JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

WED-AM-185

# LayoutDM: Discrete Diffusion Model for Controllable Layout Generation

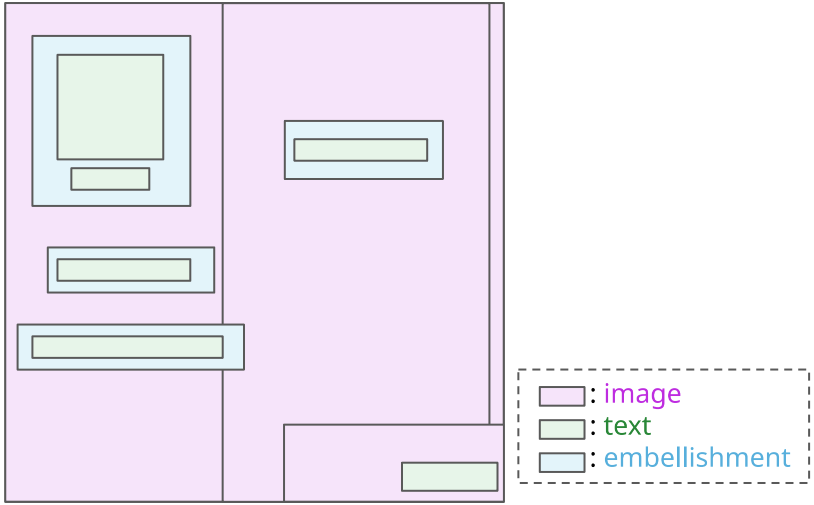

Naoto Inoue    Kotaro Kikuchi    Mayu Otani
Edgar Simo-Serra    Kota Yamaguchi

CyberAgent AI Lab

WASEDA University
早稲田大学

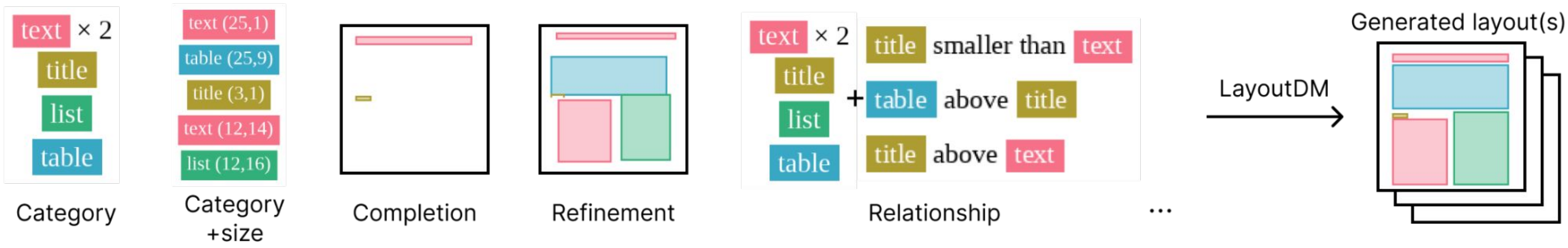
# Layout

= Simple yet essential interface to understand & control visual design

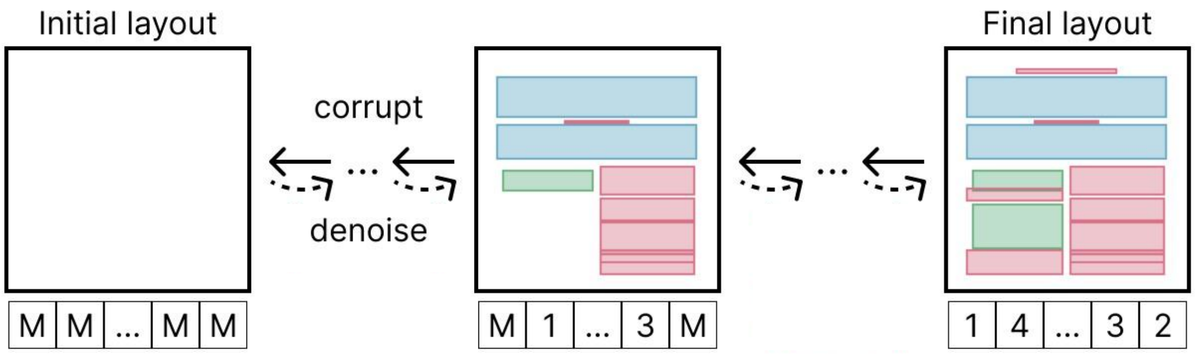
# Controllable Layout Generation

## Our work: solve a broad range of tasks in a single model



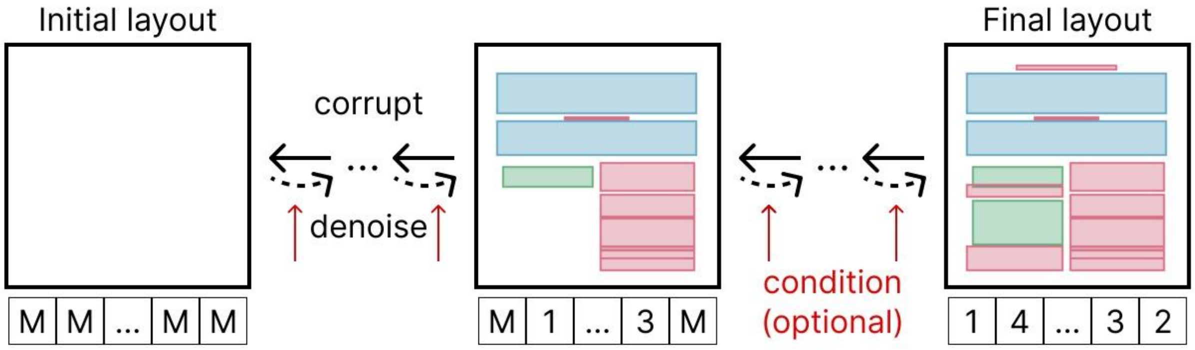
# LayoutDM

- A discrete diffusion model tamed for layout generation



# LayoutDM

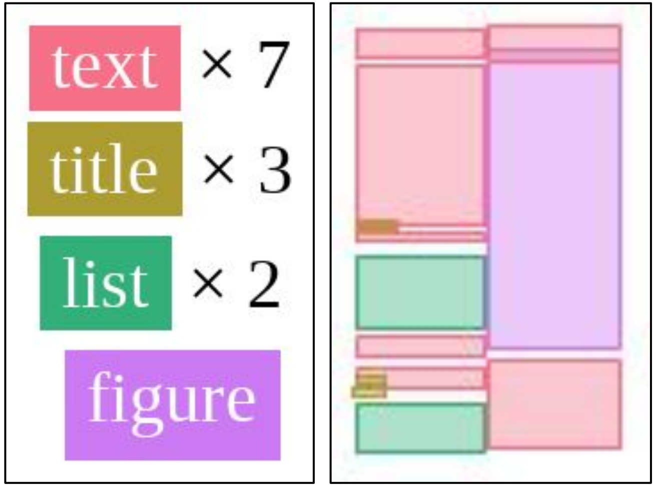
- A discrete diffusion model tamed for layout generation
- Training-free algorithm to inject various conditions during inference



# LayoutDM Results

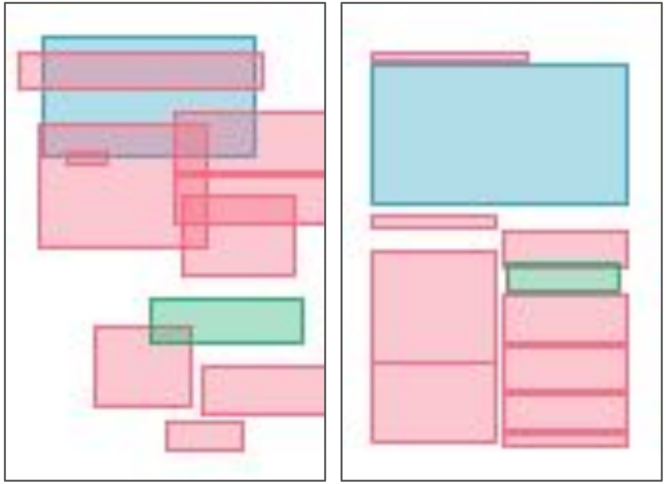
Input:
category

Output




Input:
noisy layout

Output

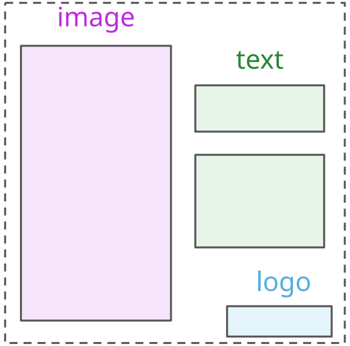


# What is Layout?

- A set of category (1-dim.) + positional info. (4-dim. e.g., xywh)
- Recent trend: layout as a sequence of discrete variables (c.f., text)

```
[
    ["image", 0.3, 0.5, 0.25, 0.9],
    ["text", 0.7, 0.35, 0.25, 0.15],
    ["text", 0.7, 0.6, 0.25, 0.4],
    ["logo", 0.85, 0.95, 0.2, 0.04],
]
```

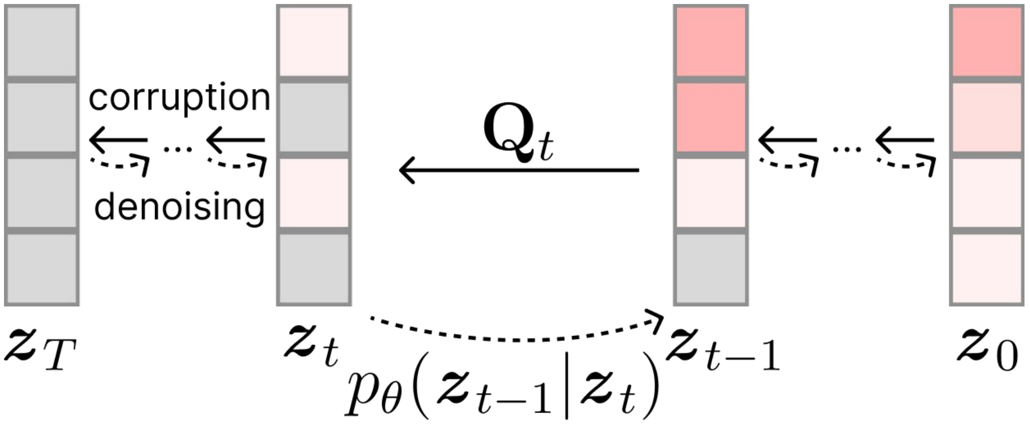
Data



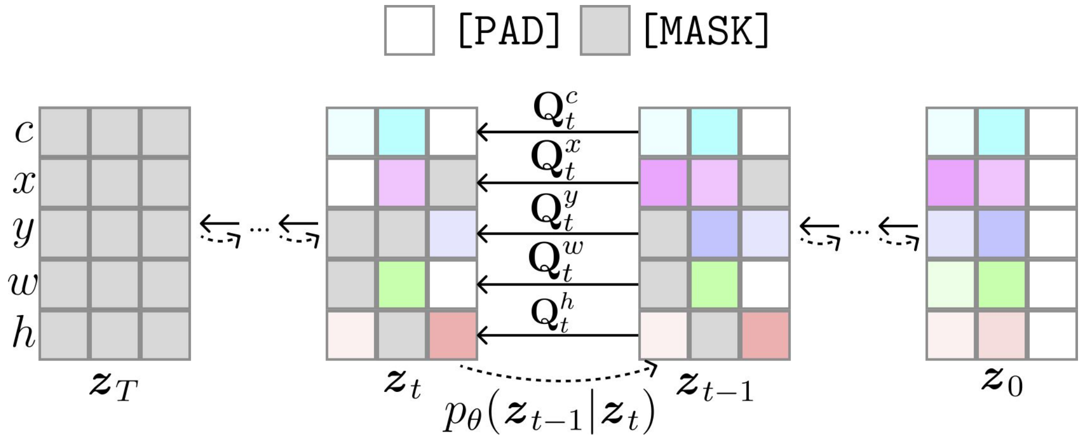

Visualization

# Discrete Diffusion Models [Austin+, NeurIPS'21]

- = diffusion models for modeling categorical variables (e.g., text)
- Corruption: a token is stochastically replaced with another in vocabulary

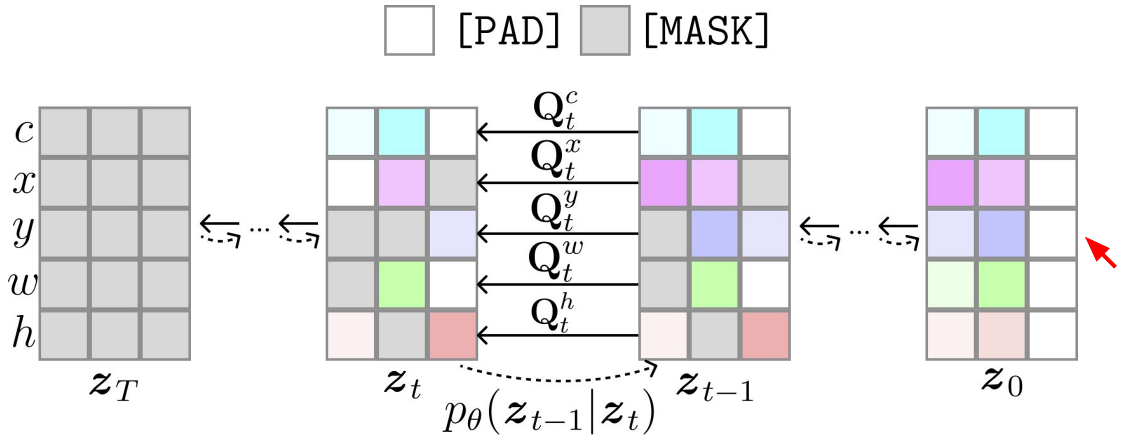


# Adapting Discrete Diffusion Models for Layout



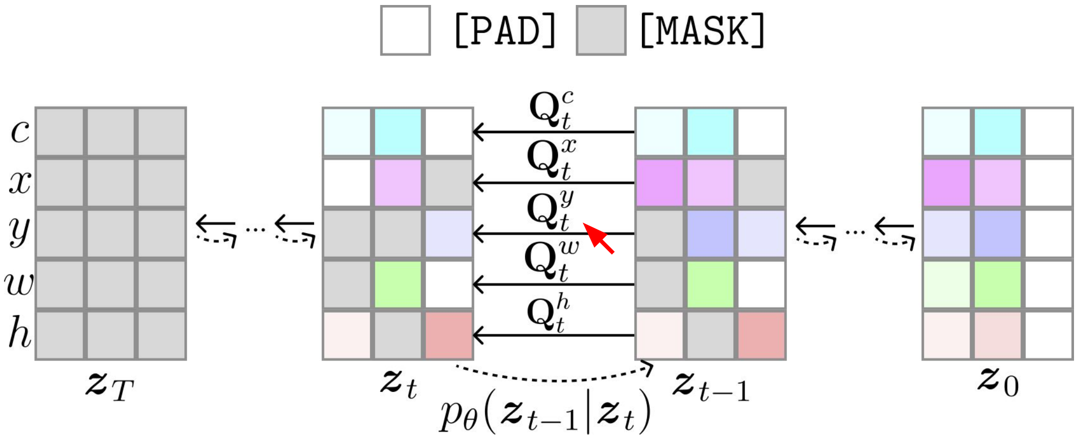
# Adapting Discrete Diffusion Models for Layout

- [PAD] token to enable variable length generation

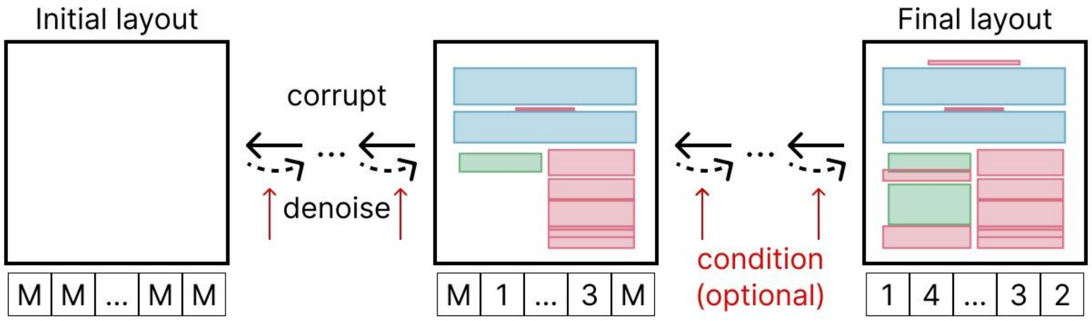


# Adapting Discrete Diffusion Models for Layout

- [PAD] token to enable variable length generation
- Modality-wise corruption process

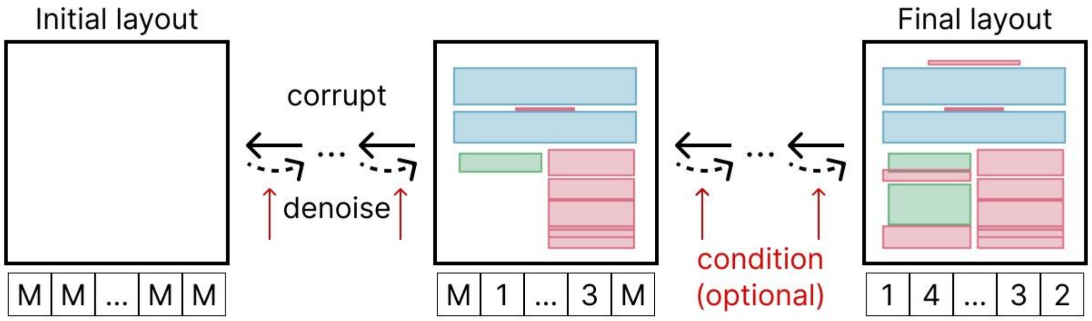


# How to Feed Conditions during Inference?



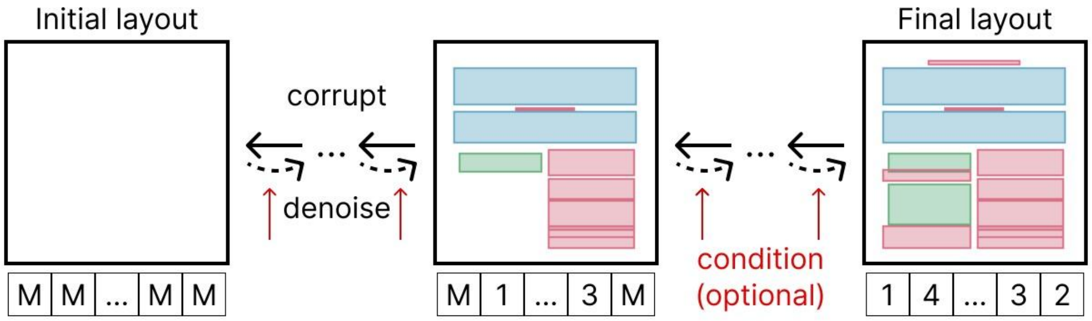
# How to Feed Conditions during Inference?

- **Hard condition: masking**
  - e.g., “i-th element’s category is C”



# How to Feed Conditions during Inference?

- **Hard condition: masking**
  - e.g., “i-th element’s category is C”
- **Soft condition: logit adjustment**
  - e.g., “an element at the top”, “an element bigger than another”



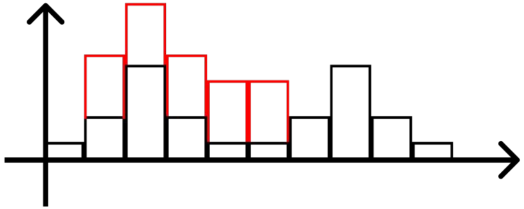
# Logit Adjustment

## Inject soft condition as a prior term

$$\log \hat{p}_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t) = \log p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t) + \lambda_\pi \boldsymbol{\pi}$$

$$\boldsymbol{z}_{t-1} \sim \hat{p}_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t)$$

Prior term

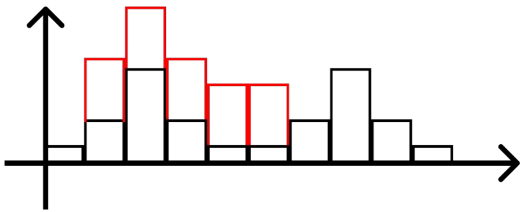


# Logit Adjustment

## Inject soft condition as a prior term

$$\log \hat{p}_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t) = \log p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t) + \underbrace{\lambda_\pi \boldsymbol{\pi}}_{\text{Prior term}}$$

$$\boldsymbol{z}_{t-1} \sim \hat{p}_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t)$$



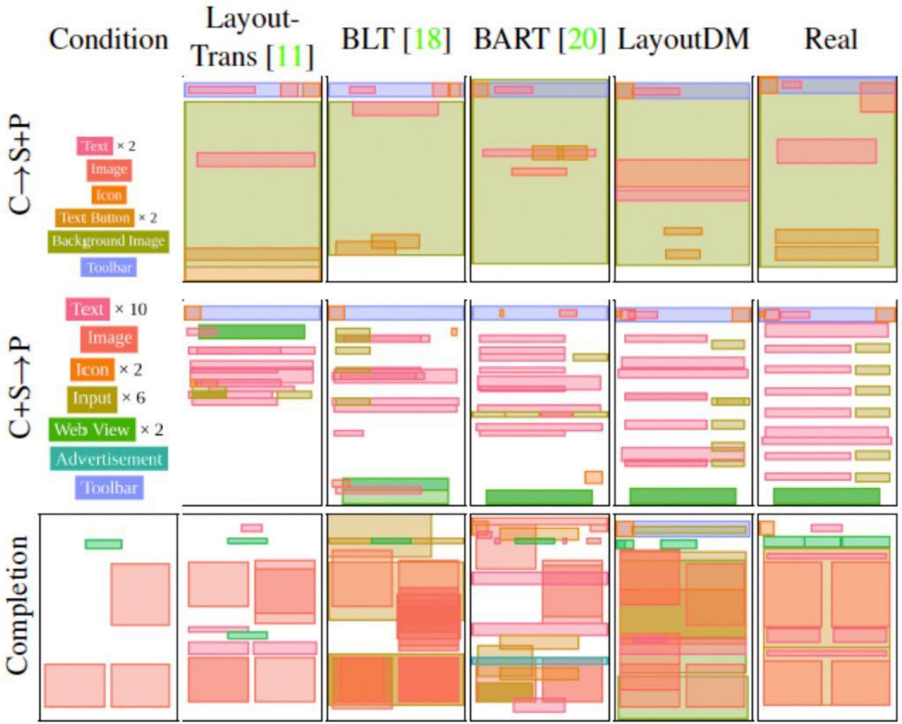
## How to implement a prior?

- Hard coding (e.g., refinement task)
- Gradients from loss functions w.r.t. the prediction (e.g., relationship task)

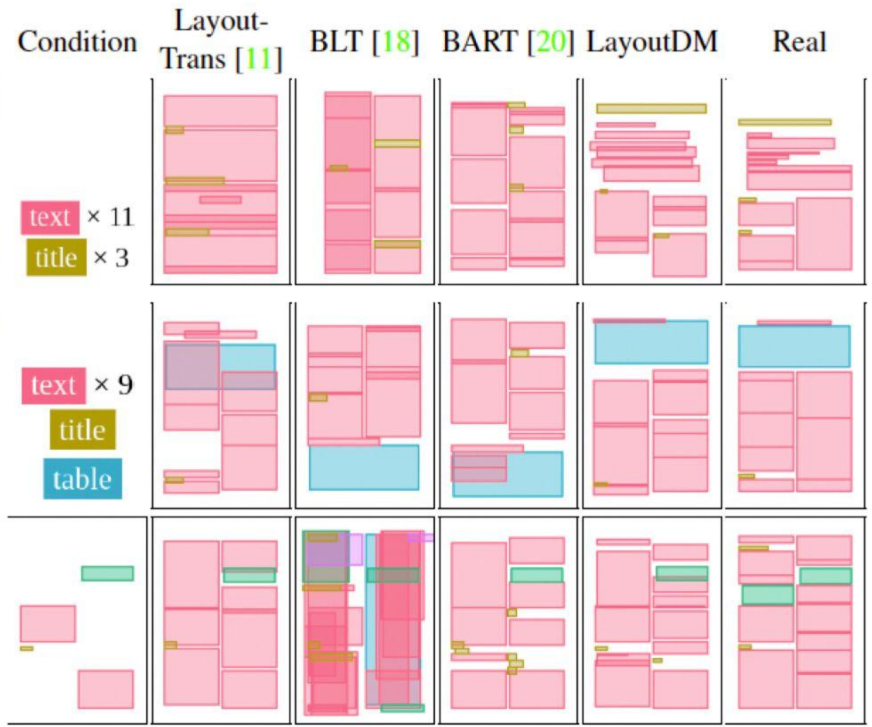
# Advantages over Existing Methods

- **No fixed generation order unlike auto-regressive models**
  - c.f., LayoutTransformer [Gupta+, ICCV'21]
- **Flexibly changing the number of elements to be generated**
  - c.f., BLT [Kong+, ECCV'22]
- **Incorporating both hard and soft conditions**
  - c.f., NDN [Lee+, ECCV'20]

# Results in Rico [Deka+, UIST'17]

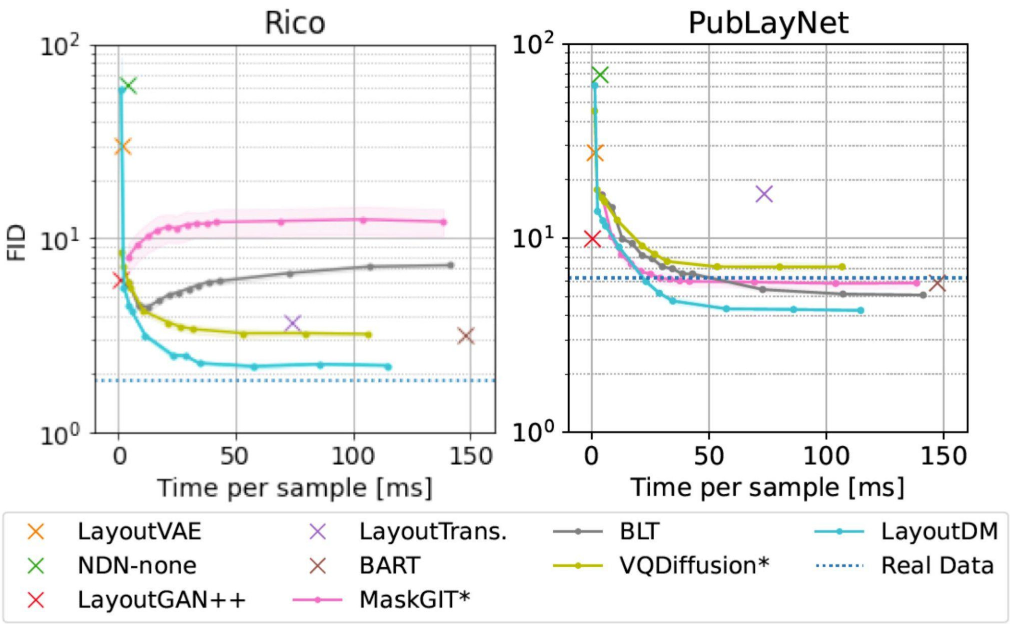


# Results in PubLayNet [Zhong+, ICDAR'19]



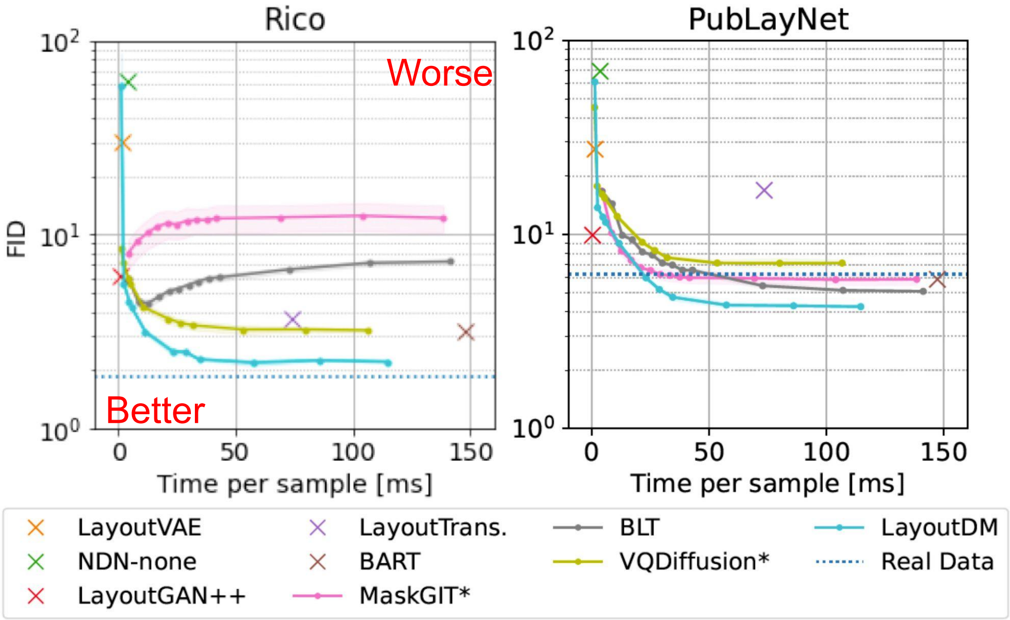
# Quantitative Evaluation (in category + size → position)

## LayoutDM achieves the best speed-quality tradeoff



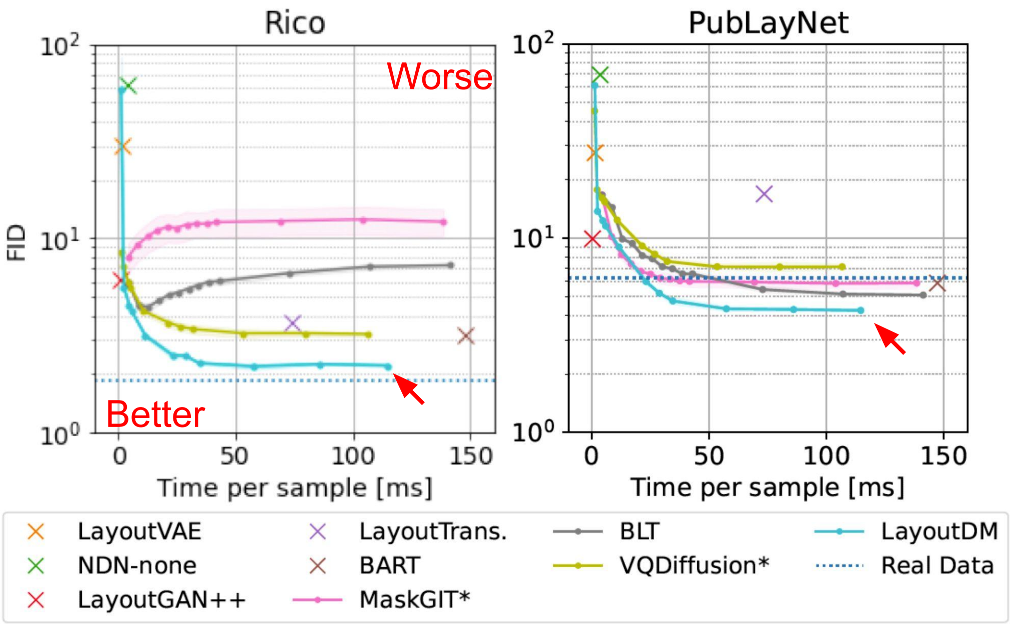
# Quantitative Evaluation (in category + size → position)

## LayoutDM achieves the best speed-quality tradeoff



# Quantitative Evaluation (in category + size → position)

## LayoutDM achieves the best speed-quality tradeoff



# Summary

- A discrete diffusion model tamed for layout generation
- Training-free algorithm to inject various conditions during inference
- Favorable performance against task-specific/agnostic baselines

Check codes and more results at

https://cyberagentailab.github.io/layout-dm/