GoR: A Unified and Extensible Generative Framework for Ordinal Regression

Anonymous authors
Paper under double-blind review

Abstract

Ordinal Regression (OR), which predicts the target values with inherent order, underpins a wide spectrum of applications from computer vision to recommendation systems. The intrinsic ordinal structure and non-stationary inter-class boundaries make OR fundamentally more challenging than conventional classification or regression. Existing approaches, predominantly based on Continuous Space Discretization (CSD), struggle to model these ordinal relationships, but are hampered by boundary ambiguity. Alternative rank-based methods, while effective, rely on implicit order dependencies and suffer from the rigidity of fixed binning. Inspired by the advances of generative language models, we propose **G**enerative **O**rdinal **R**egression (**GoR**), a novel generative paradigm that reframes OR as a sequential generation task. GoR autoregressively predicts ordinal segments until a dynamic $\langle \text{EOS} \rangle$, explicitly capturing ordinal dependencies while enabling adaptive resolution and interpretable step-wise refinement. To support this process, we theoretically establish a bias–variance decomposed error bound and propose the **Co**verage–**Di**stinctiveness Index (**CoDi**), a principled metric for vocabulary construction that balances quantization bias against statistical variance. The GoR framework is model-agnostic, ensuring broad compatibility with arbitrary task-specific architectures. Moreover, it can be seamlessly integrated with established optimization strategies for generative models at a negligible adaptation cost. Extensive experiments on **17** diverse ordinal regression benchmarks across **six** major domains demonstrate GoR's powerful generalization and consistent superiority over state-of-the-art OR methods.

1 Introduction

Ordinal Regression (OR), also referred to as ordinal classification, addresses the prediction tasks where the target categories (or values) exhibit inherent ordinal relationships. As shown in Fig. 1(a), this paradigm has broad applications across various domains such as computer vision (e.g., facial age estimation (Niu et al., 2016a; Li et al., 2022b), image aesthetic assessment (She et al., 2021; He et al., 2022)) and recommendation systems (e.g., watch time prediction (Sun et al., 2024; Zhao et al., 2024), lifetime value prediction (Drachen et al., 2018; Ma et al., 2018)). Unlike conventional multi-class classification and continuous regression, the fundamental challenge in OR lies in explicitly modeling two critical properties: (1) the inherent ordinal structure among output labels, and (2) the non-stationary nature of semantic boundaries between adjacent categories.

Previous OR works have predominantly relied on Continuous Space Discretization (CSD) (Wang et al., 2025), as illustrated in Fig. 1(b). This strategy quantizes the target output space, potentially continuous or fine-grained ordinal, into a finite set of ordered discrete bins. The model typically outputs a softmax probability distribution over these bins, mapped to a prediction via probability-weighted expectation. Essentially, CSD simplifies learning by transforming the problem into a multi-class classification. Under this framework, subsequent research mainly explores in two directions.

As shown in Fig. 1(c), one is to tackle ambiguous inter-class boundaries by enhancing discrimination of boundary-proximal samples through reference comparisons (Li et al., 2021; Shin et al., 2022). However, its performance critically depends on efficient reference selection, which is often governed by unstable heuristics and limits the gains in wide-range scenarios where combinatorial reference points escalate selection complexity. Another is rank-based that implicitly encodes the ordinality via

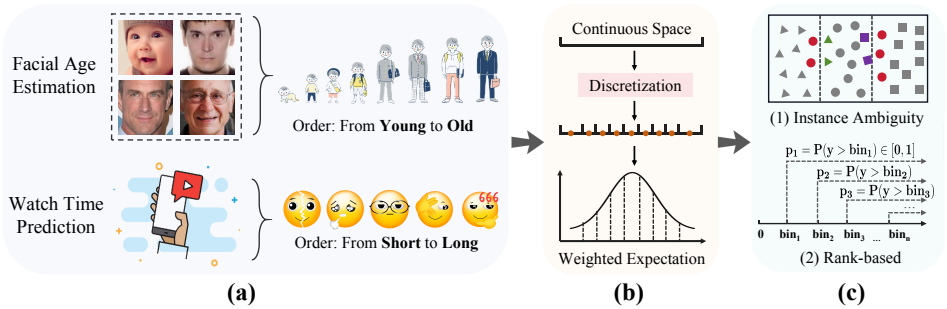


Figure 1: Overview of Universal Ordinal Regression. (a) Representative Ordinal Regression tasks with ordered labels. (b) The Continuous Space Discretization (CSD) workflow: discretizing continuous space into bins and using weighted expectation for prediction. (c) Two key research directions explored under the CSD framework.

label transformation, reframing OR into sequential binary subtasks (Niu et al., 2016a; Wang et al., 2023a). Despite empirical success with theoretical guarantees (Chen et al., 2017), its order dependency resides solely in label definitions, leaving bin-wise predictions independent (See Proposition 1). Besides, the predefined discretization introduces rigidity: it amplifies head-category errors in long-tailed distributions, frequently seen in real-world tasks, and makes performance highly sensitive to bin granularity — wide intervals blur semantics, while narrow ones induce sparsity (Sun et al., 2024).

Inspired by the advances in generative language models (Liu et al., 2024a), we propose **G**enerative **O**rdinal **R**egression (**GoR**), a novel framework that reformulates ordinal regression as sequential token generation. GoR autoregressively predicts tokens representing ordinal value segments, whose cumulative summation yields the final prediction upon generating the ⟨EOS⟩ token. This design explicitly models sequential ordinal dependencies through conditional generation while enabling adaptive resolution via dynamic ⟨EOS⟩ prediction, circumventing the rigidity of fixed binning. For instance, in facial age estimation, the model may first predict a coarse token (`50`), then finer adjustments (`+5`, `+3`), culminating in the estimate (`50+5+3=58` years). Each step progressively reduces prediction error by selecting tokens that provide an increasingly precise approximation. This step-wise refinement process mirrors human cognitive progression from coarse to precise estimation and implements successive approximation, offering interpretable intermediate predictions.

However, adapting this paradigm to general ordinal regression poses two key challenges. First, unlike purely compositional tokens in nature language process (NLP) tasks, GoR tokens uniquely encode dual semantics, *i.e.*, sequential ordinality and additive numerical relationships, necessitating specialized mechanisms to disentangle and exploit these intertwined properties. Since numerical values are infinitely decomposable and combinable, vocabulary design demands principled strategies to balance expressiveness and efficiency. Second, domain heterogeneity in ordinal label distributions, spanning diverse value ranges (e.g., $[0, 1]$ for quality scores vs. $[0, 100]$ for age estimation) and label granularity types (discrete vs. continuous), requires robust cross-domain generalization capabilities. Hence, effectively adapting the autoregression mechanism to universal OR tasks requires systematic vocabulary design and label decomposition strategies.

Through bias-variance decomposition, we derive a closed-form Mean Squared Error (MSE) bound that quantifies token selection trade-off between quantization bias and statistical variance, providing a theoretical foundation for vocabulary design. Building on this, we further propose the **Co**verage–**Di**stinctiveness Index (**CoDi**) to optimize token selection — maximizing coverage (bias minimization) while suppressing common segments (variance reduction) — yielding a compact, recoverable vocabulary that enhances cross-task adaptability. Besides, as a model-agnostic framework, GoR's core component, i.e., the encoder-decoder architecture, permits flexible substitution with task-specific implementations and can be seamlessly integrated with existing optimization strategies (Bengio et al., 2009; Goodman et al., 2020; Shao et al., 2024) at a negligible adaptation cost, thereby ensuring its flexibility and extensibility for broader applications.

Our contributions are fourfold: (i) We expose the theoretical limitations of prevailing rank-based methods under the CSD paradigm and, in turn, propose the first generative formulation of ordinal regression as an autoregressive sequence generation task. (ii) We introduce GoR, a unified framework that models sequential ordinal dependencies via dynamic ⟨EOS⟩-terminated token generation,

offering adaptive resolution and interpretable step-wise refinement. (iii) We establish a theoretical foundation based on MSE decomposition, accompanied by the Coverage–Distinctiveness Index (CoDi) to optimize the token vocabulary by bias-variance trade-off. (iv) We perform extensive experiments across 17 ordinal regression benchmarks spanning six domains, demonstrating GoR's strong generalization and consistent superiority over SOTA baselines.

The remainder of this paper is organized as follows. Section 2 formalizes the problem, theoretically analyzes rank-based methods' limitation, and introduces the proposed GoR framework, including its error-bound formulation, vocabulary construction, ordinal target sequencing, and encoder–decoder design. Section 3 reports extensive experiments and analyses across diverse domains. Section 4 concludes the paper. Due to space limit, we provide related work review in Appendix. B.

2 METHOD

2.1 PROBLEM DEFINITION

We formalize the learning problem on a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i denotes an input instance of heterogeneous modalities (e.g., visual data in computer vision or multimodal embeddings in recommendation systems), $y_i \in \mathbb{R}_{\geq 0}$ represents its ordinal label and N is the number of instances. The fundamental objective of the ordinal regression task is to learn a function $g(\cdot)$ that accurately maps the input x_i to its associated ordinal label y_i, i.e., $y_i = g(x_i)$. **GoR** reformulates ordinal regression via sequence generation by establishing two key mappings:

- **Ordinal target sequencing.** Encode label y_i into variable-length token sequence $s_i = \{s_i^t\}_{t=1}^{T_i}, s_i^t \in \mathcal{V}$, where $\mathcal{V}$ is a predefined vocabulary $\{w_j\}_{j=1}^{V}$, T_i is the length of s_i, each token w_j represents a value segment and V denotes the vocabulary size.
- **Sequence scoring.** Decode sequences back to the label space via $y_i = r(s_i) = \sum_{t=1}^{T_i} \phi(s_i^t)$, where $\phi : \mathcal{V} \to \mathbb{R}$ is a token-value lookup table.

Following standard sequence modeling practice, we extend the vocabulary $\mathcal{V}$ with three control tokens: ⟨SOS⟩, ⟨EOS⟩, and ⟨PAD⟩. ⟨SOS⟩ and ⟨EOS⟩ are appended at the beginning and end of the sequence, respectively. ⟨PAD⟩ is utilized to pad sequence length for parallel processing. As these tokens do not carry semantic meaning within the label space (i.e., $\phi(w) = 0, w \in \{\langle\text{SOS}\rangle, \langle\text{EOS}\rangle, \langle\text{PAD}\rangle\}$), they are omitted from the following mathematical formulations for enhanced clarity.

2.2 THEORETICAL ANALYSIS

Our theoretical analysis consists of two parts: characterizing the limitation of rank-based CSD approaches, and establishing GoR's theoretical foundation via MSE decomposition, which derives a closed-form MSE bound to guide vocabulary design.

Limitation of rank-based methods.

Assumption 1. *Define a sequence of binary random variables $\mathbf{B}_i^m = 1(y_i > c_m)$, where $1(\cdot)$ denotes the indicator function and c_m is the right boundary of the m-th interval. Then $\mathbf{B}_i = \{\mathbf{B}_i^1, \ldots, \mathbf{B}_i^M\}$ constitutes a set of non-mutually exclusive binary decisions that together describe the position of y_i. Rank-based methods approximate the true conditional distribution $P_{true}(\mathbf{B}_i \mid \mathbf{x}_i) = \prod_{m=1}^{M} P(\mathbf{B}_i^m | \mathbf{B}_i^{<m}, \mathbf{x}_i)$ by assuming conditional independence across all binary decisions: $P_{naive}(\mathbf{B}_i | \mathbf{x}_i) = \prod_{m=1}^{M} P(\mathbf{B}_i^m | \mathbf{x}_i)$. Based on this factorization, the final prediction is obtained as $\hat{y}_i = \sum_{m=1}^{M} P(B_i^m = 1) \cdot (c_m - c_{m-1})$.*

Building on this conditional independence assumption, we quantify the resulting approximation error:

Proposition 1 (Independence Limitation in Rank-based CSD methods)**.**

$$D_{\text{KL}}\left(P_{true}(\mathbf{B}_i|\mathbf{x}_i) \| P_{naive}(\mathbf{B}_i|\mathbf{x}_i)\right) = \sum_{m=1}^{M} \mathbb{E}_{\mathbf{B}_i^{<m}}[D_{KL}^{(m)}], \tag{1}$$

where $D_{\text{KL}}^{(m)}$ measures the divergence between $P(\mathbf{B}_i^m|\mathbf{x}_i)$ and $P(\mathbf{B}_i^m|\mathbf{B}_i^{<m}, \mathbf{x}_i)$.

The complete derivations are provided in Appendix C. Proposition 1 reveals that naive discretization exhibits systematic modeling errors stemming from its inability to capture inter-interval dependencies.

This phenomenon manifests as an approximation error quantified by cumulative KL divergence scaling with the conditional mutual information between adjacent intervals. To overcome this limitation, we recast ordinal regression through an autoregressive framework that explicitly captures sequential dependencies among latent tokens.

MSE bound. We denote $\phi(s_i^t)$ as a discrete random variable C_i^t, which takes values in $\phi(w_j)_{j=1}^V$. The model aims to approximate C_i^t with its prediction $\hat{C}_i^t$. Let $B = \max_t |\mathbb{E}[\hat{C}_i^t \mid \theta] - C_i^t|$ denote the maximum per-step bias, and define the variance term $V_{var} = \max \mathbb{V}(C_i^t) \leq \frac{(w_{\max} - w_{\min})^2}{4}$, where $[w_{\min}, w_{\max}]$ is the range of $\{w_j\}_{j=1}^V$.

Theorem 1 (Error Bound of Generative Ordinal Regression)**.** *By a bias-variance decomposition (ignoring irreducible noise), the mean squared error of GoR satisfies:*

$$\mathbb{E}\big[(\hat{y}_i - y_i)^2\big] \leq T_i^2 B^2 + T_i^2 V_{var} \leq T_i^2 B^2 + T_i^2 \frac{(w_{max} - w_{min})^2}{4} \tag{2}$$

The detailed derivation and illustrations is in Appendix D. This error bound demonstrates that minimizing prediction error requires coordinated control of three critical factors: (i) token-sequence length T_i, (ii) maximum per-step bias B, and (iii) per-step variance V_{var}. Guided by these findings, we formulate three axiomatic principles for vocabulary design: (1) $\mathcal{V}$ must support approximation of all target values $\{y_i\}_{i=1}^N$ through finite unique tokens, ensuring bounded approximation bias. (2) Joint optimization of bias and variance via dual mechanisms—preventing sample imbalance bias through coverage constraint while suppressing variance via vocabulary sparsity control. (3) Parametric invariance across datasets, enforcing robustness to distribution shifts through scale-agnostic token contributions. These principles collectively ensure rigorous error control while maintaining practical applicability.

2.3 Vocabulary Construction

We initialize the vocabulary $\mathcal{W} = \{w_j\}_{j=1}^W$ through a quantile-based selection strategy, which iteratively selects tokens based on a fixed percentile of the remaining label values, and subtracts them from the exceeding labels until residuals are negligible (Details in Alg. 2 of Appendix E). This initialization ensures comprehensive coverage of the observed value distribution while introducing computational challenges due to excessive vocabulary size. We therefore develop a principled pruning strategy based on our proposed *Coverage-Distinctiveness Index* (CoDi) for tokens as follows:

$$\mathrm{CoDi}_j = \underbrace{\left(\frac{1}{N}\sum_{i=1}^{N}\frac{\mathrm{count}(w_j, s_i)}{T_i}\right)}_{\text{Coverage}} \cdot \underbrace{\log\frac{N}{|\{\, i \mid w_j \in s_i\}| + 1}}_{\text{Distinctiveness}} \tag{3}$$

Here $\mathrm{count}(w_j, s_i)$ is the count of token w_j in the sequence s_i while $|\{\, i \mid w_j \in s_i\}|$ denotes the number of sequences containing w_j. In CoDi, the *Coverage* term measures token usage frequency, which affects approximation bias, while the *Distinctiveness* term evaluates token uniqueness, influencing model variance.

Based on CoDi, we design a top-down vocabulary pruning strategy in Alg. 1: starting with the initial vocabulary, we iteratively remove tokens with the lowest CoDi in the initial vocabulary $\mathcal{W}$. After each removal, the retained percentage β and threshold ϵ are utilized to control vocabulary size while preserving representational fidelity, achieving a favorable trade-off between computational efficiency and modeling power. The refined vocabulary $\mathcal{V}$[1] then serves as the foundation for formalizing *Ordinal Target Sequencing*.

Algorithm 1: Vocabulary pruning with CoDi

Input: Label set $Y = \{y_i\}_{i=1}^N$; Sequence set $\{s_i\}_{i=1}^N$; Threshold ϵ; Initial vocabulary $\mathcal{W} = \{w_j\}_{j=1}^W$; Minimum percentage of initial vocabulary retained β

Output: Pruned vocabulary $\mathcal{V}$ with $err < \epsilon$

$\mathcal{V} \leftarrow \mathcal{W}$;
Compute CoDi for all $w_j \in \mathcal{V}$;
$err \leftarrow$ evaluate($\mathcal{V}$);
while $err < \epsilon$ ***and*** $|\mathcal{V}| \geq \beta|\mathcal{W}|$ **do**
  $w^- \leftarrow \arg\min_{w_j \in \mathcal{V}} \mathrm{CoDi}_j$;
  $\mathcal{V} \leftarrow \mathcal{V} \setminus \{w^-\}$;
  Update sequence set $\{s_i\}_{i=1}^N$ based on $\mathcal{V}$;
  Update error metric $err \leftarrow \max\{\frac{y_i - r(s_i)}{y_i}\}_{i=1}^N$;
end
return $\mathcal{V}$;

[1]Without loss of generality, the token indices in the vocabulary are also sorted in descending numerical order.

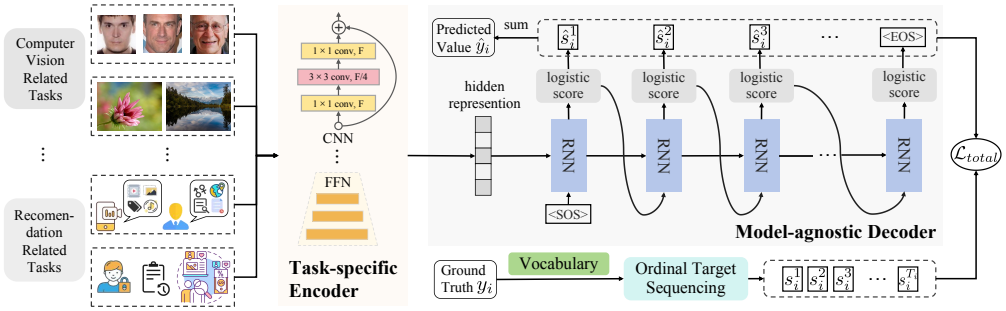


Figure 2: The framework of our proposed *Generative Ordinal Regression* (GoR), which adopts a flexible encoder-decoder architecture with the model-agnostic nature of both the encoder and decoder.

2.4 Ordinal Target Sequencing

Ordinal target sequencing aims to encode each target y_i into a token sequence $s_i = \{s_i^1, ..., s_i^{T_i}\}$ that preserves semantic fidelity while ensuring efficient learning, guided by three axiomatic principles:

1. **Accuracy**: $\frac{|y_i - r(s_i)|}{y_i} \leq \epsilon$, where ϵ balances precision and practical feasibility;
2. **Efficiency**: use the shortest possible sequence length T_i to simplify the learning diffculties;
3. **Monotonicity**: enforce coarse-to-fine refinement with $s_i^t \geq s_i^{t+1}$.

Building on these principles, we develop a greedy decomposition algorithm that iteratively selects the largest admissible token $s_i^t \in \mathcal{V}$ by satisfying $s_i^t = \max\left\{w \in \mathcal{V} \mid w \leq y_i - \sum_{k=1}^{t-1} \phi(s_i^k)\right\}$, which terminates when the residual falls below ϵ. This procedure guarantees: (i) minimal T_i for given ϵ by design, (ii) monotonic token values ensured by the decomposition process, and (iii) $\mathcal{O}(|\mathcal{V}|)$ time complexity via pre-sorted vocabulary search. The resultant sequences provide compact yet precise representations while maintaining generation consistency across the dataset.

2.5 Framework

As illustrated in Fig. 2, GoR employs a model-agnostic encoder-decoder architecture with two key components: (1) a task-specific encoder for feature extraction, and (2) an architecture-agnostic autoregressive decoder for sequential prediction. The encoder adapts to arbitrary input modalities (e.g., text, images, or tabular), while the decoder generalizes across sequence modeling paradigms — compatible with both RNN-based (Schuster & Paliwal, 1997; Chung et al., 2014) and Transformer-based architectures (Vaswani et al., 2017)[2].

2.5.1 Architecture

Encoder. The encoder is input modality-specific. Structured feature vectors employ Feedforward Network (FFN), while images use convolutional networks like ResNet (He et al., 2016) or ViT (Dosovitskiy et al., 2020). Formally, the encoder maps x_i to a latent representation $h_i = \text{Encoder}(x_i) \in \mathbb{R}^{L \times D}$ where D denotes the feature dimension and L is the input length expected by the decoder.

Decoder. Conditioned on h_i, the decoder generates a token sequence $\hat{s}_i = (\hat{s}_i^1, \ldots, \hat{s}_i^{T_i})$ through autoregressive factorization as follows:

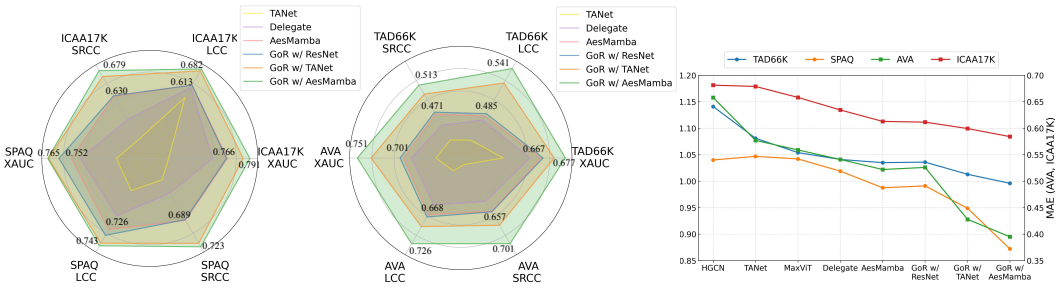
$$P_\theta(s_i \mid h_i) = P_\theta(s_i^1, ..., s_i^{T_i} \mid h_i) = \prod_{t=1}^{T_i} P_\theta(s_i^t \mid h_i, \hat{s}_i^{<t}) \tag{4}$$

where θ denotes the model parameters. At each step t, the predicted token $\hat{s}_i^t$ is sampled as:

$$\hat{s}_i^t = \arg\max_{w \in \mathcal{V}} P_\theta(w | h_i, \hat{s}_i^{<t}) = \arg\max_{w \in \mathcal{V}} \text{Softmax}\big(f_\theta(h_i, \hat{s}_i^{<t})\big) \tag{5}$$

Here, f_θ outputs the unnormalized logits, while P_θ denotes the normalized probability distribution.

[2]To intuitively illustrate the generative regression process, Fig. 2 depicts an RNN-based decoder, which can be replaced with any other architecture.



(a) (b)

Figure 3: Aesthetics assessment results on four benchmarks (Best viewed in color).

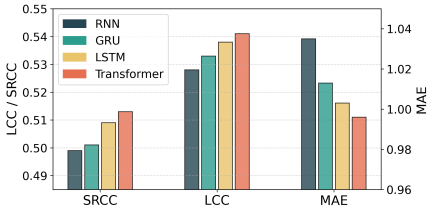


Figure 4: Performanceon the TAD66K dataset with different decoder architectures.

Method	Criteo-SSC		Kaggle	
	MAE ↓	SRCC ↑	MAE ↓	SRCC ↑
Two-stage (Drachen et al., 2018)	21.719	0.2386	74.782	0.431
MTL-MSE (Ma et al., 2018)	21.190	0.2478	74.065	0.433
ZILN (Wang et al., 2019b)	20.880	0.2434	72.528	0.524
MDME (Li et al., 2022a)	16.598	0.2269	72.900	0.516
MDAN (Liu et al., 2024b)	20.030	0.2470	73.940	0.437
OptDist (Weng et al., 2024)	15.784	0.2505	70.929	<u>0.525</u>
HiLTV (Xu et al., 2025)	<u>14.764</u>	<u>0.2645</u>	<u>69.331</u>	0.512
Our GoR	**12.996**	**0.3026**	**67.035**	**0.533**

Table 1: Performance comparison on LTV datasets with the MAE and SRCC metrics.

Extensibility. Beyond flexible substitution with task-specific implementations, GoR can integrate existing optimization strategies with negligible adaptation cost, including curriculum learning (Bengio et al., 2009), N-gram (Goodman et al., 2020), and reinforcement learning such as GRPO (Shao et al., 2024). The detailed discussion of these extensions is provided in Sec. 3.2.4.

2.5.2 Training and Inference

Training loss. The primary objective is to minimize the negative likelihood:

$$\mathcal{L}_{ce} = -\sum_{i=1}^{N} \log P_\theta(s_i \mid h_i) = -\sum_{i=1}^{N}\sum_{t=1}^{T_i} \log P_\theta(s_i^t \mid h_i, \hat{s}_i^{<t}) \tag{6}$$

To incorporate ordinal relationships into the predictions, we follow (Liu et al., 2018b) to employ the Huber loss (Huber, 1992) as:

$$\mathcal{L}_{huber} = \mathcal{L}_\delta(y_i, \hat{y}_i) = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2 & \text{if} |y_i - \hat{y}_i| \le \delta, \\ \delta \cdot (|y_i - \hat{y}_i| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \tag{7}$$

where δ balances sensitivity and robustness to outliers, and the final objective is:

$$\mathcal{L}_{total} = \mathcal{L}_{ce} + \lambda \cdot \mathcal{L}_{huber} \tag{8}$$

where λ is a hyperparameter that balances the two losses.

Inference process. The encoder processes the input x_i to derive a hidden representation h_i analogous to the training phase. The decoder then initiates the generation of sequence $\hat{s}_i$ autoregressively, beginning with the $\langle$SOS$\rangle$ token and proceeding until the $\langle$EOS$\rangle$ token is generated. Finally, the predicted value is computed by $\hat{y}_i = \sum_{t=1}^{T_i} \phi(\hat{s}_i^t)$.

3 Experiments

We first present GoR's overall performance across multiple domains, then introduce in-depth analyses of architectural choices, interval-wise performance, distributional visualization, extensibility, vocabulary ablation, and token semantics to elucidate its underlying mechanisms. Evaluation metrics include Mean Absolute Error (MAE), Cumulative Score (CS), XAUC (Zhan et al., 2022), Linear Correlation Coefficient (LCC), and Spearman's Rank Correlation Coefficient (SRCC). Due to space limit, detailed metric definitions, implementation settings, and additional results are moved to Appendix F.

Table 2: Performance comparison among different approaches on WTP task for four datasets.

Method	KuaiRec		KuaiRand		CIKM16		Indust.	
	MAE ↓	XAUC ↑	MAE ↓	XAUC ↑	MAE ↓	XAUC ↑	MAE ↓	XAUC ↑
VR	7.634	0.534	12.349	0.521	1.039	0.641	46.343	0.588
WLR (Covington et al., 2016)	6.047	0.545	11.582	0.529	0.998	0.672	-	-
D2Q (Zhan et al., 2022)	5.426	0.565	10.564	0.537	0.899	0.661	-	-
CWM (Zhao et al., 2024)	3.452	0.580	8.696	0.561	0.891	0.662	-	-
TPM (Lin et al., 2023)	3.456	0.571	9.573	0.542	0.850	0.676	41.486	0.593
CREAD (Sun et al., 2024)	3.307	0.594	9.487	0.549	0.865	0.678	39.979	0.597
SWaT (Yang et al., 2025)	3.438	0.585	9.553	0.544	0.857	0.685	40.995	0.591
Our GoR	**3.194**	**0.616**	**7.032**	**0.567**	**0.812**	**0.694**	**38.528**	**0.604**

3.1 OVERALL PERFORMANCE

3.1.1 IMAGE AESTHETICS ASSESSMENT (IAA)

Following (He et al., 2023), we select 14 representative and state-of-the-art (SOTA) baselines for evaluation on four datasets: TAD66K (He et al., 2022), AVA (Murray et al., 2012), ICAA17K (He et al., 2023), and SPAQ (Fang et al., 2020), with four metrics: MAE, XAUC, LCC, and SRCC. Recognizing the critical importance of visual features in the IAA task (He et al., 2022), we evaluate GoR by employing three distinct encoder backbones: ResNet50 (He et al., 2016), a representative legacy architecture (He et al., 2022), and a recent SOTA model (Gao et al., 2024).

Performance. Fig. 3 demonstrates GoR's superiority over SOTA methods across four metrics, with three critical observations: (1) Compatibility: GoR with standard ResNet50 matches SOTA models with expert-designed architectures; (2) Robustness: Even paired with outdated TANet, GoR significant improvements over the SOTA methods—a compelling result given the critical dependence of IAA tasks on feature quality; (3) Synergy: Combined with a modern AesMamba encoder, GoR achieves new SOTA (detailed metrics in Appendix F.2.2). These results validate the universal efficacy of our generative ordinal modeling paradigm across different encoder architectures.

3.1.2 LIFE TIME VALUE PREDICTION (LTV)

Following (Weng et al., 2024), we evaluate GoR on the Criteo-SSC and Kaggle datasets with MAE and SRCC metrics. A Feed-Forward Network (FFN) serves as the encoder of GoR. Details regarding the encoder architecture, the baseline methods, and datasets are in Appendix F.4.

Performance. Across both ordinal (SRCC) and numeric (MAE) metrics, GoR consistently surpasses existing methods (see Tab. 1). It improves HiLTV by 3.43% in MAE (reduction) and 3.94% in SRCC on Kaggle. Notably, for Criteo-SSC, GoR achieves a 13.6% reduction in MAE and a 12.59% improvement in SRCC compared to the SOTA method HiLTV, substantiating the superiority of GoR.

3.1.3 WATCH TIME PREDICTION (WTP)

Following (Lin et al., 2023; Zhao et al., 2024), three publicly available datasets (CIKM16, KuaiRec (Gao et al., 2022a) and KuaiRand (Gao et al., 2022b)) and one industrial dataset from a real-world short-video app are used to evaluate the proposed GoR, with the metrics of MAE and XUAC. The encoder is consistent with the LTV task implementation in Sec. 3.1.2, and comprehensive details about datasets and compared baselines are in Appendix F.3.1.

Performance. We compare GoR with 6 existing methods and the results are presented in Tab. 2. Compared to the second-best method (marked in *underline*), GoR achieves relative reductions in MAE of 3.36% (KuaiRec), 1.91% (KuaiRand), and 4.12% (CIKM16), alongside relative improvements in XAUC of 1.07% (KuaiRec), 3.37% (KuaiRand), and 1.92% (CIKM16). The comprehensive improvements in both MAE and XAUC substantiate the superiority of the GoR method. Besides, GoR exhibits a 3.629% relative decrease in MAE and a 1.001% improvement in XAUC on the Indust dataset, demonstrating its potential to significantly enhance real-world user experiences.

3.1.4 FACIAL AGE ESTIMATION (FAE)

FAE is a discrete ordinal regression problem (e.g. 0-100 years old) unlike some previous tasks that involve continuous ordinal labels. Consistent with the datasets, baselines, and evaluation protocol used in (Paplhám et al., 2024), we evaluate GoR on four FAE datasets (UTKFace (Zhang et al., 2017), FG-NET (Lanitis et al., 2002), MORPH (Ricanek & Tesafaye, 2006), and CACD (Chen et al., 2014)) with MAE and CS (tolerance $L = 5$) metrics, and use FaRL (Zheng et al., 2022) as the encoder.

Table 3: Facial age estimation results on four benchmarks

Method	UTKFace		FG-NET		MORPH		CACD	
	MAE ↓	CS(%) ↑	MAE ↓	CS(%) ↑	MAE ↓	CS(%) ↑	MAE ↓	CS(%) ↑
OR-CNN (Niu et al., 2016a)	4.40	63.67	5.09	83.80	2.83	61.97	4.01	73.41
DLDL (Gao et al., 2017)	4.39	63.65	5.26	83.83	2.81	62.43	3.96	73.37
SORD (Diaz & Marathe, 2019)	4.36	64.25	5.59	82.83	2.81	61.31	3.96	73.48
Mean-Var. (Pan et al., 2018)	4.42	63.36	5.45	83.43	2.83	62.87	4.07	72.98
Unimodal (Li et al., 2022b)	4.47	62.67	5.13	83.97	2.78	63.15	4.10	73.55
POE (Li et al., 2021)	4.43	63.37	5.24	83.56	2.83	62.45	4.02	73.08
FaRL (Paplhám et al., 2024)	3.87	65.38	4.95	84.52	3.04	63.49	3.96	74.18
Our GoR	**3.43**	**66.58**	**4.68**	**85.66**	**2.69**	**64.95**	**3.73**	**75.29**

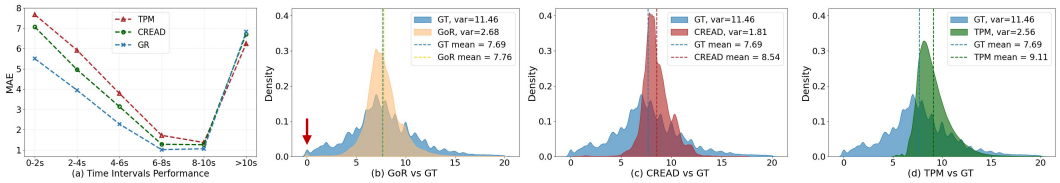


Figure 5: (a) MAE comparison on the KuaiRec dataset across time intervals. (b-d) The distribution of predicted values by GoR, CREAD, and TPM, compared against that of the Ground Truth (GT).

Performance. The results in Tab. 3 show that our GoR model achieves SOTA performance, significantly surpassing all baselines across all datasets in both evaluation metrics. GoR delivers consistent and substantial improvements — MAE reductions between 5.8% (MORPH) and 14.1% (FG-NET), and CS improvements between 1.5% (CACD) and 2.7% (FG-NET) — demonstrating its generality over diverse ordinal regression tasks, from continuous to discrete labels. Due to space constraint, experiments of Historical Image Dating, another discrete OR task, are presented in Appendix F.6.

3.2 Further Analysis

3.2.1 Architecture-Agnostic Analysis

Fig. 4 shows the results of decoder architecture ablation on TAD66K under IAA task: All variants including RNN (Schuster & Paliwal, 1997), GRU (Chung et al., 2014), LSTM (Hochreiter & Schmidhuber, 1997), and Transformer (Vaswani et al., 2017) surpass the SOTA methods, with Transformer achieving the optimal performance, validating GoR's architectural independence.

3.2.2 Performance Gain Analysis of GoR

We analyze model performance across ground truth (GT) time intervals on KuaiRec, where approximately 80% of videos have GT $\leq$10s. Fig. 5(a) shows that GoR significantly outperforms CREAD and TPM on frequent short and medium watch times, with slightly lower performance only on the last >10s interval, where it lags behind TPM (detailed reasons explained in Appendix F.3.3). Further insights are observed by examining the predicted value distributions (Fig. 5(b-d)). GoR achieves superior distribution alignment through its coarse-to-fine tokenization. GoR's $\langle$EOS$\rangle$ token enables precise near-zero GT predictions, highlighting its flexibility for adaptive resolution. Conversely, the overestimation in CREAD and TPM stems from rigid discretization buckets, where large tail bucket spans disproportionately amplify errors for shorter GTs.

3.2.3 Vocabulary Analysis

Ablation study. Vocabulary initialization analysis in Tab. 4 reveals: (1) Quantile-based method outperforms manual (intervals $[1,3,5,7]\times10^n$), and binary strategies (intervals 2^n from 1) benefit from balanced token distributions (See Fig. 6(a)); (2) CoDi enhances all initialization strategies, which is linked to CoDi yielding a more balanced token frequency distribution by reducing token-level frequency variance and per-step bias B, consistent with theoretical expectations. (3) β sensitivity analysis in Fig. 6(b) reflects that decreasing β filters more tokens, and performance varies non-monotonically. This trend supports the bias-variance trade-off, showing initial size reduction suppresses variance, while excessive compactness increases bias, aligning with Theorem. 1.

Analysis of learned token semantics. We analyze token semantics via derived weighted numerical embeddings $e_i = \sum_{t=1}^{T_i} r_t \cdot E[\hat{s}_i^t, :]$, where $r_t = \frac{\phi(\hat{s}_i^t)}{y_i}$ weights token contributions and E represents

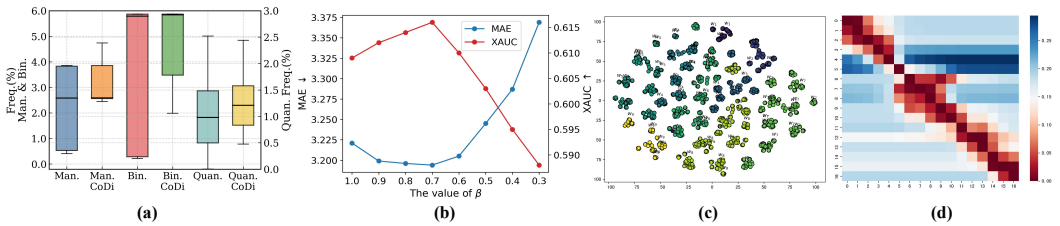


Figure 6: Vocabulary analysis on KuaiRec dataset for WTP task: (a) Token frequency distributions under different vocabulary strategies. (b) The sensitivity analysis of β. (c) t-SNE of weighted token embeddings. (d) Tokens similarity heatmap against numerical distance.

Table 4: Performance Across Vocabulary Construction Strategies (w/ vs. w/o CoDi)

Vocabulary design	KuaiRec		CIKM16	
	MAE ↓	XAUC ↑	MAE ↓	XAUC ↑
Manual (Man.)	3.281	0.604	0.825	0.685
Binary (Bin.)	3.268	0.605	0.821	0.687
Quantile (Quan.)	3.221	0.609	0.820	0.688
Man.+CoDi	3.253	0.610	0.819	0.689
Bin.+CoDi	3.239	0.611	0.815	0.691
Quan.+CoDi	3.194	0.616	0.812	0.694

Table 5: Compatibility with existing generative optimization strategies on WTP and LTV tasks.

	Strategy			KuaiRec		Criteo-SSC	
	TF	CL	NP	MAE ↓	XAUC ↑	MAE ↓	SRCC ↑
(a)	✓			3.359	0.588	16.198	0.252
(b)		✓		3.299	0.592	14.893	0.264
(c)			✓	3.241	0.604	13.996	0.276
(d)	✓		✓	3.208	0.612	13.068	0.292
(e)		✓	✓	3.194	0.616	12.996	0.303
(e) w/ DPO	-	-	-	3.185	0.620	12.438	0.309

token embeddings matrix. As shown in Fig. 6(c), instances predicting the same initial token form distinct clusters, and clusters for numerically similar initial tokens appear closer, indicating that GoR effectively captures the magnitude relationships among tokens. Fig. 6(d) visualizes pairwise token similarity based on probabilistic outputs, revealing a strong correlation: smaller numerical differences correspond to smaller predicted probabilities differences, indicating GoR effectively learns the intended magnitude relationships across the vocabulary. This structural encoding of ordinal relationships validates the generative paradigm's suitability for ordinal regression, positioning GoR as a promising foundation for future research in this direction.

3.2.4 Compatibility with Generative Optimization Strategies.

As discussed in Sec. 1, GoR can seamlessly accommodate optimization strategies for generative language model with negligible adaptation cost. Here, we assess this compatibility by incorporating several representative strategies prevalent in language models: 1) Teacher Forcing (TF) (Sutskever, 2014) is a training method that feeds the ground-truth token at step $t-1$ into the decoder at step t. 2) Curriculum Learning (CL) (Bengio et al., 2015) progressively shifts the training strategy from full TF to a strategy that closely mimics autoregressive inference. 3) N-gram Prediction (NP) (Goodman et al., 2020) simultaneously predicts N tokens to improve predictive lookahead and compensate for output-to-input gradients. 4) DPO[3] (Rafailov et al., 2023) is a reinforcement-learning-based strategy that optimizes model outputs via preference alignment. Our GoR requires no explicit reward model, instead using beam-search candidates with MAE against labels as implicit preference signals. The results in Tab. 5 indicate that this compatibility significantly enhances model performance. Crucially, these improvements are achieved without introducing additional model parameters, demonstrating a cost-effective and scalable enhancement that facilitates GoR's future exploration in wider applications.

4 Conclusion

This paper proposes **GoR**, the first generative framework for Ordinal Regression (OR), formulating OR as an autoregressive sequence generation task by predicting tokens for ordinal value segments autoregressively. This explicitly models sequential dependencies and enables adaptive resolution, overcoming the drawback of rigid binning. Supported by a bias-variance theoretical analysis and the CoDi metric for vocabulary optimization, GoR demonstrates SOTA performance on 17 diverse benchmarks spanning 6 domains, providing a strong baseline for future generative OR research.

[3]To avoid the prolonged training time associated with reinforcement-learning-based post-training, while effective, all results are reported using the efficient setup from line (e).

REFERENCES

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in Neural Information Processing Systems*, 2015.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 2009.

Bor-Chun Chen, Chu-Song Chen, and Winston H Hsu. Cross-age reference coding for age-invariant face recognition and retrieval. In *Computer Vision–ECCV 2014*, 2014.

Shixing Chen, Caojin Zhang, Ming Dong, et al. Using ranking-cnn for age estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

Kyunghyun Cho. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, 2016.

Raul Diaz and Amit Marathe. Soft labels for ordinal regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

Pusen Dong, Chenglong Cao, Xinyu Zhou, Jirong You, Linhe Xu, Feifan Xu, and Shuo Yuan. Halo: Hindsight-augmented learning for online auto-bidding. 2025.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Anders Drachen, Mari Pastor, Aron Liu, Dylan Jack Fontaine, Yuan Chang, Julian Runge, Rafet Sifa, and Diego Klabjan. To be or not to be... social: Incorporating simple social features in mobile game customer lifetime value predictions. In *Proceedings of the Australasian Computer Science Week Multiconference*, 2018.

Yao Du, Qiang Zhai, Weihang Dai, and Xiaomeng Li. Teach clip to develop a number sense for ordinal regression. In *European Conference on Computer Vision*, pp. 1–17. Springer, 2024.

Yuming Fang, Hanwei Zhu, Yan Zeng, Kede Ma, and Zhou Wang. Perceptual quality assessment of smartphone photography. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

Bin-Bin Gao, Chao Xing, Chen-Wei Xie, Jianxin Wu, and Xin Geng. Deep label distribution learning with label ambiguity. *IEEE Transactions on Image Processing*, 2017.

Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. Kuairec: A fully-observed dataset and insights for evaluating recommender systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022a.

Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. Kuairand: An unbiased sequential recommendation dataset with randomly exposed videos. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022b.

Fei Gao, Yuhao Lin, Jiaqi Shi, Maoying Qiao, and Nannan Wang. Aesmamba: Universal image aesthetic assessment with state space models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024.

Zhen Gong, Lvyin Niu, Yang Zhao, Miao Xu, Haoqi Zhang, Zhenzhe Zheng, Zhilin Zhang, Rongquan Bai, Chuan Yu, Jian Xu, et al. Mebs: Multi-task end-to-end bid shading for multi-slot display advertising. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2024.

Sebastian Goodman, Nan Ding, and Radu Soricut. Teaforn: Teacher-forcing with n-grams. *arXiv e-prints*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Shuai He, Yongchang Zhang, Rui Xie, Dongxiang Jiang, and Anlong Ming. Rethinking image aesthetics assessment: Models, datasets and benchmarks. In *IJCAI*, 2022.

Shuai He, Anlong Ming, Yaqi Li, Jinyuan Sun, ShunTian Zheng, and Huadong Ma. Thinking image color aesthetics assessment: Models, datasets and benchmarks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

B Hidasi. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

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

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

Valentin Hofmann, Hinrich Schuetze, and Janet Pierrehumbert. An embarrassingly simple method to mitigate undesirable properties of pretrained language model tokenizers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, May 2022.

Vlad Hosu, Bastian Goldlucke, and Dietmar Saupe. Effective aesthetics prediction with multi-level spatially pooled features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*. 1992.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Shu Kong, Xiaohui Shen, Zhe Lin, Radomir Mech, and Charless Fowlkes. Photo aesthetics ranking network with attributes and content adaptation. In *Computer Vision–ECCV 2016*, 2016.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2018.

A. Lanitis, C.J. Taylor, and T.F. Cootes. Toward automatic simulation of aging effects on face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

Kunpeng Li, Guangcui Shao, Naijun Yang, Xiao Fang, and Yang Song. Billion-user customer lifetime value prediction: an industrial-scale solution from kuaishou. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022a.

Qiang Li, Jingjing Wang, Zhaoliang Yao, Yachun Li, Pengju Yang, Jingwei Yan, Chunmao Wang, and Shiliang Pu. Unimodal-concentrated loss: Fully adaptive label distribution learning for ordinal regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022b.

Wanhua Li, Xiaoke Huang, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Learning probabilistic ordinal embeddings for uncertainty-aware regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021.

Wanhua Li, Xiaoke Huang, Zheng Zhu, Yansong Tang, Xiu Li, Jie Zhou, and Jiwen Lu. Ordinalclip: Learning rank prompts for language-guided ordinal regression. *Advances in Neural Information Processing Systems*, 35:35313–35325, 2022c.

Xu Li, Michelle Ma Zhang, Zhenya Wang, and Youjun Tong. Arbitrary distribution modeling with censorship in real-time bidding advertising. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023.

Hairen Liao, Lingxiao Peng, Zhenchuan Liu, and Xuehua Shen. ipinyou global rtb bidding algorithm competition dataset. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, 2014.

Xiao Lin, Xiaokai Chen, Linfeng Song, Jingwei Liu, Biao Li, and Peng Jiang. Tree based progressive regression model for watch-time prediction in short-video recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Wenshuang Liu, Guoqiang Xu, Bada Ye, Xinji Luo, Yancheng He, and Cunxiang Yin. Mdan: Multi-distribution adaptive networks for ltv prediction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2024b.

Xiaofeng Liu, Yang Zou, Yuhang Song, Chao Yang, Jane You, and BV K Vijaya Kumar. Ordinal regression with neuron stick-breaking for medical diagnosis. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018a.

Yanzhu Liu, Adams Wai Kin Kong, and Chi Keong Goh. A constrained deep neural network for ordinal regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018b.

Yanzhu Liu, Fan Wang, and Adams Wai Kin Kong. Probabilistic deep ordinal regression based on gaussian processes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019a.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019b.

Xin Lu, Zhe Lin, Hailin Jin, et al. Rapid: Rating pictorial aesthetics using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia*, 2014.

Hongxu Ma, Kai Tian, Tao Zhang, Xuefeng Zhang, Chunjie Chen, Han Li, Jihong Guan, and Shuigeng Zhou. Generative regression based watch time prediction for video recommendation: Model and performance. *arXiv e-prints*, 2024.

Shuang Ma, Jing Liu, and Chang Wen Chen. A-lamp: Adaptive layout-aware multi-patch deep convolutional neural network for photo aesthetic assessment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018.

Naila Murray, Luca Marchesotti, and Florent Perronnin. Ava: A large-scale database for aesthetic visual analysis. In *2012 IEEE conference on computer vision and pattern recognition*, 2012.

Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal regression with multiple output cnn for age estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016a.

Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal regression with multiple output cnn for age estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016b.

Frank Palermo, James Hays, and Alexei A Efros. Dating historical color images. In *Computer Vision–ECCV 2012*, 2012.

Hongyu Pan, Hu Han, Shiguang Shan, and Xilin Chen. Mean-variance loss for deep age estimation from a face. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

Shengjun Pan, Brendan Kitts, Tian Zhou, Hao He, Bharatbhushan Shetty, Aaron Flores, Djordje Gligorijevic, Junwei Pan, Tingyu Mao, San Gultekin, et al. Bid shading by win-rate estimation and surplus maximization. 2020.

Jakub Paplhám, Vojt Franc, et al. A call to reflect on evaluation practices for age estimation: comparative analysis of the state-of-the-art and a unified benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 2023.

Jian Ren, Xiaohui Shen, Zhe Lin, Radomir Mech, and David J Foran. Personalized image aesthetics. In *Proceedings of the IEEE international conference on computer vision*, 2017.

Kan Ren, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, and Yong Yu. Deep landscape forecasting for real-time bidding advertising. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019.

Karl Ricanek and Tamirat Tesafaye. Morph: A longitudinal image database of normal adult age-progression. In *7th international conference on automatic face and gesture recognition (FGR06)*, 2006.

Craig W. Schmidt, Varshini Reddy, Haoran Zhang, Alec Alameddine, Omri Uzan, Yuval Pinter, and Chris Tanner. Tokenization is more than compression, 2024.

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 1997.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Dongyu She, Yu-Kun Lai, Gaoxiong Yi, and Kun Xu. Hierarchical layout-aware graph convolutional network for unified aesthetics assessment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021.

Kekai Sheng, Weiming Dong, Chongyang Ma, Xing Mei, Feiyue Huang, and Bao-Gang Hu. Attention-based multi-patch aggregation for image aesthetic assessment. In *Proceedings of the 26th ACM international conference on Multimedia*, 2018.

Nyeong-Ho Shin, Seon-Ho Lee, and Chang-Su Kim. Moving window regression: A novel approach to ordinal regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 2015.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019.

Jie Sun, Zhaoying Ding, Xiaoshuang Chen, Qi Chen, Yincheng Wang, Kaiqiao Zhan, and Ben Wang. Cread: A classification-restoration framework with error adaptive discretization for watch time prediction in video recommender systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

I Sutskever. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014.

Hossein Talebi and Peyman Milanfar. Nima: Neural image assessment. *IEEE Transactions on Image Processing*, 2018.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in Neural Information Processing Systems*, 2024.

Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *European conference on computer vision*, 2022.

Ashish Vaswani, Noam Shazeer, et al. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Jinhong Wang, Yi Cheng, Jintai Chen, TingTing Chen, Danny Chen, and Jian Wu. Ord2seq: regarding ordinal regression as label sequence prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023a.

Jinhong Wang, Jian Liu, Dongqi Tang, Weiqiang Wang, Wentong Li, Danny Chen, Jintai Chen, et al. Scalable autoregressive monocular depth estimation. *arXiv preprint arXiv:2411.11361*, 2024.

Jinhong Wang, Jintai Chen, Jian Liu, Dongqi Tang, Danny Z Chen, and Jian Wu. A survey on ordinal regression: Applications, advances and prospects. *arXiv preprint arXiv:2503.00952*, 2025.

Kai Wang, Xiaojun Quan, and Rui Wang. Biset: Bi-directional selective encoding with template for abstractive summarization. *arXiv preprint arXiv:1906.05012*, 2019a.

Rui Wang, Peipei Li, Huaibo Huang, Chunshui Cao, Ran He, and Zhaofeng He. Learning-to-rank meets language: Boosting language-driven ordering alignment for ordinal classification. *Advances in Neural Information Processing Systems*, 36:76908–76922, 2023b.

Xiaojing Wang, Tianqi Liu, and Jingang Miao. A deep probabilistic model for customer lifetime value prediction. *arXiv preprint arXiv:1912.07753*, 2019b.

Yunpeng Weng, Xing Tang, Zhenhao Xu, Fuyuan Lyu, Dugang Liu, Zexu Sun, and Xiuqiang He. Optdist: Learning optimal distribution for customer lifetime value prediction. *arXiv preprint arXiv:2408.08585*, 2024.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Dongling Xiao, Han Zhang, Yukun Li, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-gen: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. *arXiv preprint arXiv:2001.11314*, 2020.

Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. Vocabulary learning via optimal transport for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, August 2021.

Junwei Xu, Aisi Zheng, Ling Ding, Huangbin Zhang, Zhengwei Deng, Qun Yu, and Xiao-Ping Zhang. Hiltv: Hierarchical multi-distribution modeling for lifetime value prediction in online games. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025.

Shentao Yang, Haichuan Yang, Linna Du, Adithya Ganesh, Bo Peng, Boying Liu, Serena Li, and Ji Liu. Swat: Statistical modeling of video watch time through user behavior analysis. 2025.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 2019.

Shaked Yehezkel and Yuval Pinter. Incorporating context into subword vocabularies. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

Qinkai Yu, Jianyang Xie, Anh Nguyen, He Zhao, Jiong Zhang, Huazhu Fu, Yitian Zhao, Yalin Zheng, and Yanda Meng. Clip-dr: Textual knowledge-guided diabetic retinopathy grading with ranking-aware prompting. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 667–677. Springer, 2024.

Hui Zeng, Zisheng Cao, Lei Zhang, and Alan C Bovik. A unified probabilistic formulation of image aesthetic assessment. *IEEE Transactions on Image Processing*, 2019.

Ruohan Zhan, Changhua Pei, Qiang Su, Jianfeng Wen, Xueliang Wang, Guanyu Mu, et al. Deconfounding duration bias in watch-time prediction for video recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

Zhifei Zhang, Yang Song, et al. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

Haiyuan Zhao, Guohao Cai, Jieming Zhu, Zhenhua Dong, Jun Xu, and Ji-Rong Wen. Counteracting duration bias in video recommendation via counterfactual watch time. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024.

Jiyang Zheng, Yu Yao, Bo Han, Dadong Wang, and Tongliang Liu. Enhancing contrastive learning for ordinal regression via ordinal content preserved data augmentation. In *The Twelfth International Conference on Learning Representations*, 2024.

Yinglin Zheng, Hao Yang, Ting Zhang, Jianmin Bao, Dongdong Chen, Yangyu Huang, Lu Yuan, Dong Chen, et al. General facial representation learning in a visual-linguistic manner. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

Tian Zhou, Hao He, Shengjun Pan, Niklas Karlsson, Bharatbhushan Shetty, Brendan Kitts, Djordje Gligorijevic, San Gultekin, Tingyu Mao, Junwei Pan, et al. An efficient deep distribution network for bid shading in first-price auctions. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021.

Hancheng Zhu, Leida Li, Jinjian Wu, Sicheng Zhao, Guiguang Ding, and Guangming Shi. Personalized image aesthetics assessment via meta-learning with bilevel gradient optimization. *IEEE Transactions on Cybernetics*, 2020.

APPENDIX

A THE USE OF LARGE LANGUAGE MODELS (LLMS)

Generative AI tools (e.g., ChatGPT) were used solely to improve the manuscript's clarity and readability during the writing stage. These tools were not employed for generating any novel content, such as text, figures, tables, code, or experimental results. No generative AI was used in the conception, implementation, analysis, or evaluation of the research itself. The authors take full responsibility for the integrity and accuracy of the final manuscript.

B RELATED WORK

B.1 ORDINAL REGRESSION (OR)

OR addresses prediction tasks with ordered targets, widely applied in diverse domains like facial age estimation (Niu et al., 2016b; Chen et al., 2017), image aesthetic/quality assessment (He et al., 2022; 2023), watch-time prediction Ma et al. (2024); Sun et al. (2024); Lin et al. (2023), life-time value prediction (Wang et al., 2019b; Li et al., 2022a; Weng et al., 2024). Prior OR works have predominantly relied on Continuous Space Discretization (CSD) (Wang et al., 2025), transforming OR into classification. Within the CSD paradigm, key directions include methods that enhance boundary discrimination via reference comparisons (Shin et al., 2022; Li et al., 2021; Zheng et al., 2024), but are sensitive to heuristic reference selection. Another prevalent rank-based approach implicitly encodes ordinality via label transformation into sequential binary subtasks (Niu et al., 2016b; Lin et al., 2023; Sun et al., 2024; Chen et al., 2017; Liu et al., 2018b;a). While effective, the fixed discretization leads to prediction rigidity and can amplify errors for head categories, particularly in long-tailed distributions. Recent CLIP-based works align image features with textual ordinal descriptions to enhance the semantic understanding and generalization of ordinal relationships, representing a rapidly growing and promising research direction. The Learnable Prompts strategy (Li et al., 2022c; Yu et al., 2024) employs trainable context vectors to automatically capture ordinal relationships and extract rank concepts from CLIP's latent space. Conversely, Semantic Alignment approaches (Wang et al., 2023b; Du et al., 2024) focus on constructing rank-specific textual descriptions. In contrast, GoR adopts a fundamentally different, generative perspective of autoregressive sequence modeling in language models, which inherently captures explicit sequential dependencies and employs a dynamic termination mechanism to generate sequences of variable lengths. This grants significantly greater flexibility in output granularity compared to methods constrained by fixed bins, enabling adaptation to varying data distributions and prediction requirements.

B.2 SEQUENCE PREDICTION

Sequence prediction, which necessitates models to comprehend input context and produce output sequences, initially focuses on natural language processing (NLP) tasks such as machine translation (Sutskever, 2014; Cho, 2014) and text summarization (Wang et al., 2019a; Xiao et al., 2020). The advent of the Transformer architecture (Vaswani et al., 2017) significantly improves sequence prediction capabilities, leading to numerous derivative models (Liu et al., 2019b; Yang et al., 2019) and expanding its application to fields such as computer vision (CV) (Tian et al., 2024; Wang et al., 2024) and recommendation systems (Hidasi, 2015; Sun et al., 2019) through the successful reformulation of their tasks as effective end-to-end sequence prediction problems. However, sequence prediction has not yet been applied to OR, and our work pioneers a sequence prediction perspective for OR, offering a fundamentally novel modeling paradigm. A related work is Ord2Seq (Wang et al., 2023a). It maps generated sequences to fixed bins and, in essence, remains sequential binary subtasks under the rank-based paradigm and does not scale well when the number of categories or the value range is large—its reported validation has been limited to at most eight ordinal groups. In contrast, GoR employs a generative autoregressive formulation with a dynamic $\langle \text{EOS} \rangle$, enabling adaptive ordinal segmentation rather than relying on predefined bins.

B.3 TOKENIZER DESIGN

Tokenizer design is widely employed in generative language models for compact vocabulary representation and broadly falls into two categories: bottom-up merging and top-down pruning. The former, exemplified by BPE (Sennrich et al., 2016) and WordPiece (Wu et al., 2016), iteratively combines subword units based on statistical criteria. Conversely, top-down pruning methods, represented by Unigram (Kudo & Richardson, 2018), reduce large initial sets by evaluating and eliminating subwords according to their contributions. Building upon these foundational strategies, recent literature has explored various enhancements and adaptations (Xu et al., 2021; Hofmann et al., 2022; Yehezkel & Pinter, 2023; Schmidt et al., 2024), further improving tokenization efficiency. However, these traditional NLP tokenization strategies are not directly applicable to our GoR, where tokens inherently exhibit dual sequential and numerical additive semantics, necessitating customized methodologies.

C PROOF OF PROPOSITION 1

C.1 THEORETICAL ANALYSIS

This section provides a theoretical analysis of these limitations, demonstrating the importance of capturing temporal dependencies to improve prediction accuracy.

Let $\{(x_i, y_i)\}_{i=1}^N$ be the training set. Discretize the value range of y_i into M intervals $d_m = [c_{m-1}, c_m]$ with boundaries $c_0 < \cdots < c_M$. Define binary variables $\mathbf{B}_i^m = 1(y_i > c_m)$ for $m = 1, \ldots, M$, and let $\mathbf{B}_i = \{\mathbf{B}_i^1, \ldots, \mathbf{B}_i^M\}$ with decision history $\mathbf{B}_i^{<m} = (\mathbf{B}_i^1, \ldots, \mathbf{B}_i^{m-1})$.

The label transformation methods with sequential binary subtasks implicitly assume conditional independence across these discretized intervals:

$$P_{\text{naive}}(\mathbf{B}_i \mid x_i) = \prod_{m=1}^{M} P(\mathbf{B}_i^m \mid x_i). \tag{9}$$

In contrast, the true conditional distribution factorizes sequentially:

$$P_{\text{true}}(\mathbf{B}_i \mid x_i) = \prod_{m=1}^{M} P(\mathbf{B}_i^m \mid \mathbf{B}_i^{<m}, x_i). \tag{10}$$

The KL divergence between these two distributions is given by:

$$\begin{aligned} D_{KL}(P_{\text{true}} \| P_{\text{naive}}) &= \sum_{\mathbf{B}_i} P_{\text{true}}(\mathbf{B}_i \mid x_i) \log \frac{P_{\text{true}}(\mathbf{B}_i \mid x_i)}{P_{\text{naive}}(\mathbf{B}_i \mid x_i)} \\ &= \sum_{\mathbf{B}_i} P_{\text{true}}(\mathbf{B}_i \mid x_i) \log \frac{\prod_{m=1}^{M} P(\mathbf{B}_i^m \mid \mathbf{B}_i^{<m}, x_i)}{\prod_{m=1}^{M} P(\mathbf{B}_i^m \mid x_i)} \\ &= \sum_{\mathbf{B}_i} P_{\text{true}}(\mathbf{B}_i \mid x_i) \sum_{m=1}^{M} \log \frac{P(\mathbf{B}_i^m \mid \mathbf{B}_i^{<m}, x_i)}{P(\mathbf{B}_i^m \mid x_i)}. \end{aligned} \tag{11}$$

Rearranging the summation terms, we derive the total KL divergence that quantifies the error introduced by ignoring dependencies among discretized intervals:

$$\begin{aligned} D_{KL}(P_{\text{true}} \| P_{\text{naive}}) &= \sum_{m=1}^{M} \sum_{\mathbf{B}_i} P_{\text{true}}(\mathbf{B}_i \mid x_i) \log \frac{P(\mathbf{B}_i^m \mid \mathbf{B}_i^{<m}, x_i)}{P(\mathbf{B}_i^m \mid x_i)} \\ &= \sum_{m=1}^{M} \mathbb{E}_{\mathbf{B}_i \sim P_{\text{true}}} \left[\log \frac{P(\mathbf{B}_i^m \mid \mathbf{B}_i^{<m}, x_i)}{P(\mathbf{B}_i^m \mid x_i)} \right]. \end{aligned} \tag{12}$$

We can decompose this expectation using the Law of Iterated Expectations:

$$
\begin{aligned}
D_{KL}(P_{\text{true}} \| P_{\text{naive}}) &= \sum_{m=1}^{M} \mathbb{E}_{\mathbf{B}_i^{<m}} \left[\mathbb{E}_{\mathbf{B}_i^{\geq m} | \mathbf{B}_i^{<m}} \left[\log \frac{P(\mathbf{B}_i^m | \mathbf{B}_i^{<m}, \mathbf{x}_i)}{P(\mathbf{B}_i^m | \mathbf{x}_i)} \right] \right] \\
&= \sum_{m=1}^{M} \mathbb{E}_{\mathbf{B}_i^{<m}} \left[\sum_{b^m \in \{0,1\}} P(\mathbf{B}_i^m = b^m | \mathbf{B}_i^{<m}, \mathbf{x}_i) \log \frac{P(\mathbf{B}_i^m = b^m | \mathbf{B}_i^{<m}, \mathbf{x}_i)}{P(\mathbf{B}_i^m = b^m | \mathbf{x}_i)} \right] \\
&= \sum_{m=1}^{M} \mathbb{E}_{\mathbf{B}_i^{<m}} \left[D_{KL}^{(m)} \right].
\end{aligned} \tag{13}
$$

This can be explicitly expressed as the conditional KL divergence for each bucket:

$$
\begin{aligned}
D_{KL}^{(m)} &= D_{KL} \left(P(\mathbf{B}_i^m \mid x_i, \mathbf{B}_i^{<m}) \,\|\, P(\mathbf{B}_i^m \mid x_i) \right) \\
&= \sum_{b^m \in \{0,1\}} P(\mathbf{B}_i^m = b^m \mid x_i, \mathbf{B}_i^{<m}) \log \frac{P(\mathbf{B}_i^m = b^m \mid x_i, \mathbf{B}_i^{<m})}{P(\mathbf{B}_i^m = b^m \mid x_i)}.
\end{aligned} \tag{14}
$$

This expectation can be expressed as:

$$
\mathbb{E}_{\mathbf{B}_i^{<m}} \left[D_{KL}^{(m)} \right] = I(\mathbf{B}_i^m; \mathbf{B}_i^{<m} \mid x_i), \tag{15}
$$

where $I(\mathbf{B}_i^m; \mathbf{B}_i^{<m} \mid x_i)$ denotes the conditional mutual information between the current decision $\mathbf{B}_i^m$ and all preceding decisions $\mathbf{B}_i^{<m}$, given the features x_i. The detailed proof is in the following:

$$
\begin{aligned}
I(\mathbf{B}_i^m; \mathbf{B}_i^{<m} \mid x_i) &\triangleq \sum_{\mathbf{b}^{<m}, b^m} P(\mathbf{b}^{<m}, b^m \mid x_i) \log \frac{P(b^m \mid \mathbf{b}^{<m}, x_i)}{P(b^m \mid x_i)} \\
&= \sum_{\mathbf{b}^{<m}} P(\mathbf{b}^{<m} \mid x_i) \sum_{b^m} P(b^m \mid \mathbf{b}^{<m}, x_i) \log \frac{P(b^m \mid \mathbf{b}^{<m}, x_i)}{P(b^m \mid x_i)} \\
&= \sum_{\mathbf{b}^{<m}} P(\mathbf{b}^{<m} \mid x_i) \left(D_{KL} \left(P(\mathbf{B}_i^m \mid x_i, \mathbf{B}_i^{<m}) \,\|\, P(\mathbf{B}_i^m \mid x_i) \right) \right) \\
&\triangleq \mathbb{E}_{\mathbf{B}_i^{<m}} \left[D_{KL}^{(m)} \right].
\end{aligned} \tag{16}
$$

The derived KL divergence decomposition illustrates that the error introduced by the naive discretized modeling approach which ignores dependencies across intervals, can be quantified precisely as the cumulative sum of conditional mutual information across all discretized intervals. Specifically, if intervals are entirely independent (i.e., mutual information $I = 0$), the resulting KL divergence error is zero; conversely, if strong dependencies exist among intervals ($I > 0$), the error increases proportionally to the strength of these dependencies.

C.2 Empirical Validation

To empirically validate Proposition 1, we conduct additional experiments using an SOTA baseline SWaT (Yang et al., 2025) on the CIKM16 dataset of watch-time prediction task. Specifically, we introduce explicit sequential dependencies by modeling previous bin features with an RNN for joint prediction. The results in Tab. 6 demonstrate performance gains, substantiating our theoretical claims.

Table 6: Empirical validation of Proposition 1 on the CIKM16 dataset.

Method	MAE ↓	XAUC ↑
SWaT	0.857	0.685
SWaT+RNN	0.831	0.689

D PROOF OF THEOREM 1

We provide a formal bias-variance decomposition of the expected squared error in GoR. Our goal is to upper bound the prediction error of a model that generates a sequence of value tokens.

Let the true label be defined as:

$$y_i = \sum_{t=1}^{T_i} \phi(s_i^t), \tag{17}$$

and the predicted label be:

$$\hat{y}_i = \sum_{t=1}^{T_i} \phi(\hat{s}_i^t), \tag{18}$$

where T_i represents the overall length of the token sequence for sample i, s_i^t and $\hat{s}_i^t$ denote the ground-truth and predicted tokens at step t, and $\phi(\cdot)$ maps a token to its corresponding numeric value.

We model $\phi(s_i^t)$ as a discrete random variable, denoted as C_i^t. The probability distribution of C_i^t is given by: $P(C_i^t = \omega), \omega \in \{\phi(w_j)\}_{j=1}^{V}$,which denotes the probability of C_i^t taking the value ω. So, the predicted token output be modeled as $\hat{y}_i = \sum_{t=1}^{T_i} \hat{C}i^t$

We now analyze the expected squared error:

$$\mathbb{E}[(\hat{y}_i - y_i)^2] = \mathbb{E}\left[\left(\sum_{t=1}^{T_i} \hat{C}_i^t - \sum_{t=1}^{T_i} C_i^t\right)^2\right]. \tag{19}$$

Let $\Delta_t := \hat{C}_i^t - C_i^t$. Then we can write:

$$\mathbb{E}\left[\left(\sum_{t=1}^{T_i} \Delta_t\right)^2\right] = \underbrace{\left(\sum_{t=1}^{T_i} \mathbb{E}[\Delta_t]\right)^2}_{\text{Bias}^2} + \underbrace{\mathbb{V}\left(\sum_{t=1}^{T_i} \Delta_t\right)}_{\text{Variance}} \tag{20}$$

$$= \left(\sum_{t=1}^{T_i} b_t\right)^2 + \sum_{t=1}^{T_i} \mathbb{V}(\Delta_t) + \sum_{t \neq t'} \text{Cov}(\Delta_t, \Delta_{t'}), \tag{21}$$

where $b_t := \mathbb{E}[\Delta_t] = \mathbb{E}[\hat{C}_i^t] - C_i^t$. This formulation captures the compounding errors of autoregressive models through the covariance term, which reflects correlations between errors at different steps. Next, we analyze the bias and variance terms separately to derive their upper bounds.

Bias Upper Bound. If the model is unbiased at each step, $b_t = 0$. Otherwise, we assume:

$$|b_t| \leq B, \quad \forall t, \tag{22}$$

where B represents the maximum bias across all time steps in the prediction sequence, mainly governed by the model's predictive accuracy. In practice, B corresponds to the extreme case where the predicted token deviates most from the ground-truth token at the current step. Hence, it can be bounded by the largest token value at the current step $B \leq \max_{1 \leq t \leq T_i} C_i^t$. Then,

$$\left(\sum_{t=1}^{T_i} b_t\right)^2 \leq T_i^2 B^2. \tag{23}$$

Variance Upper Bound. Applying the Cauchy–Schwarz inequality:

$$\sum_{t \neq t'} \text{Cov}(\Delta_t, \Delta_{t'}) \leq \sum_{t \neq t'} \sqrt{\mathbb{V}(\Delta_t)\mathbb{V}(\Delta_{t'})} \leq \frac{T_i(T_i - 1)}{2} \cdot \max_t \mathbb{V}(\Delta_t). \tag{24}$$

We then analyze the item $\max_t \mathbb{V}(\Delta_t)$.

$$\mathbb{V}(\Delta_t) = \mathbb{V}\left(\hat{C}_i^t - C_i^t\right) = \mathbb{V}\left(\hat{C}_i^t\right) + \mathbb{V}\left(C_i^t\right) - 2\text{Cov}\left(\hat{C}_i^t, C_i^t\right) \tag{25}$$

Algorithm 2: Quantile-based Vocabulary Initialization

Input: Dataset labels $Y = \{y_i\}_{i=1}^{N}$; initially empty initial vocabulary $\mathcal{W} = \varnothing$; precision threshold ε; fixed percentile q
Output: initial vocabulary $\mathcal{W}$

Sort Y in descending order to obtain $\hat{Y}$
Initialize iteration counter $iter \leftarrow 1$, error metric $err \leftarrow \infty$;
while $err > \varepsilon$ **do**
  Compute the q-percentile z_{iter} of Y
  if $z_{iter} = 0$ **then break** `// terminate if percentile value is zero`
  Insert z_{iter} into vocabulary $\mathcal{W}$
  foreach $\hat{y}_i \in \hat{Y}$ **do**
    if $\hat{y}_i \geq z_{iter}$ **then**
      $\hat{y}_i \leftarrow \hat{y}_i - z_{iter}$
  end
  Update error metric $err \leftarrow \max_i \frac{\hat{y}_i}{y_i}$
  $iter \leftarrow iter + 1$
end
return $\mathcal{W}$

Assume that the predicted variable and the true variable are two independent random variables. Since the vocabulary is the same, the range of values for both the predicted and the true items is identical. Assuming token values are bounded in $[w_{\text{min}}, w_{\text{max}}]$, we apply Popoviciu's inequality:

$$\mathbb{V}(\hat{C}_i^t) \leq \frac{(w_{\text{max}} - w_{\text{min}})^2}{4}. \tag{26}$$

Let:

$$V_{var} := \max_t \mathbb{V}(\Delta_t) \leq \frac{(w_{\text{max}} - w_{\text{min}})^2}{4}. \tag{27}$$

Then total variance becomes:

$$\mathbb{V}(\sum_t \Delta_t) \leq T_i^2 V_{var}. \tag{28}$$

Final Bound. Combining both components, we obtain:

$$\mathbb{E}\left[(\hat{y}_i - y_i)^2\right] \leq T_i^2 B^2 + T_i^2 \cdot \frac{(w_{\text{max}} - w_{\text{min}})^2}{4} \tag{29}$$

This theoretical bound reveals three critical insights for GoR optimization: (1) prediction error grows quadratically with sequence length T_i, suggesting shorter sequences are preferable when possible; (2) both bias and variance contribute proportionally to overall error, necessitating balanced optimization; and (3) token value range $(w_{\text{max}} - w_{\text{min}})$ directly impacts variance, indicating that carefully designed vocabularies with appropriate value distributions can substantially improve model performance. These findings provide a principled foundation for our vocabulary construction strategy.

E Quantile-based Vocabulary Initialization Strategy

This section details the Quantile-based Vocabulary Initialization Strategy adopted in Sec. 2.3. As shown in Alg. 2, this iterative strategy constructs the vocabulary by selecting tokens based on a fixed percentile q of the remaining label values, subtracting them from exceeding values, and repeating until residuals are negligible. Alg. 2 serves as the initialization stage: it provides a coarse yet comprehensive token set. Building on this, Alg. 1 in Sec. 2.3 further prunes and refines the vocabulary via the CoDi criterion, yielding a compact and task-adaptive representation.

F Additional Experiments

F.1 Experimental Settings

F.1.1 Metrics.

A set of performance metrics is utilized to evaluate the proposed method across various tasks. Task requirements determine the specific metrics applied for each task:

- **MAE (Mean Absolute Error)**: This regression precision is measured as the average absolute error between the value prediction $\{\hat{y}_i\}_{i=1}^N$ and the ground truth $\{y_i\}_{i=1}^N$ and is formulated as $\frac{1}{N}\sum_{i=1}^N |y_i - \hat{y}_i|$.
- **CS (Cumulative Score)**: This metric quantifies the proportion of instances in the test set for which the absolute error between the predicted value $\hat{y}_i$ and the ground truth value y_i is less than or equal to a specified tolerance L.
- **XAUC** (Zhan et al., 2022): This metric measures the agreement between the predicted ranks and the ground truth order for pairs of samples. Calculated over uniformly sampled pairs, XAUC represents the proportion of pairs where the predicted relative order is consistent with the true relative order. Higher XAUC indicates superior performance in capturing ordinal relationships.
- **LCC (Linear Correlation Coefficient) (Talebi & Milanfar, 2018)**: This metric quantifies the linear relationship between the predicted values $\{\hat{y}_i\}_{i=1}^N$ and the ground truth values $\{y_i\}_{i=1}^N$. It is computed as their covariance divided by the product of their standard deviations. The LCC ranges in $[-1, 1]$, with values closer to ± 1 indicating a stronger linear correlation.
- **SRCC (Spearman's Rank Correlation Coefficient) (Talebi & Milanfar, 2018)**: SRCC assesses the monotonic relationship between the ranks of the predicted values $\{\hat{y}_i\}_{i=1}^N$ and the ranks of the ground truth values $\{y_i\}_{i=1}^N$. As a non-parametric measure of rank correlation, it ranges in $[-1, 1]$. Values closer to ± 1 indicate a stronger monotonic correlation. SRCC is less sensitive to outliers compared to LCC.

F.1.2 Implementation Details.

Unless otherwise specified in the respective experimental sections, the following training protocol is adopted. The proposed GoR architecture employs an encoder-decoder framework. The encoder architecture in GoR is tailored to the specific task, with details regarding data processing and encoder configurations provided in the corresponding experimental sections. The decoder in GoR is a two-layer Transformer decoder utilizing a 4-head attention mechanism. The hyperparameter λ in Eq. (8) is set to 10. For vocabulary construction, in Alg. 2 for initial vocabulary, q is set to 0.9, ε is set to 0.005. In Alg. 1, β is set to 0.7 and ϵ is set to 0.005. To mitigate overfitting, a dropout rate of 0.1 is applied. The Adam optimizer (Kingma & Ba, 2014) with default parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$) and a learning rate of 5e-4 are used to minimize the objective function. For experiments involving image data (common in computer vision tasks), training is conducted for 100 epochs with a batch size of 128. For the tasks involving structured data (WTP, LTV and Bid), training is performed for 20 epochs using a batch size of 1024. Experiments are conducted on a system equipped with an NVIDIA RTX 4090 GPU.

F.1.3 Reproducibility and Fair Comparison.

Each experiment is repeated five times, and we report both the mean and standard deviation (See Appendix. F.7). As noted in prior work (Paplhám et al., 2024), state-of-the-art methods in the FAE field often exhibit large performance variance due to inconsistent dataset splits, preprocessing protocols, and evaluation criteria, rendering many results incomparable and irreproducible. Motivated by this, (Paplhám et al., 2024) proposed a standardized evaluation protocol. We observe the same issue in the IAA domain Consequently, we reproduced all baselines under their original hyperparameter settings in their respective papers and averaged the results. This strategy offers two key benefits: (1) it ensures fair and consistent comparisons; and (2) it establishes GoR as a reliable baseline to facilitate standardized evaluations in future research. To ensure reproducibility, we will release the complete codebase, including all baseline implementations.

Table 7: The results of Image Aesthetics Assessment task on TAD66K and AVA datasets

Method	TAD66K				AVA			
	MAE ↓	XAUC ↑	LCC ↑	SRCC ↑	MAE ↓	XAUC ↑	LCC ↑	SRCC ↑
RAPID (Lu et al., 2014)	1.766	0.510	0.332	0.314	0.978	0.513	0.336	0.327
AADB (Kong et al., 2016)	1.463	0.523	0.400	0.379	0.784	0.534	0.431	0.408
PAM (Ren et al., 2017)	1.314	0.534	0.440	0.422	0.614	0.619	0.531	0.521
NIMA (Talebi & Milanfar, 2018)	1.422	0.511	0.405	0.390	0.715	0.532	0.472	0.447
ALamp (Ma et al., 2017)	1.349	0.523	0.422	0.411	0.657	0.579	0.498	0.487
MP_{ada} (Sheng et al., 2018)	1.191	0.589	0.408	0.389	0.602	0.632	0.543	0.531
MLSP (Hosu et al., 2019)	1.132	0.620	0.432	0.409	0.579	0.657	0.563	0.553
BIAA (Zhu et al., 2020)	1.329	0.538	0.431	0.348	0.672	0.566	0.496	0.476
UIAA (Zeng et al., 2019)	1.281	0.548	0.441	0.361	0.608	0.626	0.535	0.525
POE (Li et al., 2021)	1.185	0.588	0.420	0.377	0.633	0.608	0.524	0.506
HGCN (She et al., 2021)	1.141	0.615	0.419	0.406	0.658	0.578	0.511	0.486
TANet (He et al., 2022)	1.081	0.649	0.452	0.428	0.577	0.659	0.568	0.554
MaxViT (Tu et al., 2022)	1.054	0.659	0.472	0.441	0.559	0.679	0.594	0.571
Delegate (He et al., 2023)	1.041	0.661	0.477	0.451	0.541	0.688	0.642	0.634
AesMamba (Gao et al., 2024)	1.035	0.666	0.482	0.468	0.522	0.697	0.663	0.656
GoR with ResNet	1.036	0.667	0.485	0.471	0.526	0.701	0.668	0.657
GoR with TANet	1.013	0.672	0.523	0.499	0.428	0.735	0.689	0.686
GoR with AesMamba	0.996	0.677	0.541	0.513	0.395	0.751	0.726	0.701

F.2 Image Aesthetics Assessment (IAA)

F.2.1 Datasets, Baselines, and Experimental Setup.

GoR is evaluated on four widely used IAA datasets: TAD66K (He et al., 2022), AVA (Murray et al., 2012), ICAA17K (He et al., 2023), and SPAQ (Fang et al., 2020). Data was randomly split into 80% for training, 10% for validation, and 10% for testing. Due to the relatively small range of aesthetics scores (typically 0-10), labels were scaled by 100 for GoR's vocabulary construction and ordinal target sequencing. Predictions were scaled back by 100 for evaluation metric computation to ensure fair comparison.

Baselines were chosen based on two criteria: 1) classical architectures with available code, and 2) state-of-the-art (SOTA) performance in specific areas, such as personalized IAA. For a fair comparison, these baselines were trained using their recommended hyperparameter settings and evaluated under identical training and testing configurations. Consistent with the approach in (He et al., 2023), all compared baselines were subjected to identical data preprocessing.

Given the critical role of visual features in image aesthetics assessment, we evaluate GoR by employing three different encoder backbones: ResNet50 (He et al., 2016) as a widely recognized standard, a representative older architecture (He et al., 2022), and a recent SOTA model (Gao et al., 2024). This strategy also helps to ensure that observed performance gains are due to the GoR framework itself, rather than simply an increase in model parameters.

F.2.2 Comprehensive Baselines Comparison for Image Aesthetics Assessment.

Due to space constraints in Sec. 3.1.1 of the main paper, comprehensive baseline comparisons for the Image Aesthetics Assessment task are presented here. Tab. 7 and Tab. 8 detail the performance of all compared methods on the TAD66K/AVA and ICAA17K/SPAQ datasets, respectively.

F.3 Watch Time Prediction (WTP)

F.3.1 Datasets and Experimental Setup

Three publicly available datasets and one industrial dataset are used to evaluate the proposed method. The large-scale industrial dataset (Indust. for short) is sourced from a real-world streaming short-video app with over 400 million DAUs and multi-billion impressions each day. We collect interaction logs for 4 days and utilize the subsequent day's data for evaluation. The CIKM16[4], sourced from the CIKM16 Cup competition, is designed to predict user engagement duration in online search

[4]https://competitions.codalab.org/competitions/11161

Table 8: The results of Image Aesthetics Assessment task on ICAA17K and SPAQ datasets.

Method	ICAA17K				SPAQ			
	MAE ↓	XAUC ↑	LCC ↑	SRCC ↑	MAE ↓	XAUC ↑	LCC ↑	SRCC ↑
RAPID (Lu et al., 2014)	0.7415	0.6416	0.5164	0.5083	1.0890	0.6997	0.6565	0.6128
AADB (Kong et al., 2016)	0.7142	0.6661	0.5311	0.5195	1.083	0.7036	0.6646	0.6162
PAM (Ren et al., 2017)	0.7070	0.6729	0.5385	0.5247	1.0726	0.7104	0.6691	0.6222
ALamp (Ma et al., 2017)	0.6948	0.6847	0.5478	0.5339	1.0511	0.7250	0.6835	0.6349
NIMA (Talebi & Milanfar, 2018)	0.6957	0.6839	0.5458	0.5333	1.0756	0.7084	0.6709	0.6204
MP_{ada} (Sheng et al., 2018)	0.6948	0.6848	0.5485	0.5340	1.0525	0.7240	0.6808	0.6341
MLSP (Hosu et al., 2019)	0.6814	0.6983	0.5606	0.5445	1.0428	0.7306	0.6952	0.6402
MT-A (Fang et al., 2020)	0.6855	0.6940	0.5558	0.5412	1.0455	0.7289	0.6862	0.6384
BIAA (Zhu et al., 2020)	0.6864	0.6932	0.5552	0.5405	1.0497	0.7259	0.6826	0.6358
UIAA (Zeng et al., 2019)	0.6889	0.6907	0.5559	0.5386	1.0469	0.7278	0.6862	0.6376
POE (Li et al., 2021)	0.6808	0.6966	0.5583	0.5432	1.0456	0.7307	0.6877	0.6368
MUSIQ (Ke et al., 2021)	0.6740	0.7059	0.5632	0.5504	1.0427	0.7308	0.6925	0.6401
HGCN (She et al., 2021)	0.6813	0.6983	0.5566	0.5445	1.040	0.7328	0.6934	0.6417
TANet (He et al., 2022)	0.6789	0.7008	0.5599	0.5465	1.0469	0.7279	0.6844	0.6375
MaxViT (Tu et al., 2022)	0.6582	0.7227	0.5853	0.5636	1.042	0.7308	0.6925	0.6401
Delegate (He et al., 2023)	0.6345	0.7498	0.6034	0.5847	1.019	0.7473	0.7114	0.6545
AesMamba (Gao et al., 2024)	0.6129	0.7663	0.6137	0.6294	0.9875	0.7522	0.7261	0.6895
GoR with ResNet	0.6115	0.7653	0.6133	0.6305	0.9911	0.7588	0.7322	0.6886
GoR with TANet	0.5994	0.7838	0.6744	0.6675	0.9489	0.7644	0.7406	0.7189
GoR with AesMamba	0.5842	0.7913	0.6823	0.6789	0.8722	0.7648	0.7434	0.7233

sessions. It contains 310,302 sessions, 122,991 items, and an average session length of 3.981. Both KuaiRand (Gao et al., 2022b) and KuaiRec (Gao et al., 2022a) are real-world datasets collected from Kuaishou app video view logs. KuaiRand comprises 26,988 users, 6,598 items, and 1,266,560 impressions, while the larger KuaiRec dataset consists of 7,176 users, 10,728 items, and 12,530,806 impressions.

Unlike traditional user behavior modeling tasks in recommendation systems, WTP does not inherently depend on history action sequences. Consequently, we employ a two-layer Multi-Layer Perceptron (MLP) as the encoder, maintaining the same configuration as the compared baseline methods.

F.3.2 BASELINES

We evaluate our method with several existing state-of-the-art WTP methods. Here, we provide more detailed information about these compared methods as follows:

1. VR (Value Regression): This method employs direct regression fitting to predict the absolute value of watch time, evaluating model accuracy by mean square error (MSE).
2. WLR (Covington et al., 2016): This method reformulates watch time regression as a binary classification problem and incorporates the watch time as the weight in the loss function.
3. D2Q (Zhan et al., 2022): This method segments data based on the video duration and then employs a regression model to predict the watch time quantile within each group. The final prediction is obtained by converting the predicted quantile back into actual watch time.
4. CWM (Zhao et al., 2024): It models counterfactual watch time (CWT) by estimating user interest via a cost-based transform function. The final prediction is derived by optimizing a counterfactual likelihood function over observed watch times.
5. TPM (Lin et al., 2023): It uses a tree structure to model relationships between different granularities of time intervals by ordinal regression. The watch time is calculated as the weighted sum of the products of probabilities and time intervals along the tree path.
6. CREAD (Sun et al., 2024): Utilizing an error-adaptive discretization technique based on ordinal regression, this method constructs dynamic time intervals. Within each interval, a specialized classifier determines whether the watch time exceeds the interval's threshold, and the final prediction is derived from the weighted sum of probabilistic estimates across all intervals.
7. SWaT (Yang et al., 2025): User-centric statistical framework modeling watch time with behavioral assumptions. It employs bucketization for non-stationary viewing probabilities, with prediction via a weighted sum of probabilistic estimates.

F.3.3 Analysis of Performance in the >10s Interval

TPM (Lin et al., 2023) is a tree-like binary model executing sequential binary subtasks for ordinal regression. As shown in Fig. 5(d), the prediction distribution of TPM exhibits a notable skew towards higher values, attributable to the model's tendency (observed during case analysis) to learn probabilities greater than 0.5 at the root node of its tree structure. This can result in an overall overestimation of the predicted outcomes, thereby explaining why GoR's performance is marginally surpassed by TPM in the >10s interval. However, given the characteristic long-tail distribution of real-world watch time data, the superior overall performance and distributional fidelity achieved by GoR represent a favorable trade-off for this minor discrepancy in the high-value range.

F.3.4 Parameter-controlled comparison

To ascertain that the enhancement observed in the GoR model is not merely a consequence of an increase in model parameters, we conducted a comparative analysis, as presented in Tab. 9. Specifically, we standardized the parameters of the SOTA models (CREAD (Sun et al., 2024), TPM (Lin et al., 2023) and SWat (Yang et al., 2025)) to a uniform level. The results indicate that, when evaluated with an equivalent parameter scale, GoR consistently surpasses the previous SOTA methods across a range of metrics. This finding confirms that the observed performance improvement is attributable not to the number of parameters, but to the efficacy of utilizing conditional dependencies and the flexibility to generate a wider range of potential sequences.

Table 9: The impact of model parameters on the performance between different methods on KuaiRec.

Method	Parameters	MAE	XAUC
VR	0.86M	7.634	0.534
TPM	0.86M	3.456	0.571
CREAD	0.86M	3.307	0.594
SWaT	0.88M	3.438	0.585
VR-large	4.34M	7.556	0.545
TPM-large	4.34M	3.432	0.577
CREAD-large	4.34M	3.293	0.599
SWaT-large	4.35M	3.406	0.591
GoR (ours)	**4.18M**	**3.194**	**0.616**

F.3.5 Online Experiment

We also conduct an online A/B test of watch time prediction task on a leading short-video platform to demonstrate GoR's real-world efficacy. Considering the platform serves over 400 million users daily, doing experiments from 10% of traffic involves a huge population of more than 40 million users, which can yield highly reliable results. Most recommendation systems follow a two-stage framework where a set of candidate items is retrieved in the first stage and the top-ranking items are selected from the candidates in the ranking stage. The predicted watch times are used in the ranking stage to prioritize items with higher predicted watch times, making them more likely to be recommended.

The online experiment has been launched on the system for six days, with evaluation metrics including app usage time, average app usage per user, and video consumption time (accumulated watch time). The control group utilized the CREAD model, and the proposed GoR framework exhibited a 10.2% reduction in average queries per second (QPS) during online serving. Despite this computational overhead, the overall return on investment (ROI) met the threshold for full deployment, indicating favorable trade-offs between operational costs and business value enhancement.

As shown in Tab. 10, the results demonstrate that GoR consistently boosts performance in watch time related metrics, with an improvement by 0.087% on average app usage per user, significant **0.129%** on video consumption time and **0.112%** on app usage time with p-value[5] $= 0.01$, substantiating its potential to significantly enhance real-world user experiences.

[5]Lower p-values mean greater statistical significance (e.g., p=0.01 implies a 1% likelihood of gain occurring by chance).

Table 10: Performance gain on online A/B testing.

A/B test	APP Usage Time	+0.112% (p-value=0.01)
	Average App Usage Per User	+0.087%
	Video Consumption Time	+0.129%

In a stable video recommendation system, a **0.1% increase is significant.**

F.4 Life Time Value Prediction (LTV)

F.4.1 Datasets and Experimental Setup

We evaluate GoR on the Criteo-SSC[6] and Kaggle[7] datasets. For both datasets, a random split of 7:1:2 is used for training, validation, and testing, respectively. Criteo-SSC is a large-scale public dataset derived from Criteo Predictive Search (CPS) logs. Each instance represents a user's click behavior, with the task being to predict conversion and associated 30-day revenue. The product price feature was excluded from the inputs. The Kaggle Dataset contains transaction records. Following (Weng et al., 2024), the task involves predicting a user's total purchase value from a specific company in the year following their initial purchase. Our experiments focus on initial purchases within 2012-03-01 and 2012-07-01, using data from the three companies with the highest transaction volume.

F.4.2 Baselines

We evaluate our method with several existing state-of-the-art LTV methods (Drachen et al., 2018; Ma et al., 2018; Wang et al., 2019b; Li et al., 2022a; Liu et al., 2024b; Weng et al., 2024). Here, we provide more detailed information about these compared methods as follows:

1. Two-stage (Drachen et al., 2018) decomposes the CLTV prediction into two tasks: the first task is a classification task predicting whether a user will churn or not, and the second task is a regression task predicting the revenue that the user brings.
2. MTL-MSE (Ma et al., 2018) estimates conversion rate and CLTV with MSE loss according to the multi-task learning paradigm.
3. ZILN (Wang et al., 2019b) assumes that the long-tailed CLTV distribution follows a zero-inflated log-normal distribution and uses a DNN to estimate the mean μ, standard deviation σ and conversion rate p for the samples.
4. MDME (Li et al., 2022a) divides the training samples by CLTV into multiple sub-distributions and buckets, and constructs corresponding classification problems to predict the bucket a sample belongs to. In the next stage, the bias within the bucket is estimated so that the samples obtain a fine-grained CLTV value.
5. MDAN (Liu et al., 2024b) predicts predefined LTV bucket labels using a multi-classification network and leverages a multi-channel learning network to derive embeddings for each bucket. The final sample representation is obtained by fusing these embeddings with the classification network's output through a weighted sum, which is then utilized for CLTV prediction.
6. OptDist (Weng et al., 2024) employs an adaptive mechanism to model and select optimal sub-distributions for individual samples, consisting of a distribution learning module (DLM) that trains multiple sub-distribution networks, and a distribution selection module (DSM) that dynamically chooses the appropriate sub-distribution for each customer.
7. HiLTV (Xu et al., 2025) is a hierarchical framework for game LTV prediction that models multi-modal recharge behaviors with a Zero-Inflated Mixture-of-Logistic loss and introduces a calibration module for robust new-user prediction.

For this task, we employ the same encoder architecture for GoR in Appendix F.3.1.

F.5 Bid Shading for Real-Time Bidding (RTB)

Bid shading in Real-Time Bidding aims to dynamically adjust advertiser bids to avoid overspending under First-Price Auction (FPA) settings. Given bid requests with user and contextual features xi and

[6]https://ailab.criteo.com/criteo-sponsored-search-conversion-log-dataset/

[7]https://www.kaggle.com/c/acquire-valued-shoppers-challenge

first estimated values v_i, the model generates shading ratios α_i to obtain final bids $b_i = v_i \cdot \alpha_i$. The objective is to maximize the surplus $(v_i - b_i) \cdot Wr(b_i|x_i)$, where $Wr(\cdot)$ denotes the winning rate.

F.5.1 Datasets and Metrics

We evaluate GoR on both a public benchmark and a large-scale industrial dataset. The public iPinYou dataset (Liao et al., 2014), derived from Second-Price Auctions (SPA), treats the advertiser's bid as the actual value and the paid price as the winning price. It consists of 10.6 million samples (29.7% win rate) with 18 features, randomly split 7:3 for training and testing. In contrast, the industrial dataset (Indust_RTB for short) from a real-world app platform is substantially larger, containing 162.5 million samples with a lower win rate of 3.89% and 197 features.

In First-Price Auctions (FPA), offline evaluation must prioritize business-centric metrics to assess an algorithm's viability for deployment. Aligned with the objective of bid shading, we select surplus $(V - b) \cdot \mathbb{I}(b > z)$ and surplus rate as primary metrics, as they directly quantify business impact. The surplus rate, defined as the proportion of realized surplus to the total optimal surplus, is computed as $SurplusRate(SR) = \frac{\sum_i (v_i - b_i) \cdot \mathbb{I}(b_i > z_i)}{\sum_i (v_i - z_i)}$.

F.5.2 Baselines

We evaluate our method with six existing state-of-the-art RTB methods. Here, we provide more detailed information about these compared methods as follows:

1. CVAE (Sohn et al., 2015): An extended model of variational autoencoder, which achieves sample generation based on specific inputs by integrating conditional variables in the encoder and decoder.
2. TSBS-DLF (Ren et al., 2019): A two-stage bid shading method uses the DLF model in the machine learning stage, which models the bid landscape without distributional assumptions, employing the conditional probability chain rule and LSTM.
3. DF (Ho et al., 2020): A generative diffusion model that corrupts the data distribution by gradually adding noise and then learns the inverse denoising process to generate the shading ratio.
4. WR (Pan et al., 2020): A two-stage bid shading method, which optimizes the surplus using a bisection algorithm in the operations research stage.
5. EDDN (Zhou et al., 2021): A two-stage bid shading method, which optimizes the surplus using golden section search in the operations research stage.
6. TSBS-ADM (Li et al., 2023): A two-stage bid shading method uses the ADM model in the machine learning stage, which uses neighborhood likelihood loss for accurate prediction.
7. MEBS (Gong et al., 2024): An end-to-end bid shading method, which jointly optimizes the shading model and the win rate model, performs supervised learning through the negative logarithm of the surplus as the loss.
8. HALO (Dong et al., 2025) is a hindsight-augmented auto-bidding framework that leverages trajectory reorientation and B-spline functional representation to enable efficient, generalizable adaptation to diverse budget–ROI constraints in RTB systems.

F.5.3 Performance

As summarized in Tab. 15, GoR consistently outperforms all baseline methods across both datasets, achieving the highest surplus rate (SR) and total surplus. On the public iPinYou dataset, GoR attains an SR of 60.48% and a surplus of 114.07 million, surpassing the strongest baseline by approximately 4 absolute percentage points in SR. On the more challenging Indust_RTB dataset—characterized by a significantly lower win rate and higher feature dimensionality—GoR achieves an SR of 41.74% and a surplus of 30.58 million, demonstrating a clear improvement over existing methods and highlighting its robustness in large-scale, real-world environments.

F.6 Historical Image Dating (HID)

F.6.1 Datasets, Baselines, and Performance

Here, we use the widely recognized Historical Color Image (HCI) dataset (Palermo et al., 2012). Consistent with the established evaluation protocol adopted in numerous prior studies (Palermo et al., 2012; Liu et al., 2018b; 2019a; Diaz & Marathe, 2019; Li et al., 2021; Shin et al., 2022; Wang

Table 11: Performance comparison on RTB datasets.

Method	iPinYou		Indust_RTB	
	SR ↑	Surplus ↑	SR ↑	Surplus ↑
CVAE (Sohn et al., 2015)	51.42%	96,981,112	36.45%	26,698,064
TSBS-DLF (Ren et al., 2019)	55.22%	104,164,484	27.25%	19,986,132
DF (Ho et al., 2020)	48.04%	90,617,280	31.28%	22,912,038
EDDN (Zhou et al., 2021)	47.82%	90,199,768	19.88%	14,582,074
WR (Pan et al., 2020)	54.90%	103,560,851	26.13%	19,164,882
TSBS-ADM (Li et al., 2023)	55.49%	104,673,800	32.67%	24,183,220
MEBS (Gong et al., 2024)	54.46%	102,716,112	31.45%	23,041,120
HALO (Dong et al., 2025)	56.58%	105,531,332	36.67%	26,243,657
Our GoR	**60.48%**	**114,068,392**	**41.74%**	**30,580,080**

Table 12: Historical Image Dating Results

Datasets \ Methods		Palermo et al. (Palermo et al., 2012)	CNNPOR (Liu et al., 2018b)	GP-DNNOR (Liu et al., 2019a)	SORD (Diaz & Marathe, 2019)	POE (Li et al., 2021)	MWR (Shin et al., 2022)	Ord2Seq (Wang et al., 2023a)	GoR
HCI	MAE ↓	0.93	0.82	0.76	0.70	0.66	0.58	0.53	**0.51**

et al., 2023a), we randomly partition the images within each decade into training (80%), validation (5%), and testing (15%) subsets. Subsequently, 10-fold cross-validation is performed, and the mean Mean Absolute Error (MAE) results for different methods are reported in Tab. 12. To ensure a fair comparison, all evaluated methods utilize the ResNet50 (He et al., 2016) architecture as the backbone. Our GoR model achieves state-of-the-art performance on the HCI dataset, yielding a significant improvement over previous methods with a 3.77% reduction in MAE, indicating the superiority of our approach.

F.7 Additional Results with Mean and Standard Deviation

To complement the main results, we report here the complete performance with mean and standard deviation across five runs for all datasets and tasks. Significance is assessed using paired t-tests against the strongest baseline, with improvements marked as * ($p<0.05$) and ** ($p<0.01$). These results provide a more comprehensive view of GoR's robustness and stability across domains.

Table 13: Image Aesthetics Assessment Results.

Method	**TAD66K**				**AVA**			
	MAE ↓	XUAC ↑	LCC ↑	SRCC ↑	MAE ↓	XAUC ↑	LCC ↑	SRCC ↑
GoR	0.996±0.003**	0.677±0.003**	0.541±0.012*	0.513±0.011**	0.395±0.015**	0.751±0.012*	0.726±0.028*	0.701±0.016**
Method	**ICAA**				**SPAQ**			
	MAE ↓	XUAC ↑	LCC ↑	SRCC ↑	MAE ↓	XAUC ↑	LCC ↑	SRCC ↑
GoR	0.5842±0.0081**	0.7913±0.0102**	0.6823±0.012**	0.6789±0.0132*	0.8722±0.0223**	0.7648±0.0035**	0.7434±0.0028**	0.7233±0.0135*

Table 14: Life Time Value Prediction Results.

Method	**Criteo-SSC**		**Kaggle**	
	MAE ↓	SRCC ↑	MAE ↓	SRCC ↑
GoR	12.996±0.353**	0.3026±0.094*	67.035±0.112**	0.5334±0.041**

Table 15: Real-Time Bid Shading Results.

Method	**iPinYou**		**Indust_RTB**	
	SR ↑	Surplus ↑	SR ↑	Surplus ↑
GoR (ours)	60.48±0.15%**	114,068,392±132582*	41.74±0.22%**	30,580,080±43543*

Table 16: Watch Time Prediction Results.

Method	**KuaiRec**		**KuaiRand**		**CIKM16**	
	MAE ↓	XAUC ↑	MAE ↓	XAUC ↑	MAE ↓	XAUC ↑
GoR	3.194±0.031**	0.616±0.007**	7.032±0.132*	0.567±0.006**	0.812±0.019**	0.694±0.006**

Table 17: Facial Age Estimation Results.

Method	**UTKFace**		**FG-NET**		**MORPH**		**CACD**	
	MAE ↓	CS ↑	MAE ↓	CS ↑	MAE ↓	CS ↑	MAE ↓	CS ↑
GoR	3.43±0.092**	66.58±0.145**	4.68±0.115**	85.66±0.032**	2.69±0.028**	64.95±0.232**	3.73±0.068*	75.29±0.105**

G LIMITATION AND FUTURE

While GoR establishes a novel generative paradigm for ordinal regression and achieves state-of-the-art performance, several limitations exist, opening promising avenues for future research.

First, the autoregressive nature of GoR, while enabling sequential modeling, incurs an inference latency cost that is proportional to the output sequence length. This poses a challenge for tasks requiring rapid prediction of long sequences.

However, similar to the early challenges faced by initial Transformer models with non-linear time complexities, it is important to emphasize that GoR achieves SOTA results across 17 diverse OR datasets spanning six domains, which represents a foundational step toward a more powerful generative paradigm. Furthermore, even in industrial-scale recommendation systems with stringent real-time requirements, GoR achieves a +0.112% increase in app usage time (p-value = 0.01) in A/B tests on a real-world platform with over 400 million DAUs (See Appendix. F.3.5), demonstrating that the trade-off between latency and performance is acceptable. Nevertheless, exploring non-autoregressive generative architectures remains a meaningful direction for further reducing inference overhead.

Second, GoR, like other language generation models, is susceptible to the risk of error accumulation. Errors in predicting earlier tokens can compound in subsequent steps, potentially leading to larger deviations in the final prediction. Exploring sequence-level optimization or calibration strategies, such as those based on reinforcement learning, could help alleviate this issue.