopoBalance is shown in Figure 3. Overall, TopoBalance remains robust across different mode-selection probabilities. The diversity-related metrics are stable: H_2 remains unchanged at 14.28, H_1 varies only slightly from 6.32 to 6.39, and the number of represented complexity labels remains within 95–96. However, legality and distributional balance are more sensitive to the probability configuration. Among all settings, P3 achieves the strongest overall performance, with the highest legality (98.66%), highest H_1 (6.39), highest $H_{1,\text{norm}}$ (0.97), and lowest Top-10 concentration (15.61%). Increasing the probability of Exploration mode beyond this setting does not further improve complexity-label coverage and may reduce generation feasibility. Therefore, we adopt P3 as the default configuration in all main experiments.

6 Conclusion

We present TopoBalance, a distribution-aware framework for legal layout-pattern generation under limited training data. Its key innovation is to jointly address training-stage support sparsity and generation-stage distribution imbalance. TV-RegSLERP expands complexity-label support through topology-valid cross-complexity latent interpolation, while DBAS-Gen adaptively coordinates complexity-label sampling, layout generation and legalization, and feedback update to balance the evolving legal-layout distribution. Together, these two components enable broader and more balanced exploration of the legal layout space while maintaining high generation feasibility. Future work will extend TopoBalance to multi-layer layouts, via connections, and more complex process rules, and evaluate its generalization across additional technology nodes and downstream DFM tasks.

AI USE STATEMENT

Generative AI tools were used solely for language polishing to improve the clarity, grammar, and readability of the manuscript. They were not used for study design, methodology development, data analysis, result generation, or scientific interpretation. All AI-assisted language edits were reviewed and approved by the authors, who take full responsibility for the final content of the manuscript.

REPRODUCIBILITY STATEMENT

To support reproducibility, we publicly release the source code of our method on GitHub, including the complete implementation, training and evaluation pipelines. The dataset used in this work is also provided with an accessible link.

REFERENCES

Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: On the curvature of deep generative models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJzRZ-WCZ.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=3itjR9QxFw.

Hyungjin Chung, Jeongsol Kim, Geon Yeong Park, Hyelin Nam, and Jong Chul Ye. CFG++: Manifold-constrained classifier-free guidance for diffusion models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=E77uvbOTtp.

Yaobin Feng, Zhiyang Song, Jiaman Liu, Zhiqin Li, Feng Yang, Hao Jiang, and Shiyuan Liu. Layout pattern analysis and coverage evaluation in computational lithography. *Optics Express*, 31(5): 8897–8913, 2023. doi: 10.1364/OE.485206.

Frank Gennari and Ya-Chieh Lai. Topology design using squish patterns. U.S. Patent 8,832,621, September 2014.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, pp. 2672–2680, 2014.

Genquan Han and Yue Hao. Design technology co-optimization towards sub-3 nm technology nodes. *Journal of Semiconductors*, 42(2):020301, 2021. doi: 10.1088/1674-4926/42/2/020301.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Pervaiz Kareem, Yonghwi Kwon, and Youngsoo Shin. Layout pattern synthesis for lithography optimizations. *IEEE Transactions on Semiconductor Manufacturing*, 33(2):283–290, 2020. doi: 10.1109/TSM.2020.2982989.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pp. 52–59, 2011.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pp. 8024–8035, 2019.

Amyn Poonawala and Peyman Milanfar. Mask design for optical microlithography—an inverse imaging problem. *IEEE Transactions on Image Processing*, 16(3):774–788, 2007. doi: 10.1109/TIP.2006.891332.

Gaurav Rajavendra Reddy, Constantinos Xanthopoulos, and Yiorgos Makris. Enhanced hotspot detection through synthetic pattern generation and design of experiments. In *Proceedings of the IEEE VLSI Test Symposium (VTS)*, pp. 1–6, 2018. doi: 10.1109/VTS.2018.8368646.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022. doi: 10.1109/CVPR52688.2022.01042.

Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27 (3):379–423, July 1948.

Soobin Um, Suhyeon Lee, and Jong Chul Ye. Don't play favorites: Minority guidance for diffusion models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3NmO9lY4Jn.

Zixiao Wang, Yunheng Shen, Wenqian Zhao, Yang Bai, Guojin Chen, Farzan Farnia, and Bei Yu. DiffPattern: Layout pattern generation via discrete diffusion. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2023. doi: 10.1109/DAC56929.2023.10248009.

Zixiao Wang, Wenqian Zhao, Yunheng Shen, Yang Bai, Guojin Chen, Farzan Farnia, and Bei Yu. DiffPattern-Flex: Efficient layout pattern generation via discrete diffusion. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 44(12):4679–4690, 2025. doi: 10.1109/TCAD.2025.3569493.

Haoyu Yang, Jing Su, Yi Zou, Bei Yu, and Evangeline F. Y. Young. Layout hotspot detection with feature tensor generation and deep biased learning. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2017. doi: 10.1145/3061639.3062270.

Haoyu Yang, Piyush Pathak, Frank Gennari, Ya-Chieh Lai, and Bei Yu. Detecting multi-layer layout hotspots with adaptive squish patterns. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 299–304, 2019. doi: 10.1145/3287624.3288747.

Haoyu Yang, Wei Zhong, Yuzhe Ma, Hao Geng, Ran Chen, Wanli Chen, and Bei Yu. VLSI mask optimization: From shallow to deep learning. *Integration, the VLSI Journal*, 77:96–103, March 2021. doi: 10.1016/j.vlsi.2020.11.001.

Haoyu Yang, Shuhe Li, Wen Chen, Piyush Pathak, Frank Gennari, Ya-Chieh Lai, and Bei Yu. DeePattern: Layout pattern generation with transforming convolutional auto-encoder. *IEEE Transactions on Semiconductor Manufacturing*, 35(1):67–77, February 2022. doi: 10.1109/TSM.2021.3139354.

Zeqi Ye, Qijie Zhu, Molei Tao, and Minshuo Chen. Provable separations between memorization and generalization in diffusion models. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=42gfTZzyvV.

Xiaopeng Zhang, James Shiely, and Evangeline F. Y. Young. Layout pattern generation and legalization with generative learning models. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–9, 2020. doi: 10.1145/3400302.3415607.

Zekai Zhang, Xiao Li, Xiang Li, Lianghe Shi, Meng Wu, Molei Tao, and Qing Qu. Generalization of diffusion models arises with a balanced representation space. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=57THeGgNAN.

Guanglei Zhou, Bhargav Korrapati, Gaurav Rajavendra Reddy, Chen-Chia Chang, Jingyu Pan, Jiang Hu, Yiran Chen, and Dipto G. Thakurta. PatternPaint: Practical layout pattern generation using diffusion-based inpainting. In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, pp. 1–7, 2025. doi: 10.1109/DAC63849.2025.11132857.

A Full Problem Statement

This section provides the detailed definitions underlying the compact formulation in Section 3.

Definition 1 (Layout Representation). A layout pattern is represented as

$$\mathcal{L} = (B, d^x, d^y), \tag{17}$$

where

$$B = [b_{j,i}] \in \{0,1\}^{c_y \times c_x} \tag{18}$$

is the canonical binary topology matrix. Here, $b_{j,i} = 1$ indicates metal occupancy at the intersection of row interval j and column interval i, whereas $b_{j,i} = 0$ denotes an empty region. The geometry vectors

$$d^x = (d^x_1, \ldots, d^x_{c_x}), \qquad d^y = (d^y_1, \ldots, d^y_{c_y}) \tag{19}$$

specify the physical dimensions of the corresponding column and row intervals.

The canonical representation is obtained by merging adjacent rows or columns with identical occupancy patterns until no further merge is possible. Throughout this paper, B denotes the resulting canonical topology matrix. Thus, B captures the discrete layout structure, whereas (d^x, d^y) specifies its continuous physical geometry.

Definition 2 (Complexity Label and Complexity-Label Distribution). Following prior layout-generation studies (Yang et al., 2022; Wang et al., 2023; 2025; Zhou et al., 2025), each canonical topology is assigned a complexity label

$$g = \kappa(B) = (c_x, c_y), \tag{20}$$

where c_x and c_y are the numbers of canonical column and row intervals, respectively. The complexity label is a coarse structural descriptor and does not uniquely identify a topology; multiple distinct canonical topologies may therefore share the same label.

For a layout library $\mathcal{D} = \{\mathcal{L}_n\}_{n=1}^{|\mathcal{D}|}$, the number of layout samples associated with complexity label g is

$$n_{\mathcal{D}}(g) = \sum_{\mathcal{L}_n \in \mathcal{D}} \mathbf{1}[\kappa(B_n) = g], \tag{21}$$

and the empirical complexity-label distribution is

$$p_{\mathcal{D}}(g) = \frac{n_{\mathcal{D}}(g)}{|\mathcal{D}|}. \tag{22}$$

The corresponding complexity-label support is

$$\mathcal{G}(\mathcal{D}) = \{g \mid n_{\mathcal{D}}(g) > 0\}, \tag{23}$$

where $|\mathcal{G}(\mathcal{D})|$ denotes the number of represented complexity labels. The complexity-label distribution entropy is defined as

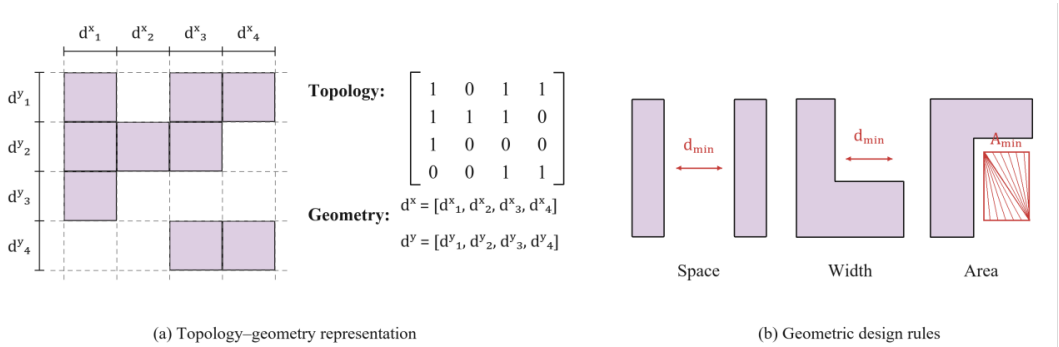
$$H_1(\mathcal{D}) = -\sum_{g \in \mathcal{G}(\mathcal{D})} p_{\mathcal{D}}(g) \log_2 p_{\mathcal{D}}(g). \tag{24}$$

A larger H_1 reflects greater complexity-label distributional diversity, which depends jointly on label coverage and distribution evenness.

Definition 3 (Legal Layout Space). Let Ω_{topo} denote the set of topology-valid canonical matrices and $\mathcal{R}$ the target physical design-rule set. For $B \in \Omega_{\text{topo}}$, the feasible geometry set is defined as

$$\begin{aligned}\Omega_{\text{geo}}(B; \mathcal{R}) = \big\{(d^x, d^y) \in \mathbb{R}_+^{c_x} \times \mathbb{R}_+^{c_y} \mid \\ \phi_{\mathcal{R}}(B, d^x, d^y) \le 0,\ \psi_{\mathcal{R}}(B, d^x, d^y) = 0\big\},\end{aligned} \tag{25}$$

where $\phi_{\mathcal{R}}$ and $\psi_{\mathcal{R}}$ represent the inequality and equality constraints induced by the target design rules and layout window.



(a) Topology–geometry representation (b) Geometric design rules

Figure 4: Topology–geometry representation and representative geometric design-rule constraints. (a) Canonical binary topology matrix and corresponding geometry vectors. (b) Minimum spacing, minimum width, and minimum area constraints.

The legal layout space is therefore

$$\Omega_{\mathcal{R}} = \{(B, d^x, d^y) \mid B \in \Omega_{\text{topo}},\ (d^x, d^y) \in \Omega_{\text{geo}}(B; \mathcal{R})\}\,. \tag{26}$$

A generated layout sample is included in the final legal library only after topology verification, geometric legalization, and independent design-rule checking (DRC).

Figure 4 illustrates the topology–geometry representation and representative geometric constraints considered in this work.

Problem Definition. Given a limited legal training set

$$\mathcal{D}_{\text{train}} = \{\mathcal{L}_n\}_{n=1}^{N} \subseteq \Omega_{\mathcal{R}}, \tag{27}$$

the goal is to construct

$$\mathcal{D}_{\text{out}}(\pi) = \{\mathcal{L}_m^{\text{out}}\}_{m=1}^{M} \subseteq \Omega_{\mathcal{R}} \tag{28}$$

with broad complexity-label coverage and a balanced complexity-label distribution.

For a generation policy π, let $C(\pi)$ denote the total number of generated candidate layout samples required to obtain M accepted legal layouts. The end-to-end legality rate is defined as

$$R_{\text{legal}}(\pi) = \frac{M}{C(\pi)}. \tag{29}$$

Legal layout-library generation is formulated as the multi-objective problem

$$\max_{\pi} \left(H_1(\mathcal{D}_{\text{out}}(\pi)), R_{\text{legal}}(\pi)\right), \tag{30}$$

subject to

$$|\mathcal{D}_{\text{out}}(\pi)| = M, \qquad \mathcal{D}_{\text{out}}(\pi) \subseteq \Omega_{\mathcal{R}}. \tag{31}$$

The objective therefore seeks greater complexity-label distributional diversity while maintaining a high end-to-end legality rate.

B Additional Method Details

B.1 Detailed Procedure of Topology-Valid RegSLERP

This appendix provides the detailed procedure of *Topology-Valid RegSLERP* (TV-RegSLERP), which expands the complexity-label support of the limited training set before diffusion training. The procedure consists of four steps: (1) construction of candidate cross-complexity parent-label

pairs; (2) construction and prioritization of target complexity labels; (3) latent interpolation; and (4) topology-validity screening and iterative augmentation.

(1) Construction of Candidate Parent-Label Pairs. Let

$$\mathcal{G}_{\text{train}} = \mathcal{G}(\mathcal{D}_{\text{train}}) \tag{32}$$

denote the set of complexity labels observed in the original training set, where each label is

$$g = (c_x, c_y). \tag{33}$$

The original training samples are grouped according to their complexity labels.

TV-RegSLERP constructs the candidate parent-label set as

$$\mathcal{P} = \left\{ (g^a, g^b) \in \mathcal{G}_{\text{train}} \times \mathcal{G}_{\text{train}} \;\middle|\; \delta_1^{\min} \le \|g^a - g^b\|_1,\ \|g^a - g^b\|_\infty \le \delta_\infty^{\max} \right\}. \tag{34}$$

The lower bound prevents the two parent labels from being overly similar, whereas the upper bound avoids excessively large cross-complexity transitions.

In our implementation, we set

$$\delta_1^{\min} = 2, \qquad \delta_\infty^{\max} = 8. \tag{35}$$

(2) Target-Label Construction and Prioritization. For each valid parent-label pair $(g^a, g^b) \in \mathcal{P}$, we consider the predefined interpolation-position set

$$\Lambda = \{0.15, 0.20, \ldots, 0.85\}, \tag{36}$$

corresponding to $\lambda \in [0.15, 0.85]$ with a step size of 0.05.

Each tuple (g^a, g^b, λ) defines an expected target complexity label

$$g^* = \text{round}\left[(1-\lambda)g^a + \lambda g^b\right]. \tag{37}$$

TV-RegSLERP then constructs a target-to-parent mapping

$$\mathcal{T}(g^*) = \left\{ (g^a, g^b, \lambda) \;\middle|\; (g^a, g^b) \in \mathcal{P},\ \lambda \in \Lambda,\ g^* = \text{round}\left[(1-\lambda)g^a + \lambda g^b\right] \right\}. \tag{38}$$

The set $\mathcal{T}(g^*)$ collects all admissible parent-label pairs and interpolation positions capable of reaching the same target label g^*.

The mapping is used to direct interpolation toward insufficiently represented regions of the complexity-label space. Target labels that are absent from the current training set receive the highest augmentation priority, followed by labels with relatively low sample frequencies. Thus, TV-RegSLERP does not uniformly sample interpolation paths, but preferentially allocates augmentation attempts toward missing or underrepresented target labels.

To avoid repeatedly spending the augmentation budget on targets that are difficult to realize, target priorities are updated according to both their current frequencies and interpolation histories. Repeated unsuccessful attempts reduce the priority of the corresponding target, progressively shifting augmentation effort toward underrepresented targets that remain practically reachable.

(3) Latent Interpolation. Once a target label g^* is selected, a compatible tuple

$$(g^a, g^b, \lambda) \in \mathcal{T}(g^*) \tag{39}$$

is sampled. One layout is then sampled from each training group associated with g^a and g^b, respectively. Let x^a and x^b denote their corresponding VAE input representations.

A pretrained VAE maps the two parent samples into the continuous latent space:

$$z^a = E_{\text{VAE}}(x^a), \qquad z^b = E_{\text{VAE}}(x^b), \tag{40}$$

where E_{VAE} denotes the VAE encoder.

Following RegSLERP, interpolation is performed by separately controlling the angular and radial transitions in the latent space. Let

$$r_a = \|z^a\|_2, \qquad r_b = \|z^b\|_2, \tag{41}$$

and

$$u_a = \frac{z^a}{r_a}, \qquad u_b = \frac{z^b}{r_b}. \tag{42}$$

The angular distance between the two latent directions is

$$\theta = \arccos\left(u_a^\top u_b\right). \tag{43}$$

The interpolated direction is obtained by spherical interpolation:

$$u_\lambda = \frac{\sin[(1-\lambda)\theta]}{\sin\theta} u_a + \frac{\sin(\lambda\theta)}{\sin\theta} u_b, \tag{44}$$

while the latent radius is linearly interpolated as

$$r_\lambda = (1-\lambda) r_a + \lambda r_b. \tag{45}$$

The final interpolated latent representation is therefore

$$z_\lambda = r_\lambda \frac{u_\lambda}{\|u_\lambda\|_2}. \tag{46}$$

The interpolated latent representation is decoded as

$$\hat{x}_\lambda = G_{\mathrm{VAE}}(z_\lambda), \tag{47}$$

where G_{VAE} denotes the pretrained VAE decoder.

(4) Topology-Validity Screening and Iterative Augmentation. The decoded candidate $\hat{x}_\lambda$ is not directly added to the training set. Its topology is first extracted and checked for validity. Let $\hat{B}_\lambda$ denote the canonical topology matrix extracted from $\hat{x}_\lambda$. The complexity label of the decoded candidate is then computed using the same mapping $\kappa(\cdot)$ defined in Section 3.

A candidate is accepted into $\mathcal{D}_{\text{TV-RegSLERP}}$ only if its decoded topology is valid and its resulting complexity label matches the intended target:

$$\kappa(\hat{B}_\lambda) = g^*. \tag{48}$$

Candidates failing either condition are discarded.

After each interpolation attempt, the history associated with target g^* is updated. If a valid and target-consistent candidate is obtained, it is inserted into $\mathcal{D}_{\text{TV-RegSLERP}}$ and the frequency of g^* is updated accordingly. Repeated unsuccessful attempts reduce the priority of the corresponding target. The target priorities are then recomputed before the next target is selected.

This process is repeated until the predefined augmentation budget is exhausted. Let $\mathcal{D}_{\text{TV-RegSLERP}}$ denote all topology-valid and target-consistent decoded samples accepted during this procedure. The expanded training set is

$$\mathcal{D}^+_{\text{train}} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{TV-RegSLERP}}. \tag{49}$$

In summary, TV-RegSLERP differs from unconstrained latent interpolation in three aspects: it restricts interpolation to controlled cross-complexity parent-label pairs, explicitly prioritizes missing or underrepresented target labels, and admits only topology-valid samples whose decoded complexity labels match the intended targets. These mechanisms expand the training support toward insufficiently represented regions while avoiding indiscriminate introduction of invalid interpolated samples.

B.2 Distribution-Balanced Adaptive Sampling and Generation

Section 4.3 presents DBAS-Gen as a closed loop with three stages: mode-aware complexity-label sampling, layout generation and legalization, and feedback update. This section provides only the implementation details needed to reproduce that loop.

(1) Mode-aware complexity-label sampling. DBAS-Gen first performs warm-up generation until the initial legal-layout library $\mathcal{D}_0$ contains N_0 accepted layouts. The resulting complexity-label frequencies and generation outcomes initialize the statistics used by the two sampling modes. At generation round t, DBAS-Gen computes $f_t(g)$, $p_t(g)$, $H_1(\mathcal{D}_t)$, and ΔH_t as defined in Section 4.3.1,

and determines whether the complexity-label entropy has reached a plateau. In the default configuration, the Balanced/Exploration probabilities are $(0.8, 0.2)$ before plateau detection and $(0.4, 0.6)$ after plateau detection. The transition is probabilistic, so both modes remain available in either state.

In Balanced mode, the target complexity g^* is sampled from the labels already represented in $\mathcal{D}_t$. Labels with smaller current frequency receive higher priority, while the historical generation-and-legalization feasibility term suppresses targets that are rare but repeatedly infeasible. The selected target also serves as the source complexity, i.e., $g_{\mathrm{src}} = g^*$.

In Exploration mode, DBAS-Gen constructs reachable targets and their target-to-action mappings $\mathcal{T}_t^{\mathrm{exp}}(g^*)$ from the complexity labels currently available in $\mathcal{D}_t$. The target construction follows the same complexity-space rule used by TV-RegSLERP, but the generation mechanism is different. An action $a = (g_{\mathrm{src}}, g_{\mathrm{ref}}, \lambda)$ determines the requested target g^*. Only g_{src} provides a source layout, whereas g_{ref} and λ specify the exploration direction in complexity space. Therefore, no two-parent latent interpolation is performed during generation.

(2) Layout generation and legalization. After g^* and g_{src} are selected, a source layout $\mathcal{L}_{\mathrm{src}}$ is sampled from the corresponding source group and encoded as z_{src}. The candidate is generated through the 50-step single-source diffusion process described in Section 4.3.2. The decoded output is processed according to the requested target complexity, canonicalized, and subjected to topology verification. Its discrete topology is then held fixed while the geometric dimensions are optimized under the physical rules listed in Section 5.1. Finally, an independent design-rule check is applied. A candidate is inserted into the legal-layout library only after passing topology verification, geometric legalization, and the final design-rule check.

(3) Feedback update and repeat. For every accepted layout, DBAS-Gen uses the complexity label of the realized canonical topology, rather than the requested target, to update $f_t(g)$ and the distribution of $\mathcal{D}_t$. Generation failures are not inserted into the legal-layout library, but their topology-verification and legalization outcomes are retained in the feasibility statistics. The updated distribution and feasibility information are then used in the next sampling round. This process repeats until M legal layouts have been collected, yielding the final legal-layout library defined in Section 4.3.3.

C Additional Experimental Details

C.1 Evaluation Protocol and Metric Interpretation

Unless otherwise stated, every method is evaluated on a final library of $N_{\mathrm{legal}} = 20{,}000$ layouts that have passed the complete physical evaluation protocol. All generated candidates, including rejected candidates, contribute to the denominator of the end-to-end legality rate. Only accepted legal layouts are used to compute the coverage, diversity, and distributional-balance metrics below.

Complexity coverage and diversity. For a legal-layout library $\mathcal{D}$, the number of represented complexity labels is

$$\#\mathrm{Labels}(\mathcal{D}) = |\mathcal{G}(\mathcal{D})|, \tag{50}$$

where $\mathcal{G}(\mathcal{D})$ and the empirical complexity-label probability $p_{\mathcal{D}}(g)$ are defined in Appendix A. The complexity-label entropy $H_1(\mathcal{D})$ is given by Eq. equation 24. Because H_1 depends on both support breadth and the allocation of samples within that support, it is reported together with #Labels and the distributional-balance metrics below.

To evaluate geometric diversity, the horizontal and vertical geometry vectors of every legal layout are quantized at 1 nm resolution. Let $\mathcal{C}_{\mathrm{geo}}(\mathcal{D})$ denote the set of observed quantized geometry categories and let $p_{\mathrm{geo}}(c)$ be the empirical probability of category c. Geometry entropy is defined as

$$H_2(\mathcal{D}) = -\sum_{c \in \mathcal{C}_{\mathrm{geo}}(\mathcal{D})} p_{\mathrm{geo}}(c) \log_2 p_{\mathrm{geo}}(c). \tag{51}$$

Distributional balance. The support-normalized complexity entropy is

$$H_{1,\mathrm{norm}}(\mathcal{D}) = \frac{H_1(\mathcal{D})}{\log_2 |\mathcal{G}(\mathcal{D})|}. \tag{52}$$

It equals one when the samples are uniformly distributed over the observed support. It therefore measures within-support uniformity and must be interpreted together with #Labels, which measures support breadth.

Distance to Uniform is one half of the ℓ_1 distance between the observed complexity-label distribution and the uniform distribution over the same support:

$$D_{\mathrm{unif}}(\mathcal{D}) = \frac{1}{2} \sum_{g \in \mathcal{G}(\mathcal{D})} \left| p_{\mathcal{D}}(g) - \frac{1}{|\mathcal{G}(\mathcal{D})|} \right|. \tag{53}$$

Top-10 concentration is the largest probability mass carried by any set of at most ten observed complexity labels:

$$\text{Top-10}(\mathcal{D}) = \max_{\substack{\mathcal{S} \subseteq \mathcal{G}(\mathcal{D}) \\ |\mathcal{S}| \leq 10}} \sum_{g \in \mathcal{S}} p_{\mathcal{D}}(g). \tag{54}$$

Lower values of D_{unif} and Top-10 indicate weaker concentration on dominant complexity labels.

Legality and candidate efficiency. The end-to-end legality rate is

$$R_{\mathrm{legal}} = \frac{N_{\mathrm{legal}}}{\#\mathrm{Candidates}}, \tag{55}$$

consistent with Eq. equation 29. Here, #Candidates includes every generated candidate, whether accepted or rejected.

The number of additional candidates required to construct the final legal library is

$$\#\mathrm{Extra} = \#\mathrm{Candidates} - N_{\mathrm{legal}}. \tag{56}$$

C.2 Ablation Configurations for RQ2

All RQ2 variants use the same diffusion, topology-processing, legalization, and design-rule-checking settings; only the component named by each ablation is changed.

The overall ablation compares the full method with variants that remove TV-RegSLERP, DBAS-Gen, or both. The *w/o TV-RegSLERP* variant removes training-stage support expansion while retaining DBAS-Gen during generation. The *w/o DBAS-Gen* variant retains TV-RegSLERP training augmentation but disables generation-stage distribution balancing. The *w/o Both* variant removes both proposed components.

The TV-RegSLERP ablation replaces the constrained cross-complexity parent selection with random selection of two distinct parent labels while retaining the same interpolation budget, RegSLERP operation, topology-validity screening, and training configuration.

The DBAS-Gen ablation uses the same TV-RegSLERP-trained model as the full method. *w/o Balanced* retains only Exploration-mode sampling; *w/o Exploration* retains only Balanced-mode sampling; and *w/o Both* disables both adaptive modes and uses the same single-source generation and legalization pipeline without complexity-distribution feedback. These controls isolate training-stage support expansion from generation-stage distribution balancing while keeping the remaining pipeline unchanged.

C.3 Scale Evaluation for RQ3

For each method, the legal-layout library is evaluated at sizes 2,000, 5,000, 10,000, 15,000, and 20,000. At every size, the metrics are computed on the corresponding prefix of the accepted legal-layout sequence. Therefore, the curves in Figure 2 describe how complexity coverage, distributional balance, legality, and per-layout generation time evolve as the same legal-layout library grows. The generation model, physical rules, and evaluation definitions remain unchanged across the five library sizes.

C.4 MODE-SELECTION SENSITIVITY FOR RQ4

The nine mode-selection configurations are defined in Table 5 of the main paper. Table 6 reports their exact numerical results, which complement the visual comparison in Figure 3. P3 is the default configuration used in the main experiments.

Table 6: Mode-selection sensitivity.

Config.	Legality↑	H_1 ↑	H_2 ↑	#Labels↑	$H_{1,\mathrm{norm}}$ ↑	Top-10↓
P1	98.08%	6.34	14.28	95	0.96	20.10%
P2	97.31%	6.36	14.28	96	0.96	18.44%
P3	**98.66%**	**6.39**	14.28	**96**	**0.97**	**15.61%**
P4	96.82%	6.32	14.28	95	0.96	20.21%
P5	97.48%	6.36	14.28	96	0.96	18.57%
P6	98.30%	6.37	14.28	96	0.96	18.34%
P7	97.00%	6.37	14.28	96	0.96	19.41%
P8	97.44%	6.35	14.28	95	0.96	19.81%
P9	97.98%	6.36	14.28	95	0.96	18.62%

Across the nine settings, H_2 remains unchanged and the observed support varies only between 95 and 96 labels. The main sensitivity is therefore in legality and the allocation of samples among the covered labels. P3 provides the strongest joint result, while assigning more probability to Exploration after plateau detection does not further expand the observed support and tends to reduce legality or distributional balance.