第 27 回 画像の認識・理解シンポジウム

Layout-Corrector: Alleviating Layout Sticking Phenomenon in Discrete Diffusion Model

岩井 翔真[1,a)]　長内 淳樹[2,b)]　北田 俊輔[2,c)]　大町 真一郎[1,d)]

概要

レイアウト生成は，カテゴリ・位置・サイズで記述される要素の集合を生成するタスクである．人間が試行錯誤を通じてレイアウトを洗練させるのに対し，現在主流の離散拡散モデル (DDM) では一度生成された要素が固着し，修正されないことを示す．この課題に対して，本研究では不調和な要素を検出する Layout-Corrector (LC) を提案する．LC は DDM の生成結果を評価し，評価値の低い要素を初期化することで要素の固着を防ぐ．実験の結果，様々な DDM に対して提案手法は一貫した性能改善を達成した．

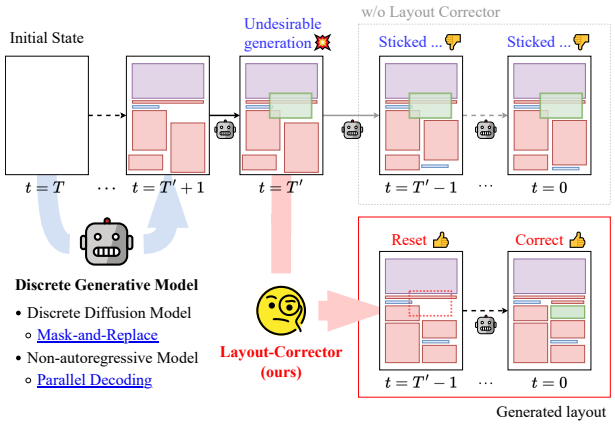


図 1: Layout-Corrector の概略．

1. はじめに

レイアウト生成はカテゴリ，位置，サイズで表現される要素の集合を生成するタスクであり，UI，広告等のデザイン自動化に向けて重要な技術である [15, 18]．このタスクでは深層学習ベースの手法が高い性能を示しており，現在は離散拡散モデル (DDM) [5, 9, 21] を用いた手法が最先端 (SoTA) である．調和のとれたレイアウトは試行錯誤を通じて作成されるが，3.2 節で示す通り，現在の DDM では，一度生成された要素が以降更新されずに固着することがわかった（図 1 上）．この修正能力の欠如に対処するため，不調和な要素を検出する Layout-Corrector (LC) を提案する（図 1 下）．LC は DDM の生成過程で暫定の生成レイアウトを評価し，不調和な要素を初期化する．初期化された要素を DDM で再生成することで，レイアウトの修正機能を実現する．実験では，LC は複数の DDM に対して性能改善を達成した．また，生成の多様性と忠実性の制御や，生成ステップ数を減らした際の性能低下の抑制に成功した．

2. 関連研究

初期のレイアウト生成の研究では，要素のアライメントなど幾何的な制約を最小化する最適化ベースの手法が提案された [16, 17]．その後，GAN [14, 23] や VAE [10, 20] といった深層学習ベースの生成モデルがレイアウトの分野でも発展した．また，画像処理，自然言語処理での成功を受け，レイアウト生成においても Transformer ベースの手法が多く提案されている [1, 6, 11]．近年では拡散モデルを用いた手法が SoTA を達成しており，レイアウト生成分野の研究の主流となっている [8, 9, 21]．特に非自己回帰 (NAR) の生成手法や拡散モデルによる手法は，条件なしの生成に加え，ユーザーが与える制約による条件付き生成の両者に対応できるため，その汎用性から研究が盛んである．

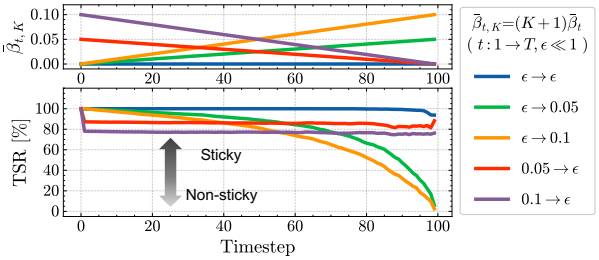
3. 提案手法

3.1 離散拡散モデル

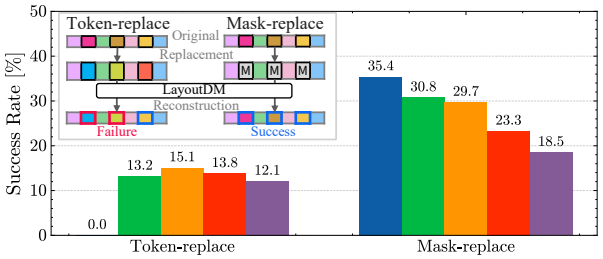
N 要素からなるレイアウトを $\boldsymbol{l}_i = (c_i, x_i, y_i, w_i, h_i), i \in [1, N]$ と表現する．ここで，$c, (x, y), (w, h)$ はカテゴリ，中心座標，サイズを表す．(x, y, w, h) は連続量であるため，$[0, 1]$ に正規化後に bin 数 B で離散化する．すなわち，$c_i \in \{1, \cdots, C\}$, $(x_i, y_i, w_i, h_i) \in \{1, \cdots, B\}^4$ となる．

次に DDM について説明する．DDM の学習は，トークン列を崩壊する拡散過程，崩壊されたトークン列を復元する生成過程からなる．時刻 t であるトークンがとる値を $z_t \in \{1, \cdots, K\}$ とする．先行研究 [9] に従い，任意長の要素を表現するための `[PAD]`，拡散過程の最終状

1 東北大学大学院工学研究科 通信工学専攻
2 LINE ヤフー株式会社
a) shoma.iwai.s4@dc.tohoku.ac.jp
b) atsuki.osanai@lycorp.co.jp
c) s.kitada@lycorp.co.jp
d) shinichiro.omachi.b5@tohoku.ac.jp



(a) 時刻 t に対する $\bar{\beta}_t$ のスケジュール (上図)，および token-sticking-rate（TSR）（下図).



(b) $\bar{\beta}_t$ スケジュールに対する，Token-replace，および Mask-replace でのランダムに置換されたトークンの修正成功率.

図 2: Rico test set における LayoutDM での事前実験結果.

態を表す [MASK] を導入し，合計 $(K+2)$ 個の状態を扱う．拡散過程は遷移行列 $\mathbf{Q}_t \in [0,1]^{(K+2)\times(K+2)}$ を用いて，$q(z_t|z_{t-1}) = \boldsymbol{v}(z_t)^\top \mathbf{Q}_t \boldsymbol{v}(z_{t-1})$ と書くことができる．ここで，$\boldsymbol{v}(z_t) \in \{0,1\}^{K+2}$ は z_t の one-hot 表現である．また，累積遷移行列 $\bar{\mathbf{Q}}_t = \mathbf{Q}_t\mathbf{Q}_{t-1}\cdots\mathbf{Q}_1$ を用いることで，時刻 0 から t への遷移を $q(z_t|z_0) = \boldsymbol{v}(z_t)^\top \bar{\mathbf{Q}}_t \boldsymbol{v}(z_0)$ と書くことができる．なお，$\mathbf{Q}_t, \bar{\mathbf{Q}}_t$ の各列の総和は 1 となる．生成過程では事後確率 $p_\theta(z_{t-1}|z_t)$ を求める．θ は生成モデルのパラメータである．ここで，re-parametrization trick を用い，$t=0$ での予測 $\tilde{p}_\theta(\tilde{z}_0|\boldsymbol{z}_t)$ を通じて，$p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t) \propto \sum_{\tilde{z}_0} q(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, \tilde{z}_0)\ \tilde{p}_\theta(\tilde{z}_0|\boldsymbol{z}_t)$ のように事後確率を求める．学習の損失関数は，変分下限 $\mathcal{L}_{\text{vlb}}$ に加え，$\tilde{p}_\theta(\tilde{z}_0|\boldsymbol{z}_t)$ に対する Cross-Entropy 損失を用いて次のように定義する．

$$\mathcal{L}_{\text{DDM}} = \mathcal{L}_{\text{vlb}} + \lambda \mathbb{E}_{\substack{\boldsymbol{z}_t\sim q(\boldsymbol{z}_t|\boldsymbol{z}_0) \\ \boldsymbol{z}_0\sim q(\boldsymbol{z}_0)}}[-\log \tilde{p}_\theta(\tilde{\boldsymbol{z}}_0|\boldsymbol{z}_t)]. \qquad (1)$$

3.2 事前実験

遷移行列の設計は DDM の挙動を決める上で重要である．同じ状態に留まる確率を α_t，[MASK] へ遷移する確率を γ_t，それ以外の状態へ遷移する確率を β_t とすると，$\mathbf{Q}'_t = \alpha_t\mathbb{I} + \beta_t\mathbf{1}\mathbf{1}^\top \in [0,1]^{(K+1)\times(K+1)}$ を用いて $\mathbf{Q}_t$ は，

$$\mathbf{Q}_t = \begin{bmatrix} \mathbf{Q}'_t & \mathbf{0} \\ \gamma_t \cdots \gamma_t & 1 \end{bmatrix}, \qquad (2)$$

となる．また，$\bar{\mathbf{Q}}_t$ に対応する各確率を $\bar{\alpha}_t, \bar{\beta}_t, \bar{\gamma}_t$ とする (導出は文献 [5] 参照)．LayoutDM [9] では，任意の時刻で $\bar{\beta}_{t,K} = (K+1)\bar{\beta}_t = \epsilon$ $(\epsilon \ll 1)$ としている．このとき，図 2a に示す通り，時刻 t と 0 でのトークン固着率 (TSR) は 100%，つまり一度トークンが [MASK] 以外の状態になると，以降更新されない．一方，$\bar{\beta}_t > \epsilon$ とすると，TSR $< 100\%$ となり固着は緩和される．

次に DDM が持つ不調和な要素の修正能力を確認する．模擬実験として，真値のレイアウトからランダムに選択した 3 トークンを別の状態でランダムに置換し，DDM が真値を復元できるか評価する．(i) [MASK] 以外の状態 (Token-replace)，または (ii) [MASK] (Mask-replace) で置換する 2 パターンの実験を行い，復元成功率で評価した（図 2b)．

Token-replace の場合，$\bar{\beta}_t = \epsilon$ では固着して修正されないが，$\bar{\beta}_t > \epsilon$ では 10% へと改善する．一方，Mask-replace では特に $\bar{\beta}_t = \epsilon$ のケースで成功率が 35% に到達する．この実験により，不調和なトークンを [MASK] に置換することで，DDM がレイアウトを修正できることが示唆された．

3.3 Layout-Corrector

本研究では不調和なトークンを特定するため，Layout-Corrector (LC) を導入する．図 3 に示すように，LC は DDM の生成過程で各トークンの正しさを評価する．このスコアが低いトークンを [MASK] に置換し，DDM による生成を続ける．これにより，LC は生成ミスの修正を明示的に促す．要素間の関係性を考慮して正しさを判定するために，LC には Transformer [19] ベースの構造を用いた．

次に LC の学習方法 (図 3 上) について述べる．LC は生成過程で誤生成されたトークンを検出する 2 値分類器として学習される．具体的には，真のレイアウト $\boldsymbol{z}_0$ と時刻 t に対し，拡散過程を経たトークン列 $q(\boldsymbol{z}_t|\boldsymbol{z}_0)$ を求め，学習済み DDM を用いて時刻 $t-1$ の分布 $p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t)$ を推定する．この分布からサンプリングを行い，[MASK] を含まない暫定の生成結果 $\hat{\boldsymbol{z}}_{t-1}$ を得る．LC は $\hat{\boldsymbol{z}}_{t-1}$ と t を受け取り，$\hat{\boldsymbol{z}}_{t-1}$ と $\boldsymbol{z}_0$ の各トークンの一致度を示すスコア $p_\phi(\hat{\boldsymbol{z}}_{t-1}, t) \in [0,1]^{5N}$ を出力する．既存研究 [13] では，$\hat{\boldsymbol{z}}_{t-1}$ の各トークンが時刻 t 時点で [MASK] であったかを推定していたが，我々は $\boldsymbol{z}_0$ との一致を評価することで，直接的にレイアウトの正しさを測定する．学習の損失関数は Binary Cross-Entropy (BCE) 損失，

$$\mathcal{L}_{\text{Corrector}} = \text{BCE}(\boldsymbol{m}, p_\phi(\hat{\boldsymbol{z}}_{t-1}, t)), \qquad (3)$$

を使用する．ここで $\hat{z}^{(i)}_{t-1} = z^{(i)}_0$ なら $m^{(i)} = 1$，そうでなければ $m^{(i)} = 0$ とする．

次に，LC と DDM を組み合わせた生成方法について述べる．条件無し生成では，図 3 下に示すように，全トークン [MASK] の状態から生成を始める．時刻 t において，DDM が分布 $p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t)$ を推定し，[MASK] を含まないトークン列 $\hat{\boldsymbol{z}}_{t-1}$ をサンプリングする．次に LC を使い，各トークン

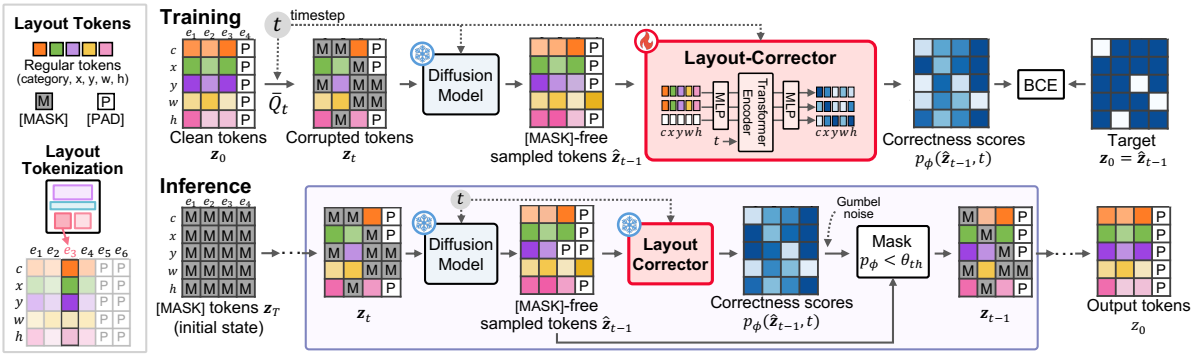


図 3: Layout-Corrector の概念図．上が学習時の流れ，下が生成時の流れを表す．

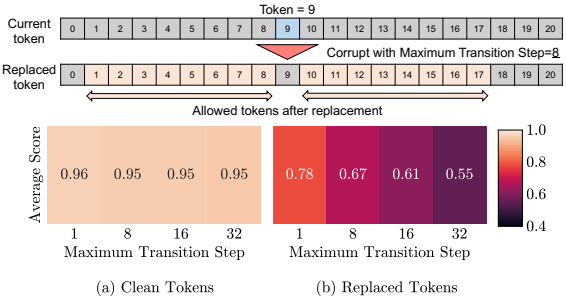


図 4: 異なる最大遷移幅に対する，(a) 置換しなかったトークンと，(b) 置換したトークンの平均スコア．

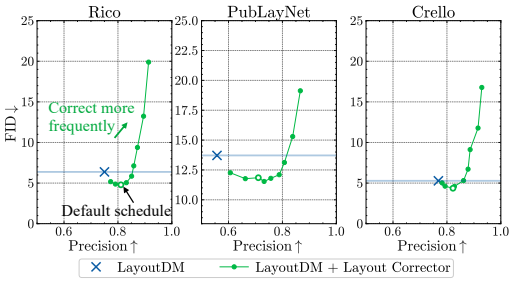


図 5: LC の適用時刻を変えた際の FID-Precision の変化．

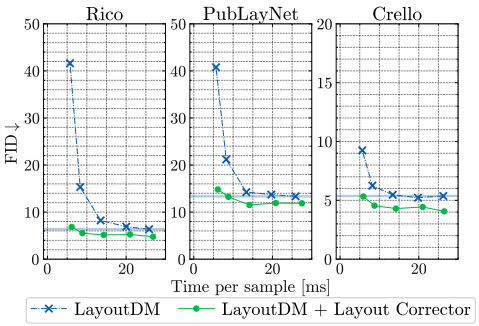


図 6: 条件無し生成の速度と品質の変化．LC は，総ステップ数を減らした際の性能低下を抑制できている．

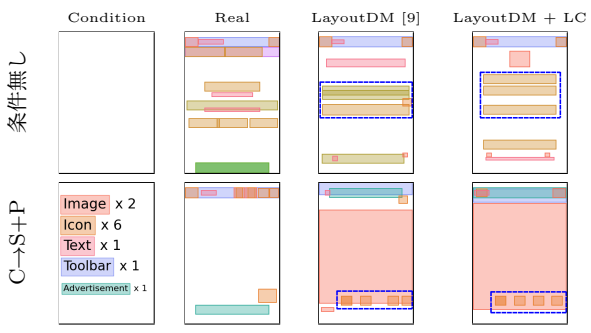


図 7: Rico の生成結果．青点線部分のように，LC は LayoutDM [9] に見られる不調和な配置を修正できている．

のスコア $p_\phi(\hat{\boldsymbol{z}}_{t-1}, t)$ を推定する．ランダム性を導入するための Gumbel ノイズを加えた後，スコアが閾値 θ_{th} を下回るトークンを `[MASK]` に置換する．条件有り生成では，レイアウトの一部（カテゴリ情報など）が条件として与えられる．井上ら [9] と同様に，与えられたトークンを初期状態とし，残りを `[MASK]` で埋める．生成の各ステップで，条件として与えられたトークンのスコアを 1 にすることで，これらが `[MASK]` で置換されないようにする．

生成時，LC は任意の時刻 t で適用できる．生成時に外部モジュールを毎時刻適用する既存手法 [13] と異なり，我々は特定時刻のみで LC を適用する．特に LayoutDM [9] では，たった 3 回の適用で性能を大きく改善することができる．さらに適用時刻を調整することで，生成の忠実性・多様性のトレードオフを制御することができる．具体的には，頻繁に LC を適用する程，不調和なトークンの初期化割合が増え，生成の忠実性が向上する．反対に適用回数を減らすことで多様性が向上する．詳細は 4.3 節で述べる．

4. 実験

4.1 実験設定

実験では，3 つの異なるドメインのデータセット (Rico [4], PubLayNet [22], Crello [20]) を使用した．レイアウト生成のタスクには，条件無しと，要素のカテゴリが条件として与えられる設定 (C→P+S) を用いる．評価指標には，生成レイアウトの分布と真の分布の類似度を計測する FID [7] と，生成レイアウトの忠実性と多様性を評価する Precision と Recall [12] を用いる．

レイアウト生成のベースラインとして，NAR から MaskGIT [3] を，DDM から LayoutDM [9] を使用する．生成の総タイムステップ数 T は，MaskGIT では 10，Lay-

表 1: 条件無し生成における， 提案手法 Corrector (LC) の有無によるベースラインモデルの性能比較．*Arch.* は離散生成モデルのアーキテクチャを表す．ベースラインモデルと比べて改善された評価指標は**太文字**で示す．

		Rico [4]			Crello [20]			PubLayNet [22]		
Model	Arch.	FID↓	Precision↑	Recall↑	FID↓	Precision↑	Recall↑	FID↓	Precision↑	Recall↑
MaskGIT [3]	Non-AR	70.37	0.793	0.437	35.32	0.802	0.376	34.23	0.587	0.460
MaskGIT + TC [13]		**15.65**	0.682	**0.843**	**7.59**	0.735	**0.815**	**17.55**	0.579	**0.825**
MaskGIT + LC (Ours)		**14.40**	**0.814**	**0.744**	**11.17**	**0.839**	**0.696**	**13.74**	0.501	**0.883**
LayoutDM [9]	DDMs	6.37	0.759	0.906	5.28	0.768	0.875	13.72	0.557	0.919
LayoutDM + TC [13]		17.97	**0.884**	0.670	9.01	**0.844**	0.678	22.27	**0.836**	0.582
LayoutDM + LC (Ours)		**4.79**	**0.811**	0.891	**4.36**	**0.822**	0.851	**11.85**	**0.711**	0.890

outDM では 100 とし，LC の適用時刻は，MaskGIT では毎時刻，LayoutDM では $t = \{10, 20, 30\}$ とする．閾値 θ_{th} は，MaskGIT では 0.3，LayoutDM では 0.7 に設定した．

4.2 Layout-Corrector の有効性

前述のベースライン手法を例に，Token-Critic (TC) [13] に対する LC の有効性を確認する．TC は画像生成のために提案された，生成過程の中間状態を評価するモジュールである．表 1 より，LC はいずれのデータセット，ベースラインに対しても一貫して FID を改善している．一方，TC は MaskGIT の FID は改善したが，LayoutDM に対しては悪化させている．これらの結果は，TC を直接レイアウト生成に適用すると，性能悪化につながることを示している．また，生成の忠実性と多様性については，NAR・DDM で異なる傾向が見られた．MaskGIT に対して，LC は忠実性と多様性の両方を改善した．生成時の確信度を基準とした MaskGIT の反復的な生成戦略は，忠実性が高く多様性が低いレイアウトを生み出す．LC はこのようなパターンをリセットするため，忠実性を保ちながら多様性を改善している．LayoutDM に対しては，多様性が僅かに低下するものの，忠実性が改善している．LayoutDM では，拡散モデルの確率的な挙動で多様性が高まる一方，分布から外れた状態がサンプリングされる場合もある．LC はこれらを `[MASK]` に戻して再生成を促し，忠実性を改善している．

4.3 Layout-Corrector の分析

スコアの分析: LC の不調和な要素を検出する能力を確かめるため，Rico test セットの真のレイアウトのうち，3 トークンをランダムに置換し，LC のスコアの値を分析した．図 4 は，ランダムにトークンの値を置換する際の最大遷移幅を変えた時の，置換されたトークンとそれ以外のトークンの平均スコアを示している．図より，置換されたトークンのスコアは，それ以外のトークンよりも低くなっている．また，遷移幅を大きくするほど，つまり真のレイアウトから大きく離れるほど出力スコアが下がる．これらの結果から，LC のスコアは不調和要素の検出だけでなく，真のレイアウトとの乖離度も表していると言える．

スケジュールと忠実性・多様性のトレードオフ: LC を様々なスケジュールで LayoutDM に適用し，FID（多様性）と Precision（忠実性）への影響を調べた．スケジュールは $t = [\{10\}, \{10, 20\}, \ldots, \{10, 20, \ldots, 90\}]$ の合計 9 パターン試し，図 5 の結果が得られた．図より，LC のスケジュール変更によって忠実性と多様性のトレードオフを制御できることが分かる．適用回数を増やすほど忠実性が上がり，減らすほど多様性が上がる傾向がある．これは不調和な要素をリセットする LC の役割と一致しており，多く適用するほど不調和な要素が減り，忠実性が上がっている．

生成速度と品質のトレードオフ: LayoutDM において生成時の総タイムステップ数を調整し，LC の有無による生成品質の変化を調べた．Fast-sampling [2] を用いて総ステップ数を $T' = \{20, 30, 50, 75, 100\}$ に調整し，それぞれの実行時間と FID の関係を図 6 に示す．LayoutDM 単体で生成した場合，T' を減らすと FID が大きく悪化しているが，LC を用いると FID の低下を抑えることができており，$T' = 20$ でも $T' = 100$ の LayoutDM と近い性能を達成している．

4.4 定性評価

Rico データセットにおける条件無し生成と C→S+P 生成の結果を図 7 に示す．LayoutDM + LC では，LayoutDM [9] の時刻 $t = \{10, 20, 30\}$ で LC を適用するため，$t = 30$ まで LayoutDM と同じ生成過程となる．よって全体の構成は類似するが，LC の適用により，要素の重なりや不揃いな配置が修正されている．これは，LC が不調和な要素を初期化し，修正を促したことを示している．

5. まとめ

本研究では，DDM における要素の固着を防ぐために Layout-Corrector を提案した．Layout-Corrector は DDM の生成過程で，不調和なトークンを初期化し，DDM による修正を促す．実験では，Layout-Corrector は NAR や DDM の手法に対して一貫した性能改善を達成した．また，生成の忠実性と多様性の制御や，生成ステップ数を減らした際の性能低下を抑えることに成功した．

参考文献

[1] Arroyo, D. M., Postels, J. and Tombari, F.: Variational transformer networks for layout generation, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13642–13652 (2021).

[2] Austin, J., Johnson, D. D., Ho, J., Tarlow, D. and Van Den Berg, R.: Structured denoising diffusion models in discrete state-spaces, *Advances in Neural Information Processing Systems*, Vol. 34, pp. 17981–17993 (2021).

[3] Chang, H., Zhang, H., Jiang, L., Liu, C. and Freeman, W. T.: Maskgit: Masked generative image transformer, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11315–11325 (2022).

[4] Deka, B., Huang, Z., Franzen, C., Hibschman, J., Afergan, D., Li, Y., Nichols, J. and Kumar, R.: Rico: A mobile app dataset for building data-driven design applications, *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pp. 845–854 (2017).

[5] Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L. and Guo, B.: Vector quantized diffusion model for text-to-image synthesis, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10696–10706 (2022).

[6] Gupta, K., Lazarow, J., Achille, A., Davis, L. S., Mahadevan, V. and Shrivastava, A.: Layouttransformer: Layout generation and completion with self-attention, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1004–1014 (2021).

[7] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. and Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium, *Advances in neural information processing systems*, Vol. 30 (2017).

[8] Hui, M., Zhang, Z., Zhang, X., Xie, W., Wang, Y. and Lu, Y.: Unifying Layout Generation with a Decoupled Diffusion Model, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1942–1951 (2023).

[9] Inoue, N., Kikuchi, K., Simo-Serra, E., Otani, M. and Yamaguchi, K.: LayoutDM: Discrete Diffusion Model for Controllable Layout Generation, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10167–10176 (2023).

[10] Jyothi, A. A., Durand, T., He, J., Sigal, L. and Mori, G.: Layoutvae: Stochastic scene layout generation from a label set, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9895–9904 (2019).

[11] Kong, X., Jiang, L., Chang, H., Zhang, H., Hao, Y., Gong, H. and Essa, I.: BLT: bidirectional layout transformer for controllable layout generation, *European Conference on Computer Vision*, pp. 474–490 (2022).

[12] Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J. and Aila, T.: Improved precision and recall metric for assessing generative models, *Advances in Neural Information Processing Systems*, Vol. 32 (2019).

[13] Lezama, J., Chang, H., Jiang, L. and Essa, I.: Improved masked image generation with token-critic, *European Conference on Computer Vision*, pp. 70–86 (2022).

[14] Li, J., Yang, J., Hertzmann, A., Zhang, J. and Xu, T.: LayoutGAN: Generating Graphic Layouts with Wireframe Discriminators, *International Conference on Learning Representations* (2019).

[15] Lok, S. and Feiner, S.: A survey of automated layout techniques for information presentations, *Proceedings of SmartGraphics*, Vol. 2001, pp. 61–68 (2001).

[16] O'Donovan, P., Agarwala, A. and Hertzmann, A.: Designscape: Design with interactive layout suggestions, *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pp. 1221–1224 (2015).

[17] PeterO' Donovan, Agarwala, A., Hertzmann, A.: Learning layouts for single-pagegraphic designs, *IEEE transactions on visualization and computer graphics*, Vol. 20, No. 8, pp. 1200–1213 (2014).

[18] Shi, Y., Shang, M. and Qi, Z.: Intelligent layout generation based on deep generative models: A comprehensive survey, *Information Fusion*, p. 101940 (2023).

[19] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I.: Attention is all you need, *Advances in neural information processing systems*, Vol. 30 (2017).

[20] Yamaguchi, K.: Canvasvae: Learning to generate vector graphic documents, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5481–5489 (2021).

[21] Zhang, J., Guo, J., Sun, S., Lou, J.-G. and Zhang, D.: LayoutDiffusion: Improving Graphic Layout Generation by Discrete Diffusion Probabilistic Models, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7226–7236 (2023).

[22] Zhong, X., Tang, J. and Yepes, A. J.: Publaynet: largest dataset ever for document layout analysis, *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1015–1022 (2019).

[23] Zhou, M., Xu, C., Ma, Y., Ge, T., Jiang, Y. and Xu, W.: Composition-aware graphic layout GAN for visual-textual presentation designs, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 4995–5001 (2022).