NonMBB: Rethinking Synchronous Denoising for Bundle Construction

Rongchao Zhang[1] **Siheng Wang**[1,2] **Haodong Jing**[3]
Yajun Liu[4] **Zhengtao Yao**[5]
[1]Peking University [2]Westlake University [3]Xi'an Jiaotong University
[4]Shanghai Jiao Tong University [5]University of Southern California

Abstract

Bundle construction (BC) is a fundamental task in recommender systems that aims to assemble coherent subsets of items from large-scale catalogs, either by generating bundles from scratch or completing partial bundles with missing items. The vast combinatorial search space and heterogeneous semantic granularities pose substantial challenges to effective bundle construction. Despite recent advances in generative bundle construction, existing diffusion-based approaches typically employ shared denoising schedules, limiting their ability to accommodate heterogeneous refinement requirements across item positions. To address these challenges, we propose NonMBB, a generative approach based on asynchronous diffusion models for BC. Specifically, NonMBB trains a time-aware denoising network under a cosine Markov bridge forward and adopts a per-token asynchronous denoising strategy for inference. During inference, predictive entropy guides the assignment of unresolved tokens to different schedules, allowing high-uncertainty positions to remain under refinement while more confident positions may resolve earlier and provide additional contextual information. A reweighting coefficient further constrains the maximum timestep disparity across slots, ensuring controlled asynchronous denoising. Experiments on music playlist continuation and fashion outfit completion demonstrate consistent improvements in exact-match performance over the diffusion baseline, with particularly pronounced gains on longer playlists.

1 Introduction

Bundle construction (BC) aims to assemble coherent sets of items from large-scale catalogs to satisfy user needs and facilitate cross-selling, with applications in entertainment, food services, and various commercial domains [7, 52, 38]. Despite its broad applicability, BC poses substantial computational and semantic challenges due to the combinatorial explosion of candidate bundles and the heterogeneous dependencies among constituent items. Extensive research has explored sequential approaches that construct bundles by selecting items one at a time [25, 49]. However, such approaches impose an artificial ordering on inherently unordered item sets, potentially limiting their ability to capture complex inter-item dependencies and efficiently explore the combinatorial solution space [52, 61]. This limitation motivates combinatorial approaches that model bundles directly as sets, enabling order-agnostic construction without relying on a predefined decoding sequence [61, 52, 51].

Unlike conventional item recommendation and general set generation, BC requires jointly modeling higher-order compatibility among constituent items [7, 52] while flexibly conditioning on arbitrary subsets of observed items [35, 34]. Item slots within a bundle exhibit rich semantic heterogeneity, encoding multi-granularity compatibility signals ranging from pairwise co-occurrence to global stylistic consistency [52], which grow exponentially as bundle length increases. If a particular slot is

E-mail: lyzdxa@outlook.com

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

highly correlated with the known items, a progressive and carefully controlled refinement process is necessary to accurately model fine-grained compatibility semantics. Conversely, weakly related slots primarily provide auxiliary contextual information rather than serving as direct targets, yet they must be determined early in the generation process to provide stable contextual anchors for refining highly compatible slots. For example, in fashion outfit completion, given a coat, the trousers slot may require careful refinement to satisfy stylistic and color compatibility, whereas accessory slots, such as belts, can be resolved earlier to provide complementary contextual cues. Such heterogeneous refinement requirements pose challenges for sequential and existing set-based generative approaches.

Sequential and autoregressive approaches have been widely explored in recommendation and bundle construction, ranging from recurrent models for outfit generation [19] to self-attention-based sequential recommendation [25] and Transformer-based generative retrieval [44]. By modeling conditional dependencies between successive predictions, these approaches can effectively capture contextual relationships among items. However, conventional sequential generation imposes a predefined ordering on inherently unordered bundles, potentially introducing ordering bias and limiting flexibility in modeling set-level dependencies [61, 52]. In contrast, diffusion-based approaches enable iterative refinement of item representations, providing a flexible framework for jointly modeling intra-bundle compatibility without relying on a fixed item generation order [52]. Nevertheless, conventional diffusion approaches typically employ a shared noise schedule across all item slots, resulting in synchronized denoising processes that may overlook heterogeneous semantic dependencies and refinement requirements. Such co-progressive denoising limits the flexibility to resolve less constrained slots earlier as contextual anchors while allowing more semantically constrained slots to undergo gradual refinement using progressively clearer inter-slot context. These observations motivate the central research question of this work.

In this paper, we introduce NonMBB, a novel **Non**-co-progressive **M**arkov **B**ridge generative model for **B**undle construction. The key idea is to decouple the denoising trajectories of individual item slots to accommodate their heterogeneous refinement requirements. Specifically, NonMBB employs a discrete Markov bridge framework [23, 70] to model conditional bundle generation through the progressive refinement of perturbed item code sequences. Building upon asynchronous diffusion strategies [22, 40, 54], NonMBB assigns slot-specific denoising timesteps rather than enforcing a shared denoising schedule across all item slots. To dynamically coordinate these trajectories, we employ an entropy-guided dynamic masking mechanism that adaptively adjusts slot-wise denoising schedules according to prediction uncertainty at each inference step. Furthermore, a schedule reweighting mechanism constrains the maximum timestep disparity across slots to mitigate residual noise [22] arising from excessive asynchrony. This design allows less constrained slots to resolve earlier and serve as contextual anchors, while more semantically constrained slots undergo gradual refinement by leveraging progressively clearer inter-slot context. Consequently, NonMBB enables fine-grained compatibility modeling under partial observability while preserving the order-agnostic nature of bundle construction. Extensive experiments on music playlist continuation and fashion outfit completion demonstrate the effectiveness of NonMBB across diverse bundle construction scenarios. Our contributions can be summarized as follows:

- We introduce NonMBB, a discrete Markov bridge framework for conditional bundle construction that supports order-agnostic generation from partially observed item sets.
- We decouple the denoising trajectories of directly relevant and background item slots via entropy-guided dynamic masking and schedule reweighting, preventing noise residuals caused by excessive timestep disparity while preserving coherent joint bundle compatibility.
- Through experiments, we demonstrate the effectiveness of NonMBB in various scenarios, such as music playlist continuation and fashion outfit completion.

2 Related Work

2.1 Bundle Construction

Early studies explored related set-level recommendation problems through combinatorial optimization, casting it as a maximum clique problem [16] or leveraging graph neural networks to model item co-occurrence patterns [7, 8]. The Transformer architecture [53] has gained widespread prominence, particularly for its pivotal role in advancing large language models [59, 1, 69]. Thus, a dominant

line of work frames BC as a sequential generation problem, wherein items are predicted one at a time conditioned on previously selected items [25, 49]. Despite their empirical success in next-item prediction settings, these sequential approaches suffer from a fundamental mismatch: bundles are inherently unordered sets, and imposing a fixed decoding order introduces spurious positional biases that harm generalization [61, 52]. Wei et al. [57] explored non-autoregressive decoding for personalized bundle creative generation, while BundleNAT [61] employs a permutation-equivariant decoder to generate personalized bundles in a single pass by incorporating user preferences and item compatibility signals. Tomasi et al. [51] apply a continuous diffusion model to slate recommendation conditioned on text prompts, while DDBC [52] adopts a masked discrete diffusion framework with residual vector quantization to address both the ordering and catalog-size challenges simultaneously. In this work, we introduce a non-co-progressive token-level timestep allocation into the diffusion BC framework, enabling slot-specific refinement under partial bundle observations.

2.2 Diffusion and Markov Bridge Models

Diffusion models [21, 5, 66] have emerged as a powerful class of generative models, originally formulated in continuous spaces as denoising diffusion probabilistic models that learn to reverse a gradual Gaussian noise corruption process. Subsequent work significantly advances sampling efficiency through deterministic DDIM [48] and score-based formulations [3], broadening their applicability to image synthesis [68, 47, 65, 11], audio generation [42, 50, 20], and trajectory planning [63, 64]. For discrete data, diffusion has been extended to categorical token spaces through absorbing-state corruption mechanisms [24, 39] and masked diffusion language models [14, 18, 60, 17, 62, 58], wherein tokens are progressively masked and recovered through a learnable denoising network, with recent applications to sequential recommendation and generative retrieval. Multiple approaches have been explored to achieve this. In recommendation, diffusion-based methods have been explored for preference modeling and item prediction [56], while recent masked discrete diffusion approaches further enable joint generation of item codes for bundle construction [52].

Closely related to diffusion-based generative modeling, Markov bridges provide a probabilistic framework for constructing stochastic transitions conditioned on prescribed endpoints [15]. Unlike conventional diffusion formulations that typically connect data to a simple reference distribution, Markov bridge models can establish conditional stochastic paths between more general endpoint distributions. Recent studies have demonstrated their potential for discrete generation. RetroBridge [23] introduces a discrete Markov bridge model for retrosynthesis, directly modeling the relationship between product and reactant distributions without requiring a tractable noise prior. Bridge-IF [70] further develops a structure-conditioned Markov bridge for inverse protein folding, progressively refining sequences from an informative prior derived from protein structures. These advances highlight the flexibility of Markov bridges in modeling conditional dependencies between structured discrete distributions, motivating their exploration for bundle construction under partial observations.

2.3 Asynchronous Diffusion

Recent studies have explored asynchronous diffusion strategies that decouple the denoising trajectories of individual components through differentiated noise schedules [22, 40, 54]. A representative approach is AsynDM [22], which introduces pixel-wise asynchronous denoising for text-to-image generation to improve alignment between generated images and textual conditions. Rather than denoising all pixels at identical timesteps, AsynDM dynamically adjusts their noise schedules according to their relevance to the input prompt. This demonstrates how differentiated denoising trajectories can facilitate context-dependent refinement beyond conventional synchronous generation. Beyond image generation, asynchronous diffusion has also been investigated in audio-driven talking-head generation [54], temporal point process modeling [37], and diffusion-based visual perception [55], illustrating its applicability to diverse generative and predictive settings. Adaptive decoding has been investigated as an effective strategy for improving masked generative models. MaskGIT [6] adopts confidence-guided token selection under an iterative masking schedule, allowing highly confident predictions to be retained earlier. Path Planning for masked diffusion sampling [43, 4] further considers the selection of token generation orders, while entropy-bounded unmasking dynamically regulates the number of tokens revealed at each decoding step. Motivated by these, NonMBB adapts slot-specific denoising schedules to a discrete Markov bridge framework, enabling uncertainty-guided refinement of item slots conditioned on partially observed bundles.

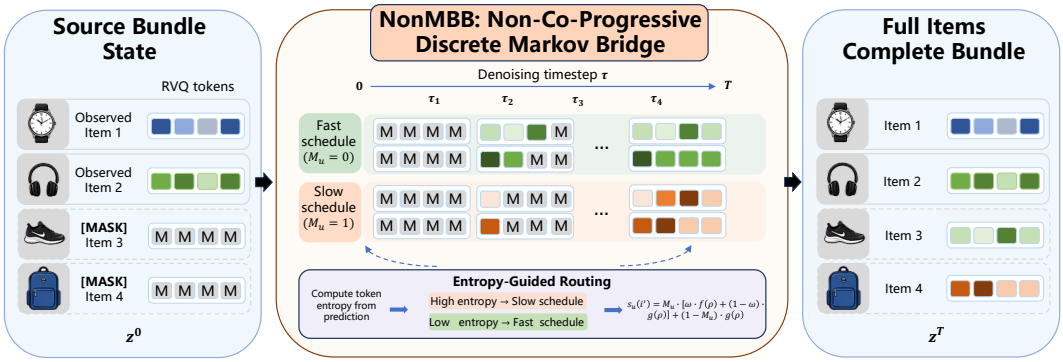


Figure 1: Overview of NonMBB. NonMBB reformulates bundle construction as a Markov bridge process, interpolating between the source bundle state and the target complete bundle via a structured forward corruption process. NonMBB employs entropy-guided token-level scheduling to accommodate heterogeneous refinement requirements. High-uncertainty slots follow slower refinement trajectories to leverage progressively clearer inter-slot context, whereas low-uncertainty slots are resolved earlier to provide contextual anchors.

3 Preliminary

3.1 Problem Statement

Bundle construction aims to assemble coherent sets of items that jointly satisfy user needs, where constituent items exhibit compatibility, semantic complementarity, and stylistic consistency [52, 34]. Formally, let $\mathcal{I} = \{\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_N\}$ denote the item catalog of size N, and $\mathcal{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_M\}$ denote the collection of M observed bundles, where each bundle $\boldsymbol{b}_m \in \mathcal{B}$ is a set of items $\{\boldsymbol{i}_{m_1}, \boldsymbol{i}_{m_2}, \ldots, \boldsymbol{i}_{m_{|b|}}\}$ with $\{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_{|b|}\} \subseteq [N]$ and $|b| \geq 2$. Following [52], each item $i_n \in \mathcal{I}$ is represented by a sequence of L discrete semantic codes, denoted as $z(i_n) = (z_{n,1}, z_{n,2}, \ldots, z_{n,L})$. These codes are obtained through residual vector quantization (RVQ), where successive quantization levels capture increasingly fine-grained semantic information. Each bundle is then serialized into a flat token sequence by prepending a <boi> token before each item's L code tokens and wrapping the sequence with <bos> and <eos> boundary tokens. Given a bundle $\hat{\boldsymbol{b}}_m \in \mathcal{B}$, we partition it into an observed subset $\boldsymbol{b}_x \subseteq \hat{\boldsymbol{b}}_m$ and a complementary target subset $\boldsymbol{b}_y = \hat{\boldsymbol{b}}_m / \boldsymbol{b}_x$. Thus, the ultimate objective of bundle construction is to learn the conditional distribution $p_\theta(\boldsymbol{b}_y \mid \boldsymbol{b}_x)$.

3.2 Discrete Markov Bridge

Discrete Markov bridges provide a probabilistic framework for modeling conditional transitions between discrete distributions through stochastic processes constrained by prescribed endpoints [70, 23, 15]. Given a pair of coupled bundle token sequences $(\boldsymbol{x}, \boldsymbol{y}) \sim p_{X,Y}$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ denote the initial and target sequences, respectively, a discrete Markov bridge defines a stochastic process $(\boldsymbol{z}^\tau)_{\tau=0}^T$ with $\boldsymbol{z}^0 = \boldsymbol{x}$ and $\boldsymbol{z}^T = \boldsymbol{y}$. Conditioned on the target endpoint $\boldsymbol{y}$, the process satisfies the Markov property

$$q(\boldsymbol{z}^\tau \mid \boldsymbol{z}^0, \boldsymbol{z}^1, \ldots, \boldsymbol{z}^{\tau-1}, \boldsymbol{y}) = q(\boldsymbol{z}^\tau \mid \boldsymbol{z}^{\tau-1}, \boldsymbol{y}), \tag{1}$$

with the terminal pinning condition

$$q(\boldsymbol{z}^T = \boldsymbol{y} \mid \boldsymbol{z}^{T-1}, \boldsymbol{y}) = 1. \tag{2}$$

Let $\mathcal{V}$ denote a finite token vocabulary of size $K = |\mathcal{V}|$. Each token state is represented by a one-hot vector in $\{0, 1\}^K$. Following the discrete transition formulation of [70], the conditional transition kernel for an individual token is defined as

$$q(\boldsymbol{z}^{\tau+1} \mid \boldsymbol{z}^\tau, \boldsymbol{y}) = \mathrm{Cat}\left(\boldsymbol{z}^{\tau+1}; \boldsymbol{Q}^\tau(\boldsymbol{y})\boldsymbol{z}^\tau\right), \tag{3}$$

where $\mathrm{Cat}(\cdot; \boldsymbol{p})$ denotes a categorical distribution parameterized by the probability vector $\boldsymbol{p}$. The transition matrix is given by

$$\boldsymbol{Q}^\tau(\boldsymbol{y}) = \alpha^\tau \boldsymbol{I}_K + (1 - \alpha^\tau)\boldsymbol{y}\boldsymbol{1}_K^\top, \tag{4}$$

where $\boldsymbol{I}_K$ is the identity matrix and $\alpha^\tau \in [0,1]$ controls the transition between preserving the current token and moving toward the target token. The schedule satisfies $\alpha^0 = 1$ and $\alpha^{T-1} = 0$, ensuring that the process reaches the target endpoint at the final timestep. The intermediate states can be sampled directly without simulating the entire Markov chain. Specifically, for $\tau \geq 1$, the cumulative transition matrix is defined as

$$\bar{\boldsymbol{Q}}^\tau = \boldsymbol{Q}^{\tau-1}\boldsymbol{Q}^{\tau-2}\cdots\boldsymbol{Q}^0 = \bar{\alpha}^\tau \boldsymbol{I}_K + (1-\bar{\alpha}^\tau)\boldsymbol{y}\mathbf{1}_K^\top, \tag{5}$$

where $\bar{\alpha}^\tau = \prod_{s=0}^{\tau-1}\alpha^s$ and $\bar{\boldsymbol{Q}}^0 = \boldsymbol{I}_K$. Accordingly, the marginal distribution at timestep τ admits the closed-form expression $q(\boldsymbol{z}^\tau \mid \boldsymbol{x}, \boldsymbol{y}) = \mathrm{Cat}\left(\boldsymbol{z}^\tau; \bar{\boldsymbol{Q}}^\tau \boldsymbol{x}\right)$. This formulation enables efficient sampling of intermediate token states at arbitrary timesteps during training. To approximate the target endpoint when only the initial bundle context is available, we employ a time-conditioned neural network ψ_θ to predict the target token distribution: $\hat{\boldsymbol{y}} = \psi_\theta(\boldsymbol{z}^\tau, \tau)$. The predicted distribution is subsequently used to parameterize the learned transition kernel.

4 Methods

In this section, we first formulate bundle construction as a discrete Markov bridge process that models the conditional transition from a partially observed bundle to its complete counterpart (§4.1). We then adapt non-co-progressive token-level denoising to accommodate heterogeneous refinement requirements across item slots, employing entropy-guided dynamic masking and schedule reweighting to adaptively coordinate their denoising trajectories (§4.2). Finally, we present the network architecture (§4.3). The overall framework is illustrated in Figure 1.

4.1 Markov Bridge Modeling for Bundle Construction

Standard discrete diffusion models typically corrupt clean data toward an uninformative all-mask absorbing state, implicitly treating the prior as an unstructured noise distribution [2, 45]. However, bundle completion differs from unconditional generation in that the observed item subset $\boldsymbol{b}_x$ provides informative semantic context for generating the missing items. Motivated by this observation, we formulate conditional bundle construction within a discrete Markov bridge framework [70, 23]. We establish a conditional stochastic pathway from a partially observed bundle state to its complete counterpart, progressively recovering missing item codes while preserving the observed bundle context.

Forward Transition via Bridge Schedule. Given a complete bundle $\hat{\boldsymbol{b}}_m$ containing $B_m = |\hat{\boldsymbol{b}}_m|$ items, let $\boldsymbol{y} \in \mathcal{V}^U$ denote its serialized token sequence, where $U = 2 + B_m(L+1)$ includes item codes and structural boundary tokens. We define the set of code-token positions as $\Omega_{\text{code}} = \{(j,\ell) \mid j = 1,\ldots,B_m,\ \ell = 1,\ldots,L\}$, and partition it into observed positions Ω_x and target positions $\Omega_y = \Omega_{\text{code}} \setminus \Omega_x$. The initial bridge state $\boldsymbol{z}^0$ preserves the observed item codes and structural tokens, while replacing the target positions with `[MASK]`. The terminal state is defined as $\boldsymbol{z}^T = \boldsymbol{y}$. Following the discrete Markov bridge formulation, the transition from timestep τ to $\tau+1$ is parameterized by $q(\boldsymbol{z}^{\tau+1} \mid \boldsymbol{z}^\tau, \boldsymbol{y})$, where the categorical transition matrix for each target token is defined as

$$\boldsymbol{Q}^\tau(\boldsymbol{y}) = \alpha^\tau \boldsymbol{I}_K + (1-\alpha^\tau)\boldsymbol{y}\mathbf{1}_K^\top. \tag{6}$$

where $\alpha^\tau \in [0,1]$ controls the probability of retaining the current token state. At target positions, the process progressively transitions from the mask state toward the clean target token, whereas observed item codes remain unchanged. To control the progression of the bridge, a cosine cumulative schedule can be adopted, inspired by [70], which concentrates the corruption mass toward intermediate timesteps and avoids the rapid collapse at early steps that can destabilize training. Accordingly, the marginal distribution at timestep τ admits the closed-form expression $q(\boldsymbol{z}^\tau \mid \boldsymbol{z}^0, \boldsymbol{y}) = \mathrm{Cat}\big(\boldsymbol{z}^\tau; \bar{\alpha}^\tau \boldsymbol{z}^0 + (1-\bar{\alpha}^\tau)\boldsymbol{y}\big)$. For each target position, the intermediate token therefore remains masked with probability $\bar{\alpha}^\tau$ and is replaced by its clean target with probability $1-\bar{\alpha}^\tau$. This closed-form marginal enables efficient sampling of intermediate bridge states at arbitrary timesteps without explicitly simulating the entire Markov chain. Crucially, structural boundary tokens in $\Omega_{\text{flag}} = \{\langle\texttt{bos}\rangle, \langle\texttt{eos}\rangle\} \cup \{\langle\texttt{boi}\rangle_j\}_{j=1}^{|\hat{\boldsymbol{b}}_m|}$ are exempt from corruption throughout, and their indices are pinned to original values, ensuring that the bundle serialization structure is preserved across all timesteps.

Bridge-Based Reconstruction and Training Objective. To approximate the target endpoint of the bridge, we employ a time-conditioned denoising network $p_\theta(\boldsymbol{y} \mid \boldsymbol{z}^\tau, \tau)$ that predicts the clean token sequence from a corrupted intermediate state at timestep τ.

Following the simple reconstruction objective of masked diffusion language models [45], the network is trained to recover the ground-truth tokens at masked target positions while conditioning on the observed item codes and the already recovered target tokens. Let $\Omega^\tau_{\text{mask}} \subseteq \Omega_y$ denote the set of target positions remaining masked at timestep τ. We optimize the following masked reconstruction objective:

$$\mathcal{L}_{\text{CE}} = \mathbb{E}_{\tau \sim \mathcal{U}\{0,\ldots,T-1\}} \mathbb{E}_{q(\boldsymbol{z}^\tau \mid \boldsymbol{z}^0, \boldsymbol{y})} \left[w_\tau \sum_{u \in \Omega^\tau_{\text{mask}}} -\log p_\theta(y_u \mid \boldsymbol{z}^\tau, \tau) \right], \tag{7}$$

where w_τ denotes a timestep-dependent weighting coefficient. In our implementation, we follow the ELBO-derived weighting adopted by masked diffusion language models [45], rather than using a uniformly weighted reconstruction loss. To ensure that generated code tuples decode to valid catalog items [52], we impose token-level validity constraints at inference time: for each code position (j, ℓ), the predicted logits are masked to $-\infty$ for all tokens outside the ℓ-th level codebook $\mathcal{V}^{(\ell)}$ before softmax, restricting generation to the legally admissible token subset at each RVQ level and preventing the model from emitting code combinations that cannot be mapped back to any catalog item.

Inference for Partial Bundle Completion. At inference time, the initial state is constructed from the observed partial bundle $\boldsymbol{b}_x$. Specifically, the observed code positions Ω_x and structural boundary tokens are preserved, while all missing item code positions Ω_y are initialized to `[MASK]`. Starting from $\boldsymbol{z}^0$, NonMBB iteratively predicts the target token distributions and samples the next bridge state according to learned bridge. Once a target position is recovered, its token value is retained throughout subsequent transitions, resulting in a monotonic mask-to-token generation process. To ensure valid token predictions, we restrict each RVQ code position (j, ℓ) to the corresponding level-specific vocabulary $\mathcal{V}^{(\ell)}$. After completing the bridge transitions, the resulting token sequence $\hat{\boldsymbol{z}}^T$ is converted into item identities through an inverted RVQ index, yielding the completed bundle $\hat{\boldsymbol{b}}_m = \boldsymbol{b}_x \cup \hat{\boldsymbol{b}}_y$.

4.2 Non-Co-Progressive Denoising for Heterogeneous Bundle Slots

Heterogeneity-Aware Timestep Scheduling. The Markov bridge formulation described above employs a shared transition schedule, under which all target item tokens progress toward the clean endpoint at the same rate. However, such co-progressive refinement overlooks the heterogeneous semantic dependencies and prediction uncertainties of individual item slots. Inspired by asynchronous diffusion strategies [22, 40, 54], we adapt token-wise timestep scheduling to the discrete Markov bridge framework, allowing different target positions to follow distinct refinement trajectories.

Formally, we replace the shared scalar noise level with a position-dependent noise-level vector $\boldsymbol{s} \in [0,1]^U$. The normalized noise level is related to the bridge timestep by $\tau_u = T(1 - s_u)$. We define the per-slot level directly as a monotone function of the global inference step $\rho = i'/T \in [0,1]$. Let $f(\rho) = 1 - \rho^2$ and $g(\rho) = 1 - \rho$. Given a binary routing mask $\boldsymbol{M} \in \{0,1\}^U$, where $M_u = 1$ indicates the slow track and $M_u = 0$ indicates the fast track, the candidate noise level of token u is defined as

$$s_u(i') = M_u \cdot [\omega \cdot f(\rho) + (1 - \omega) \cdot g(\rho)] + (1 - M_u) \cdot g(\rho), \tag{8}$$

where $\omega \in (0,1)$ controls the separation between the two trajectories. The slow track maintains a higher noise level than the fast track at intermediate inference steps, allowing uncertain positions to undergo more gradual refinement while earlier-resolved positions provide increasingly informative context. Under this position-dependent schedule, each target token follows a local bridge transition determined by its current and next noise levels.

Entropy-Guided Dynamic Mask and Inference. The scheduling mask $\boldsymbol{M}$ is initialized with all unknown positions assigned to the slow schedule ($M_u = 1$ for $u \in \Omega_y$) and is dynamically updated at each denoising step based on the predictive entropy of the denoising network. High predictive entropy is used as a heuristic indicator of unresolved uncertainty, motivating additional refinement

while allowing more confident positions to be resolved earlier. Formally, we compute the Shannon entropy of the predicted marginal at each masked position:

$$H_u = -\sum_{v \in \mathcal{V}} p_\theta(y_u = v \mid \boldsymbol{z}, \boldsymbol{s}) \, \log p_\theta(y_u = v \mid \boldsymbol{z}, \boldsymbol{s}), \tag{9}$$

The mask is then updated by thresholding against the mean entropy over currently masked positions $\Omega_y^{\text{mask}} = \{u \in \Omega_y : z_u = \texttt{[MASK]}\}$:

$$\begin{aligned} \bar{H} &= \frac{1}{|\Omega_y^{\text{mask}}|} \sum_{u \in \Omega_y^{\text{mask}}} H_u, \\ M_u &\leftarrow \mathbb{I}[H_u > \bar{H}], \qquad u \in \Omega_y^{\text{mask}}, \end{aligned} \tag{10}$$

with $M_u = 0$ for all resolved positions. Positions that the model struggles to predict are precisely those that would benefit most from the clearer inter-slot context provided by the faster-resolving background positions.

4.3 Network Architecture

NonMBB employs a lightweight Diffusion Transformer [41] as the denoising backbone. The denoising network is time-aware: a scalar slot-averaged global timestep is injected through adaptive layer normalization, conditioning the model on the current global denoising progress of the unknown slots. The backbone is trained under this time-aware, cosine schedule forward. Finally, output logits are projected to the full $\mathcal{V}$ and processed through the SUBS parameterization [45].

5 Experiments

5.1 Experimental Setup

Datasets. Following the experimental protocol of [52], we evaluate the performance of NonMBB on two bundle construction benchmarks from distinct application domains: Spotify [9] for music playlist continuation and POG [10] for fashion outfit completion. Spotify contains a large item catalog of $N = 254{,}155$ tracks. Since playlists vary widely in length and the discrete diffusion backbone requires fixed-length sequences, we derive three subsets by capping playlist length at $k \in \{30, 60, 90\}$ tracks (denoted Spotify$_{k=30/60/90}$). Following DDBC, playlists longer than k are truncated by uniformly sampling a valid starting position and retaining the corresponding contiguous subsequence of length k, while shorter playlists are discarded. POG contains $N = 31{,}217$ fashion items, and we use its dense variant with a fixed bundle length of $k = 4$, denoted as POG$_{k=4}$. We adopt the same non-overlapping training, validation, and test splits as DDBC and apply its item-swapping augmentation strategy during training to reduce sensitivity to arbitrary item orderings. We also follow DDBC's candidate-set construction protocol. Given a ground-truth target set b_y, a candidate ratio ρ determines the size of the evaluation pool: $\mathcal{S} = b_y \cup \text{Random}_{(\rho-1)|b_y|}(\mathcal{I} \setminus b)$, where $\text{Random}_n(\cdot)$ denotes the uniform sampling of n distinct negative items.

Implementation Details. Item embeddings are discretized via $L = 4$-level RVQ with a fixed per-level codebook size. The denoising backbone is a lightweight Diffusion Transformer comprising 6 Transformer blocks with hidden dimension 64 and 8 self-attention heads, conditioned on timestep via adaptive layer normalization. The denoising mechanism is applied exclusively at inference time with schedule reweighting coefficient $\omega = 0.5$ and entropy threshold computed dynamically as the mean entropy across all unresolved slots at each step. Unless stated otherwise, the input-predict ratio $|b_x| : |b_y|$ is set to $1 : 1$. All models are trained on A100 GPUs with the AdamW optimizer [33].

Evaluation Metrics. We report three complementary metrics to evaluate bundle construction quality from both exact-match and semantic perspectives [52]. Specifically, i) **F1** and ii) **Jaccard** (Jacc) [36, 12] measure set-level retrieval performance based on exact item overlap between the predicted set $\hat{b}_y$ and the ground-truth, where Jacc $= \frac{|\hat{b}_y \cap b_y|}{|\hat{b}_y \cup b_y|}$. Latent-space similarity-based metric iii) **OAS** [27] complements exact-match metrics by measuring semantic alignment in the embedding space via optimal bipartite matching: OAS $= \frac{1}{|b_y|} \max_{\mathcal{P}} \sum_{(i,j) \in \mathcal{P}} \cos(\mathbf{E}(i), \mathbf{E}(j))$, where $\mathcal{P}$ denotes the

Table 1: Performance on Spotify ($k \in \{30, 60, 90\}$) and POG ($k = 4$). Best results are **bold**.

Model	**Spotify** ($k = 30$)			**Spotify** ($k = 60$)			**Spotify** ($k = 90$)			**POG** ($k = 4$)		
	F1 ↑	Jacc. ↑	OAS ↑	F1 ↑	Jacc. ↑	OAS ↑	F1 ↑	Jacc. ↑	OAS ↑	F1 ↑	Jacc. ↑	OAS ↑
CLHE [35]	0.071	0.039	0.373	0.100	0.054	0.446	0.119	0.065	0.486	0.140	0.096	0.446
Bi-LSTM [19]	0.124	0.071	0.489	0.062	0.034	0.430	0.047	0.025	0.426	0.035	0.024	0.390
SASRec [25]	0.070	0.043	0.318	0.089	0.054	0.310	0.050	0.029	0.285	0.169	0.114	0.468
DDBC [52]	0.282	0.176	0.660	0.285	0.178	0.662	0.287	0.177	0.684	0.139	0.098	0.526
TIGER [44]	0.093	0.053	0.329	0.129	0.076	0.413	0.123	0.070	0.480	**0.213**	**0.157**	**0.546**
BundleNAT [61]	0.153	0.090	0.454	0.101	0.056	0.438	0.095	0.052	0.446	0.145	0.097	0.462
BundleMLLM [32]	0.046	0.024	0.296	0.045	0.024	0.324	0.052	0.027	0.355	0.070	0.047	0.322
NonMBB	**0.352**	**0.230**	**0.660**	**0.369**	**0.239**	**0.701**	**0.350**	**0.223**	**0.715**	0.153	0.108	0.523

optimal one-to-one matching between $\hat{b}_y$ and b_y solved via the Hungarian algorithm [27], and $\mathbf{E}(\cdot)$ is the CLHE encoder [35]. OAS captures semantically plausible predictions that exact-match metrics penalize, and is particularly informative for longer bundles.

5.2 Music Playlist Bundle Generation

We evaluate NonMBB against representative baselines spanning sequential recommendation and generation methods, including Bi-LSTM [19], SASRec [25], and TIGER [44], as well as set-oriented and non-autoregressive approaches, including CLHE [35], BundleNAT [61], BundleMLLM [32], and the diffusion-based DDBC [52]. Table 1 summarizes the results on Spotify$_{k=30/60/90}$. NonMBB consistently achieves the best F1 and Jaccard scores across all three settings, outperforming DDBC, the strongest baseline on Spotify, by substantial margins. These results demonstrate the effectiveness of NonMBB in recovering ground-truth items across different bundle lengths. Beyond exact-match performance, NonMBB also achieves competitive semantic alignment, matching DDBC's OAS of 0.660 at $k = 30$ and improving it from 0.662 to 0.701 and from 0.684 to 0.715 at $k = 60$ and $k = 90$, respectively. Since OAS evaluates embedding-space similarity through optimal bipartite matching, these improvements suggest that NonMBB produces semantically closer alternatives to ground-truth items, particularly for longer playlists where multiple plausible completions may exist. Overall, the results are consistent with the motivation of non-co-progressive denoising, which allows item tokens to follow differentiated refinement schedules and potentially benefit from progressively clearer inter-item context.

5.3 Fashion Outfit Bundle Generation

Table 1 also reports the results on POG$_{k=4}$. Unlike Spotify playlist continuation, fashion outfit completion involves substantially shorter bundles, with only four items per outfit and two target items under the default $1:1$ observed-to-target ratio. This setting provides fewer unknown item slots for differentiated refinement, potentially limiting the benefits of non-co-progressive denoising. Indeed, its OAS slightly decreases from 0.526 to 0.523, and TIGER achieves the best performance across all three metrics. These results suggest that NonMBB remains competitive with diffusion-based bundle generation on short outfits, although its advantages are less pronounced than those observed on longer Spotify playlists.

5.4 Effect of Input-Predict Ratio

Table 2 reports robustness to the input-predict ratio $|b_x| : |b_y|$ on Spotify$_{k=60}$. NonMBB consistently outperforms all competing methods across the four evaluated ratios, demonstrating its effectiveness under different levels of partial observability. NonMBB achieves particularly pronounced improvements when only a small number of items are observed. These results suggest that NonMBB is particularly effective in settings with limited observed context, where recovering compatible target items is challenging. The uncertainty-guided asynchronous refinement allows different item positions to follow adaptive denoising trajectories and progressively exploit contextual information from

Table 2: Effect of input-predict ratio on Spotify$_{k=60}$.

Model	5/55			10/50			30/30			45/15		
	F1	Jacc	OAS	F1	Jacc	OAS	F1	Jacc	OAS	F1	Jacc	OAS
BundleNAT	0.106	0.059	0.443	0.128	0.072	0.463	0.101	0.056	0.438	0.084	0.046	0.359
SASRec	0.119	0.070	0.425	0.131	0.078	0.442	0.089	0.054	0.310	0.095	0.055	0.315
DDBC	0.237	0.144	0.637	0.268	0.164	0.664	0.285	0.178	0.662	0.260	0.161	0.614
TIGER	0.087	0.050	0.365	0.100	0.059	0.381	0.129	0.076	0.413	0.154	0.091	0.426
NonMBB	**0.332**	**0.213**	**0.676**	**0.359**	**0.232**	**0.700**	**0.369**	**0.239**	**0.701**	**0.313**	**0.200**	**0.643**

earlier-resolved positions. Although the relative improvements become smaller as the number of observed items increases, NonMBB maintains a consistent advantage across all evaluated settings, highlighting its robustness to varying degrees of bundle incompleteness.

5.5 Effect of Non-Co-Progressive Scheduling Hyperparameter ω

Table 3 presents the effect of the schedule reweighting coefficient ω on Spotify$_{k=60}$. As defined in §4.2, ω controls the separation between the slow and fast denoising trajectories. It shows that NonMBB maintains relatively stable performance across the evaluated settings. Specifically, $\omega = 0.5$ achieves the highest F1 and Jaccard, while matching the best OAS obtained at $\omega = 0.3$. Increasing ω to 1.0 slightly reduces all three metrics, suggesting that excessively separated denoising trajectories may be less favorable for coherent bundle generation. These results indicate that moderate schedule separation provides a reasonable balance between heterogeneous token refinement and coordinated bundle generation, motivating the default choice of $\omega = 0.5$.

Table 3: Effect of schedule reweighting coefficient ω on Spotify$_{k=60}$.

ω	F1↑	Jacc↑	OAS↑
0.3	0.367	0.238	0.701
0.5	0.369	0.239	0.701
0.7	0.366	0.238	0.700
1.0	0.364	0.235	0.700

5.6 Multi-Sampling Performance

Table 4 reports Jaccard under multiple sampling attempts $A \in \{1, 5, 20\}$ across the three Spotify subsets. NonMBB consistently outperforms the conventional diffusion baseline at all sampling budgets and across all subsets. Although Jaccard decreases as the number of sampling attempts increases, NonMBB maintains its advantage at $A = 20$, achieving scores of 0.060, 0.062, and 0.058 across the three subsets. This decline can be attributed in part to the union-based evaluation protocol, under which additional sampling attempts may introduce non-target items and enlarge the predicted set. Overall, these results demonstrate that NonMBB maintains stronger set-level retrieval performance across different sampling budgets, although the relative advantage over DDBC becomes smaller with more generation attempts.

5.7 Ablation Studies

Table 5 presents a component ablation on Spotify$_{k=60}$. These results demonstrate the effectiveness of combining bridge-based generation with heterogeneous timestep scheduling, even without dynamically updating the routing assignments. The full NonMBB with entropy-guided dynamic mask achieves the best results, validating that adaptively updating the mask throughout denoising, which allows the model to progressively transfer slots from the slow schedule to the fast schedule as their predictive certainty increases. Overall, the ablation results highlight the importance of non-co-progressive refinement within the proposed framework, while indicating that entropy-guided routing provides complementary benefits.

Table 4: Jaccard under multiple sampling attempts $A \in \{1, 5, 20\}$ on Spotify subsets.

Model	$\text{Spotify}_{k=30}$			$\text{Spotify}_{k=60}$			$\text{Spotify}_{k=90}$		
	@1	@5	@20	@1	@5	@20	@1	@5	@20
CLHE	0.009	0.005	0.002	0.008	0.004	0.002	0.007	0.003	0.001
SASRec	0.043	0.023	0.009	0.054	0.030	0.013	0.029	0.013	0.005
DDBC	0.177	0.130	0.053	0.185	0.137	0.055	0.177	0.132	0.054
BundleNAT	0.090	0.076	0.033	0.056	0.055	0.027	0.052	0.052	0.026
TIGER	0.053	0.028	0.011	0.076	0.036	0.013	0.070	0.034	0.013
NonMBB	**0.228**	**0.158**	**0.060**	**0.239**	**0.158**	**0.062**	**0.223**	**0.145**	**0.058**

Table 5: Ablation study of NonMBB on $\text{Spotify}_{k=60}$. Each row progressively introduces components.

w/ MB schedule	w/ Non-co-prog.	w/ Entropy-guided routing	F1↑	Jacc↑	OAS↑
✗	✗	✗	0.285	0.178	0.662
✓	✗	✗	0.292	0.184	0.674
✓	✓	✗	0.366	0.238	0.700
✓	✓	✓	**0.369**	**0.239**	**0.701**

6 Conclusion and Limitations

In this work, we presented NonMBB, a non-co-progressive discrete Markov bridge framework for bundle construction. By combining bridge-based conditional generation with token-wise asynchronous denoising, NonMBB accommodates heterogeneous refinement requirements across item slots. Specifically, entropy-guided dynamic scheduling allows uncertain positions to undergo more gradual refinement while earlier-resolved positions provide increasingly informative inter-slot context. Extensive experiments on Spotify and POG demonstrate the effectiveness of NonMBB, achieving consistent improvements in F1 and Jaccard over the diffusion-based baseline across all evaluated settings. Notably, NonMBB also achieves stronger semantic alignment on longer Spotify playlists, highlighting the potential of adaptive denoising for constructing coherent bundles.

Limitations. There are also some limitations. i) We do not implement the literal per-token state recurrence of the original formulation. A rigorously well-defined per-token state machine faithful to that recurrence is left as future work. ii) The benefits of non-co-progressive denoising are less pronounced for short bundles, where fewer target slots may limit opportunities for differentiated refinement. iii) The entropy-guided scheduling introduces additional computational overhead and relies on prediction uncertainty as an indirect indicator of refinement requirements, which may not fully capture semantic compatibility among items. Finally, the applicability of NonMBB beyond bundle construction remains unexplored. Although uncertainty-guided asynchronous refinement could potentially benefit other unordered set-generation and masked diffusion tasks, its generalizability across different data modalities and generation objectives has yet to be systematically evaluated. Extending NonMBB to these settings represents a promising direction for future research.

References

[1] X. An, J. Sun, Z. Gui, and W. He. CHOICE: benchmarking the remote sensing capabilities of large vision-language models. In *NeurIPS*, 2025.

[2] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *NeurIPS*, 2021.

[3] B. Barancikova, Z. Huang, and C. Salvi. Sigdiffusions: Score-based diffusion models for time series via log-signature embeddings. In *ICLR*, 2025.

[4] H. Ben-Hamu, I. Gat, D. Severo, N. Nolte, and B. Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. In *NeurIPS*, 2025.

[5] P. Cao, F. Zhou, Q. Song, and L. Yang. Controllable generation with text-to-image diffusion models: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 48(4):4771–4791, 2026.

[6] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022.

[7] J. Chang, C. Gao, X. He, D. Jin, and Y. Li. Bundle recommendation with graph convolutional networks. In *ACM SIGIR*, 2020.

[8] J. Chang, C. Gao, X. He, D. Jin, and Y. Li. Bundle recommendation and generation with graph neural networks. *IEEE Trans. Knowl. Data Eng.*, 35(3):2326–2340, 2023.

[9] C.-W. Chen, P. Lamere, M. Schedl, and H. Zamani. Recsys challenge 2018: Automatic music playlist continuation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018.

[10] W. Chen, P. Huang, J. Xu, X. Guo, C. Guo, F. Sun, C. Li, A. Pfadler, H. Zhao, and B. Zhao. Pog: personalized outfit generation for fashion recommendation at alibaba ifashion. In *KDD*, 2019.

[11] H. Chi, J. Li, H. Jiang, W. Sheng, Y. Ma, J. Wang, and Y. Mu. Balancing image compression and generation with bootstrapped tokenization. *arXiv preprint arXiv:2606.05552*, 2026.

[12] Y. Ding, P. Mok, Y. Ma, and Y. Bin. Personalized fashion outfit generation with user coordination preference learning. *Information Processing & Management*, 60(5):103434, 2023.

[13] R. Dong, S. Yuan, B. Luo, M. Chen, J. Zhang, L. Zhang, W. Li, J. Zheng, and H. Fu. Building bridges across spatial and temporal resolutions: Reference-based super-resolution via change priors and conditional diffusion model. In *CVPR*, 2024.

[14] Y. Du, M. Zhao, S. Fan, Z. Chen, C. Jia, and Y. Jiang. Mdiff4str: Mask diffusion model for scene text recognition. In *AAAI*, 2026.

[15] P. Fitzsimmons, J. Pitman, and M. Yor. Markovian bridges: Construction, palm interpretation, and splicing. In *Seminar on Stochastic Processes*, pages 101–134. 1993.

[16] Y. Gong, Y. Zhu, L. Duan, Q. Liu, Z. Guan, F. Sun, W. Ou, and K. Q. Zhu. Exact-k recommendation via maximal clique optimization. In *KDD*, 2019.

[17] Z. Gu, H. Chen, Q. Qin, Y. Xin, K. Gan, L. Liu, L. Cui, X. Wang, B. Zhou, Y. Zhang, Z. Zeng, C. Gao, W. Chen, R. Zhang, H. Wu, S. Shen, C. Meng, W. Wang, J. Li, and Z. Lan. Llada-ui: Bringing block-wise diffusion to vision-language GUI agents. *CoRR*, abs/2609.13287, 2026.

[18] D. Gwak, M. Jung, J. Park, M. Park, C. Park, J. Hyung, and J. Choo. Reward-weighted sampling: Enhancing non-autoregressive characteristics in masked diffusion llms. In *EMNLP*, 2025.

[19] X. Han, Z. Wu, Y.-G. Jiang, and L. S. Davis. Learning fashion compatibility with bidirectional lstms. In *MM*, 2017.

[20] Y. Han, Y. Wang, Z. Yang, L. Lin, C. Tao, X. Chu, and L. Pan. Beyond entropy: Self-diagnostic multi-role token optimization for video reasoning. *arXiv preprint arXiv:2610.03400*, 2026.

[21] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[22] Z. Hu, Y. Tong, F. Zhang, J. Yuan, J. Xiao, and K. Kuang. Asynchronous denoising diffusion models for aligning text-to-image generation. In *ICLR*, 2026.

[23] I. Igashov, A. Schneuing, M. H. S. Segler, M. M. Bronstein, and B. E. Correia. Retrobridge: Modeling retrosynthesis with markov bridges. In *ICLR*, 2024.

[24] F. Kang, X. Huang, and B. Li. Diffulex: Diffusion based lexically constrained text generation with mixed absorbing state and constraint balance. *Expert Syst. Appl.*, 281:127614, 2025.

[25] W. Kang and J. J. McAuley. Self-attentive sequential recommendation. In *ICDM*, 2018.

[26] B. Kim, G. Kwon, K. Kim, and J. C. Ye. Unpaired image-to-image translation via neural schrödinger bridge. In *ICLR*, 2024.

[27] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[28] E. Lee, S. Jeong, and K. Sohn. EBDM: exemplar-guided image translation with brownian-bridge diffusion models. In *ECCV*, 2024.

[29] W. Lei, L. Liu, and J. Wang. Bridge to non-barrier communication: Gloss-prompted fine-grained cued speech gesture generation with diffusion model. In *IJCAI*, 2024.

[30] C. L'eonard. From the schrödinger problem to the monge-kantorovich problem. *arXiv: Optimization and Control*, 2010.

[31] Z. Li, H. Dou, S. Fang, W. Han, Y. Deng, and L. Yang. Physics-aligned field reconstruction with diffusion bridge. In *ICLR*, 2025.

[32] X. Liu, J. Wu, Z. Tao, Y. Ma, Y. Wei, and T.-s. Chua. Fine-tuning multimodal large language models for product bundling. In *KDD*, 2025.

[33] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[34] G. Lv, C. Zhao, M. Chen, N. Shen, and S. Yan. Multi-view hypergraph contrastive learning for bundle recommendation. *J. Supercomput.*, 81(15):1365, 2025.

[35] Y. Ma, X. Liu, Y. Wei, Z. Tao, X. Wang, and T. Chua. Leveraging multimodal features and item-level user feedback for bundle construction. In *WSDM*, 2024.

[36] C. D. Manning, P. Raghavan, and H. Schütze. Introduction to information retrieval, 2008.

[37] A. Mukherjee, R. Deng, H. Zhao, Y. Mao, L. Sigal, and F. Tung. Adiff4tpp: Asynchronous diffusion models for temporal point processes. *Trans. Mach. Learn. Res.*, 2026, 2026.

[38] H. Nguyen, Q. Nguyen, D. Pham, D. Le, H. Le, P. Sitkrongwong, A. Takasu, and M. Mansoury. Ramen: Multi-strategy multi-modal learning for bundle construction. In *ECAI*, 2025.

[39] J. Ou, S. Nie, K. Xue, F. Zhu, J. Sun, Z. Li, and C. Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *ICLR*, 2025.

[40] Y. Pan, R. Feng, Q. Dai, Y. Wang, W. Lin, M. Guo, C. Luo, and N. Zheng. Semantics lead the way: Harmonizing semantic and texture modeling with asynchronous latent diffusion. In *CVPR*, 2026.

[41] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

[42] W. Peng and K. Zhang. Harmonidpo: Video-guided audio generation via preference-optimized diffusion. *Int. J. Comput. Vis.*, 134(2):77, 2026.

[43] Z. Peng, Z. Bezemek, S. Patel, J. Rector-Brooks, S. Yao, A. Tong, and P. Chatterjee. Path planning for masked diffusion model sampling. *CoRR*, 2025.

[44] S. Rajput, N. Mehta, A. Singh, R. Hulikal Keshavan, T. Vu, L. Heldt, L. Hong, Y. Tay, V. Tran, J. Samost, et al. Recommender systems with generative retrieval. *NeurIPS*, 2023.

[45] S. S. Sahoo, M. Arriola, Y. Schiff, A. Gokaslan, E. Marroquin, J. T. Chiu, A. Rush, and V. Kuleshov. Simple and effective masked diffusion language models. In *NeurIPS*, 2024.

[46] E. Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. *Annales de l'institut Henri Poincaré*, 2(4):269–310, 1932.

[47] Z. Shang, H. Liu, R. Zhang, Z. Wei, and T. Feng. Pmmd: A pose-guided multi-view multi-modal diffusion for person generation. In *ICASSP*, 2026.

[48] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[49] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*, 2019.

[50] Y. Sun, Z. Xu, H. Zhou, J. Guan, Q. Yang, K. Wang, B. Liang, Y. Li, H. Feng, J. Wang, Z. Liu, and H. Koike. Cosh-dit: Co-speech gesture video synthesis via hybrid audio-visual diffusion transformers. *Int. J. Comput. Vis.*, 134(3):85, 2026.

[51] F. Tomasi, F. Fabbri, M. Lalmas, and Z. Dai. Diffusion model for slate recommendation. *CoRR*, abs/2408.06883, 2024.

[52] T. Tu, A. Li, Y. Ma, S. Xu, X. Liu, H. Ma, L. Pang, and T.-S. Chua. Discrete diffusion for bundle construction. In *ICLR*, 2026.

[53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[54] H. Wang, Y. Weng, J. Du, H. Xu, X. Wu, S. He, B. Yin, C. Liu, J. Gao, and Q. Liu. READ: real-time and efficient asynchronous diffusion for audio-driven talking head generation. In *AAAI*, 2026.

[55] T. Wang, X. Xu, C. Sheng, S. Qiu, L. Liu, D. Guo, and X. Bao. Fast diffusion-based camouflaged object detection via asynchronous denoising and linear attention. *IEEE Trans. Multim.*, 28:6501–6516, 2026.

[56] W. Wang, Y. Xu, F. Feng, X. Lin, X. He, and T. Chua. Diffusion recommender model. In *SIGIR*, 2023.

[57] P. Wei, S. Liu, X. Yang, L. Wang, and B. Zheng. Towards personalized bundle creative generation with contrastive non-autoregressive decoding. In *SIGIR*, 2022.

[58] X. Xiao, C. Liu, C.-T. Liao, Y. Zhang, Q. Lan, Y. Wei, L. Zhao, J. Wang, J. Gu, M. Ye, et al. Staying vigil ant: Mitigating visual laziness via counterfactual visual alignment in mllms. In *ECCV*, 2026.

[59] X. Xiao, C. Ma, Y. Zhang, C. Liu, Z. Wang, Y. Li, L. Zhao, G. Hu, T. Wang, and H. Xu. Not all directions matter: Towards structured and task-aware low-rank model adaptation. In *ACL*, 2026.

[60] Y. Xin, S. Luo, T. Xu, Q. Qin, H. Chen, K. Zhu, Z. Zhang, Y. He, R. Zhang, J. Bai, S. Cao, B. Fu, J. He, Y. Liu, Y. Cao, and X. Liu. dmllm-tts: Self-verified and efficient test-time scaling for diffusion multi-modal large language models. In *CVPR*, 2026.

[61] W. Yang, C. Yang, J. Li, Y. Tan, X. Lu, and C. Shi. Non-autoregressive personalized bundle generation. *Inf. Process. Manag.*, 61(5):103814, 2024.

[62] Z. Yang, F. Wei, Y. Wang, Y. Han, Y. Li, Z. Zong, G. P. Hancke, X. Chu, and R. W. Lau. Towards unbiased on-policy distillation for block diffusion language models. *arXiv preprint arXiv:2610.05373*, 2026.

[63] L. Yin, R. Ju, G. Guo, and E. Cheng. Diffrefiner: Coarse to fine trajectory planning via diffusion refinement with semantic interaction for end to end autonomous driving. In *AAAI*, 2026.

[64] R. Zhang, Y. Huang, Y. Cao, and H. Wang. Molebridge: Synthetic space projecting with discrete markov bridges. In *NeurIPS*, 2025.

[65] R. Zhang, C. Li, Y. Lou, Y. Shi, H. Wang, and Y. Huang. Steering where to diffuse: Generative modeling of phenotypic response simulation with steered diffusion bridge. In *CVPR*, 2026.

[66] R. Zhang, Y. Lou, Y. Huang, Y. Xin, Y. Cao, and H. Wang. Beyond conservation: Flexible molecular assembly with unbalanced diffusion bridge. In *AAAI*, 2026.

[67] Z. Zhang, T. Li, and P. Zhou. Learning stochastic dynamics from snapshots through regularized unbalanced optimal transport. In *ICLR*, 2025.

[68] W. Zhong, J. Lin, P. Chen, F. Gao, L. Lin, and G. Li. High-fidelity and lip-synced talking face synthesis via landmark-based diffusion model. *IEEE Trans. Image Process.*, 35:3369–3380, 2026.

[69] T. Zhou, X. Zhang, and Y. Tang. Panollama: Generating endless and coherent panoramas with next-token-prediction llms. In *ICCV*, 2025.

[70] Y. Zhu, J. Wu, Q. Li, J. Yan, M. Yin, W. Wu, M. Li, J. Ye, Z. Wang, and J. Wu. Bridge-if: Learning inverse protein folding with markov bridges. In *NeurIPS*, 2024.

A Schrödinger Bridge.

The Schrödinger Bridge (SB) problem [46, 30], originating from statistical physics and closely related to entropy-regularized optimal transport, seeks a stochastic process that connects prescribed initial and terminal distributions while minimizing the relative entropy of its path measure with respect to a reference process [67]. By accommodating general endpoint distributions, the SB framework provides a flexible perspective on conditional generative modeling beyond conventional noise-to-data transformations [26]. This perspective has motivated diffusion bridge approaches in diverse applications, including image restoration [13, 28], speech synthesis [29], and scientific simulation [31]. Related developments have extended endpoint-conditioned stochastic processes to discrete state spaces through Markov bridge formulations [70, 23, 15]. Rather than relying on continuous Gaussian perturbations, these discrete formulations construct categorical transition kernels that progressively connect an initial state to a target state, providing a tractable probabilistic framework for structured discrete generation. For bundle construction, this formulation offers a natural means of conditioning generation on partially observed items. Instead of adopting the conventional clean-to-mask corruption process of absorbing-state discrete diffusion, a target-conditioned Markov bridge can be constructed from a partially observed bundle toward its complete counterpart.

B Connection to Schrödinger Bridges and Optimal Transport

The discrete Markov bridge underlying NonMBB is conceptually related to optimal transport and Schrödinger bridge formulations, which study stochastic processes connecting prescribed endpoint distributions. In particular, the Schrödinger bridge problem seeks a path measure that matches the prescribed initial and terminal marginals while minimizing its relative entropy with respect to a reference stochastic process [46]. Unlike this entropy-regularized optimization framework, NonMBB explicitly constructs a target-conditioned categorical transition kernel, following the Markov bridge formulation [70].

Specifically, for a token represented as a one-hot vector, the reference transition matrix is defined as

$$\mathbf{Q}^\tau(\mathbf{y}) = \alpha^\tau \mathbf{I}_K + (1-\alpha^\tau)\mathbf{y}\mathbf{1}_K^\top, \tag{11}$$

where $\mathbf{y}$ denotes the target token and $\alpha^\tau \in [0,1]$ controls the transition from the current state toward the target. The corresponding marginal distribution at an intermediate timestep can be expressed as

$$q(\mathbf{z}^\tau \mid \mathbf{z}^0, \mathbf{y}) = \mathrm{Cat}\left(\mathbf{z}^\tau; \bar{\alpha}^\tau \mathbf{z}^0 + (1-\bar{\alpha}^\tau)\mathbf{y}\right), \tag{12}$$

where $\bar{\alpha}^\tau = \prod_{s=0}^{\tau-1} \alpha^s$, with $\bar{\alpha}^0 = 1$. This construction provides an explicit stochastic interpolation between the initial and target categorical states. The cosine schedule determines the rate of interpolation, while the target-conditioned transition kernel ensures that the reference process reaches the prescribed target when $\bar{\alpha}^T = 0$. Its primary role in NonMBB is to provide a tractable discrete probabilistic path for conditional bundle generation, which is further adapted through token-wise non-co-progressive scheduling during inference.

C Implementation Details

C.1 Full Hyperparameter Configuration

Table 6 provides the complete hyperparameter configuration used in all experiments unless explicitly stated otherwise.

C.2 Data Preprocessing Details

Spotify. The raw Spotify Million Playlist Dataset [9] contains playlists of highly variable length. In the DDBC preprocessing pipeline, three fixed-length subsets are constructed with $k \in \{30, 60, 90\}$. Playlists containing fewer than k tracks are discarded, while longer playlists are truncated to a contiguous subsequence of length k. Specifically, the starting position is uniformly sampled from the valid range $s \sim \mathcal{U}\{1, \ldots, |b| - k + 1\}$. The resulting playlists are partitioned into non-overlapping training, validation, and test sets. Following the DDBC experimental protocol, item-swapping augmentation is applied during training with a swap ratio of 0.4 to reduce sensitivity to item ordering.

Table 6: Full hyperparameter configuration of NonMBB.

Category	Hyperparameter	Value
Architecture	Backbone	Diffusion Transformer
	Number of Transformer blocks	6
	Hidden dimension d	64
	Self-attention heads	8
	Time conditioning	Adaptive layer normalization
RVQ Tokenizer	Number of levels L	4
	Item encoder	CLHE [35]
	Indexing	Faiss residual quantizer
Training	Optimizer	AdamW
	Learning rate	3×10^{-4}
	(β_1, β_2)	$(0.9, 0.999)$
	Weight decay	0
	Gradient clipping	1.0
	Total training steps	20,000
	EMA decay	0.9999
Dataset	Input-predict ratio $\lvert b_x \rvert : \lvert b_y \rvert$	1:1
	Candidate ratio ρ	100
	Swap ratio (Spotify)	0.4
	Swap ratio (POG)	0.8

Under the default $1:1$ observed-to-target ratio, each playlist is divided into an observed subset b_x and a target subset b_y, with $|b_x| = |b_y| = k/2$.

POG. We use the preprocessed dense POG variant [10] released by DDBC [52], which contains fashion outfits represented as fixed-length bundles of $k = 4$ items. Under the default evaluation setting, each outfit contains two observed items and two target items, i.e., $|b_x| = |b_y| = 2$. Following DDBC, item-swapping augmentation is performed during training with a swap ratio of 0.8.

Candidate pool construction. For all experiments, the candidate pool is constructed as $\mathcal{S} = \boldsymbol{b}_y \cup \mathrm{Random}[(\rho-1)|\boldsymbol{b}_y|, \mathcal{I} \setminus \bar{\boldsymbol{b}}]$ with default $\rho = 100$. Negative items are sampled uniformly at random from the full item catalog excluding all items in the complete bundle $\bar{\boldsymbol{b}}$. The same candidate pool is used across all compared methods to ensure fair comparison.

D Additional Experiments

D.1 Sensitivity to Entropy Threshold Strategy

We analyze the effect of different routing strategies on NonMBB using Spotify$_{k=60}$ with a single sampling attempt ($A = 1$). Our default strategy employs a dynamic threshold defined as the mean predictive entropy over unresolved target positions at each denoising step. We compare it against two alternatives: *i)* Top-50% routing, which assigns the half of unresolved positions with the highest predictive entropy to the slow schedule; and *ii)* Random routing, which independently assigns each

Table 7: Sensitivity to entropy-guided routing strategies on Spotify$_{k=60}$ ($A = 1$). The best results are highlighted in bold.

Routing Strategy	**F1↑**	**Jacc↑**	**OAS↑**
Random routing	0.359	0.232	0.698
Top-50% routing	0.367	0.238	0.700
Dynamic mean (Ours)	**0.369**	**0.239**	**0.701**

unresolved position to the slow or fast schedule with equal probability. These alternatives allow us to examine whether the performance gains arise from uncertainty-aware routing rather than asynchronous scheduling alone.

As shown in Table 7, dynamic mean thresholding achieves the best results across all three metrics, with F1, Jaccard, and OAS scores of 0.369, 0.239, and 0.701, respectively. Top-50% routing performs comparably, yielding scores of 0.367, 0.238, and 0.700, while random routing exhibits slightly larger performance reductions. These results suggest that leveraging prediction uncertainty to guide asynchronous refinement is beneficial compared with random slot assignment.

D.2 Robustness to Observed Item Selection Strategy

We evaluate the robustness of NonMBB to different observed item selection strategies on Spotify$_{k=30}$ with a single sampling attempt ($A = 1$). Specifically, we consider three protocols: *Prefix*, which selects the first 15 items as the observed subset; *Suffix*, which selects the last 15 items; and *Random*, which uniformly samples 15 items from each playlist without replacement. All remaining items are treated as prediction targets. To isolate the effect of observed item selection, the same trained checkpoint is evaluated under all three protocols without retraining.

As shown in Table 8, NonMBB consistently outperforms DDBC in terms of F1 and Jaccard while maintaining comparable OAS across the three selection strategies. Notably, NonMBB exhibits minimal performance variation, with F1 ranging from 0.349 to 0.352, Jaccard from 0.228 to 0.230, and OAS from 0.658 to 0.660. DDBC also demonstrates relatively stable performance, suggesting that both approaches are robust to changes in observed item positions. These results indicate that NonMBB maintains reliable bundle completion performance under different observation patterns, which is particularly desirable in practical scenarios where the available items may originate from arbitrary positions within a bundle.

Table 8: Robustness to observed item selection strategy on Spotify$_{k=30}$ ($A = 1$).

Strategy	diffusion-based DDBC			NonMBB		
	F1	Jacc	OAS	F1	Jacc	OAS
Prefix	0.282	0.176	0.660	0.352	0.230	0.660
Suffix	0.278	0.172	0.658	0.349	0.228	0.658
Random	0.279	0.173	0.659	0.350	0.229	0.658

D.3 Effect of Candidate Pool Size ρ

We vary the candidate ratio $\rho \in \{10, 20, 50, 100\}$ on Spotify$_{k=60}$ ($A = 1$) for diffusion-based DDBC and NonMBB. Larger ρ corresponds to a harder retrieval task with more distractors.

As shown in Table 9, NonMBB consistently outperforms all competing methods across the evaluated candidate ratios. Although the performance of both NonMBB and DDBC decreases as the candidate pool expands, NonMBB maintains a substantial advantage, with increasingly pronounced relative improvements. Specifically, its Jaccard improvement over baselines. Similar trends are observed for F1 and OAS, indicating that NonMBB retains stronger predictive performance as the number of candidate items increases. These findings suggest that NonMBB is comparatively robust to larger

Table 9: Performance under different candidate pool ratios $\rho \in \{10, 20, 50, 100\}$ on Spotify$_{k=60}$ ($A = 1$). Best results are highlighted in bold.

Model	$\rho = 10$			$\rho = 20$			$\rho = 50$			$\rho = 100$		
	F1	Jacc	OAS	F1	Jacc	OAS	F1	Jacc	OAS	F1	Jacc	OAS
BundleNAT	0.266	0.163	0.508	0.210	0.124	0.485	0.153	0.088	0.464	0.101	0.056	0.438
SASRec	0.292	0.194	0.519	0.200	0.126	0.474	0.200	0.126	0.475	0.089	0.054	0.310
DDBC	0.599	0.447	0.763	0.503	0.355	0.727	0.380	0.250	0.689	0.285	0.178	0.662
TIGER	0.191	0.151	0.326	0.107	0.080	0.295	0.108	0.081	0.296	0.129	0.076	0.413
NonMBB	**0.665**	**0.516**	**0.798**	**0.578**	**0.425**	**0.763**	**0.456**	**0.312**	**0.724**	**0.369**	**0.239**	**0.701**

candidate pools, although the observed gains cannot be attributed solely to the non-co-progressive scheduling mechanism.

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 accurately reflect the paper's contributions and scope.

   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: See limitation section.

   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: [Yes]

Justification: The paper provide the full set of assumptions and a complete (and correct) proof.

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: This paper discloses all the information required to replicate the main experimental results of the paper.

   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: [Yes]

Justification: The main experimental results described in the supplementary materials are all achievable.

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: See experiment set section.

   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: We conducted many experiments and they were significant.

   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: We mentioned it.

   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: We mentioned it.

   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: [Yes]

    Justification: We mentioned it.

    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: [Yes]

    Justification: We mentioned it.

    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: We mentioned it.

    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 release new 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 nor 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: The paper does not involve crowdsourcing nor research with human subjects.

    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: [Yes]

Justification: Only the polishing of the article.

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.