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

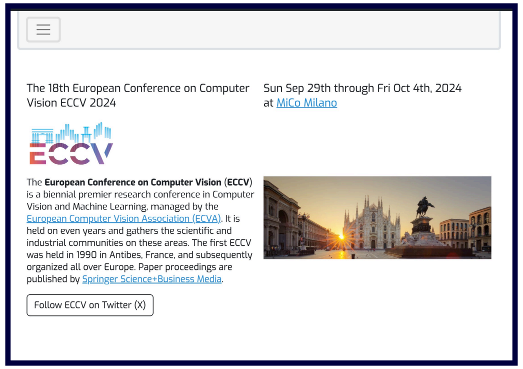
Shoma Iwai*, Atsuki Osanai, Shunsuke Kitada, Shinichiro Omachi

Tohoku University, LY Corporation

*This research was conducted during the internship at LY Corporation.

What is Layout?

ECCV2024
Website



The 18th European Conference on Computer Vision ECCV 2024

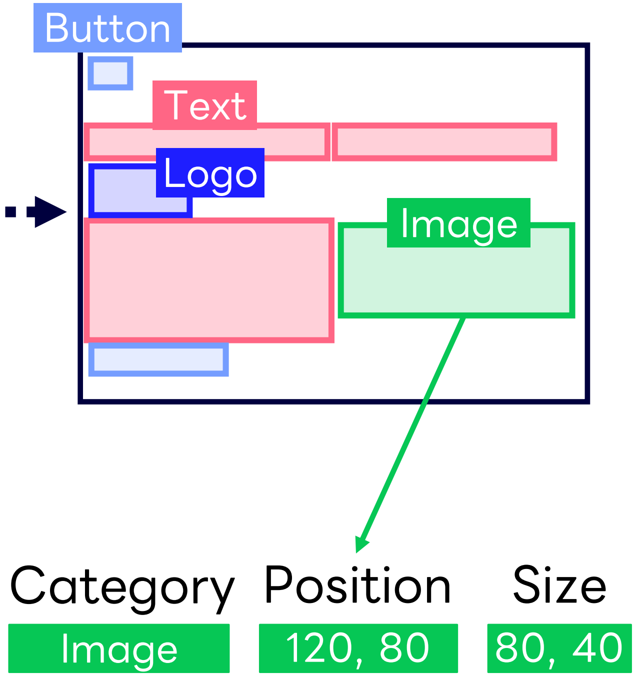
Sun Sep 29th through Fri Oct 4th, 2024 at MiCo Milano

The **European Conference on Computer Vision** (**ECCV**) is a biennial premier research conference in Computer Vision and Machine Learning, managed by the European Computer Vision Association (ECVA). It is held on even years and gathers the scientific and industrial communities on these areas. The first ECCV was held in 1990 in Antibes, France, and subsequently organized all over Europe. Paper proceedings are published by Springer Science+Business Media.

Follow ECCV on Twitter (X)

https://eccv.ecva.net/

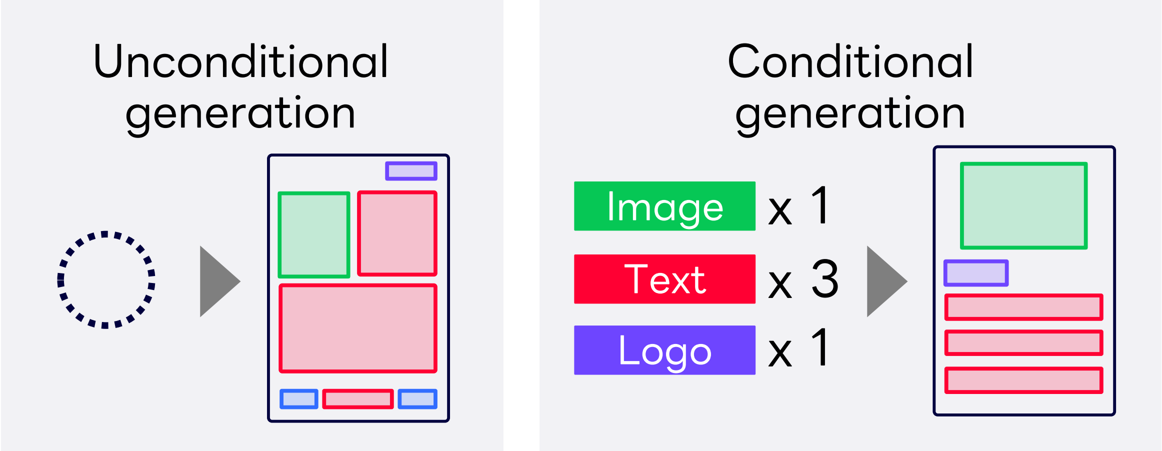
Layout:
Arrangement of elements



Category	Position	Size
Image	120, 80	80, 40

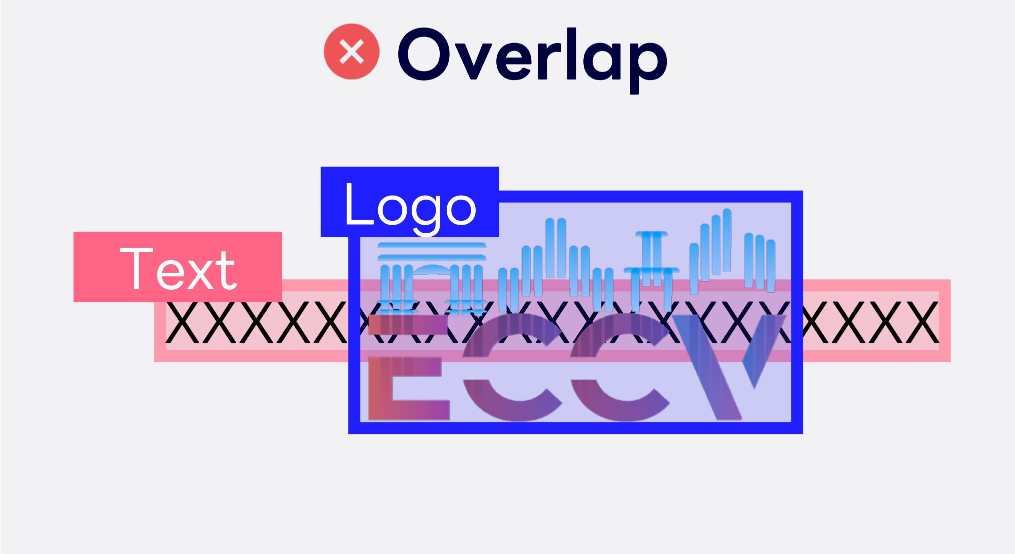
What is Layout Generation?

- Determine category, position, and size of each element.

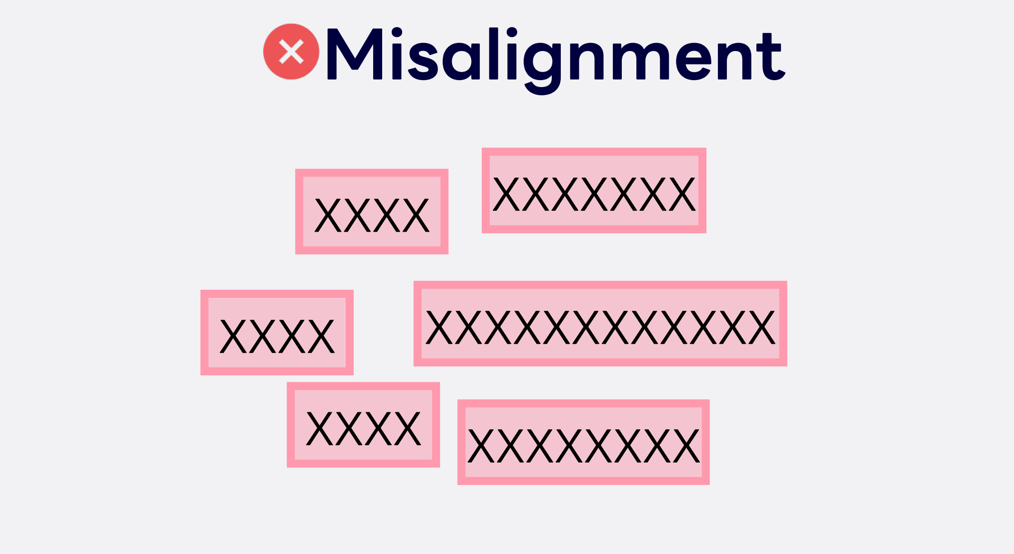


Goal: generating well-harmonized and high-quality layouts

Our goal: Avoiding Generation Error



✕ Overlap



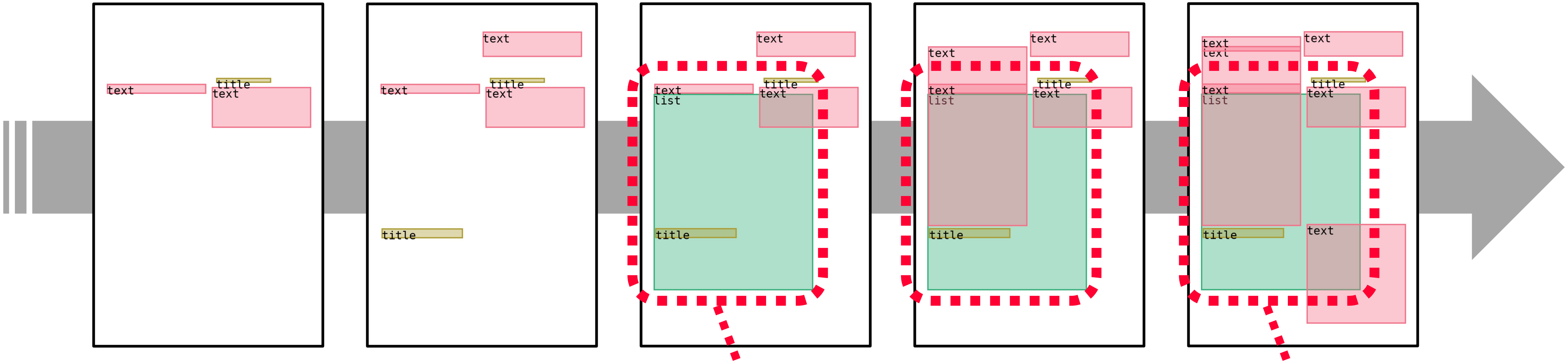
✕ Misalignment

Our goal: Avoiding Generation Error

✕ Existing Discrete Diffusion Model struggles to correct generation errors
= Generation errors got stuck! **(Layout-Sticking Problem)**

Generation process →

Final output



Error (Overlap) occurs

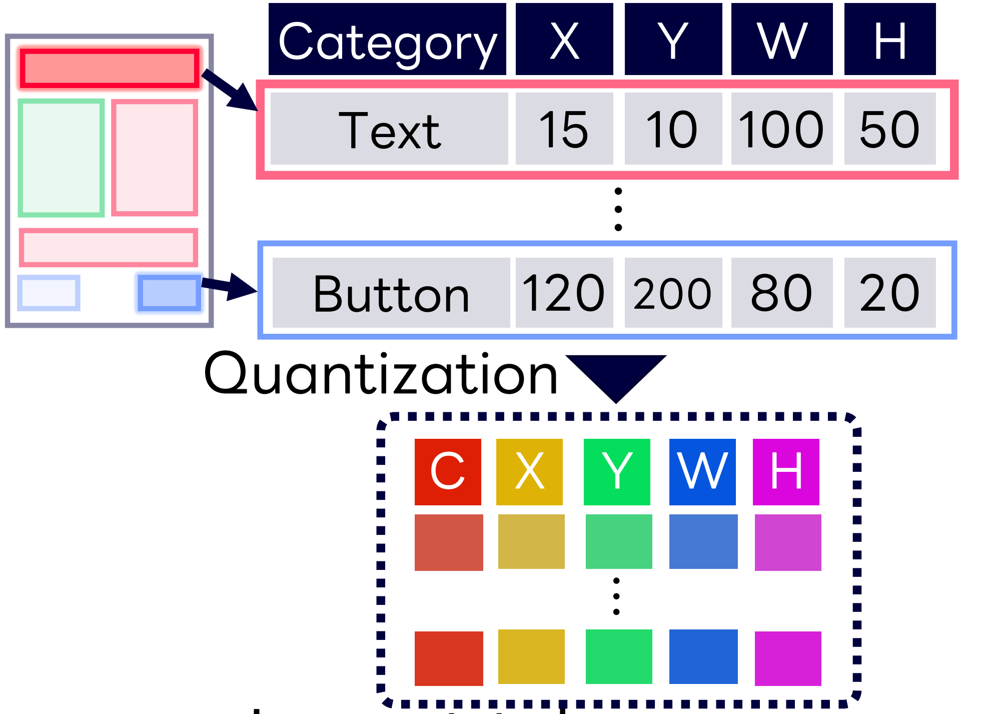
Error got stuck

→ **Our Layout-Corrector resets errors and prompts correction**

Layout Generation with Discrete Diffusion Model (DDM) [Inoue+, CVPR2023]

Layout Tokenization

Layout as Discrete Tokens

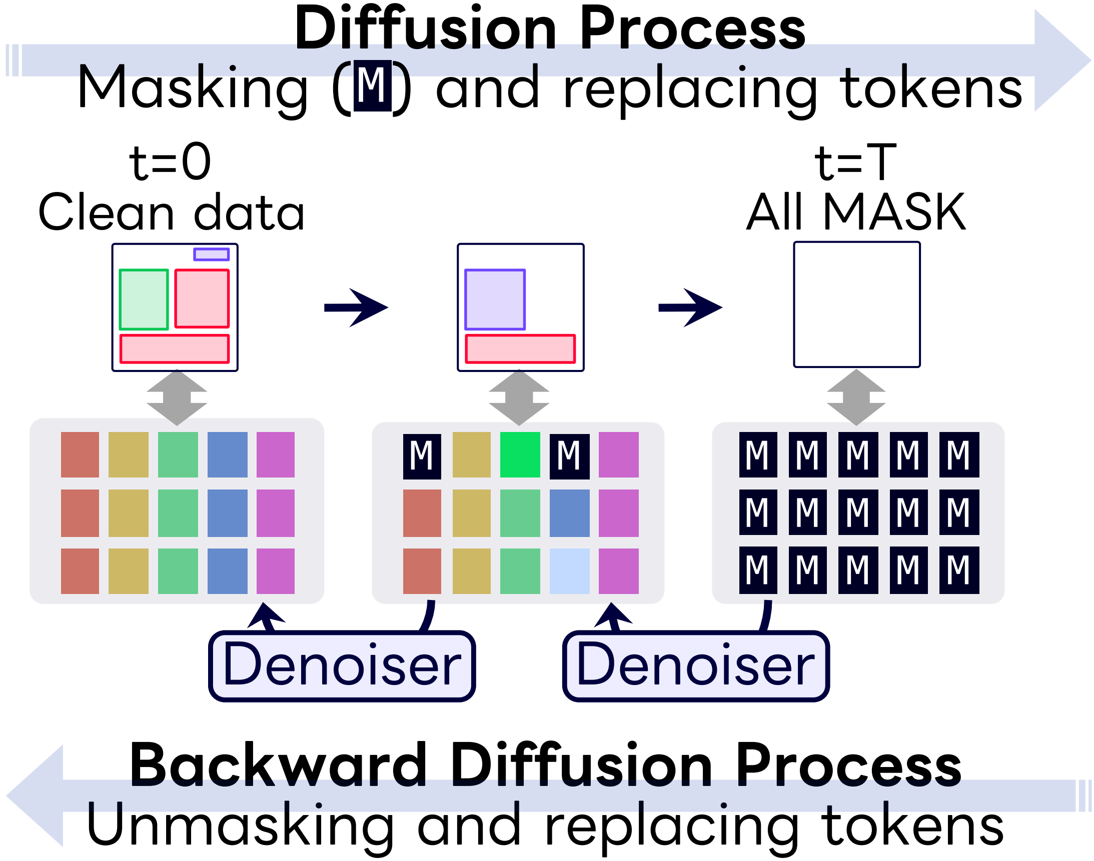


Category	X	Y	W	H
Text	15	10	100	50
⋮				
Button	120	200	80	20

Quantization

Layout token sequence

How DDM works



Diffusion Process

Masking (M) and replacing tokens

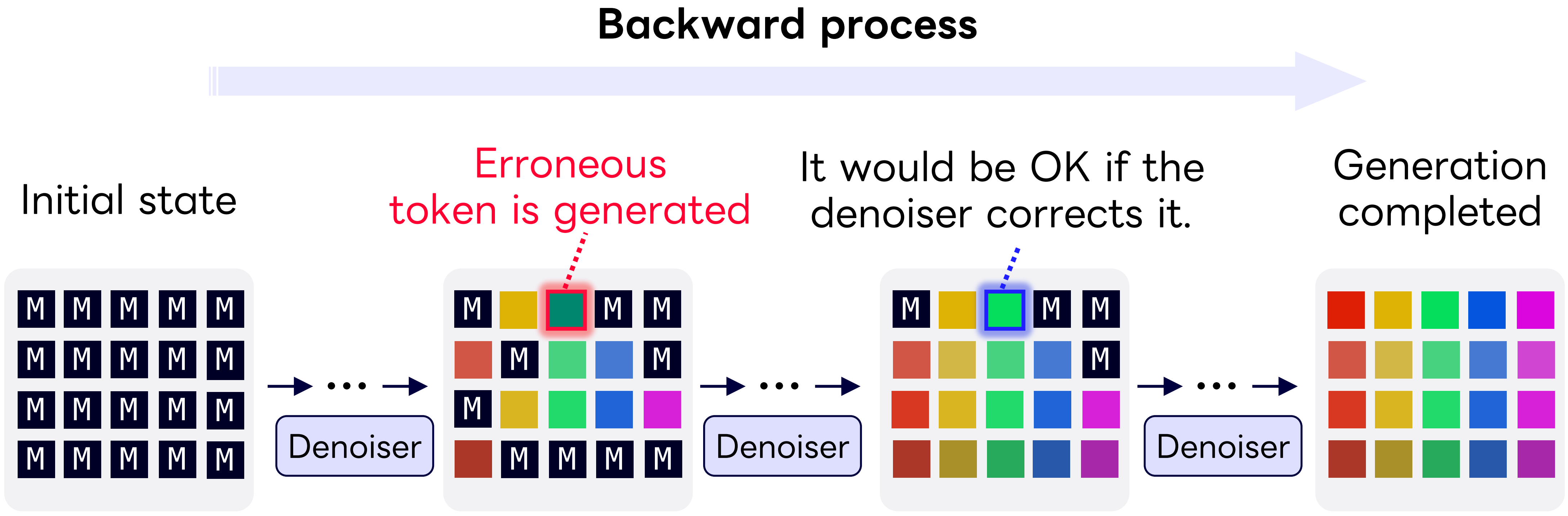
t=0
Clean data

t=T
All MASK

Backward Diffusion Process

Unmasking and replacing tokens

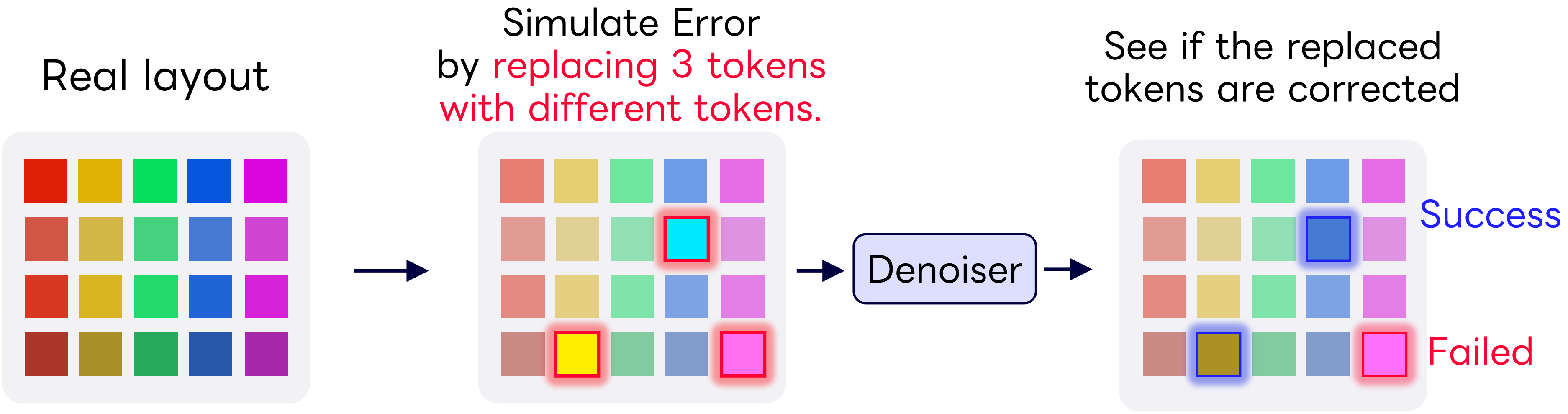
Layout Correction: A Solution to Generation Error?



Q. Can the Denoiser correct erroneous tokens?

Q. Can DDMs Correct Errors? A. Not so much

Experiment 1



Real layout

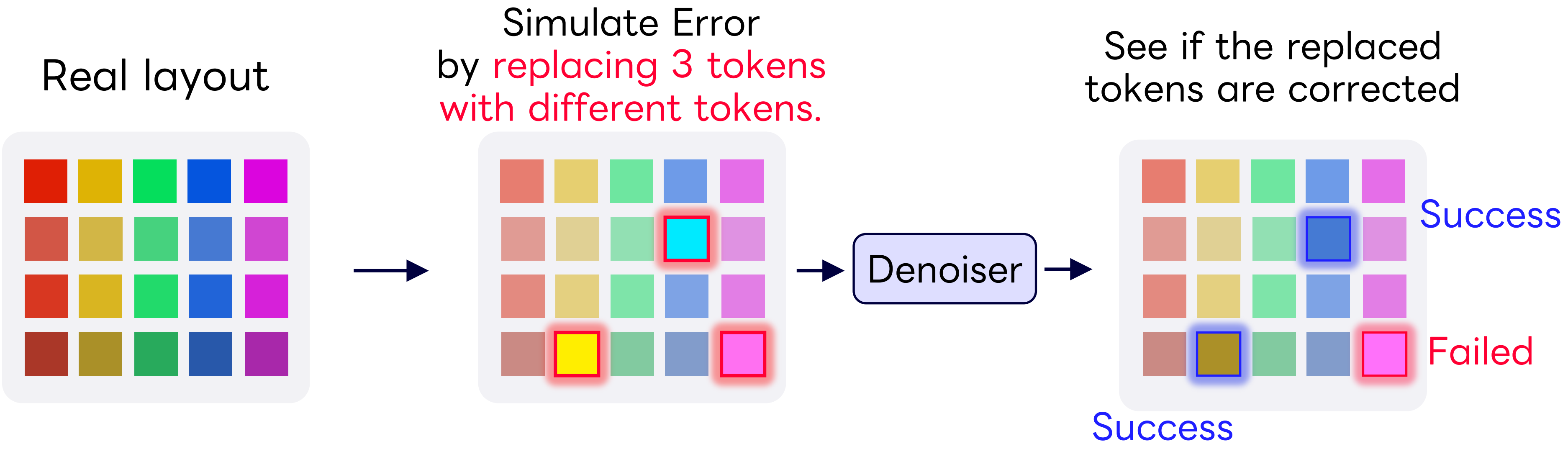
Simulate Error by replacing 3 tokens with different tokens.

Denoiser

See if the replaced tokens are corrected

Q. Can DDMs Correct Errors? A. Not so much

Experiment 1



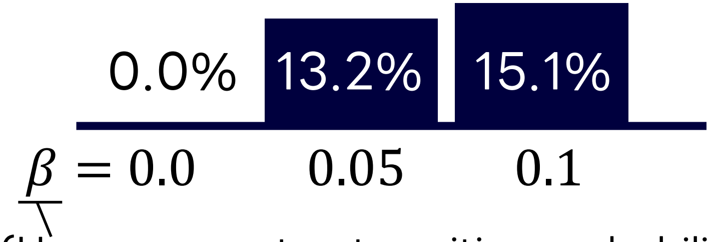
Real layout

Simulate Error by replacing 3 tokens with different tokens.

Denoiser

See if the replaced tokens are corrected

Correction success rate
(Exp. 1)



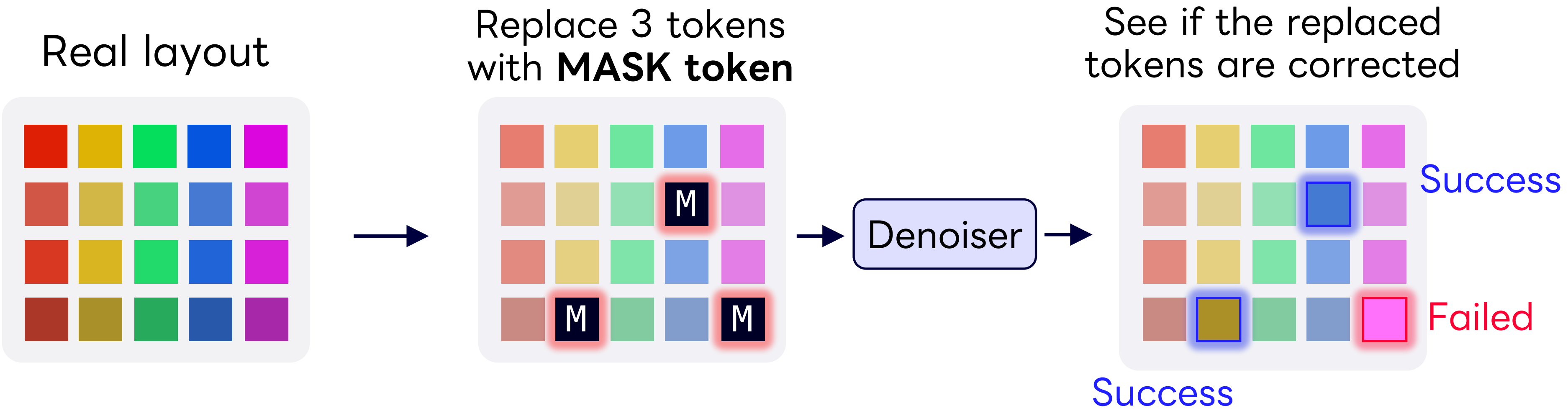
(Hyperparameter: transition probability between non-MASK tokens)

Highest Success rate: 15%

✕ DDMs struggle to correct errors, leading to layout-sticking problem

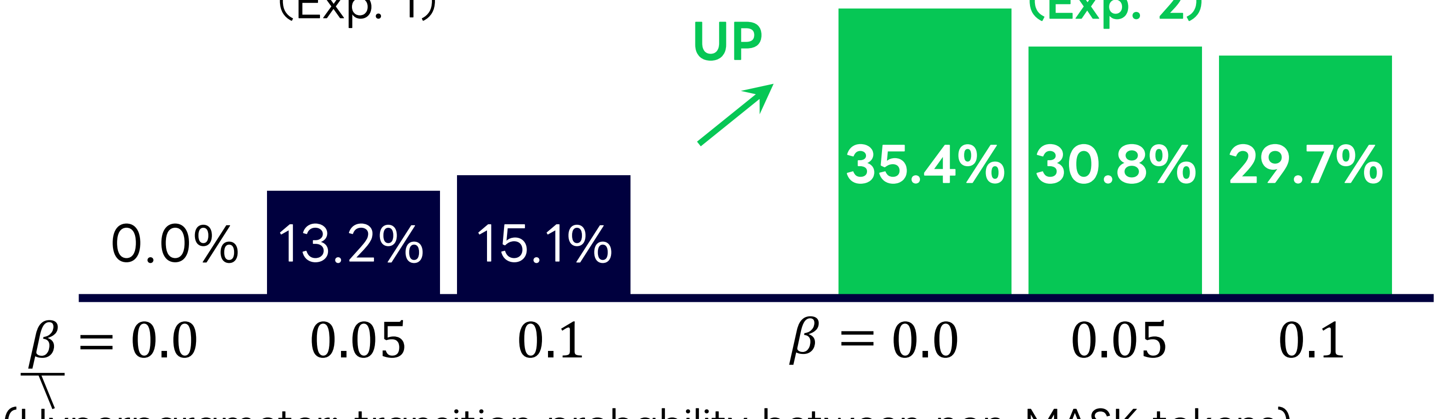
DDMs Perform Better When Reconstructing from MASK

Experiment 2


Real layout
Replace 3 tokens with **MASK token**
M
M
M
Denoiser
See if the replaced tokens are corrected
Success
Failed
Success

Correction success rate (Exp. 1)

Correction success rate (Exp. 2)



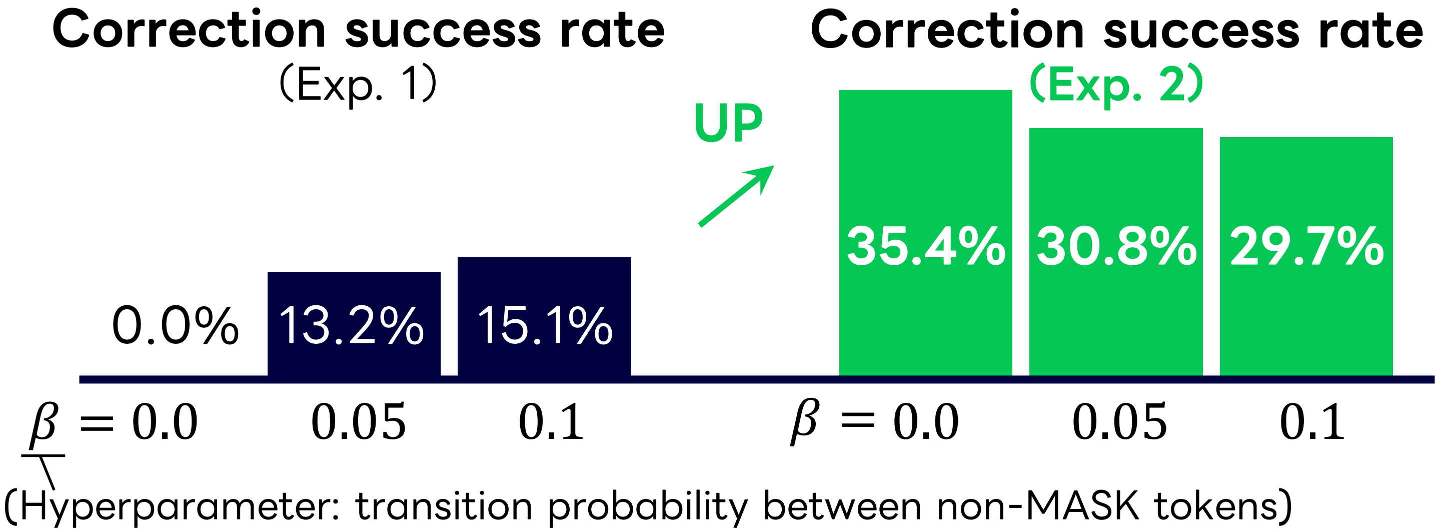
(Hyperparameter: transition probability between non-MASK tokens)

DDMs Perform Better When Reconstructing from MASK

Experiment 1 DDMs struggle to directly correct erroneous tokens

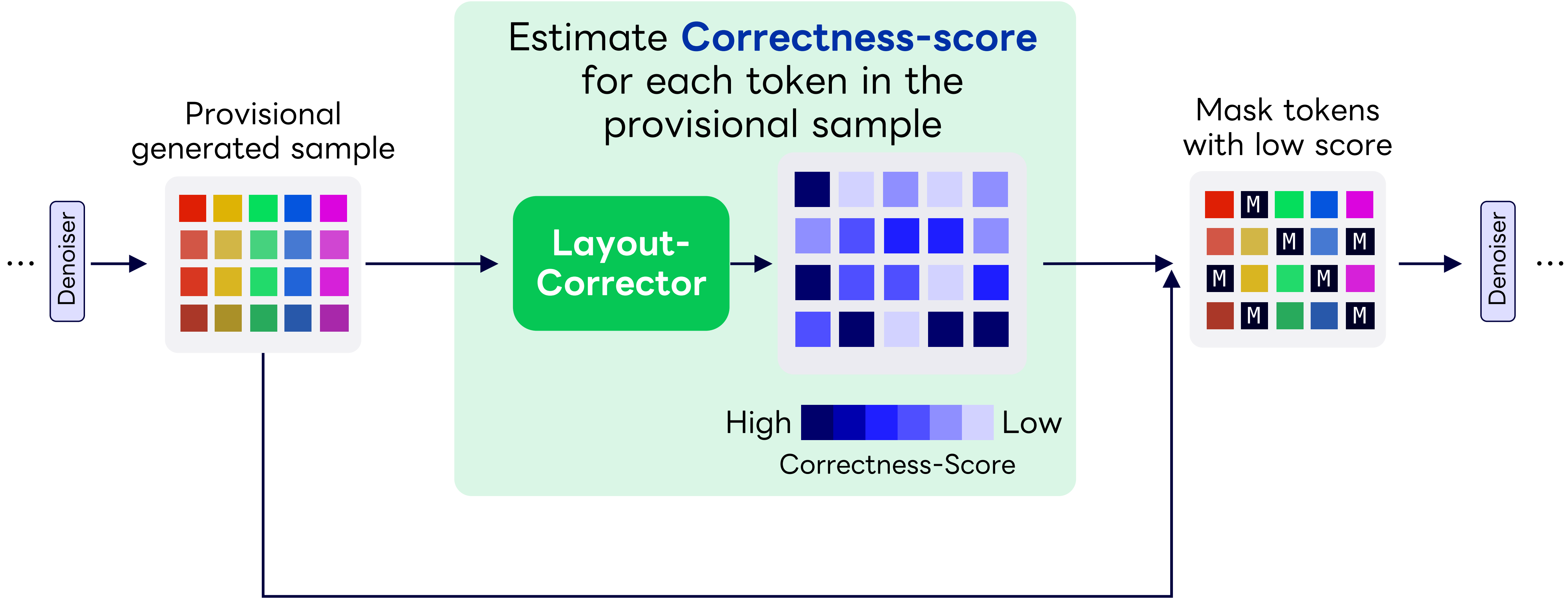
Experiment 2 If erroneous tokens are masked first, DDMs can correct them.

Propose Layout-Corrector,
which detects and masks erroneous tokens



Layout-Corrector: Overview

Detect erroneous tokens and mask them, prompting the DDM to correct them.



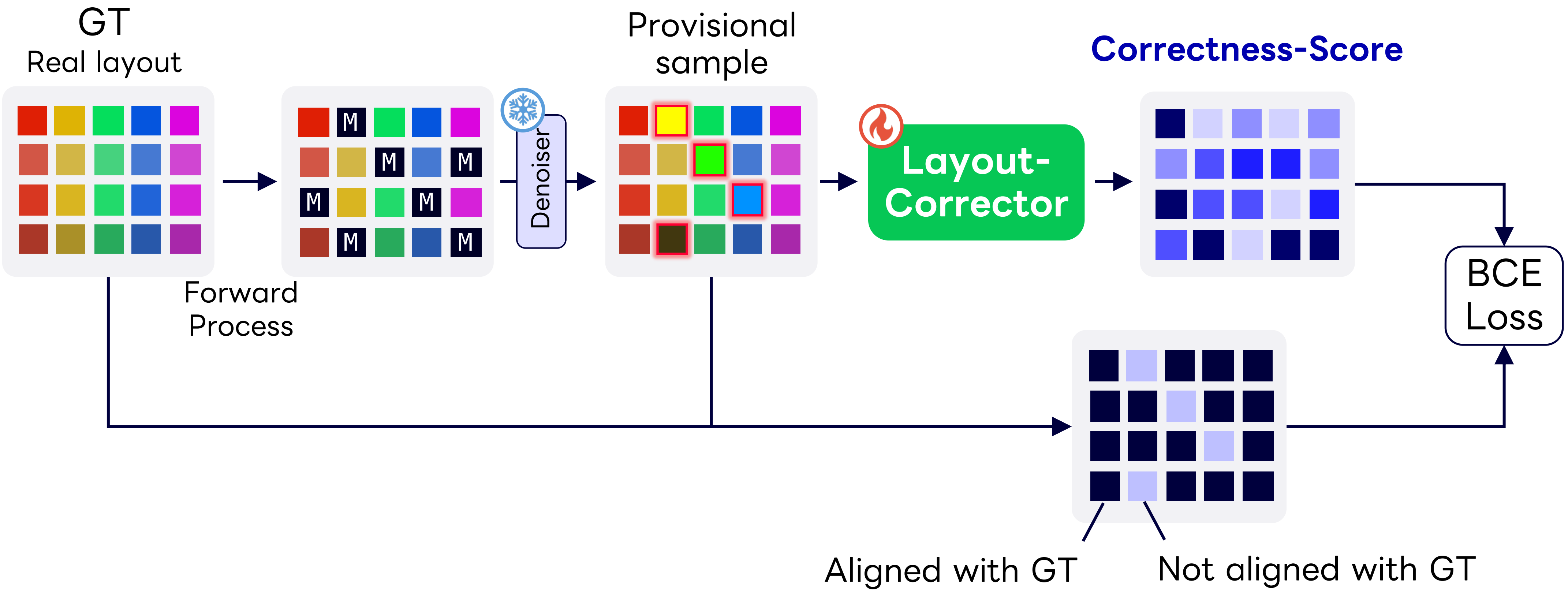
Provisional generated sample

Estimate **Correctness-score** for each token in the provisional sample

Mask tokens with low score

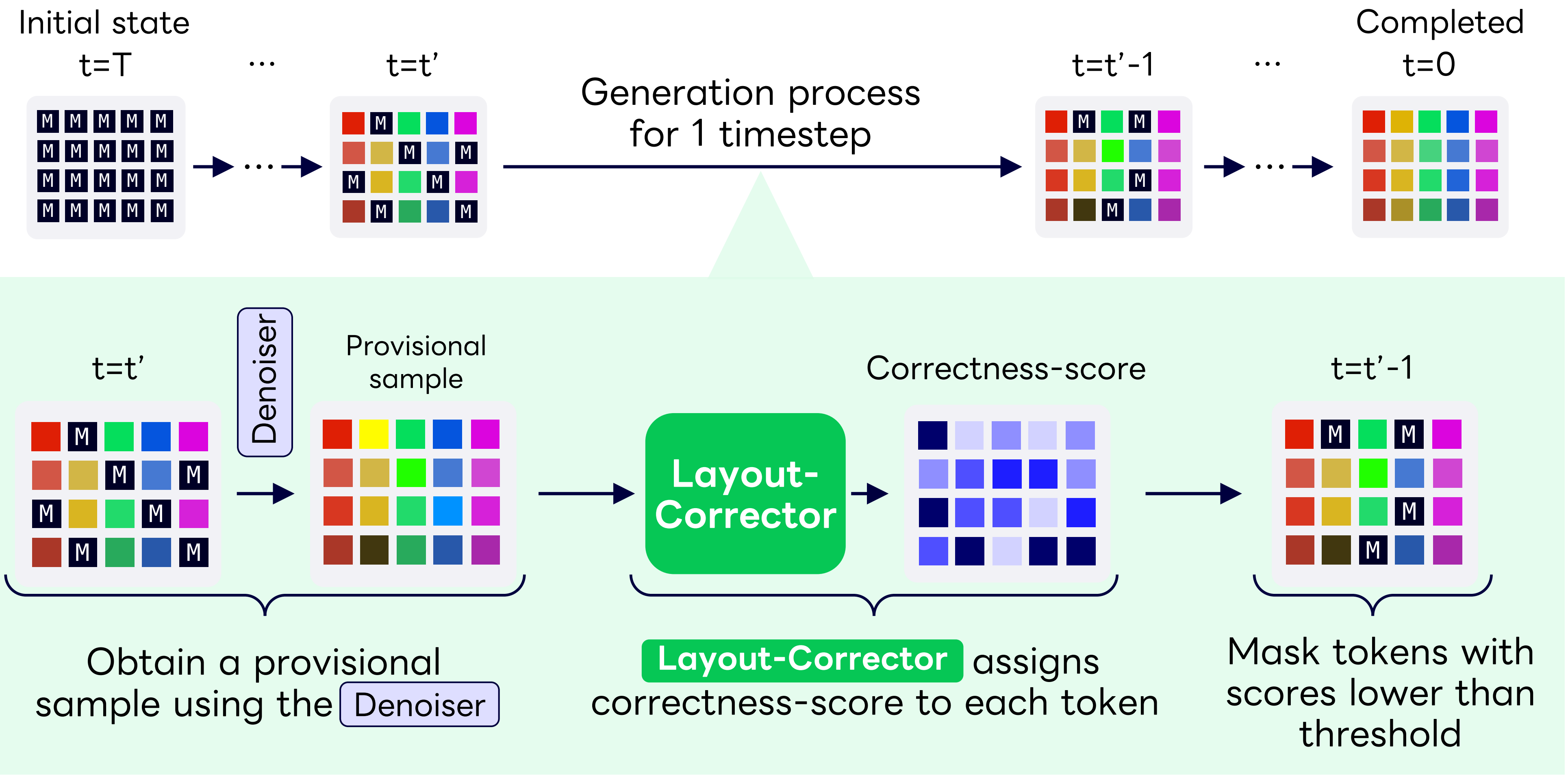
Layout-Corrector: Training

Train Layout-Corrector to predict whether each token in provisional sample with the corresponding ground-truth (real) layout.



→ Layout-Corrector learns to assign low score for tokens not aligning with GT (= erroneous tokens)

Layout-Corrector: Generating Process



Obtain a provisional sample using the Denoiser

Layout-Corrector assigns correctness-score to each token

Mask tokens with scores lower than threshold

We selectively apply Layout-Corrector at specific 3 timesteps out of T=100 timesteps

Layout-Corrector (LC) improved performance of multiple baselines

Quantitative Evaluation (FID↓)

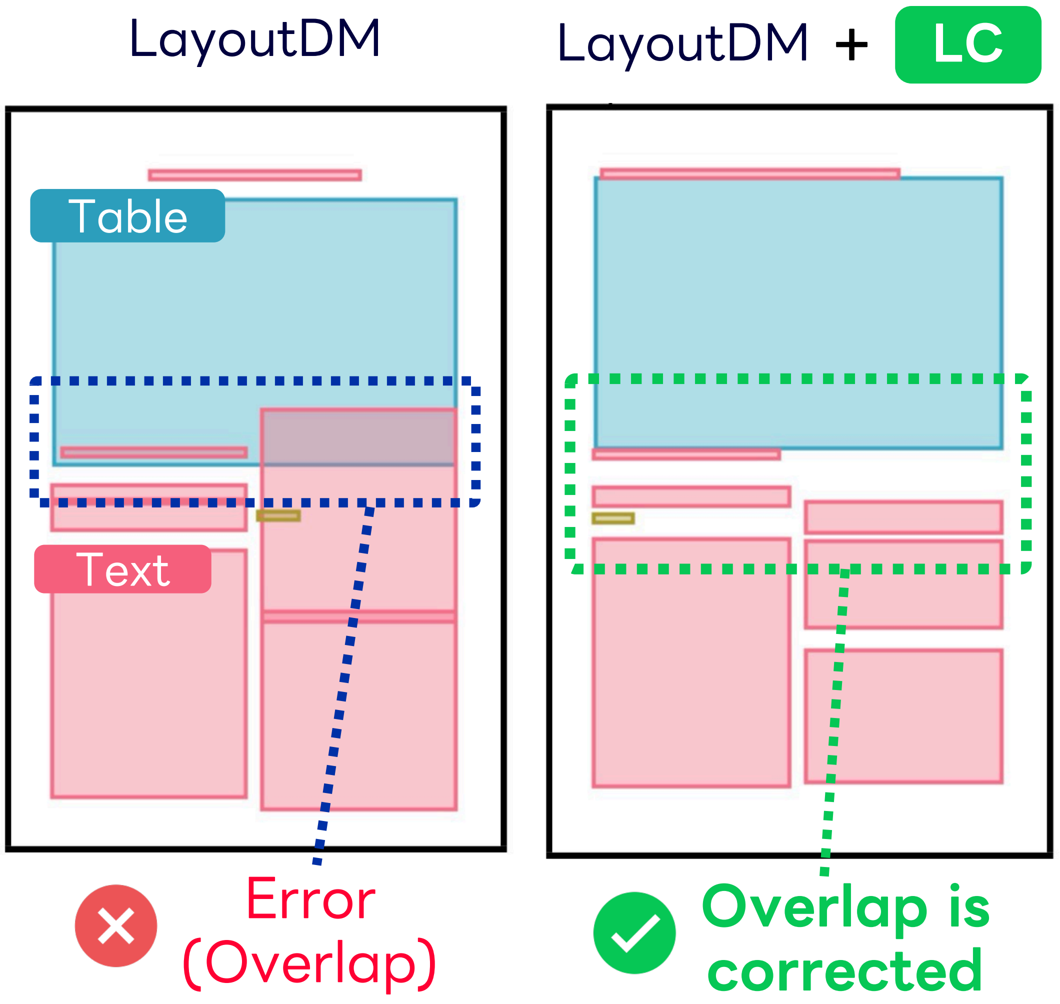
dataset:	Rico	PubLayNet
Baseline generation model 1		
MaskGIT only	70.4	34.2
MaskGIT + TC	15.7	17.6
MaskGIT + LC	**14.4**	**13.7**
Baseline generation model 2		
LayoutDM only	6.37	13.7
LayoutDM + TC	17.97	22.3
LayoutDM + LC	**4.79**	**11.9**

✓ **Improved performance of two baselines**

✓ **Outperformed existing method, TC** (Token-Critic)

TC (Token-Critic): [Lezama+, ECCV2022]
MaskGIT: [Chang+, CVPR2022]
LayoutDM: [Inoue+, CVPR2023]

Generated sample (PubLayNet dataset)



Conclusion

Summary

Goal

Correct generation error in layout-generation

Preliminary

DDM cannot directly correct erroneous tokens

DDM can correct erroneous tokens if we could mask them

Method

Layout-Corrector detects erroneous tokens and mask them

Experiment

Improve multiple baseline layout-generation models

More results in our paper

Layout-Corrector can also:

✓ Improve conditional generation

✓ Control fidelity-diversity trade-off

✓ Mitigate performance drop caused by fast-sampling

✓ Evaluate discrepancy between generated and real layouts