Cyclic Discrete Diffusion: A Novel Multi-Class Segmentation Refinement Technique

Libo Chen
Department of Mathematical Sciences
University of Bath
Bath, United Kingdom
lc2762@bath.ac.uk

Souvik Ghosh
International Institute of Information Technology
Hyderabad, India
souvik.g@research.iiit.ac.in

Vinay P. Namboodiri
Department of Computer Science
University of Bath
Bath, United Kingdom
vpn22@bath.ac.uk

Abstract

We cast image segmentation refinement as a *discrete* diffusion process. To solve this task we introduce Cyclic Discrete Diffusion (CDD), a novel forward–reverse formulation with a cyclic label topology that maps a discrete label set to the uniform distribution and back. Instead of injecting Gaussian noise as is done for continuous diffusion, CDD evolves segmentation masks through label jumps: at each step, every pixel either remains in its current class or jumps to the next label with a fixed rate. This jump-or-stay mechanism defines a finite-state Markov process with bounded forward and reverse steps, featuring analytically consistent and tractable reverse dynamics. CDD operates natively in a multi-class label space, enabling joint refinement of all classes in a single pass without requiring class-wise decomposition, as commonly needed in prior binary refinement approaches. The proposed formulation admits a finite and controllable number of refinement steps, for which we analyze the convergence properties through spectral gap analysis. We evaluate CDD across diverse segmentation tasks, including multi-class semantic, instance-level, and binary segmentation refinement. Our method consistently improves both region accuracy and boundary quality across multiple backbones and datasets.

1 Introduction

Segmentation is a fundamental task in computer vision, providing pixel-level understanding of visual scenes and object structures Long et al. [2015], Ronneberger et al. [2015], Chen et al. [2017]. Despite substantial advances He et al. [2017], Ke et al. [2022], Kirillov et al. [2020], generating precise and coherent segmentation masks remains a challenge, especially in maintaining boundary precision and structural consistency across regions. The difficulty is particularly acute in high-resolution and multi-class scenes, where complex boundaries, overlapping semantics, and subtle visual transitions must be handled simultaneously. While general methods like Mask-RCNN are widely applicable and used, we argue that they are not precise enough, especially at high resolutions Peng et al. [2022], Yuan et al. [2020]. This observation has led to a number of methods that are focused on refinement of segmentation Yuan et al. [2020], Peng et al. [2022], Wu et al. [2025], Wang et al. [2023]. A number of these methods are specific to particular architectures or modalities- for instance BPR Peng et al.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

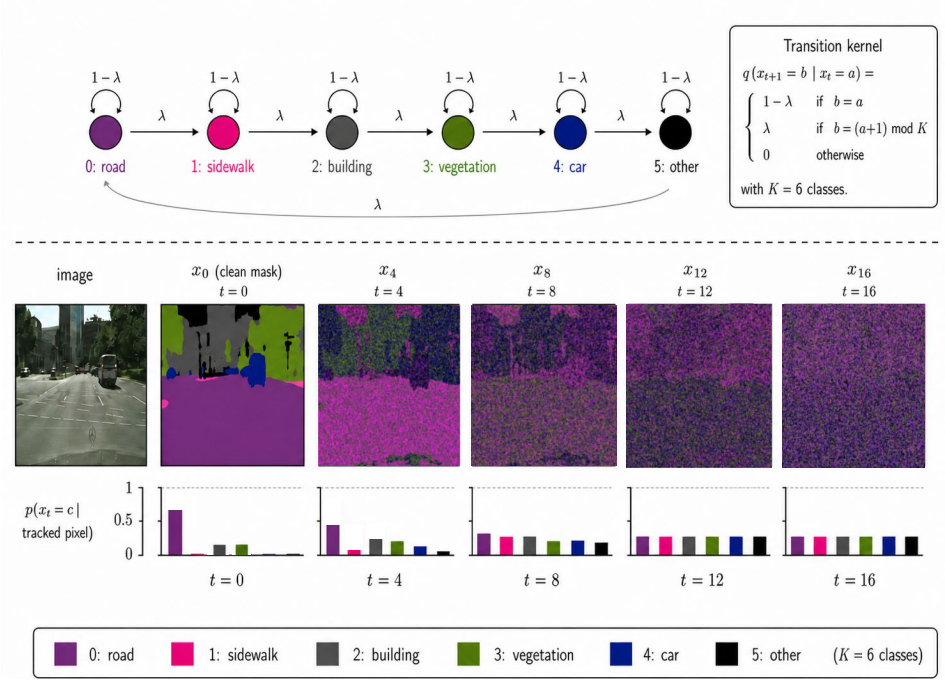

Figure 1: Method illustration of the proposed cyclic discrete diffusion (CDD). Each pixel label evolves under the cyclic transition kernel $q(x_{t+1} \mid x_t)$: it stays in the current class with probability $1 - \lambda$ or jumps to the next class (mod K) with probability λ. Starting from a clean mask (x_0), the forward process progressively corrupts the segmentation over time ($t = 0, 4, 8, 12, 16$), driving per-pixel class probabilities to an approximately uniform mixture (shown by the bars). Our model learns a reverse transition $p_\theta(x_{t-1} \mid x_t)$ to recover a refined mask.

[2022] is specific to Mask-RCNN while Wu et al. [2025] is specific to the medical domain. This limits their generalization and extensibility.

An aspect that has been recently studied in this context is model-agnostic segmentation refinement Yuan et al. [2020], Shen et al. [2022], Wang et al. [2023]. This task enables us to refine the outputs of any pretrained segmentation network, making it widely applicable and effective in practice. Among these, SegRefiner Wang et al. [2023], is closest to our work. They show that this problem can be solved using a conditional discrete diffusion framework, enabling gradual refinement of coarse predictions into fine-grained masks.
While this approach makes an interesting contribution, we identify that it could result in practical challenges for usage. For instance, typically one uses multi-class semantic segmentation. In order to use SegRefiner as designed, one has to do *binary segmentation refinement* by taking each class into account and refining the mask. This poses a significant practical challenge. To address this challenge, we pose a novel discrete diffusion based *multi-class segmentation refinement* approach that solves the problem in a single pass.

Our design follows a Markov jump process termed a cyclic discrete diffusion (CDD). To enable us to solve for multi-class refinement, we follow this jump process over the label space. In our formulation, each pixel follows a jump rule at every time step, either jumping to the next discrete state with a fixed probability or remaining in its current state. This process is illustrated in fig. 1. To ensure stability in discrete space, we further define a cyclic state connection, allowing transitions to wrap around the label space. This cyclic construction guarantees that the process converges to a uniform stationary distribution, establishing a mathematically consistent and well-posed Markov chain. Intuitively, the jump-or-stay mechanism captures the stochastic evolution of segmentation boundaries, where each pixel probabilistically decides whether to update its class assignment, aligning closely with the iterative refinement nature of segmentation. Second, CDD provides an analytically tractable forward process with a finite number of explicitly controllable steps. We further derive a theoretical

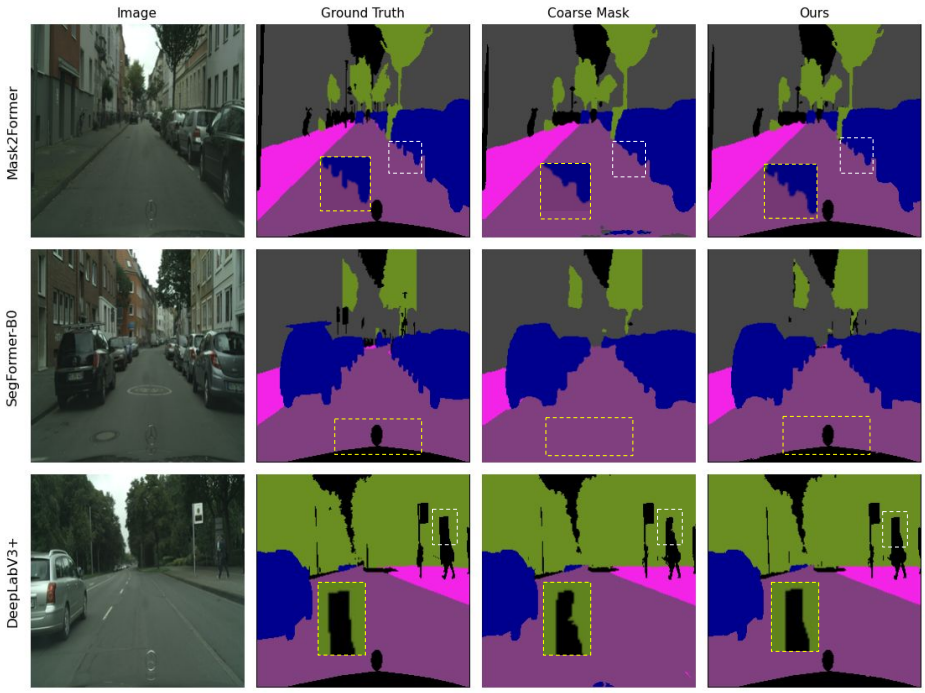

Figure 2: Qualitative comparison of multi-class segmentation refinement on Cityscapes. For each backbone, we show the input image, ground truth, coarse mask, and our CDD-refined result. CDD consistently improves coarse predictions from different segmentation models, yielding cleaner boundaries and better local structural consistency. Highlighted regions show representative boundary and fine-structure corrections.

analysis to calculate the number of refinement steps, offering principled guidance for step selection and ensuring efficient convergence. Together, these properties form a principled bridge between discrete probabilistic modeling and real-world segmentation refinement.

We extensively evaluate CDD across diverse segmentation tasks and backbones. We first evaluate CDD on multi-class semantic segmentation benchmarks, where it performs joint refinement across all classes in a single pass, leading to substantial improvements in both IoU and boundary quality. These results highlight the advantage of modeling segmentation refinement directly in a multi-class discrete space. To enable direct comparison with prior work such as SegRefiner, which is designed for binary mask refinement, we additionally evaluate CDD in the binary setting on standard benchmarks. On the COCO validation set with LVIS annotations Lin et al. [2014], Gupta et al. [2019], CDD consistently improves over strong refinement baselines and outperforms SegRefiner across multiple backbones in both Mask AP and Boundary AP (Tab. 3, Tab. 4). These results confirm the effectiveness, generality, and strong boundary precision of our principled multi-class discrete diffusion framework for segmentation refinement.

Contributions: This work makes the following novel contributions:

- We generalize the idea of discrete diffusion to multi-class refinement in a single pass providing significant practical benefits using a principled framework of jumps in a Markov chain.
- We propose a novel transition matrix thereby extending the framework of D3PM Austin et al. [2021]. Our transition probabilities are derived based on the Markov Jump Process. We provide extensive results in Section 4 that demonstrate the efficacy of our method.
- We also provide a theoretical analysis that guides the estimation of the number of refinement steps and provides evidence of convergence based on the spectral gap analysis.

2 Related Work

2.1 Segmentation Refinement

Segmentation refinement aims to enhance coarse or low-quality masks by improving boundary precision and recovering fine-grained structures. Early methods often used probabilistic graphical models, such as fully connected CRFs Krähenbühl and Koltun [2011], to enforce local smoothness and align region boundaries with image edges. Deep networks like FCNs Long et al. [2015] later integrated refinement into end-to-end learning, though traditional CRF-based post-processing operates independently of deep features and has limited adaptability to complex visual patterns. With the rise of deep learning, refinement has evolved into a learnable and task-specific process. Representative works, such as the Deep Watershed Transform Bai and Urtasun [2017] and Boundary-Aware Instance Segmentation Hayder et al. [2017], train networks to predict truncated distance or boundary maps within cropped instances, achieving more precise delineation of object contours. PointRend Kirillov et al. [2020] formulated segmentation as adaptive point-based rendering, focusing computational effort on uncertain regions. There are some model-agnostic approaches Tang et al. [2021], Zhou et al. [2020], Cheng et al. [2020], Shen et al. [2022]. While these methods achieve strong empirical gains and remain efficient in practice, they are generally designed around specific segmentation pipelines and rely on deterministic updates, limiting their adaptability to multi-class or highly ambiguous regions. Several more general methods have been proposed. For instance, SegFix Yuan et al. [2020] proposed a model-agnostic boundary correction framework that explicitly predicts pixel-wise displacement vectors to repair misaligned edges. More recently, SegRefiner Wang et al. [2023] unified mask-level refinement across diverse instance segmentation models, progressively enhancing boundary details through an iterative UNet-like architecture. We show that our method outperforms these more recent general works using a principled approach.

2.2 Diffusion Models in Continuous and Discrete Spaces

Continuous diffusion models have achieved remarkable success in image synthesis and other continuous domains by parameterizing the gradient of the data distribution through score matching Song and Ermon [2019], Ho et al. [2020], Song et al. [2020]. These models define a forward process that gradually perturbs data with Gaussian noise and a reverse process learned to denoise samples step by step, resulting in high-quality and diverse generations. However, their reliance on continuous noise makes them inherently ill-suited for discrete structures such as segmentation labels, where transitions between categorical states cannot be modeled by infinitesimal Gaussian perturbations. To address this issue, discrete diffusion models Austin et al. [2021], Campbell et al. [2022], Sun et al. [2022], Lou et al. [2023] reformulate the forward–reverse dynamics as a Markov process, introducing categorical corruption and discrete transition matrices. Although these methods extend the diffusion paradigm to token- or label-based data, they often suffer from unstable training objectives and limited scalability. For example, mean prediction formulations Austin et al. [2021] and ratio-matching losses Sun et al. [2022] provide theoretical consistency but yield suboptimal empirical performance, while more recent score-entropy formulations Lou et al. [2023] improve stability at the cost of additional computational complexity. We further validate that, for segmentation refinement, Segrefiner Wang et al. [2023] that follows Austin et al. [2021] indeed demonstrates suboptimal performance. RefineCatDiff Liu et al. [2025] also employs categorical diffusion for medical segmentation refinement. Unlike general categorical transitions, CDD uses a cyclic jump-or-stay kernel with only two admissible predecessor states, yielding a tractable reverse posterior. Our work builds upon this line of research by introducing a CDD that defines a finite, cyclic, and interpretable transition structure in discrete space. Unlike prior discrete diffusion models, CDD incorporates a jump-or-stay stochastic rule governed by a controllable rate parameter, ensuring both convergence to a uniform stationary distribution and coherent coarse-to-fine refinement.

2.3 Preliminaries: Diffusion Models (Continuous and Discrete)

In continuous diffusion model, the data are gradually transformed into noise through a forward stochastic differential equation (SDE) Song et al. [2020]. x_0 denote a data sample drawn from the data distribution, and x_t its perturbed version at time t. The forward SDE is defined as

$$dx_t = f(x_t, t)\, dt + g(t)\, dw_t, \tag{1}$$

where f and g are smooth drift and diffusion coefficients, and w_t denotes standard Brownian motion. As $t \to T$, the distribution of x_t converges to a prior noise distribution, typically Gaussian.

The goal of the reverse process is to invert this stochastic dynamics and recover the original data distribution from noise. The corresponding reverse SDE is given by Anderson [1982]

$$dx_t = \big(f(x_t, t) - g^2(t)\,\nabla \log p(x_t, t)\big)\,dt + g(t)\,d\overline{w}_t, \tag{2}$$

where $\nabla \log p(x_t, t)$ is the score function, i.e., the gradient of the log-density at time t, and $\overline{w}_t$ is a Brownian motion adapted to the reverse-time filtration.

To extend diffusion modeling to categorical data, Austin et al. [2021] proposed the Structured Denoising Diffusion Model (D3PM), which generalizes continuous diffusion to finite discrete state spaces. Instead of adding Gaussian noise, the forward process applies random state transitions governed by a Markov transition matrix $Q_t \in \mathbb{R}^{K \times K}$:

$$q(x_t \mid x_{t-1}) = \mathrm{Cat}(x_t; p = x_{t-1}Q_t), \tag{3}$$

where $\mathrm{Cat}(x_t; p)$ denotes a *categorical distribution* over K discrete states with class probabilities, $x_t \in \{1, 2, \ldots, K\}$ represents a categorical variable at time t. This process progressively corrupts the original data by repeatedly sampling from Q_t, driving the distribution of x_t toward a predefined stationary distribution.

The reverse process is trained to approximate the true posterior $q(x_{t-1} \mid x_t, x_0)$, which follows directly from the Markov chain and can be expressed analytically via Bayes' rule. A neural network $f_\theta(x_t, t)$ predicts the reverse transition probabilities $p_\theta(x_{t-1} \mid x_t)$, and the model is optimized via a variational lower bound equivalent to a per-step cross-entropy objective.

3 Our Method: Cyclic Discrete Diffusion

In this work, we propose CDD, a novel discrete diffusion–based framework inspired by the dynamics of cyclic (clock-like) jumps. Unlike continuous Gaussian diffusion models that operate in Euclidean space by adding Gaussian perturbations, CDD evolves over a finite discrete label space through stochastic state transitions. Each pixel either jumps to its neighboring class or remains unchanged, according to a fixed transition probability, forming a cyclic Markov process over semantic labels. This process defines a discrete-time Markov chain that progressively mixes the categorical states and converges to a uniform terminal distribution. In the forward process, CDD gradually perturbs the ground-truth mask into a highly randomized mask by repeatedly applying these local jumps. In the reverse process, the model starts from a noisy mask conditioned on a coarse input and reconstructs the clean segmentation by learning the reverse transition probabilities. The following sections detail the forward and reverse formulations and present the overall learning objective.

3.1 Discrete Forward Process

The forward process in CDD progressively perturbs the ground-truth segmentation mask into a noisy one through a sequence of discrete stochastic jumps. Let $\mathcal{X} = \{0, 1, \ldots, K-1\}$ denote the finite set of semantic labels, and let $x_t \in \mathcal{X}$ be the class of a pixel at timestep t. At each step, the pixel either *jumps* to its next class or *remains* in the current state. Formally, the transition kernel is defined as

$$q(x_{t+1} = b \mid x_t = a) = \begin{cases} 1 - \lambda, & b = a, \\ \lambda, & b = (a+1) \bmod K, \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $\lambda \in (0, 1)$ is the jump rate. The use of the modulo operator "$\bmod K$" plays a key role in this design. It makes the label transitions cyclic: when a pixel in the last class $K-1$ performs a jump, it wraps around to the first class 0 instead of stopping at the boundary. Unlike an open random walk that terminates at the edge, this closed-loop construction ensures that all labels are mutually reachable. This forward design leads to a Markov chain that can not only traverse the entire label space, but also converge to a uniform stationary distribution. Let u denote the stationary probability of being in label x after an infinite number of jumps. Then, for the proposed cyclic process, all labels share the same probability

$$u(x) = \tfrac{1}{K}, \quad \forall x \in \mathcal{X}. \tag{5}$$

The circular design is particularly suited for segmentation tasks, which provides both theoretical tractability and practical suitability for segmentation refinement. Unlike Gaussian perturbations in continuous space, which can blur class boundaries, our discrete cyclic dynamics operate directly on categorical labels, preserving semantic consistency while enabling controllable degeneracy.

As t increases, the structured mask is gradually randomized through many jumps, eventually converging to a uniform categorical distribution. Let π_t be the current marginal distribution at time t. Formally, the total variation distance between π_t and the stationary distribution decays exponentially as

$$\|\pi_t - u\|_{\mathrm{TV}} \leq Ce^{-\gamma t}, \tag{6}$$

where the effective exponential decay rate of the cyclic transition kernel, for $0 < \lambda < 1$, is

$$\gamma = -\frac{1}{2}\log\left[1 - 2\lambda(1-\lambda)\left(1 - \cos\frac{2\pi}{K}\right)\right]. \tag{7}$$

The decay rate γ depends only on the jump rate λ and the number of classes K. This closed-form expression guarantees fast mixing and ensures that the process reaches near-uniformity within a small number of iterations, approximately $t_{\mathrm{mix}}(\epsilon) \approx \frac{1}{\gamma}\log(1/\epsilon)$ (see Appendix A.5 for derivation and empirical validation).

3.2 Discrete Reverse Process

The goal of the reverse process is to invert the forward jump dynamics defined in (4) and recover a structured segmentation mask from noise. The forward transition in (4) follows a "jump-or-stay" rule, Formally, if a pixel is in state x_t at step t, it can only come from two possible previous states: x_t (no jump) or $x_t - 1$ (one cyclic jump). [1] This discreteness makes the reverse transition analytically tractable. Let $\pi_{t-1}(x \mid Y)$ denote the marginal distribution of the forward process at step $t-1$ for label x, optionally conditioned on auxiliary information Y (e.g., coarse images). Intuitively, $\pi_{t-1}(x \mid Y)$ quantifies the likelihood that a pixel takes label x just before the next jump in the forward direction.

For a given target label x_t, the total incoming probability mass to x_t from step $t-1$ is

$$D_t(x_t \mid Y) = \lambda\,\pi_{t-1}(x_t{-}1 \mid Y) + (1-\lambda)\,\pi_{t-1}(x_t \mid Y), \tag{8}$$

where the two terms respectively correspond to (i) arriving at x_t via a jump from x_t-1 with rate λ, and (ii) remaining in x_t with probability $1-\lambda$.

By Bayes' rule, the reverse transition kernel can then be expressed in closed form as the posterior over these two admissible predecessors:

$$q(x_{t-1} = x_t{-}1 \mid x_t, Y) = \frac{\lambda\,\pi_{t-1}(x_t{-}1 \mid Y)}{D_t(x_t \mid Y)}, \quad q(x_{t-1} = x_t \mid x_t, Y) = \frac{(1-\lambda)\,\pi_{t-1}(x_t \mid Y)}{D_t(x_t \mid Y)}. \tag{9}$$

Equation (9) gives the exact reverse transition probabilities of the forward jump process in (4). It specifies, for each pixel and timestep t, how likely we are to "undo" the last jump or to "stay" when stepping backward from x_t to x_{t-1}.

3.3 Training Objectives

Given the image I, coarse mask Y, noisy mask x_t, and timestep t, the network predicts per-pixel categorical probabilities $p_\theta(x_0 \mid x_t, I, Y)$. The RGB image, one-hot coarse mask, and one-hot noisy mask are concatenated along the channel dimension, while the coarse mask remains fixed throughout reverse sampling. Following Wang et al. [2023], we use a reconstruction loss and a texture-preserving regularization, and further add a CDD posterior matching term. The overall objective is

$$\mathcal{L} = \alpha_{\mathrm{CE}}\mathcal{L}_{\text{CE-X0}} + \alpha_{\mathrm{tex}}\mathcal{L}_{\mathrm{texture}} + \alpha_{\mathrm{KL}}\mathcal{L}_{\mathrm{KL}}. \tag{10}$$

The reconstruction loss is a weighted cross-entropy:

$$\mathcal{L}_{\text{CE-X0}} = -\mathbb{E}_{q(x_0,t,x_t)}\left[\frac{1}{|\Omega|}\sum_{u\in\Omega} w(u)\log p_\theta(x_0(u) \mid x_t, I, Y)\right], \tag{11}$$

[1] All label operations are computed modulo K, consistent with the forward process.

where $w(u) = 4$ if $Y(u) \neq x_0(u)$ and $w(u) = 1$ otherwise.

The texture loss is computed class-wise between the softmax probabilities P_θ and the one-hot ground-truth mask G:

$$\mathcal{L}_{\text{texture}} = \|S(P_\theta) - S(G)\|, \tag{12}$$

where $S(\cdot)$ denotes Sobel gradient magnitudes after 3×3 mean-filter smoothing.

To enforce consistency with the analytical CDD reverse dynamics, we use a posterior matching loss. Since one reverse step only allows $x_{t-1} \in \{x_t, (x_t - 1) \bmod K\}$, the true posterior $q(x_{t-1} \mid x_t, x_0)$ is a two-point distribution. Following the x_0-parameterization of discrete diffusion models Austin et al. [2021], we induce the reverse posterior by marginalizing over the predicted clean label:

$$p_\theta(x_{t-1} \mid x_t, I, Y) \propto \sum_{\tilde{x}_0=0}^{K-1} q(x_{t-1} \mid x_t, \tilde{x}_0)\, p_\theta(\tilde{x}_0 \mid x_t, I, Y), \tag{13}$$

and minimize

$$\mathcal{L}_{\text{KL}} = \mathbb{E}_{x_0 \sim p_{\text{data}},\, t \sim \text{Unif}(\{1,\ldots,T\}),\, x_t \sim q(\cdot \mid x_0, t)} \left[D_{\text{KL}} \left(q(x_{t-1} \mid x_t, x_0) \,\|\, p_\theta(x_{t-1} \mid x_t, I, Y) \right) \right]. \tag{14}$$

We set $\alpha_{\text{CE}} = 1$, $\alpha_{\text{KL}} = 10^{-3}$, and $\alpha_{\text{tex}} = 5$. During training, half of the samples use x_t generated by the CDD forward process from x_0, while the other half use $x_t = Y$ and $t = T$, so that the model directly observes the real coarse-mask distribution used at inference time. The ablation in Table 8 shows that the 50/50 training strategy maintains strong performance under both single-pass and iterative inference.

4 Segmentation Refinement Experiments

4.1 Multi-Class Semantic Segmentation Refinement

We evaluate multi-class segmentation refinement on the Cityscapes dataset Cordts et al. [2016]. Cityscapes is a standard benchmark for urban-scene semantic segmentation, containing complex street scenes with multiple interacting object categories and fine boundary structures. The training split is used for model optimization, and all results are reported on the validation split. In our multi-class refinement setting, we map the original semantic labels into six categories: road, sidewalk, building, vegetation, car, and background/others. This six-class setting allows us to test whether CDD can handle multi-class label refinement, while avoiding the excessive class imbalance and optimization difficulty of the full Cityscapes label space. Ground-truth semantic masks serve as the refinement targets, and coarse masks are produced by different segmentation backbones, including Mask2Former Cheng et al. [2022], SegFormer-B0 Xie et al. [2021], and DeepLabV3+ Chen et al. [2018]. All masks are resized to 256×256 for both training and evaluation.

For CDD, we set the number of classes to $K = 6$, the number of diffusion steps to $T = 16$(see Appendix A.5) , and the jump rate to $\lambda = 0.3$. The model is trained with the objective in Eq. (10), using Adam with a learning rate of 10^{-4} and a batch size of 8. Training is run for at most 120 epochs with early stopping, where the patience is set to 15 epochs and the relative improvement threshold on the training loss is 10^{-3}. We report Intersection-over-Union (IoU) Everingham et al. [2010] for region-level segmentation quality and BFScore Perazzi et al. [2016] for boundary accuracy.

4.1.1 Quantitative Results

Table 1 evaluates whether CDD can consistently improve coarse masks from different segmentation backbones. Averaged over Mask2Former, SegFormer-B0, and DeepLabV3+, our method improves IoU by 11.04 points and BFScore by 17.09 points. Across all three backbones, the gain in BFScore is larger than the gain in IoU. This suggests that the improvement is not only due to correcting large semantic regions, but also comes from better boundary alignment. This behavior can be explained by the loss design. The texture-preserving loss explicitly constrains spatial gradients of the predicted mask, encouraging sharper transitions at object boundaries, while the weighted CE loss emphasizes regions where the coarse mask disagrees with the ground truth, which are typically concentrated around boundaries. Figure 2 provides qualitative visualization results. We further evaluate zero-shot refinement on stronger backbones, SegNeXt-L Guo et al. [2022] and SegMAN-L Fu et al. [2025],

Table 1: Multi-class segmentation refinement on Cityscapes. We report IoU and BFScore (%) across different backbones. For zero-shot evaluation, CDD is trained only on Mask2Former coarse masks without retraining.

Coarse model	**IoU (%)**			**BFScore (%)**		
	Coarse	Ours	Δ	Coarse	Ours	Δ
Original evaluation backbones						
Mask2Former	73.00	**84.55**	+11.55	63.46	**82.20**	+18.74
SegFormer-B0	71.17	**82.47**	+11.30	60.44	**78.93**	+18.49
DeepLabV3+	75.41	**85.69**	+10.28	70.32	**84.35**	+14.03
Average (above three)	73.19	84.24	+11.04	64.74	81.83	+17.09
Zero-shot transfer (unseen backbones)						
SegNeXt-L	77.27	**85.09**	+7.82	76.90	**85.42**	+8.52
SegMAN-L	78.04	**85.47**	+7.43	76.11	**85.07**	+8.96

Table 2: Comparison of multi-class segmentation refinement methods on Cityscapes using Mask2Former coarse masks. We report IoU and BFScore (%) and their improvements over the coarse baseline.

Method	IoU	ΔIoU	BFScore	ΔBF
Coarse mask	73.00	–	63.46	–
CNN post-processing	77.75	+4.75	75.14	+11.68
D3PM	72.80	-0.20	64.56	+1.10
SegRefiner	78.48	+5.48	71.35	+7.89
CDD (Ours)	**84.55**	**+11.55**	**82.20**	**+18.74**

using CDD trained only on Mask2Former coarse masks. As shown in Table 1, CDD consistently improves both IoU and BFScore, demonstrating its transferability to unseen backbones.

Table 2 compares CDD with representative refinement baselines on the Cityscapes six-class setting, using Mask2Former coarse masks as input. To assess the contribution of diffusion-based refinement, we include a non-diffusion CNN post-processing baseline that directly refines the coarse mask using the RGB image. As shown in Table 2, CDD outperforms this baseline by 6.80 IoU and 7.06 BFScore points under single-pass inference, demonstrating its effectiveness beyond conventional CNN-based refinement. D3PM Austin et al. [2021] serves as a generic discrete diffusion baseline for multi-class refinement. In its standard form, D3PM performs conditional refinement by starting from a noisy state sampled from the terminal distribution and conditioning on the coarse mask during reverse sampling, but it yields only marginal improvements over the coarse baseline. SegRefiner Wang et al. [2023] is originally designed for binary mask refinement, so we adapt it to our six-class setting by training one model per class and merging the class-wise predictions into a multi-class mask. Although SegRefiner improves over the coarse mask, its gains are smaller than those of CDD, suggesting that binary refinement strategies do not directly transfer optimally to multi-class semantic refinement. CDD obtains the best IoU and BFScore among all compared methods. We further analyze the two discrete diffusion processes in the Appendix A.1, where CDD shows clearer advantages in multi-step reverse sampling from uniform noisy masks, with improved reverse-process stability and boundary preservation.

4.2 Binary Mask Refinement

Although our main contribution is native multi-class segmentation refinement, the existing diffusion-based segmentation refinement method Wang et al. [2023] is designed and evaluated in the binary mask setting. Following SegRefiner Wang et al. [2023], we evaluate our method on three representative benchmarks that cover both instance-level and pixel-level segmentation scenarios: COCO Lin et al. [2014], LVIS Gupta et al. [2019], and DIS5K Qin et al. [2022](see appendix(A.2.1)). COCO provides large-scale and diverse everyday scenes for general instance segmentation, while LVIS focuses on fine-grained, large-vocabulary segmentation with challenging boundary details. DIS5K contains

high-resolution images with delicate object contours and serves as a strong benchmark for evaluating fine-grained segmentation refinement. For all datasets, we follow the same experimental protocol as SegRefiner Wang et al. [2023], including the official training and validation splits, coarse predictions, training losses, and evaluation metrics, to ensure a fair comparison. We evaluate model performance using IoU, mBA(mean Boundary Accuracy) Cheng et al. [2020], Mask Average Precision (Mask AP), and Boundary Average Precision (Boundary AP) Cheng et al. [2021]. Mask AP measures region-level accuracy based on the IoU overlap between predicted and ground-truth masks, whereas Boundary AP focuses on contour alignment within a narrow band around object edges, making it particularly sensitive to fine-grained boundary details and small structures.

4.2.1 Quantitative Results

Table 3: Comparison with refinement baselines on the COCO val set using LVIS annotations.

Method	Mask AP						Boundary AP					
	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN-R50	39.8	61.4	42.3	24.9	47.0	55.1	27.3	53.3	25.2	24.9	41.8	27.4
+ SegFix Yuan et al. [2020]	40.6	61.4	42.8	25.0	48.4	56.6	29.1	53.7	28.0	24.9	43.6	30.8
+ BPR Tang et al. [2021]	41.0	61.4	43.1	24.8	48.5	57.8	30.4	55.2	29.5	24.7	43.8	33.7
+ SegRefiner Wang et al. [2023]	41.9	61.6	43.2	25.7	49.4	58.8	32.6	55.7	**32.5**	25.6	45.0	37.3
+ Ours	**42.4**	**61.9**	**44.3**	**25.9**	**50.1**	**59.1**	**32.6**	**56.2**	32.2	**25.8**	**45.7**	**37.5**
Mask R-CNN-R101	41.6	63.3	44.4	26.5	49.5	57.8	29.0	55.2	26.7	26.3	44.4	29.8
+ SegFix Yuan et al. [2020]	42.2	63.4	44.7	26.5	50.9	59.1	30.6	56.1	30.0	26.3	46.0	33.1
+ BPR Tang et al. [2021]	42.8	63.3	45.3	26.1	51.0	60.6	32.0	57.3	31.6	25.9	46.3	36.3
+ SegRefiner Wang et al. [2023]	43.6	63.3	45.2	27.4	51.4	61.6	34.1	57.2	34.9	27.2	47.1	**39.9**
+ Ours	**44.2**	**63.7**	**46.7**	**27.4**	**52.5**	**61.6**	**34.3**	**58.3**	**35.0**	**27.2**	**48.3**	39.3

Instance-level Refinement on COCO We evaluate CDD on the COCO validation set with LVIS annotations. For comparison, we select the three representative model-agnostic refinement baselines: SegFix Yuan et al. [2020], BPR Tang et al. [2021], and SegRefiner Wang et al. [2023]. These methods are designed to refine the coarse masks produced by segmentation models without retraining the underlying detectors, forming a fair basis for evaluating the refinement quality of our CDD. Following their setup, the coarse masks used in Table 3 are obtained from Mask R-CNN to ensure consistency with the original experiments in Tang et al. [2021], Yuan et al. [2020]. As shown in Table 3, CDD achieves the strongest overall performance across both Mask R-CNN backbones. Compared with SegRefiner, which already provides a strong refinement baseline, CDD further improves most Mask AP and Boundary AP metrics, indicating that the improvements are consistent across both region-level mask quality and boundary alignment.

Cross-model Generalization on LVIS To further evaluate transferability, we apply CDD as a plug-and-play refinement module to 7 representative instance segmentation models. As shown in Table 4, our method consistently improves performance across all baselines. For example, CDD achieves a gain of up to +3.6 AP and +6.5 Boundary AP on SOLO, and significantly boosts all model's Boundary AP_L by +9.4 on CondInst, demonstrating its strong capability in refining large-object boundaries. Compared with SegRefiner, our approach maintains superior results across all models, confirming that the proposed cyclic discrete diffusion mechanism can be effectively integrated into diverse segmentation frameworks. The underlying jump-and-stay mechanism enables more accurate recovery of fine boundary details, leading to higher segmentation precision and sharper mask contours. Qualitative comparisons in Fig. 11 further illustrate these improvements, where CDD refines complex shapes and restores more consistent object boundaries than existing refinement methods.

5 Conclusion and Discussion

We presented Cyclic Discrete Diffusion (CDD), a principled discrete diffusion framework for segmentation refinement. Unlike prior binary refinement approaches, CDD operates directly in a multi-class label space and performs joint refinement of all classes in a single pass. The proposed cyclic jump process defines a structured Markov transition over discrete labels, yielding a mathematically tractable forward process with a finite number of controllable refinement steps and a uniform distribution. We further derived the corresponding reverse dynamics and provided a spectral-gap-based analysis of convergence. Experiments across semantic, instance-level, and binary refinement benchmarks demonstrate that CDD consistently improves both region accuracy and boundary quality across

Table 4: Comparison of baseline models and our Cyclic Discrete Diffusion refinement ("+ Ours") on the LVIS val set (%). green numbers indicate improvements over the base model.

Method	Mask AP (%)				Boundary AP (%)			
	AP	AP_S	AP_M	AP_L	AP	AP_S	AP_M	AP_L
CondInst Tian et al. [2020]	39.8	24.3	47.5	54.8	29.2	24.1	42.6	30.2
+ SegRefiner	41.1	25.5	48.8	56.8	32.2	25.3	44.5	36.1
+ Ours	**41.8** (+2.0)	**26.2** (+1.9)	**50.0** (+2.5)	**56.4** (+1.6)	**33.2** (+4.0)	**24.6** (+0.5)	**45.4** (+2.8)	**39.6** (+9.4)
SOLO Wang et al. [2020]	37.4	19.3	45.5	56.6	24.7	19.0	39.3	27.8
+ SegRefiner	40.5	21.7	49.3	60.9	31.3	21.4	44.8	39.8
+ Ours	**41.0** (+3.6)	**22.4** (+3.1)	**49.7** (+4.2)	**60.2** (+3.6)	**31.2** (+6.5)	**22.2** (+3.2)	**45.2** (+5.9)	**37.9** (+10.1)
PointRend Kirillov et al. [2020]	41.5	25.1	49.0	59.3	30.6	25.0	44.2	34.1
+ SegRefiner	42.8	25.9	50.4	61.3	33.7	25.8	46.2	40.1
+ Ours	**43.0** (+1.5)	**25.7** (+0.6)	**50.8** (+1.8)	**60.9** (+1.6)	**33.7** (+3.1)	**25.7** (+0.7)	**46.5** (+2.3)	**39.2** (+5.1)
QueryInst Fang et al. [2021]	42.4	26.7	49.5	61.4	29.9	26.4	44.5	32.3
+ SegRefiner	44.2	27.5	51.4	64.7	34.9	27.3	47.2	41.6
+ Ours	**45.1** (+2.7)	**27.9** (+1.2)	**52.2** (+2.7)	**65.0** (+3.6)	**35.5** (+5.6)	**28.1** (+1.7)	**48.1** (+3.6)	**41.7** (+9.4)
Mask Transfiner Ke et al. [2022]	43.2	25.9	49.0	60.1	31.6	25.8	44.5	35.8
+ SegRefiner	43.3	26.8	49.9	62.0	34.4	26.6	45.9	41.3
+ Ours	**44.5** (+1.3)	**27.9** (+2.0)	**52.3** (+3.3)	**63.0** (+2.9)	**35.1** (+3.5)	**27.8** (+2.0)	**48.1** (+3.6)	**41.3** (+5.5)
Mask2Former Cheng et al. [2022]	46.8	27.9	54.9	69.1	37.0	27.8	50.1	44.6
+ SegRefiner	47.4	28.5	55.3	69.8	38.8	28.4	51.0	48.5
+ Ours	**48.1** (+1.3)	**29.0** (+1.1)	**56.3** (+1.4)	**70.1** (+1.0)	**39.1** (+2.1)	**28.7** (+0.9)	**51.9** (+1.8)	**48.9** (+4.3)
RefineMask Wang et al. [2020]	41.2	24.0	48.1	59.2	30.5	23.8	43.5	34.1
+ SegRefiner	41.9	24.6	48.7	60.7	33.0	24.5	44.7	39.4
+ Ours	**42.4** (+1.2)	**24.6** (+0.6)	**49.6** (+1.5)	**61.0** (+1.8)	**33.2** (+2.7)	**24.6** (+0.8)	**45.4** (+1.9)	**39.6** (+5.5)

diverse segmentation backbones. Comparisons against existing discrete diffusion formulations further suggest that the structured cyclic transition provides a suitable inductive bias for segmentation refinement by preserving partial semantic structure during corruption and reverse denoising. While our primary focus is segmentation refinement, the proposed framework also opens directions for broader discrete generative modeling in structured label spaces. Future work may explore scaling the framework to larger semantic datasets.

Computational costs and limitations. CDD requires only 1.15 GPU-hours for training (120 epochs, NVIDIA H200), compared with 1.20 GPU-hours for a matched feed-forward U-Net. Single-pass inference takes 16.5 ms per image on an RTX A5000, compared with 12.1 s for SegRefiner under its full refinement protocol on the same GPU. These results demonstrate the computational efficiency of CDD, although its scalability to higher resolutions and larger label spaces remains to be investigated.

Acknowledgments and Disclosure of Funding

Funding. Libo Chen acknowledges support from the University of Bath through the Vice-Chancellor's International Scholarship. Computational resources for this work were provided by the Department of Computer Science at the University of Bath. No dedicated project funding was received for this research.

Competing Interests. The authors declare no competing interests.

References

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5221–5229, 2017.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15334–15342, 2021.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.

Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8890–8899, 2020.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6910–6919, 2021.

Yunxiang Fu, Meng Lou, and Yizhou Yu. Segman: Omni-scale context modeling with state space models and local attention for semantic segmentation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19077–19087. IEEE, 2025.

Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zhengning Liu, Ming-Ming Cheng, and Shi-Min Hu. Segnext: Rethinking convolutional attention design for semantic segmentation. *Advances in neural information processing systems*, 35:1140–1156, 2022.

Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019.

Zeeshan Hayder, Xuming He, and Mathieu Salzmann. Boundary-aware instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5696–5704, 2017.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Lei Ke, Martin Danelljan, Xia Li, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Mask transfiner for high-quality instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4412–4421, 2022.

Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9799–9808, 2020.

Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, 24, 2011.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

Feng Liu, Hongjian Yu, Jiaan Huang, Xin Hua, Hengjia Liu, Zhimian Ma, Zhen Wang, Zhijiang Du, and Lining Sun. Refinecatdiff: Toward high-quality medical image segmentation via a categorical diffusion refinement framework. *Advanced Intelligent Systems*, 7(12):e202401125, 2025.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Juncai Peng, Yi Liu, Shiyu Tang, Yuying Hao, Lutao Chu, Guowei Chen, Zewu Wu, Zeyu Chen, Zhiliang Yu, Yuning Du, et al. Pp-liteseg: A superior real-time semantic segmentation model. *arXiv preprint arXiv:2204.02681*, 2022.

Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016.

Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. Highly accurate dichotomous image segmentation. In *European Conference on Computer Vision*, pages 38–56. Springer, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Tiancheng Shen, Yuechen Zhang, Lu Qi, Jason Kuen, Xingyu Xie, Jianlong Wu, Zhe Lin, and Jiaya Jia. High quality segmentation for ultra high-resolution images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1310–1319, 2022.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. *arXiv preprint arXiv:2211.16750*, 2022.

Chufeng Tang, Hang Chen, Xiao Li, Jianmin Li, Zhaoxiang Zhang, and Xiaolin Hu. Look closer to segment better: Boundary patch refinement for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13926–13935, 2021.

Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *European conference on computer vision*, pages 282–298. Springer, 2020.

Mengyu Wang, Henghui Ding, Jun Hao Liew, Jiajun Liu, Yao Zhao, and Yunchao Wei. Segrefiner: Towards model-agnostic segmentation refinement with discrete diffusion process. *arXiv preprint arXiv:2312.12425*, 2023.

Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *European conference on computer vision*, pages 649–665. Springer, 2020.

Junde Wu, Ziyue Wang, Mingxuan Hong, Wei Ji, Huazhu Fu, Yanwu Xu, Min Xu, and Yueming Jin. Medical sam adapter: Adapting segment anything model for medical image segmentation. *Medical image analysis*, 102:103547, 2025.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

Yuhui Yuan, Jingyi Xie, Xilin Chen, and Jingdong Wang. Segfix: Model-agnostic boundary refinement for segmentation. In *European conference on computer vision*, pages 489–506. Springer, 2020.

Peng Zhou, Brian Price, Scott Cohen, Gregg Wilensky, and Larry S Davis. Deepstrip: High-resolution boundary refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10558–10567, 2020.

A Appendix

A.1 Multi-Class Refinement

Multi-step reverse sampling We additionally evaluate the full reverse diffusion process and include a strengthened D3PM variant, denoted as D3PM (modified), which adopts the same mixed training inputs as CDD, combining noisy masks and coarse masks during training. Starting from a random noisy mask sampled from a uniform distribution, we iteratively apply each method's learned reverse transition from $t = T$ to $t = 1$. At each step, each pixel is updated by selecting the state with the highest predicted reverse probability. The coarse mask is used only as conditioning input during sampling. This setting evaluates the model's ability to generate refined masks from noise via iterative denoising. Table 5 shows that CDD provides more stable multi-step reverse refinement than D3PM, particularly at early training stages. At 50 epochs, D3PM exhibits less stable reverse sampling from noise, whereas CDD still produces better masks. As training progresses, both methods improve substantially at 100 epochs. However, CDD consistently achieves higher IoU and BFScore, indicating more reliable performance during iterative reverse sampling. Experiments were conducted on NVIDIA RTX 4090 GPUs with 24GB memory.

We further visualize the forward and reverse diffusion trajectories of CDD and D3PM in Fig. 3. CDD corrupts masks through local cyclic transitions, which preserve partial spatial layouts and semantic structures along the forward trajectory. This structured degradation path makes the reverse process more interpretable, as semantic regions and boundaries are progressively recovered from noise. By contrast, D3PM rapidly approaches near-uniform noise, weakening the correspondence between intermediate states and the underlying semantic layout and leading to less structured reverse trajectories. These observations suggest that CDD provides a structured local corruption process that preserves intermediate semantic structure, which is useful for segmentation refinement where spatial continuity and boundary preservation are essential.

Table 5: Appendix comparison between D3PM and CDD under multi-step reverse refinement at different training epochs. Best results among refinement methods are in **bold**.

Method	Epochs	IoU	BFScore	ΔIoU / ΔBF
Coarse mask	–	73.00	63.46	– / –
D3PM(modified)	50	13.30	26.39	-59.70 / -37.07
CDD	50	**62.50**	**57.65**	**-10.50 / -5.82**
D3PM(modified)	100	81.78	77.52	+8.78 / +14.06
CDD	100	**82.10**	**78.94**	**+9.10 / +15.48**

Effect of label ordering. Since CDD defines a cyclic transition over the discrete label space, the order of semantic labels may influence the structure of the forward corruption process. We therefore test a less natural label order, where semantically related classes are separated in the cycle, For

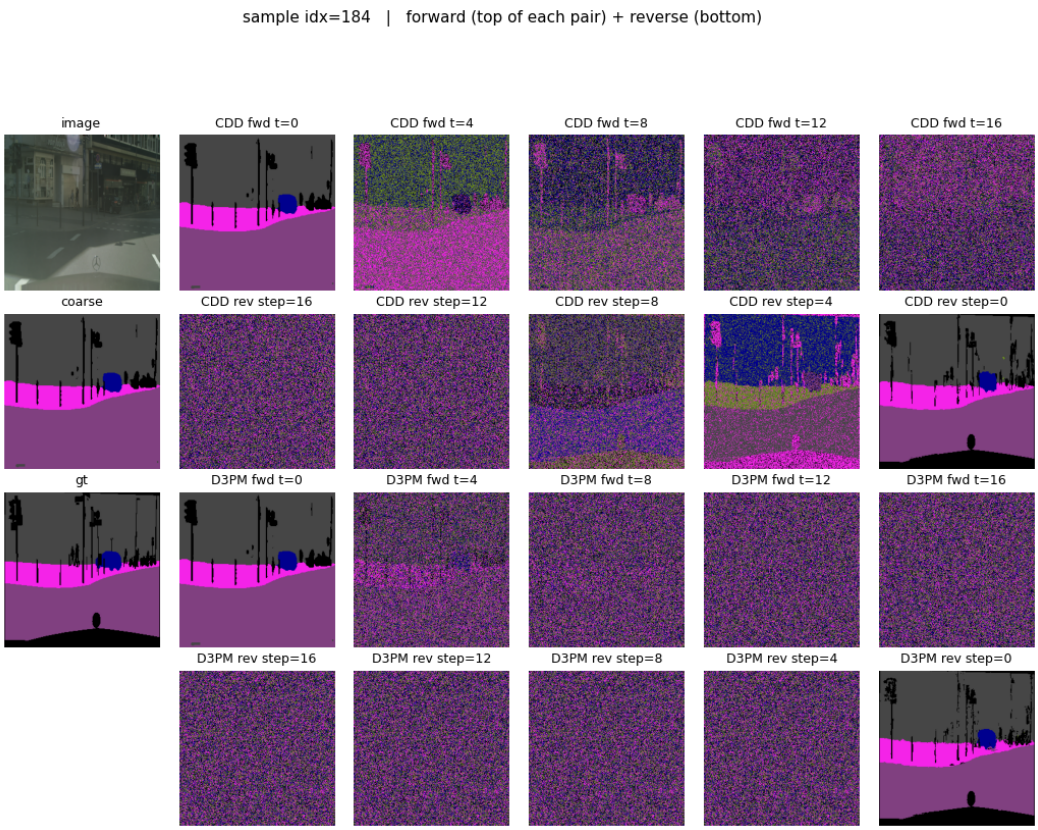

Figure 3: Visualization of forward and reverse dynamics for CDD and D3PM. The left column shows the reference images: input image, coarse mask, and ground-truth mask, while the fourth row is left empty. The first row shows the CDD forward process applied to the coarse mask at $t = 0, 4, 8, 12, 16$. The second row shows snapshots of the CDD reverse sampling trajectory at steps $16, 12, 8, 4, 0$. The third row shows the D3PM forward corruption process at the same timesteps. The fourth row shows snapshots of the D3PM reverse sampling trajectory at the corresponding reverse steps. Both methods are visualized using identical timesteps and sampling schedules.

example, road is followed by vegetation rather than sidewalk. This makes the cyclic transition less semantically meaningful.

Table 6 shows that CDD achieves nearly identical performance under the default and bad label orders. This suggests that although a semantically meaningful order provides a more interpretable transition structure, the learned refinement model is robust to moderate changes in label ordering.

Table 6: Ablation on label ordering using Mask2Former coarse masks. Best results are in **bold**.

Label order	IoU	BFScore	ΔIoU	ΔBF
Coarse mask	73.00	63.46	–	–
Original order	**84.55**	82.20	**+11.55**	+18.74
Bad order	84.52	**82.23**	+11.52	**+18.76**

Per-class performance. Table 7 reports the per-class IoU on Cityscapes. CDD improves all six categories, with the largest gain observed in the heterogeneous "other" class.

Stability across random seeds. Across three independent training runs (seeds 0, 1, 2), CDD achieves 84.52 ± 0.10 IoU and 82.15 ± 0.11 BFScore, indicating stable refinement performance.

Table 7: Per-class IoU (%) on Cityscapes using Mask2Former coarse masks.

Method	mIoU	Road	Sidewalk	Building	Vegetation	Car	Other
Coarse	73.00	80.96	71.42	76.09	83.14	87.37	39.01
CDD	**84.55**	**93.07**	**73.67**	**85.34**	**88.45**	**91.52**	**75.24**

Effect of training mixing ratio. We evaluate the proportion of real coarse masks used during training, with the remaining samples generated by the CDD forward process. As shown in Table 8, training exclusively on CDD noise severely degrades single-pass inference, while training exclusively on coarse masks reduces iterative refinement performance. The 50/50 strategy achieves the best balance across both inference modes.

Table 8: Effect of the training mixing ratio on Cityscapes using Mask2Former coarse masks. Results are reported as IoU/BFScore (%).

Real coarse ratio	Single-pass	Iterative ($T = 16$)
0%	29.01 / 34.69	75.55 / 69.97
50%	**84.55 / 82.20**	**84.23 / 81.85**
100%	84.53 / 82.18	81.31 / 76.60

A.2 Binary Refinement

Implementation Details Our CDD refiner adopts the same UNet backbone and feature pyramid structure as SegRefiner Wang et al. [2023], ensuring that performance differences arise solely from the diffusion mechanism. For binary segmentation refinement, we set the number of diffusion steps to T=6 and the jump rate to λ=0.3, following the theoretical bound derived in Appendix A.5.

A.2.1 Fine-grained Boundary Refinement on DIS5K

We evaluate CDD on the high-resolution DIS5K benchmark to assess fine-grained boundary accuracy. DIS5K is a recently introduced dataset, and to our knowledge, SegRefiner Wang et al. [2023] is the only method that has reported results on it so far. The objective of this experiment is to assess how well our approach generalizes across different backbone architectures and to benchmark it fairly against SegRefiner. As reported in Table 9, CDD achieves higher IoU and mBA compared to segrefiner across multiple backbone architectures, demonstrating superior capability in preserving fine object contours and boundary details. These results establish CDD as a strong and general model-agnostic refinement framework for both instance- and pixel-level segmentation tasks. Qualitative results in Fig. 10 further illustrate that CDD recovers fine object structures with sharper contours and cleaner shapes than SegRefiner, particularly on thin and intricate regions.

Table 9: Comparison between Ground Truth, SegRefiner Wang et al. [2023], and our CDD on DIS5K. Mean IoU (%) and mean Boundary Accuracy (mBA, %) are reported for the validation (VD) and four test sets (TE1–TE4). Green numbers indicate the improvement of Ours over SegRefiner.

Dataset	Metric	U-Net			PFNet			PSPNet			ICNet			HRNet			ISNet		
		GT	SegR	Ours	GT	SegR	Ours	GT	SegR	Ours	GT	SegR	Ours	GT	SegR	Ours	GT	SegR	Ours
DIS-VD	IoU	54.8	58.7	**59.9**(+1.2)	56.2	60.4	**61.3**(+0.9)	56.4	60.4	**63.0**(+2.6)	56.5	61.4	**61.7**(+0.3)	61.0	64.4	**64.1**(-0.3)	67.1	68.3	**67.3**(-1.0)
	mBA	69.8	75.4	**78.0**(+2.6)	63.7	74.8	**77.6**(+2.8)	62.8	74.8	**77.3**(+2.5)	66.5	75.6	**77.9**(+2.3)	68.9	76.6	**78.6**(+2.0)	74.1	76.6	**78.4**(+1.8)
DIS-TE1	IoU	44.1	47.1	**51.9**(+4.8)	47.0	49.0	**53.8**(+4.7)	47.6	49.0	**55.3**(+6.3)	46.2	49.7	**53.7**(+4.0)	51.0	53.0	**55.6**(+2.7)	56.2	57.0	**59.1**(+2.1)
	mBA	70.1	75.0	**78.8**(+3.8)	65.1	74.7	**78.3**(+3.6)	64.4	74.7	**78.2**(+3.6)	67.2	75.0	**78.6**(+3.6)	69.3	75.6	**78.9**(+3.3)	73.6	75.5	**78.5**(+3.1)
DIS-TE2	IoU	54.4	58.1	**62.4**(+4.3)	57.0	60.3	**64.8**(+4.6)	57.7	60.3	**66.3**(+6.0)	56.5	60.3	**64.1**(+3.8)	61.2	64.7	**66.1**(+1.4)	67.0	67.8	**68.6**(+0.7)
	mBA	69.9	75.6	**79.0**(+3.4)	64.6	75.1	**78.6**(+3.6)	63.7	75.1	**78.4**(+3.3)	66.8	75.6	**78.8**(+3.2)	69.5	76.6	**79.2**(+2.6)	74.0	76.4	**78.0**(+1.6)
DIS-TE3	IoU	59.3	63.3	**66.1**(+2.8)	59.5	63.4	**67.4**(+4.0)	59.9	63.4	**67.8**(+4.4)	59.8	64.2	**67.1**(+2.9)	64.4	67.6	**69.2**(+1.6)	69.9	70.9	**71.7**(+0.8)
	mBA	70.6	76.7	**79.3**(+2.6)	64.0	76.1	**78.7**(+2.6)	62.7	76.1	**78.5**(+2.4)	66.7	76.6	**79.0**(+2.4)	69.5	77.6	**79.6**(+2.0)	74.6	77.4	**79.2**(+1.8)
DIS-TE4	IoU	61.1	66.3	**67.2**(+0.9)	59.1	66.6	**67.7**(+1.1)	57.6	66.6	**68.4**(+1.8)	60.5	67.0	**68.0**(+1.0)	64.7	70.5	**68.8**(-1.7)	70.1	72.2	**70.9**(-1.3)
	mBA	70.7	77.1	**78.1**(+1.0)	62.8	76.4	**77.6**(+1.2)	61.9	76.4	**77.2**(+0.8)	66.4	77.1	**77.9**(+0.8)	68.1	77.9	**78.4**(+0.6)	74.3	77.7	**78.1**(+0.5)
Average	IoU	54.9	58.7	**61.5**(+2.8)	56.0	60.0	**63.0**(+3.0)	56.1	60.0	**64.1**(+4.1)	56.7	60.5	**62.9**(+2.4)	61.1	64.0	**64.8**(+0.7)	67.0	67.2	**67.5**(+0.3)
	mBA	70.2	76.0	**78.6**(+2.6)	64.9	75.6	**78.2**(+2.6)	64.1	75.6	**77.9**(+2.3)	66.6	76.1	**78.4**(+2.3)	69.3	76.9	**78.9**(+2.0)	74.5	76.7	**78.4**(+1.7)

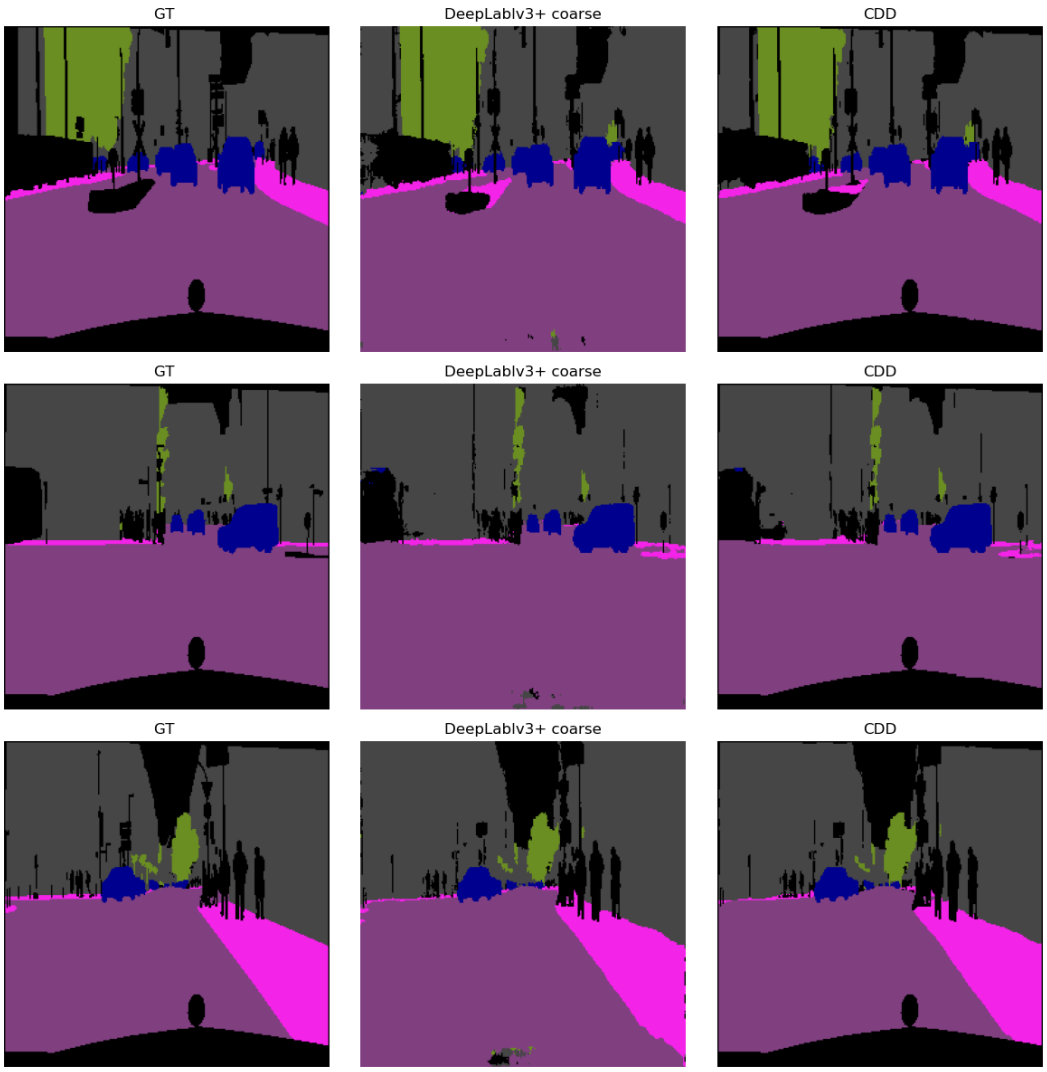

Figure 4: Visualization of multi-class segmentation refinement on Cityscapes validation images. Each row shows the ground-truth mask, the DeepLabv3+ coarse prediction, and the CDD-refined output. CDD corrects local misclassifications and sharpens object boundaries while preserving the overall semantic structure of the coarse mask.

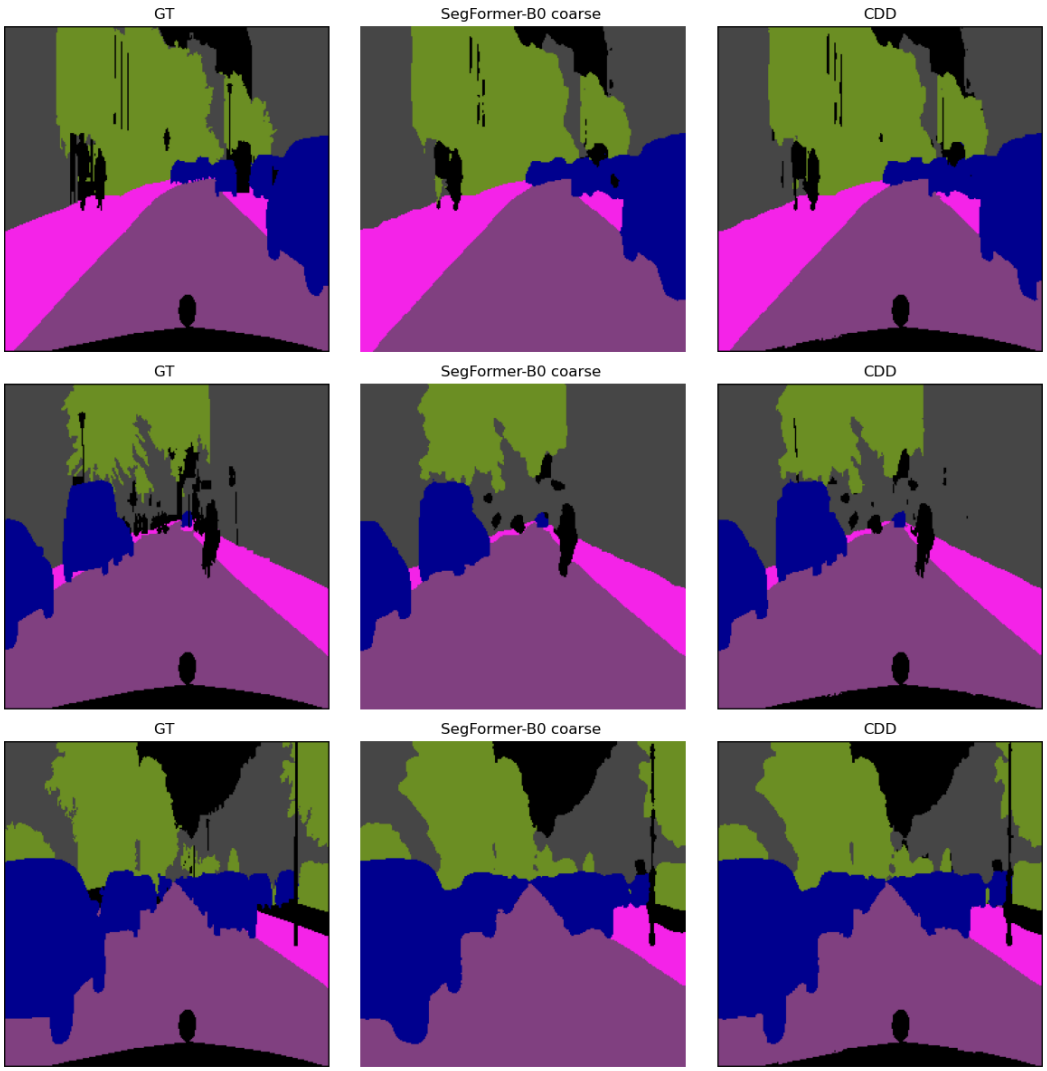

Figure 5: Qualitative visualization of CDD refinement using SegFormer-B0 coarse masks on Cityscapes validation images. From left to right, each row shows the ground-truth segmentation, the SegFormer-B0 coarse prediction, and the CDD-refined result.

A.3 Effectiveness of KL Loss

KL Loss (posterior matching). We train f_θ to match the model-induced reverse transition probabilities to the analytical posterior derived from the CDD forward process. Since the reverse transition at each pixel has only two admissible predecessors, namely staying at the current label x_t or moving one step backward to $x_t - 1$ in the cyclic chain, the model predicts a binary reverse distribution:

$$p_\theta(x_{t-1} = x_t - 1 \mid x_t, I, Y) = \sigma(z_\theta), \qquad p_\theta(x_{t-1} = x_t \mid x_t, I, Y) = 1 - \sigma(z_\theta).$$

We then minimize

$$\mathcal{L}_{\mathrm{KL}} = \mathbb{E}_{x_0,t,x_t}\left[D_{\mathrm{KL}}\left(q(x_{t-1} \mid x_t, x_0) \,\|\, p_\theta(x_{t-1} \mid x_t, I, Y)\right)\right].$$

Equivalently, this can be written as

$$\mathcal{L}_{\mathrm{KL}} = \mathbb{E}_{x_0,t,x_t}\Big[q(x_{t-1} = x_t - 1 \mid x_t, x_0) \log \frac{q(x_{t-1} = x_t - 1 \mid x_t, x_0)}{p_\theta(x_{t-1} = x_t - 1 \mid x_t, I, Y)} + q(x_{t-1} = x_t \mid x_t, x_0) \log \frac{q(x_{t-1} = x_t \mid x_t, x_0)}{p_\theta(x_{t-1} = x_t \mid x_t, I, Y)}\Big].$$

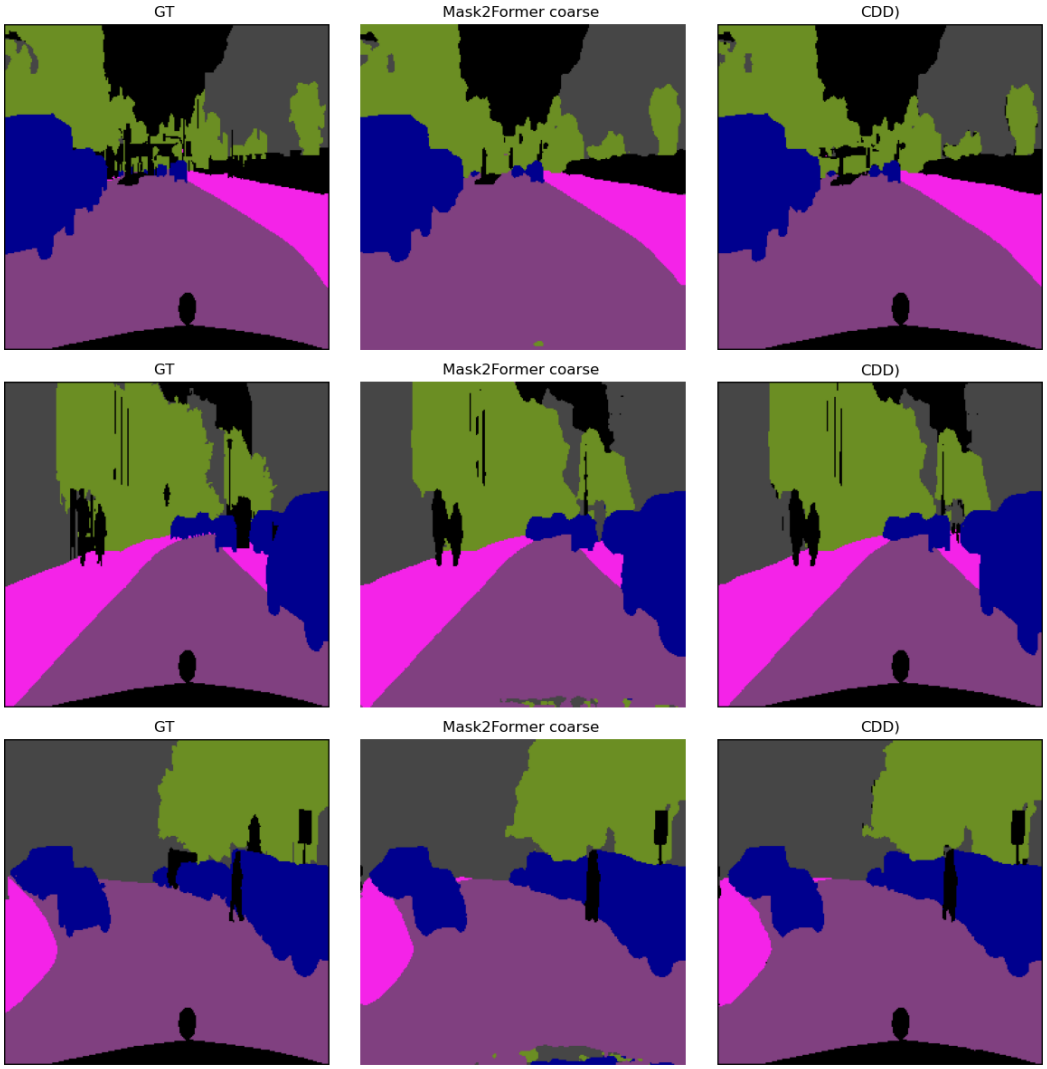

Figure 6: Qualitative visualization of CDD refinement using Mask2Former coarse masks on Cityscapes validation images. From left to right, each row shows the ground-truth segmentation, the Mask2Former coarse prediction, and the CDD-refined result.

For clarity, we distinguish between two training objectives:

(1) Original binary segmentation objective. In all binary refinement experiments on COCO, LVIS, and DIS5K reported in the main paper, we train f_θ to directly reconstruct the clean mask x_0 from an intermediate noisy state x_t using the standard cross-entropy loss. This setting is particularly effective for binary refinement.

(2) KL Loss. In the following section (toy experiments), we additionally consider a KL-augmented objective derived from the analytically known reverse transition of CDD. This objective regularises the model to match the true stay-or-jump probabilities. We empirically observe that this KL objective becomes more beneficial as the number of labels K increases, highlighting its potential for multi-class refinement.

Ablation on Original objective vs. KL Loss We evaluate two training objectives for our discrete diffusion model:

- **Original objective**: the standard cross-entropy loss for direct reconstruction.

• **Original objective+KL**: a KL-augmented objective that adds a KL term (0.001) to the CE loss.

This ablation clarifies why we include the additional KL loss in our multi-class experiments and shows that posterior matching is important for general multi-class segmentation refinement. For fairness, all ablations are performed under strictly controlled and reproducible conditions. We use a toy dataset of length 2000 containing 64×64 masks composed of synthetic shapes, and corrupt them using our forward process with a fixed jump rate $\lambda = 0.3$. The time index t for each corrupted sample is computed exactly using our analytical formula, ensuring that all models are trained under *identical forward tolerances*. This fully controlled setup eliminates confounders such as semantics and appearance variability, allowing Original objective and Original objective+KL to be compared purely in terms of their learning behavior. For each value of K, we train Original objective and Original objective+KL side-by-side using the same U-Net architecture, optimizer, and hyperparameters. As K increases, the reverse state space becomes more complex and optimization becomes more difficult.

Results in Table 10 show a clear trend. For small K, Original objective yields higher IoU, BFScore, and Boundary AP, confirming that direct supervision toward x_0 is a stronger learning signal for refinement, where the mapping from x_t to x_0 is nearly deterministic. However, as K grows, Original objective+KL becomes increasingly competitive and surpasses Original objective when $K \geq 4$, demonstrating that the KL component provides meaningful structural guidance when the reverse transition spans a richer discrete label space. In this paper, all our multi class refinement results are obtained using the Original objective+KL loss, since it provides the most stable and accurate supervision for multi class settings.

K	T	Loss	IoU ↑	BFScore ↑	Boundary AP ↑
2	4	**Original**	**0.9991**	**0.9915**	**0.9956**
		Original+KL	0.9989	0.9888	0.9943
4	8	Original	0.9961	0.9766	0.9878
		Original+KL	**0.9962**	**0.9820**	**0.9907**
6	16	Original	0.9894	**0.9701**	**0.9844**
		Original+KL	**0.9905**	0.9686	0.9834
8	38	Original	**0.9866**	0.9489	0.9728
		Original+KL	0.9852	**0.9587**	**0.9782**
10	42	Original	0.9842	0.9461	0.9715
		Original+KL	**0.9855**	**0.9550**	**0.9762**

Table 10: Ablation of the KL posterior matching objective under different numbers of classes K. For each K, T denotes the number of diffusion steps. Original+KL denotes the original objective combined with the KL posterior matching loss.

A.4 Binary Segmentation Generation Example

We also perform a preliminary experiment on unconditional binary mask generation using the same model trained *without* coarse-mask guidance on DIS. This experiment is not part of our main contribution; it merely illustrates that the proposed CDD process can in principle be used as a standalone discrete generative model for segmentation masks. Quantitative results are reported in Table 11, and representative generations are shown in Figure 7. As can be noted, the results in the unconditional case are worse as compared to the conditional results in the (Table 9). For instance, in VD, we obtained Mean IoU of 0.673 and mean mBA of 0.784 which is significantly better than these results in Table 11.

A.5 Mixing rate

To make the mixing behavior explicit, we write the forward process in matrix form. For a single pixel, the Markov chain over the K labels in $\mathcal{X}$ is governed by the $K \times K$ transition matrix Π, whose

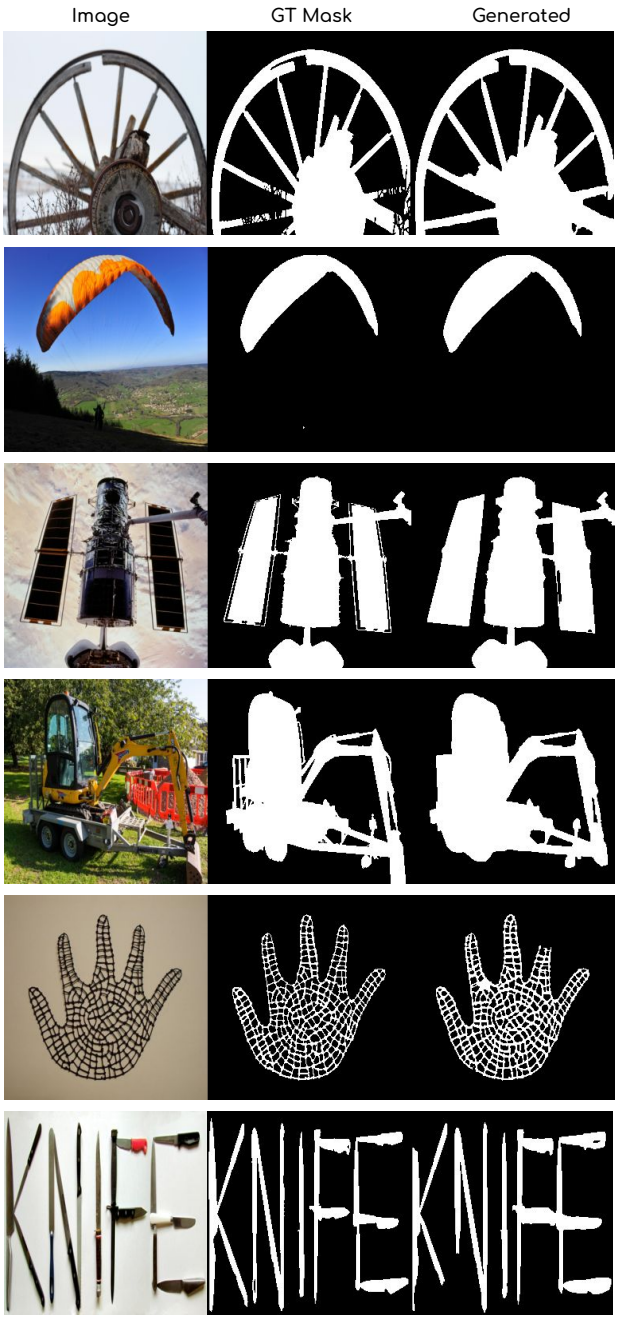

Figure 7: Results from our preliminary experiments on binary mask generation on the DIS5K dataset.

Dataset Split	Mean IoU	Mean mBA
VD	0.5107	0.6201
TE1	0.4677	0.6200
TE2	0.5450	0.6299
TE3	0.5692	0.6250

Table 11: Binary Generation Results on DIS5K.

(a, b)-th entry is $q(x_{t+1} = b \mid x_t = a)$ from Eq. (4). Concretely,

$$\Pi = \begin{bmatrix} 1-\lambda & \lambda & 0 & \cdots & 0 \\ 0 & 1-\lambda & \lambda & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1-\lambda & \lambda \\ \lambda & 0 & \cdots & 0 & 1-\lambda \end{bmatrix}, \tag{15}$$

where each row corresponds to "stay with probability $1 - \lambda$" and "jump forward with probability λ", and the last row wraps around to the first column due to the modulo-K design.

This matrix is circulant: each row is a cyclic shift of the previous row. As a result, its eigenvalues can be written in closed form as

$$\mu_m = (1 - \lambda) + \lambda\, e^{2\pi i m/K}, \quad m = 0, 1, \ldots, K - 1, \tag{16}$$

where $e^{2\pi i m/K}$ are the K-th roots of unity. The largest eigenvalue is $\mu_0 = 1$, which corresponds to the stationary (uniform) state of the chain. The remaining eigenvalues μ_m for $m \neq 0$ determine how fast the chain forgets its initial label and moves toward the stationary distribution.

The effective exponential decay rate is determined by the largest modulus among the nontrivial eigenvalues. From (16), we obtain

$$\gamma = -\log\left(\max_{m\neq 0} |\mu_m|\right) = -\frac{1}{2}\log\left[1 - 2\lambda(1-\lambda)\left(1 - \cos\frac{2\pi}{K}\right)\right]. \tag{17}$$

Indeed, the squared modulus of each eigenvalue satisfies

$$|\mu_m|^2 = 1 - 2\lambda(1-\lambda)\left(1 - \cos\frac{2\pi m}{K}\right). \tag{18}$$

Here, γ measures how quickly the chain mixes: a larger γ implies faster convergence toward the uniform distribution.

This decay rate directly controls the decay of the distance between the current marginal distribution π_t at time t and the uniform distribution u. In particular,

$$\|\pi_t - u\|_{\mathrm{TV}} \;\lesssim\; C\, e^{-\gamma t}, \qquad t_{\mathrm{mix}}(\epsilon) \;\approx\; \frac{1}{\gamma}\,\log\frac{1}{\epsilon}, \tag{19}$$

where $t_{\mathrm{mix}}(\epsilon)$ is the number of forward diffusion steps required for the chain to become ϵ-close to uniform in total variation distance. In practice, this means that the proposed cyclic jump process reaches an approximately uniform (high-entropy) state after a finite and predictable number of jumps, rather than requiring an unbounded number of diffusion steps.

For illustration, consider $K = 5$ and $\lambda = 0.3$. The dominant nontrivial eigenvalues of the transition matrix form a complex conjugate pair, with

$$\mu_1 = 0.7 + 0.3e^{2\pi i/5} \approx 0.7927 + 0.2853i.$$

This gives an effective exponential decay rate of $\gamma \approx 0.1714$. Direct computation of the transition matrix powers, starting from a deterministic label, gives $t_{\mathrm{mix}}(0.1) = 11$ and $t_{\mathrm{mix}}(0.01) = 25$. These results confirm geometric convergence toward the uniform stationary distribution.

A.5.1 Impact of Jump Probability

The jump probability λ directly controls the corruption strength of the forward process. A small λ produces mild perturbations, where most labels remain unchanged and the refinement problem becomes almost deterministic. Conversely, a large λ aggressively mixes the label space, quickly destroying boundary structure and making the reverse prediction substantially harder. From a spectral perspective, the mixing rate depends on $\lambda(1-\lambda)$ rather than increasing monotonically with λ. Within our tested range, excessively small values lead to under-corruption, while larger values promote faster mixing and may lead to overly corrupted masks.

As shown in Table 12, this trade-off is reflected in performance: $\lambda = 0.1$ preserves structure too strongly, yielding slightly higher IoU but weaker boundary accuracy; $\lambda = 0.5$ over-corrupts the masks and degrades both metrics. The best overall performance is achieved at $\lambda = 0.3$, which provides a balanced corruption level that preserves global shape while injecting sufficient boundary noise for effective refinement on DIS5K.

Table 12: Ablation on the jump rate λ (DIS validation set, $T = 6$).

Setting	**Mean IoU** $\uparrow$	**Mean mBA** $\uparrow$
T=6, λ=0.1	0.5847	0.7702
T=6, λ=0.2	0.5779	0.7716
T=6, λ=0.3	**0.5809**	**0.7732**
T=6, λ=0.4	0.5791	0.7731
T=6, λ=0.5	0.5736	0.7668

B Qualitative Analysis

We provide further qualitative analysis for our method and provide more comparisons. We do that for the refinement on DIS dataset in fig. 8. As can be seen our refined mask substantially improves the coarse mask obtained from UNet model Ronneberger et al. [2015]. Particularly the thin wires in the first row, the wired seat of the chair in the second row, wired strussses in the bridge, the collection of objects in the fourth row and the equation on the board are all better segmented. However, challenges still remain on these very challenging cases. For instance, the road part of the bridge is not yet improved in the refined mask.

In fig. 9, we consider the comparison with standard segmentation methods. In these examples we adopt the refinement of SOLO Wang et al. [2020]. However, we observe similar refinement performance for the refinement of any of these standard segmentation methods. Particularly, we have also provided results with respect to MaskRCNN in table 3 in the main paper. As can be observed from the set of examples, we do a better job for challenging cases like the fingers in the first case, facial contours and hair in the second case, and the object edges in the third case.

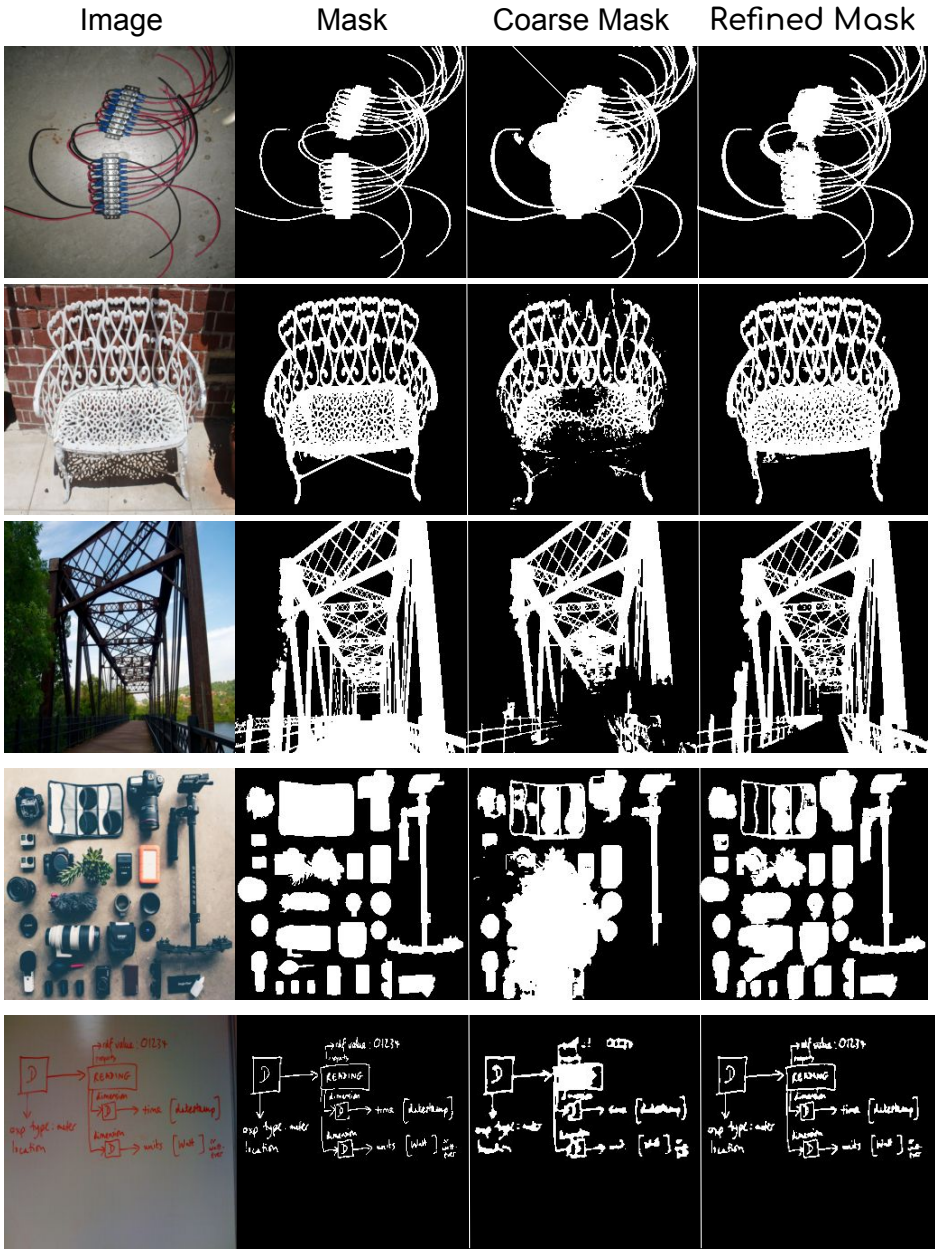

Figure 8: Refinement results on DIS dataset. From left to right: input image, ground-truth mask, coarse mask, and our refined mask. The coarse masks are taken from the UNet model and refined using our method. Notably, our method selectively corrects only the tampered regions while preserving the fine structures already present in the coarse mask.

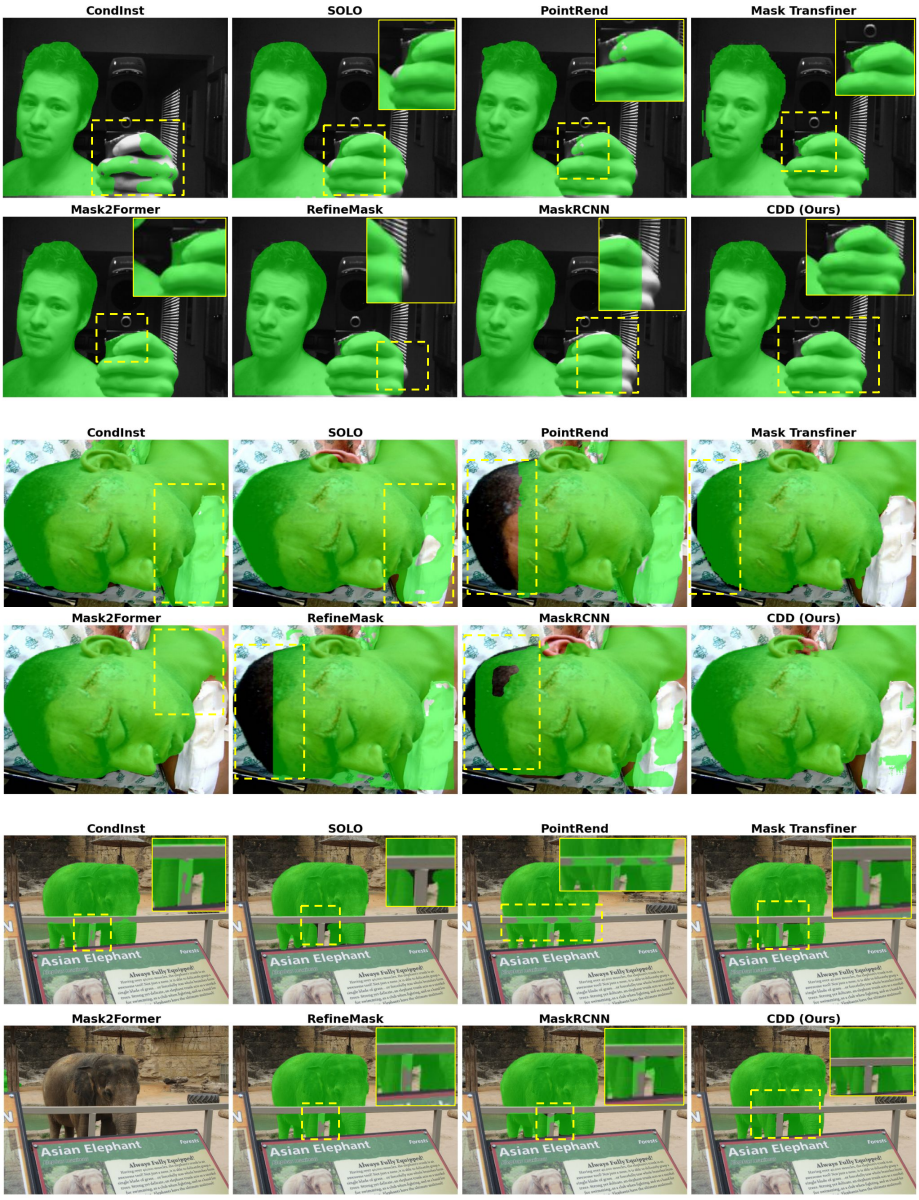

Figure 9: Refinement results on the COCO validation set. Instance segementation results are shown from 7 different mask generation models which are further refined by our method to produce the refined masks.

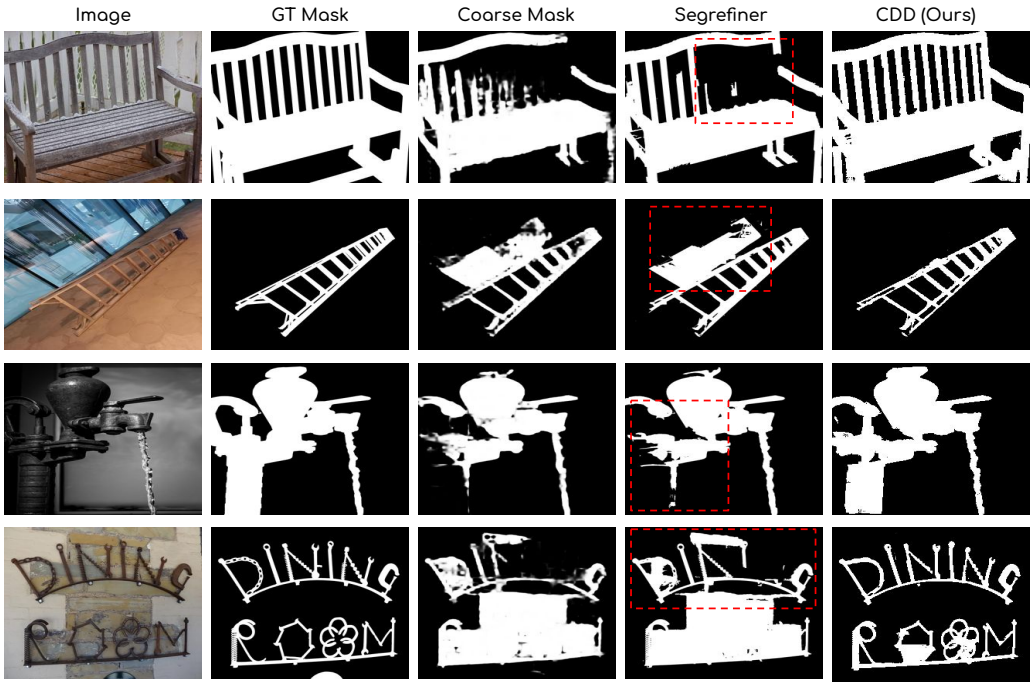

Figure 10: Qualitative comparison on the DIS dataset. From left to right: input image, ground-truth mask, coarse mask, SegRefiner result, and our CDD refinement. Compared with the coarse and SegRefiner predictions, CDD produces cleaner boundaries and more consistent fine structures, especially for thin objects and intricate details such as ladder and text patterns.

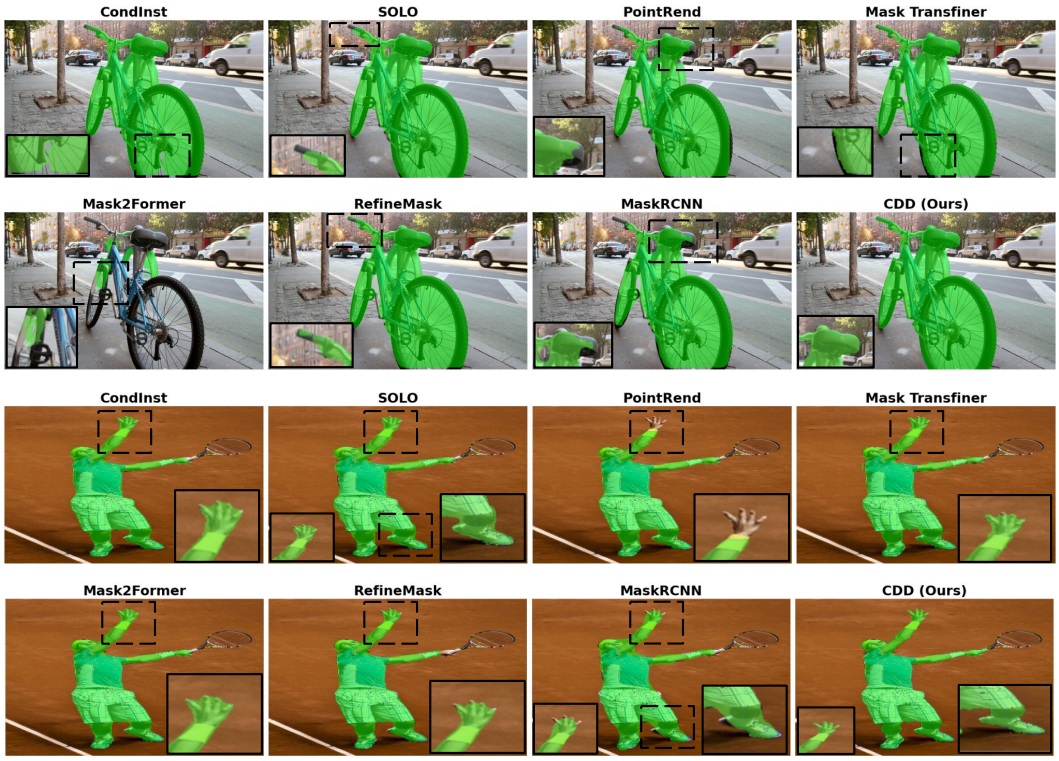

Figure 11: Visual comparison of refinement results on COCO corresponding to Table 4. We provide qualitative comparisons with strong baseline instance segmentation methods. We can observe that the proposed method is more precise in comparison to all these baselines. (Best viewed with zoom.)

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions, assumptions, and experimental scope of the proposed method. Relevant discussions can be found in the main paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations and future directions of the proposed framework in the conclusion and appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [No]

Justification: The paper provides theoretical analysis and derivations for the proposed cyclic diffusion process, including convergence and mixing-rate analysis. Key assumptions and derivations are described in the main paper and appendix, although some proofs are presented at a sketch level rather than as fully formal theorem-proof statements.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides sufficient experimental details, implementation settings, and evaluation protocols for reproducing the main results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper provides implementation and experimental details to support reproducibility. The datasets used are publicly available. The code and additional resources are intended to be released publicly.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies datasets, training settings, hyperparameters, optimization details, and evaluation protocols necessary to understand the reported results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Appendix A.1 reports results across three independent training runs with different random seeds (0, 1, and 2). The model achieves 84.52 ± 0.10 IoU and 82.15 ± 0.11 BFScore, where the reported variation represents the standard deviation across runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Computational resources and runtime measurements are reported in Section 5 and Appendix A.1. Training costs are measured using an NVIDIA H200 GPU, inference latency is evaluated on an NVIDIA RTX A5000 GPU, and the multi-step reverse sampling experiments in Appendix A.1 are conducted on NVIDIA RTX 4090 GPUs with 24 GB memory.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research presented in this paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

Justification: The paper focuses on methodological advances in segmentation refinement and does not include a dedicated discussion of broader societal impacts. Potential applications and misuse scenarios are expected to be limited due to the task-specific nature of the method.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The proposed method focuses on segmentation refinement and does not involve the release of high-risk generative models or sensitive datasets requiring additional safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites and credits the datasets, models, and prior methods used in the experiments. Publicly available assets are used according to their respective terms and licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce or release new datasets, benchmarks, or other standalone assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This research does not involve human subjects or crowdsourcing and therefore does not require IRB approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The proposed method does not use LLMs as part of the core research methodology.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.