NAVIGATING USER BEHAVIOR TOWARD PERSONALIZED MULTIMODAL GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern AIGC pipelines deliver high-fidelity images and videos but presuppose a well-formed creation instruction, while end users rarely articulate visual details, so personalized content generation must turn a user's interaction history into an executable instruction that aligns downstream synthesis with user demand. Serializing behavior is largely settled—discrete item identifiers from generative recommendation already provide a serviceable interface—so the open obstacles lie in learning: the desired instruction is unobserved and underdetermined, and no signal ties a written instruction back to user intent. We therefore propose **NaviGen**. An evolution-guided induction phase manufactures its own supervision, using evolutionary search to discover history-to-instruction traces from which the model distills preference reasoning and instruction writing. A closed-loop alignment phase then optimizes a triangular self-consistency reward that couples the written instruction, the model's own preference prediction, and the target semantics, so instruction quality earns credit only when it is user-grounded. Experiments across diverse domains show that NaviGen improves personalized image and video generation and yields more specific, relevant, and visually generatable instructions, with next-item prediction reported as an intermediate diagnostic. Our code is released at: `https://anonymous.4open.science/r/NaviGen-08F8/`.

1 INTRODUCTION

Multimodal content generation, such as text-to-image posters and short videos, is rapidly becoming a core productivity layer for media, marketing, and e-commerce (Xu et al., 2025; Ling et al., 2026). A modern creation pipeline refines a textual instruction with a language model and renders it into an image or video via a text-to-vision generator (Yang et al., 2025; Seedance et al., 2026), where the instruction serves as the central control signal of what is generated and how it looks.

Recent progress along this pipeline strengthens generation backbones (Yang et al., 2025; Seedance et al., 2026), enriches short user inputs into detailed instructions that better exploit generator capacity (Xu et al., 2025; An et al., 2026; Dang et al., 2026), and injects reference, layout, or identity conditions for fine-grained control (Zhao et al., 2025b; Ling et al., 2026).

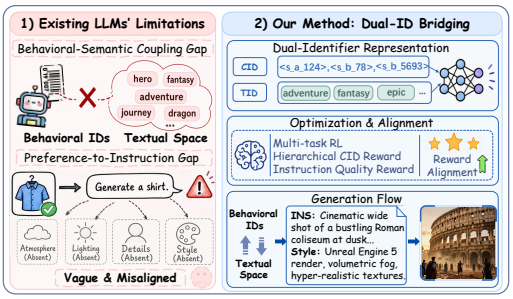


Figure 1: Personalized Multimodal Generation.

However, all of these methods presuppose **a well-formed creation instruction as input**, leaving a fundamental question unanswered: *whose taste is the content actually for?* End consumers differ widely in taste and rarely articulate visual details, so without a path from user signals to a concrete instruction, even the strongest generators produce content that is generic or misaligned with real demand. This motivates a new problem we term ***personalized content generation***: translating a user's implicit behavior history into a creation instruction that steers the downstream generator toward content the user truly wants.

Realizing this paradigm first requires serializing behavior into the language model's vocabulary, and we treat this as an interface question with an off-the-shelf answer rather than a research contribution:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

generative recommenders already compress items into discrete residual codes (Rajput et al., 2023; Deng et al., 2025), and we reuse that recipe unchanged alongside LLM-compressed textual terms so that behavioral structure and semantics enter one token stream. With the interface settled, two genuine challenges remain, both on the learning side. **(C1) The supervision available for instruction writing grades the wrong thing.** Learning to write text from behavior is not unexplored, but existing supervision does not transfer. Personalized generation pipelines tailor the caption of a reference item (Ling et al., 2026), a target that is unavailable once deployed (Liu et al., 2025) and that teaches description of a known item rather than anticipation of an unknown one. The deeper obstacle is that a good instruction is underdetermined: many texts serve the same user well, so supervision must convey *relative* quality among plausible candidates instead of nominating one of them as correct. **(C2) No signal closes the loop from instruction back to user intent.** The two signals that could supervise an instruction are optimized in isolation: AIGC quality scorers grade prompt alignment and fidelity, blind to whose taste is served (Sun et al., 2025), while generative recommenders align to preference but stop at a ranked item, never at the text that renders it (Zhang et al., 2026). Summing them changes little, as each is separately satisfiable: vivid detail scores well for any user, a right item for any text. Closing the loop requires the two to *certify each other*: quality should count only when the instruction, the preference it implies, and the user's next interest agree.

We propose NaviGen, which learns the missing path from behavior to generation-ready instructions in two stages. **Evolution-guided instruction induction** manufactures the relative signal that C1 finds missing: for each history, an evolutionary search seeded with three founder strategies and steered by a multi-dimensional LLM judge refines candidate instructions across rounds, and the model is trained not on the surviving instruction alone but on a distilled trace of how the search reached it, so it learns which refinements make an instruction better and how preference evolves rather than paraphrasing history. **Closed-loop reward alignment** then answers C2 through group-relative optimization under a triangular reward: the instruction must anchor to target semantics, stay consistent with the model's own preference prediction, and that prediction must in turn match the user's next interest, so surface quality earns no credit when the loop fails to close. Target semantics enter only as an offline training signal rather than as an input, so deployment reduces to a single policy pass over the user's history.

Our contributions are summarized as follows:

- We formulate **personalized content generation**, turning an implicit interaction history into an executable creation instruction, and instantiate it in NaviGen, a single pipeline that connects behavioral modeling to controllable multimodal generation. Behavior serialization reuses standard discrete identifiers from generative recommendation and serves only as a supporting interface.
- We propose Evolution-guided instruction induction, which converts an LLM-judged evolutionary search into history-to-instruction targets and distilled traces, so the model learns instruction writing and preference-evolution reasoning from relative quality among candidates.
- We close the loop between instruction and user intent with a triangular self-consistency reward, optimized jointly with a coarse-to-fine behavioral reward under group-relative policy optimization, crediting instruction quality only when implied preference agrees with the user's next interest.
- Across different domains, NaviGen improves personalized image and video generation quality and instruction specificity, relevance, and visual generatability, while answering each user in one policy pass; behavior-space prediction is reported as an intermediate diagnostic and remains competitive.

2 METHODOLOGY

This section presents the technical details of NaviGen. Figure 2 shows the overall architecture.

2.1 PRELIMINARIES: THE DUAL-IDENTIFIER INTERFACE

Given a user history $\mathcal{H}_u = \langle x_1, \ldots, x_n \rangle$ of engaged items carrying visual and semantic metadata, NaviGen produces a creation instruction $\mathcal{I}$, which a fixed multimodal generator g_ϕ renders into an image or video. Before an LLM can reason over behavior, each entry $x_k \in \mathcal{H}_u$ must be serialized into tokens within the model's vocabulary. Feeding captions or metadata directly is verbose and injects redundant noise that slows optimization and blurs preference signals, so we encode each entry with a compact *dual identifier* that decouples sequence-level identity from semantic grounding.

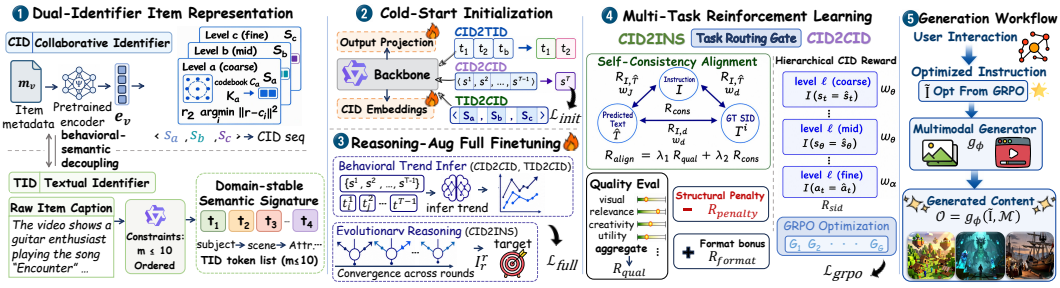


Figure 2: Architecture of the proposed NaviGen framework for personalized multimodal generation.

Collaborative Identifier (CID). We adopt the residual-quantized identifier construction standard in generative recommendation (Deng et al., 2025) to encode collaborative behavior. For an item v, its metadata m_v is mapped to an embedding $\mathbf{e}_v = \psi(m_v)$ by a pretrained embedding model ψ, then quantized by multi-layer residual K-means into a three-level code:

$$s_\ell = \arg\min_k \|\mathbf{r}_\ell - \mathbf{c}_\ell^k\|^2, \quad \mathbf{r}_{\ell+1} = \mathbf{r}_\ell - \mathbf{c}_\ell^{s_\ell}, \quad \mathrm{cid}(v) = \langle s_1(v), s_2(v), s_3(v)\rangle, \tag{1}$$

where $\mathbf{r}_1 = \mathbf{e}_v$ is the initial residual, $\mathbf{c}_\ell^k$ is the k-th centroid of the ℓ-th codebook, and $s_\ell \in \{1, \ldots, K_{\mathrm{cb}}\}$ is the code assigned at level ℓ. The property we rely on is the coarse-to-fine ordering of the three levels, which admits partial-credit supervision because matching any level yields a signal. Each CID token is added to the vocabulary and initialized via dedicated training (Section 2.2.1).

Textual Identifier (TID). Unlike existing work that treats TIDs as fixed-length prediction targets (Zhang et al., 2026), we note that semantically equivalent item variants may use different terms, which makes exact next-TID prediction overly restrictive as supervision, and that entries differ widely in semantic complexity. We therefore allow a variable number of terms,

$$\mathrm{tid}(v) = [t_1, t_2, \ldots, t_L], \quad L \le L_{\max}, \tag{2}$$

where each t_k is a concise phrase capturing a core semantic dimension (*e.g.*, subject category, key attribute). Terms are ordered by importance and produced by compressing captions through an LLM under controlled output constraints, giving a compact, deduplicated, domain-stable semantic signature that bridges preference modeling and instruction generation.

2.2 Evolution-Guided Instruction Induction

NaviGen uses two supervised stages: first adapting the newly introduced identifier tokens, then learning the instruction-induction tasks and auxiliary preference objectives. The second stage is where the supervision missing from behavior data is manufactured by search.

2.2.1 Cold-Start Embedding Initialization

To prevent random CID embeddings from destabilizing pretrained weights through noisy gradients, we decouple representation acquisition from backbone adaptation: all pretrained weights are frozen, and only parameters tied to the new tokens are updated. Let $\mathcal{D}_{\mathrm{init}}$ denote the auxiliary training set, pairing an input sequence $\mathbf{p}_i$ with target tokens y_i. The initialization loss is:

$$\mathcal{L}_{\mathrm{init}} = -\sum_i \sum_t \log p(y_{i,t} \mid \mathbf{p}_i, y_{i,<t}; \mathcal{E}_{\mathrm{CID}}, \mathbf{W}_{\mathrm{out}}) \tag{3}$$

where $\mathcal{E}_{\mathrm{CID}}$ and $\mathbf{W}_{\mathrm{out}}$ are the learnable CID embeddings and output projection layer. Three auxiliary tasks establish bidirectional CID–TID alignment: **CID2TID** maps a CID to its TID, grounding behavioral codes in semantic terms; **TID2CID** inverts that mapping, constructing behavioral identifiers from semantic signals; and **CID2CID** predicts a target CID from a history of CIDs, capturing sequential behavioral patterns that drive user preference.

2.2.2 FULL FINETUNING WITH DISTILLED TRACES

The initialization stage equips the model with stable CIDs, but two capabilities essential for behavior-conditioned instruction generation remain absent: (i) translating inferred preferences into generation-ready instructions, and (ii) reasoning over how user interests evolve along the interaction history. To instill both capabilities, we unfreeze all parameters and augment the existing objectives with a new **CID2INS** task and traces distilled from evolutionary search.

CID2INS. Since no dataset pairs a behavior history with a creation instruction, and since many instructions can serve the same user, we do not look for a single reference but synthesize a scored family of candidates by evolutionary search and keep its best member. Starting from three founder strategies (conservative, balanced, exploratory), each of R rounds selects the two strongest candidates via a multi-dimensional judge and produces two offspring through crossover and controlled mutation. The search leaves behind a candidate trajectory $\mathcal{C}$ and a single retained target $\mathcal{I}^\star$:

$$\mathcal{C} = \{\mathcal{I}^{(r)}\}_{r=0}^{R}, \qquad \mathcal{I}^\star = \arg\max_{\mathcal{I}\in\mathcal{P}^{(R)}} \mathcal{S}_{\text{judge}}(\mathcal{I}), \tag{4}$$

where $\mathcal{I}^{(r)}$ is the highest-scoring instruction of round r, so $\mathcal{I}^{(0)}$ is the strongest founder and $\mathcal{C}$ records how quality improves as the search proceeds; $\mathcal{P}^{(R)}$ is the population surviving the final round, and $\mathcal{S}_{\text{judge}}$ is the LLM-based multi-dimensional scorer that ranks candidates. Only $\mathcal{I}^\star$ serves as the instruction target, and the model is jointly supervised to predict the target TID, anchoring the generated instruction to the intended visual semantics and thereby preventing semantic drift.

Reasoning. Chain-of-thought traces are distilled from a teacher model. For CID2CID, the teacher articulates how preferences evolve along the historical identifier sequence $h_{1:n}$, where $h_i = \text{cid}(x_i)$, without referencing the target item, aggregating consecutive shifts into a reasoning chain $\mathcal{T}\big(\{h_{i-1} \to h_i\}_{i=2}^{n}\big)$. For CID2INS, the reasoning summarizes how candidate instructions converged toward the target semantics across rounds. Formally, starting from the founder population $\mathcal{P}^{(0)}$, the round-r update combines elitism, crossover, and mutation:

$$\mathcal{P}^{(r)} = \big\{\, \mathcal{I}^{(r)}_{\text{elite}},\ \mathcal{I}^{(r)}_{\text{elite}} \oplus \mathcal{I}^{(r)}_{\text{mate}},\ \mu(\mathcal{I}^{(r)}_{\text{elite}}) \,\big\}, \tag{5}$$

where $\mathcal{I}^{(r)}_{\text{elite}} = \mathcal{I}^{(r)}$ and $\mathcal{I}^{(r)}_{\text{mate}}$ are the top-2 candidates in $\mathcal{P}^{(r-1)}$, $\oplus$ fuses the elite's anchor with the mate's visual expressiveness, and $\mu(\cdot)$ mutates while preserving the semantic core. Exposing the trajectory $\mathcal{C}$ and its scores teaches the model to connect search dynamics with the output.

Let $\mathcal{D}_{\text{full}}$ denote the reasoning-augmented training set, pairing an input sequence $\mathbf{p}_i$ with an output sequence $\mathbf{y}_i$ that contains a reasoning trace and a structured answer. The full finetuning objective is:

$$\mathcal{L}_{\text{full}} = -\sum_i \sum_t \log p(\mathbf{y}_{i,t} \mid \mathbf{p}_i, \mathbf{y}_{i,<t}; \theta), \tag{6}$$

where $p(\cdot)$ denotes the token-level probability under all trainable parameters θ. The resulting model jointly performs preference-grounded reasoning and behavior-conditioned instruction generation as the control signal for downstream multimodal content synthesis.

2.3 CLOSED-LOOP REWARD ALIGNMENT

Supervised fine-tuning produces a competent generator but does not directly optimize for the quality of personalized outputs, and imitating one reference instruction per history cannot express whether that particular instruction is right *for this user*. NaviGen therefore applies Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to refine the policy π_θ from intra-group relative advantages under rewards that close the loop from the written instruction back to the user's true intent (the multi-task reinforcement-learning stage of Figure 2).

Hierarchical CID Reward. The CID encodes item behavior through three residual levels of decreasing granularity. A match at any level provides a meaningful partial signal, but coarse agreement is weighted more heavily than fine precision, reflecting the intuition that predicting the right item family matters more than nailing the exact variant of the next item. Formally, we compare the predicted and ground-truth codes level by level:

$$\hat{s} = (\hat{s}_1, \hat{s}_2, \hat{s}_3), \quad \tilde{s} = (\tilde{s}_1, \tilde{s}_2, \tilde{s}_3), \quad R_{\text{cid}} = \sum_{\ell=1}^{3} w_\ell \cdot \mathbb{I}[\hat{s}_\ell = \tilde{s}_\ell], \tag{7}$$

where $\hat{s}$ is the CID emitted by the policy and $\tilde{s}$ the CID of the user's next item, both decomposed over the three levels of Eq. (1); $\mathbb{I}[\cdot]$ is the indicator function; and the level weights satisfy $w_1 > w_2 > w_3$ with $\sum_\ell w_\ell = 1$, so a coarse hit earns partial credit while an exact three-level match yields $R_{\text{cid}} = 1$. A small bonus rewards predictions that remain within the CID vocabulary even when they do not match the ground truth, encouraging the model to stay within the identifier space.

Instruction-Aware Reward. For CID2INS, the reward combines instruction quality assessment with a triangular self-consistency check. An LLM-based judge evaluates the instruction along four dimensions—*specificity*, *creativity*, *content quality*, and *visual generatability*, aggregated into R_{qual}. Beyond standalone quality, a closed-loop alignment enforces three mutually reinforcing signals over the ground-truth target **TID** $\tilde{t}$ and the model's own predicted **TID** $\hat{t}$: the instruction must anchor to $\tilde{t}$, remain self-consistent with $\hat{t}$, and $\hat{t}$ must itself align with $\tilde{t}$. The combined reward is:

$$R_{\text{align}} = \gamma_1 R_{\text{ins}\leftrightarrow\tilde{t}} + \gamma_2 R_{\text{ins}\leftrightarrow\hat{t}} + \gamma_3 R_{\hat{t}\leftrightarrow\tilde{t}}, \quad R_{\text{ins}} = \lambda_1 R_{\text{qual}} + \lambda_2 R_{\text{align}}, \tag{8}$$

where $\gamma_1, \gamma_2, \gamma_3$ balance the three alignment signals, and λ_1, λ_2 weight quality against consistency. In implementation, the judge is used only for the instruction-side quality term during offline training; CID validity, hierarchical matching, triangular alignment, format/reasoning checks, and the KL penalty provide independent structural constraints. The training judge evaluates textual instructions, whereas the final visual evaluation uses a separate VLM and fixed-output similarity metrics (Section 3.1).

Optimization Objective. We optimize the GRPO objective over a group of G completions:

$$\mathcal{J}_{\text{grpo}} = \mathbb{E}_{\mathbf{p}}[\, \mathcal{L}_{\text{clip}} - \kappa\, \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \,], \tag{9}$$

where the expectation is over training inputs $\mathbf{p}$, $\mathcal{L}_{\text{clip}}$ is the group-averaged clipped loss, and κ the KL penalty weight. The group-relative advantage $\hat{A}_i$ is driven by a composite reward:

$$R_{\text{task}} = \begin{cases} w_{\text{cid}} R_{\text{cid}} + R_{\text{bonus}}, & \text{for CID2CID,} \\ w_{\text{ins}} R_{\text{ins}} + R_{\text{format}} - R_{\text{penalty}}, & \text{for CID2INS,} \end{cases} \tag{10}$$

where R_{bonus} encourages vocabulary adherence, R_{format} verifies JSON parseability and reasoning completeness, and R_{penalty} penalizes violations. The optimized instruction $\tilde{\mathcal{I}}$ from this stage acts as the control signal for the fixed multimodal generator g_ϕ.

3 Evaluation

We evaluate NaviGen by answering six research questions. **RQ1** compares personalized AIGC generation performance. **RQ2** studies module ablations. **RQ3** uses collaborative identifier prediction as a diagnostic of the behavioral channel. **RQ4** studies robustness and offline cost. **RQ5** studies SFT and GRPO hyperparameter sensitivity. **RQ6** examines qualitative instruction cases.

3.1 Experimental Settings

3.1.1 Datasets and Evaluation Protocols

Table 1 reports statistics of the three datasets: **Product** and **Games** are Amazon review domains with captions derived from product metadata, while **Short Videos** comes from the OpenOneRec short-video benchmark, where each item has a textual caption (Zhou et al., 2025).

Table 1: Statistics of the experimental datasets.

Dataset	#Users	#Items	#Interactions	Density
Product	30161	74487	161405	7.18e-5
Games	25912	35443	144337	1.57e-4
Short Videos	35259	153333	209172	3.87e-5

Each instance pairs a user's history with a target item; instances are split 8:1:1 into train/validation/test and evaluated by Recall@K/NDCG@K. For personalized generation we sample 1,000 image and 100 video cases from each held-out split, keeping the instruction modality identical across methods, and compare along four dimensions: **Consistency**, image-instruction CLIPScore and, for videos, average CLIPScore over one frame per second (Hessel et al., 2021); **Relevance**, cosine similarity between the generated instruction and the equal-weight embedding of history and target captions; **Aesthetic**, a VLM judge's visual-quality

Table 2: Personalized AIGC generation performance. Bold marks the best non-Oracle image result; bold* marks the best non-Oracle video result. Aesthetic and Novelty are three-run means.

Method	Output	Games				Product				Short Videos			
		Cons.	Nov.	Aes.	Rel.	Cons.	Nov.	Aes.	Rel.	Cons.	Nov.	Aes.	Rel.
PMG	Image	0.3281	0.5084	0.7424	0.4053	**0.3282**	0.5091	0.7677	0.4044	0.3316	0.5725	0.7376	0.3249
PIGEON	Image	0.3121	0.5449	0.7352	0.4382	0.3104	0.4662	0.7086	0.4347	0.3150	0.5510	0.7291	0.3571
RAGAR	Image	0.3148	0.5002	0.7696	0.4475	0.3122	0.4223	0.7171	0.4204	0.3195	0.5713	0.7410	0.3330
Cipher	Image	0.3213	0.5403	0.7501	0.4273	0.3193	0.4595	0.7362	0.4038	0.3265	0.5936	0.7311	0.3963
	Video	0.6315	0.5245	0.5855	0.4307	0.6104	0.4020	0.4423	0.3890	0.6379	**0.6280***	**0.6830***	0.3595
Prose	Image	0.3173	0.5796	0.7804	0.4052	0.3130	0.4958	0.7398	0.4298	0.3202	0.5936	0.7475	0.3034
	Video	0.6325	0.4740	0.5215	**0.4560***	0.6282	0.4640	0.4185	0.4199	0.6459	0.5355	0.6245	0.3554
TRIPLE	Image	0.3152	0.5082	0.7472	0.4273	0.3093	0.4273	0.7182	0.4038	0.3164	0.5654	0.7236	0.3963
	Video	0.6203	0.4170	0.4290	0.4559	0.6217	0.3510	0.3905	0.4093	0.6419	0.4305	0.5330	0.3315
NPC	Image	0.3238	0.5535	0.7797	0.3728	0.3188	0.4564	0.7338	0.3699	0.3234	0.5825	0.7448	0.2866
	Video	0.5643	0.4935	0.4890	0.3936	0.6006	0.3615	0.4080	0.4012	0.6079	0.5590	0.6335	0.3436
Oracle	Image	0.3441	0.6830	0.8448	0.6034	0.3296	0.6114	0.8341	0.6143	0.3422	0.6785	0.8155	0.5396
	Video	0.6473	0.5440	0.6275	0.5532	0.6355	0.4460	0.5000	0.5647	0.6510	0.5790	0.6800	0.5248
NaviGen	Image	**0.3346**	**0.6479**	**0.8423**	**0.4960**	0.3173	**0.5879**	**0.8350**	**0.4839**	**0.3390**	**0.6686**	**0.8272**	**0.4180**
	Video	**0.6449***	**0.5343***	**0.5997***	0.4322	**0.6336***	**0.4718***	**0.4468***	**0.4834***	**0.6478***	0.5528	0.6636	**0.4002***

score in $[0, 1]$ (Kao et al., 2017); and **Novelty**, the same judge's novelty and interestingness score in $[0, 1]$ (Vargas & Castells, 2011). The two judged metrics average three independent calls on fixed outputs; the other two are deterministic. Judge prompts and task templates are in Appendix A.8.

3.1.2 Baseline Methods

NaviGen is compared with three groups of representative baselines: **i) Personalized Generation Methods:** PMG (Shen et al., 2024), Pigeon (Xu et al., 2025), RAGAR (Ling et al., 2026), CIPHER (Gao et al., 2024), PROSE (Aroca-Ouellette et al., 2025), TRIPLE (Noh et al., 2026); **ii) Collaborative Filtering Methods:** SASRec (Kang & McAuley, 2018), TIGER (Rajput et al., 2023), LC-Rec (Zheng et al., 2024), OpenOneRec-Pretrain (Zhou et al., 2025); and **iii) Prompting and Reference Baselines: NPC** omits user-specific evidence and uses a generic prompt, while **Oracle** conditions on the target caption or TID as a non-deployable upper-bound reference.

Implementation details of NaviGen and the baseline methods are provided in Appendix A.3.

3.2 Overall Performance Comparison (RQ1)

Table 2 reports personalized image and video generation, with Oracle as a non-deployable reference. **Image-Level Gains.** NaviGen wins most non-oracle image comparisons, leading on all metrics for Games and Short Videos and on Novelty, Aesthetic, and Relevance for Product; Product Consistency is the sole miss. Search-distilled instructions are more specific and executable than generic or caption-style prompts, so the frozen generator yields stronger preference alignment and appearance, with that Product gap the cost of a richer prompt. **Video-Level Trade-offs.** NaviGen is best in 9 of 12 non-oracle video comparisons, winning every Product metric and most Games and Short-Video scores, yet trailing Cipher on Short-Video Novelty/Aesthetic and Prose on Games Relevance. Video generators are more sensitive to phrasing and temporal constraints, so user-grounded instructions raise frame consistency and alignment more reliably than decorative novelty, which accounts for the remaining losses. **Oracle Reference.** Oracle leads Relevance by conditioning on target semantics unseen at test time, yet NaviGen surpasses it on image Aesthetic for Product and Short Videos and nearly matches its video Consistency. A raw target caption is a semantic label, not a generation-ready instruction, so perceptual quality need not trail a target-conditioned reference even when Relevance is an upper bound. Human evaluation details are in Appendix A.4.

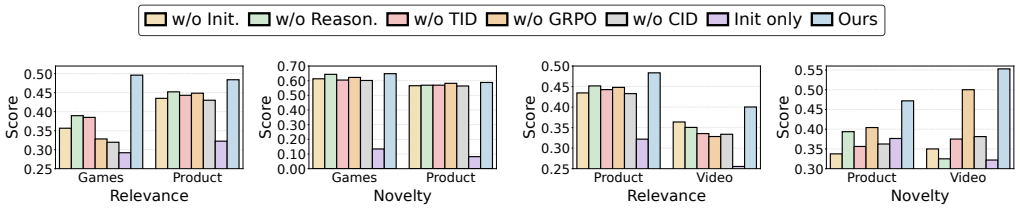
3.3 Ablation Study (RQ2)

Figure 3 and Table 3 ablate NaviGen from the generation and the behavior-modeling side. **Overall Effect.** The full variant is strongest or best balanced across generation relevance, novelty, and CID metrics, so the modules contribute complementary signals to both views. **Reasoning and Reward**

Table 5: Performance comparison on collaborative identifier prediction.

Model	Games				Product				Short Videos			
	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
SASRec	0.0281	0.0364	0.0150	0.0171	0.0017	0.0027	0.0006	0.0009	0.0022	0.0039	0.0013	0.0017
TIGER	0.0294	0.0446	0.0150	0.0190	0.0051	0.0109	0.0022	0.0037	0.0025	0.0050	0.0014	0.0021
LC-Rec	**0.0463**	0.0600	**0.0258**	0.0289	0.0072	0.0143	0.0040	0.0057	0.0042	0.0080	0.0029	0.0041
OneRec	0.0402	0.0545	0.0217	0.0288	0.0068	0.0137	0.0033	0.0050	0.0047	0.0111	0.0034	0.0053
NaviGen	0.0431	**0.0621**	0.0247	**0.0294**	**0.0126**	**0.0211**	**0.0066**	**0.0088**	**0.0155**	**0.0239**	**0.0070**	**0.0092**

Alignment. Removing reasoning supervision hurts both trajectory modeling and generation relevance; ablating GRPO leaves CID retrieval nearly intact but sharply reduces relevance while preserving novelty, *i.e.*, instructions drift toward generic creativity without user grounding. Transition-level reasoning and closed-loop reward alignment are therefore what make outputs generation-ready rather than merely fluent. **Identifier Interface.** Removing TID grounding, CID initialization, or the CID view lowers generation relevance and novelty, and the first two also weaken CID modeling; the initialization-only variant is worst on both sides. Both views of the interface are thus needed to keep generated instructions anchored to behavioral evidence.



(a) Image Generation (b) Video Generation

Figure 3: Ablation study on image and video generation.

Table 3: Ablation study on CID modeling.

Variant	Games		Product	
	R@20	N@20	R@20	N@20
w/o TID	0.0546	0.0235	0.0199	0.0083
w/o Init	0.0546	0.0242	0.0191	0.0079
w/o GRPO	0.0613	0.0292	0.0198	0.0081
Init Only	0.0530	0.0233	0.0092	0.0031
w/o Reasoning	0.0604	0.0284	0.0167	0.0066
NaviGen	**0.0621**	**0.0294**	**0.0211**	**0.0088**

Table 4: Hyperparameter study on SFT checkpoint steps. Ours corresponds to 7128 steps.

Steps	Games		Product		Short Videos	
	Cons.	R@20	Cons.	R@20	Cons.	R@20
2000	0.3499	0.0554	0.3243	0.0201	**0.3450**	0.0554
4000	0.3429	0.0613	0.3325	0.0198	0.3425	0.0613
6000	0.3502	0.0617	**0.3328**	0.0219	**0.3450**	0.0617
Ours	**0.3505**	**0.0629**	0.3319	**0.0232**	0.3409	**0.0629**

3.4 Behavioral Diagnostic: Collaborative Identifier Prediction (RQ3)

We report **CID2CID** prediction, where the model predicts the next CID from a user's CID history, as a diagnostic rather than a headline claim: the identifier construction is inherited unchanged from generative recommendation, so the question is only whether training for instruction generation leaves the behavioral channel intact. Table 5 reports Recall and NDCG. NaviGen is best on most metrics, with clear margins on Product and Short Videos, while LC-Rec keeps a narrow edge at R@10/N@10 on the small Games catalogue; the lead of NaviGen at R@20/N@20 there reflects broader top-k coverage of the identifier space rather than exact-rank precision. Relative to OneRec, which uses the same residual identifier interface, NaviGen leads on every reported CID metric, so instruction-side training does not trade away next-CID prediction skill. The larger margins on Product and Short Videos than on Games line up with those catalogues being substantially sparser, where a jointly trained semantic channel still keeps behavioral prediction usable. The added generation objectives therefore do not degrade the collaborative signal that instructions rest on.

3.5 Robustness and Cost (RQ4)

Robustness. Figure 4 reports four Recall@20 studies; the first two use Games, while the others span all domains. Scaling Qwen3 from 1.7B to 8B improves both OneRec and NaviGen, and NaviGen

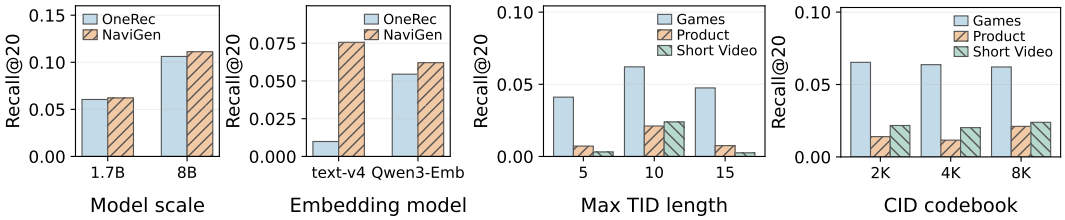


Figure 4: Four robustness studies in terms of Recall@20.

keeps its lead at 8B and across all four Games metrics at both scales, so the observed advantage is not tied to base-model size. The ordering is consistent across text-embedding-v4 and Qwen3-8B-Embedding encoders. TID length 10 is best in all three domains; Games favors a 2K codebook, whereas Product and Short Videos 8K. Full results are in Appendix A.1.

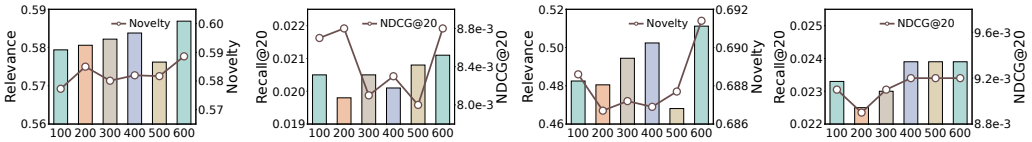
Cost. Table 6 reports the stage-wise training-time breakdown. Cold-Start Embedding Initialization (CSEI) and Reasoning-Augmented Full Finetuning (RAFF) finish in 1.5 h and 2.3 h, while multi-task reward alignment (MTRL) dominates the training budget at 18 h; the 21.8 h total is elapsed wall-clock time on four NVIDIA H20 GPUs (87.2 GPU-hours), plus roughly USD 7 in judge calls. After these three stages, inference uses no evolutionary search and no external judge call—one two-stage policy pass per served user. Appendix A.7 further compares this serving cost with TRIPLE and PROSE on a matched 1.7B serving backbone.

Table 6: Training cost of NaviGen.

Stage	Configuration	Wall-clock / cost
CSEI	3 epochs	$\approx$ 1.5 h
RAFF	3 epochs	$\approx$ 2.3 h
MTRL	600 steps	$\approx$ 18 h
Total	–	$\approx$ 21.8 h
Judge	10–20 s/call	$\approx$ USD 7

3.6 Hyperparameter Study (RQ5)

Table 4 and Figure 5 analyze training-step sensitivity on held-out splits. **SFT Steps.** More supervised tuning improves Recall@20 in all three domains while Consistency stays within a narrow band, so longer SFT strengthens behavioral-trajectory modeling without degrading instruction coherence. The Recall@20 curve rises steeply up to 4k steps and then more slowly, so the default checkpoint sits after the steep region rather than on an unstable climb. **GRPO Steps.** Relevance generally improves toward later checkpoints and novelty stays stable or rises slightly despite mid-training fluctuations, whereas CID prediction is comparatively insensitive—mild fluctuation on Product, an early plateau on Short Videos. Reward alignment therefore refines generation-side preference alignment rather than reshaping the learned identifier space. The two knobs act on different sides of the pipeline: extra SFT mainly helps next-item modeling, while extra GRPO steps mainly help the generated instruction stay user-grounded. A practical reading is therefore to pick SFT steps by CID Recall and GRPO steps by generation Relevance, rather than a single mixed metric.



(a) Product Generation (b) Product Modeling (c) Short Videos Gen (d) Short Videos Modeling

Figure 5: Hyperparameter study on GRPO steps.

3.7 Case Study (RQ6)

Figure 6 compares NaviGen with TRIPLE on a representative history-guided generation case. The recent user history moves from a humorous elf interaction, through a Qixi-festival in-store couple conflict, to a master–disciple clash in fantasy anime, while the held-out target TID further adds romance and emotional conflict on top of the anime/fantasy frame.

TRIPLE latches onto early surface cues—anime, elf, humorous family play—and writes a generic cross-domain blurb. That instruction never makes the romantic turn, so the rendered image stays on the first topic and scores lower on every metric, especially novelty and aesthetic (0.40 / 0.70 versus 0.75 / 0.90).

NaviGen follows the transitions: it keeps the anime/fantasy setting but specifies a student couple, falling cherry blossoms, and a tender atmosphere that matches the romantic shift. The prompt is visually executable rather than a bag of history tags, which is why consistency and relevance rise with appearance. The case illustrates that NaviGen distills an interaction trajectory into generation-ready visual conditions instead of copying the salient historical topic.

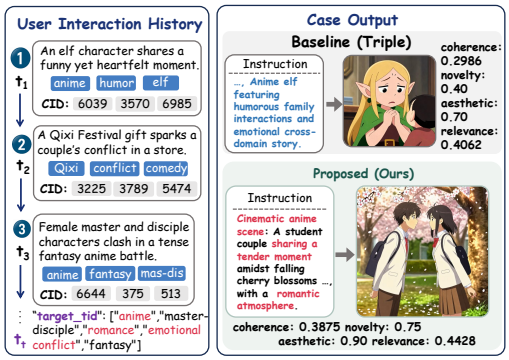


Figure 6: History-guided generation example.

4 Related Work

Personalized Generation. Personalized generation adapts synthesis to users from interaction histories, retrieved evidence, or recommendation-guided signals (Shen et al., 2024; Xu et al., 2025; Ling et al., 2026), or from profiles of edits, demonstrations, or behavioral theories (Gao et al., 2024; Aroca-Ouellette et al., 2025; Noh et al., 2026). Most of this line tailors a reference item unavailable at deployment, while an alternative generates images from history alone (Liu et al., 2025). Both emit profiles or pixels rather than generation-ready instructions; NaviGen converts implicit histories into executable AIGC instructions that any fixed generator can consume.

Multimodal Generative Models. Multimodal generative models turn natural-language instructions into images, videos, and other visual media (Xie et al., 2025; Huang et al., 2025), with VLMs increasingly supporting reasoning and content creation (Hurst et al., 2024; Jin et al., 2025) and AIGC systems improving prompt alignment, visual fidelity, temporal coherence, and motion control (Sun et al., 2025; Seedream et al., 2025; Seedance et al., 2026). These models mainly follow explicit prompts rather than infer generation conditions from sparse user behaviors; NaviGen is complementary by translating behavior-grounded preferences into executable instructions.

Behavioral Preference Modeling. Preference modeling learns user interests from graph, intent-aware, sequential, or multimodal signals (He et al., 2020; Zhang et al., 2024; Zhao et al., 2025a; Ren et al., 2024; Fu et al., 2025; Lin et al., 2025), and LLM-augmented methods interpret descriptions, refine candidates, simulate users, or calibrate decisions (Sun et al., 2023; Qin et al., 2025; Ye et al., 2025). Generative recommenders represent items as discrete residual codes and stop at next-item ranking (Rajput et al., 2023; Deng et al., 2025; Zhou et al., 2025). NaviGen reuses that code construction and, from search-synthesized supervision and a closed-loop reward, treats preference prediction as an intermediate step toward a generation-ready instruction—a target for which the recommendation literature supplies neither labels nor training signal.

5 Conclusion

We presented NaviGen, a behavior-aware framework that turns user interaction history into executable instructions for personalized AIGC. Its contribution is on the learning side: evolution-guided instruction induction manufactures history-to-instruction supervision that no dataset provides, and closed-loop reward alignment credits an instruction only when the preference it implies agrees with the user's next interest. Behavior serialization rests on standard discrete identifiers reused from generative recommendation and serves as a supporting interface, with preference prediction treated as an intermediate reasoning step rather than an end in itself. Experiments across product, game, and short-video domains show improved personalized image and video generation and instruction quality at one policy pass per user, with competitive behavior-space prediction as a diagnostic. Scale, embedding, and identifier studies confirm robustness across the tested configurations, and human and diversity analyses corroborate user-oriented quality. Interactive preference refinement, long-video generation, and broader backbone families are promising extensions.

Ethics Statement

The main evaluation uses existing public datasets and item captions. It does not infer sensitive demographic attributes or deploy personalized generation as a user-facing service. A separate owner-rating check uses purchase histories that 29 volunteers consented to share; those histories are used only to generate and score outputs for this study, are de-identified, and are not released. The generated instructions are evaluated only as experimental outputs for studying preference-to-instruction modeling. Any deployment of personalized generation should rest on explicit user opt-in, minimized retention of interaction histories, and user controls for inspecting or deleting preference profiles; we do not identify additional ethical concerns specific to this work.

AI Use Statement

We used generative AI tools to polish the writing of this paper and to improve the clarity, fluency, and consistency of the exposition. These tools served only as an aid to presentation. Every AI-assisted passage was checked and, where needed, revised by the authors. The authors take full responsibility for the final content of this work.

Reproducibility Statement

We release the code and describe the data processing, training configuration, reward and evaluator setup, human-evaluation protocol, and prompt templates in the main text and appendix. Diversity diagnostics appear in the appendix. All results use fixed held-out samples and fixed generation settings, and the appendix records the corresponding reproducibility details.

References

Ruichuan An, Sihan Yang, Renrui Zhang, Ming Lu, Gaole Dai, Hao Liang, Ziyu Guo, Shilin Yan, Yulin Luo, and Bocheng Zou. Unictokens: Boosting personalized understanding and generation via unified concept tokens. *Advances in Neural Information Processing Systems*, 38:144638–144664, 2026.

Stéphane Aroca-Ouellette, Natalie Mackraz, Barry-John Theobald, and Katherine Metcalf. Aligning LLMs by predicting preferences from user writing samples. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 1690–1721. PMLR, 2025.

Yufan Dang, Chen Qian, Xueheng Luo, Jingru Fan, Zihao Xie, Ruijie Shi, Weize Chen, Cheng Yang, Xiaoyin Che, Ye Tian, et al. Multi-agent collaboration via evolving orchestration. *Advances in neural information processing systems*, 38:165025–165059, 2026.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. OneRec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*, 2025.

Junchen Fu, Xuri Ge, Xin Xin, Alexandros Karatzoglou, Ioannis Arapakis, Kaiwen Zheng, Yongxin Ni, and Joemon M Jose Joemon. Efficient and effective adaptation of multimodal foundation models in sequential recommendation. *IEEE TKDE*, 2025.

Ge Gao, Alexey Taymanov, Eduardo Salinas, Paul Mineiro, and Dipendra Misra. Aligning llm agents by learning latent preference from user edits. *Advances in neural information processing systems*, 37:136873–136896, 2024.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*, pp. 639–648, 2020.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP*, pp. 7514–7528, 2021.

Wenxuan Huang, Shuang Chen, Zheyong Xie, Shaosheng Cao, Shixiang Tang, Yufan Shen, Qingyu Yin, Wenbo Hu, Xiaoman Wang, Yuntian Tang, et al. Interleaving reasoning for better text-to-image generation. *arXiv preprint arXiv:2509.06945*, 2025.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Zhipeng Jin, Wen Tao, Yafei Li, Yi Yang, Cong Han, Shuanglong Li, and Lin Liu. Large vison-language foundation model in baidu aigc image advertising. In *KDD*, pp. 2303–2312, 2025.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pp. 197–206. IEEE, 2018.

Yueying Kao, Ran He, and Kaiqi Huang. Deep aesthetic quality assessment with semantic information. *TIP*, 26(3):1482–1495, 2017.

Xixun Lin, Rui Liu, Yanan Cao, Lixin Zou, Qian Li, Yongxuan Wu, Yang Liu, Dawei Yin, and Guandong Xu. Contrastive modality-disentangled learning for multimodal recommendation. *ACM TOIS*, 43(3):1–31, 2025.

Run Ling, Wenji Wang, Yuting Liu, Guibing Guo, Haowei Liu, Jian Lu, Quanwei Zhang, Yexing Xu, Shuo Lu, and Yun Wang. Ragar: retrieval augmented personalized image generation guided by recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15278–15286, 2026.

Jiongnan Liu, Zhicheng Dou, Ning Hu, and Chenyan Xiong. Generate, not recommend: Personalized multimodal content generation. *arXiv preprint arXiv:2506.01704*, 2025.

Taehyung Noh, Seungwan Jin, Haein Yeo, and Kyungsik Han. Triple: Theory-driven integration of planned and habitual behaviors for llm-based personalization. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence (AAAI-26)*. AAAI Press, 2026.

Weicong Qin, Yi Xu, Weijie Yu, Chenglei Shen, Xiao Zhang, Ming He, Jianping Fan, and Jun Xu. More: A mixture of reflectors framework for large language model-based sequential recommendation. In *Recsys*, pp. 299–308, 2025.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315, 2023.

Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. Representation learning with large language models for recommendation. In *WWW*, pp. 3464–3475, 2024.

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Team Seedream, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, et al. Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Xiaoteng Shen, Rui Zhang, Xiaoyan Zhao, Jieming Zhu, and Xi Xiao. Pmg: Personalized multimodal generation with large language models. In *Proceedings of the ACM Web Conference 2024*, pp. 3833–3843, 2024.

Shangkun Sun, Xiaoyu Liang, Bowen Qu, and Wei Gao. Content-rich aigc video quality assessment via intricate text alignment and motion-aware consistency. *arXiv preprint arXiv:2502.04076*, 2025.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is chatgpt good at search? investigating large language models as re-ranking agents. *arXiv preprint arXiv:2304.09542*, 2023.

Saúl Vargas and Pablo Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *Recsys*, pp. 109–116, 2011.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *International Conference on Learning Representations*, volume 2025, pp. 28240–28264, 2025.

Yiyan Xu, Wenjie Wang, Yang Zhang, Biao Tang, Peng Yan, Fuli Feng, and Xiangnan He. Personalized image generation with large multimodal models. In *Proceedings of the ACM on Web Conference 2025*, pp. 264–274, 2025.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025.

Yuyang Ye, Zhi Zheng, Yishan Shen, Tianshu Wang, Hengruo Zhang, Peijun Zhu, Runlong Yu, Kai Zhang, and Hui Xiong. Harnessing multimodal large language models for multimodal sequential recommendation. In *AAAI*, volume 39, pp. 13069–13077, 2025.

Yi Zhang, Lei Sang, and Yiwen Zhang. Exploring the individuality and collectivity of intents behind interactions for graph collaborative filtering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1253–1262, 2024.

Zhiyang Zhang, Junda She, Kuo Cai, Bo Chen, Shiyao Wang, Xinchen Luo, Qiang Luo, Ruiming Tang, Han Li, Kun Gai, et al. Unleashing the native recommendation potential: Llm-based generative recommendation via structured term identifiers. *arXiv preprint arXiv:2601.06798*, 2026.

Chu Zhao, Enneng Yang, Yuliang Liang, Jianzhe Zhao, Guibing Guo, and Xingwei Wang. Symmetric graph contrastive learning against noisy views for recommendation. *ACM TOIS*, 43(3):1–28, 2025a.

Yibo Zhao, Liang Peng, Yang Yang, Zekai Luo, Hengjia Li, Yao Chen, Zheng Yang, Xiaofei He, Wei Zhao, Qinglin Lu, et al. Local conditional controlling for text-to-image diffusion models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 10492–10500, 2025b.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering*, pp. 1435–1448. IEEE, 2024. doi: 10.1109/ICDE60146.2024.00118.

Guorui Zhou, Honghui Bao, Jiaming Huang, Jiaxin Deng, Jinghao Zhang, Junda She, Kuo Cai, Lejian Ren, Lu Ren, Qiang Luo, et al. Openonerec technical report. *arXiv preprint arXiv:2512.24762*, 2025.

A APPENDIX



(a) Image: Game



(b) Image: Product



(c) Video: Short Video

Figure 7: Comprehensive generation cases across three representative domains (game, product, and short video), each annotated with its target TID.

Table 7: Complete CID prediction scale study.

Scale	Method	Games				Product				Short Videos			
		R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
1.7B	OneRec	0.0402	0.0545	0.0217	0.0288	0.0068	0.0137	0.0033	0.0050	0.0047	0.0111	0.0034	0.0053
1.7B	NaviGen	0.0431	0.0621	0.0247	0.0294	0.0126	0.0211	0.0066	0.0088	0.0155	0.0239	0.0070	0.0092
8B	OneRec	0.0734	0.1062	0.0413	0.0495	0.0226	0.0318	0.0132	0.0174	0.0272	0.0391	0.0136	0.0168
8B	NaviGen	0.0779	0.1112	0.0426	0.0511	0.0467	0.0597	0.0280	0.0313	0.0511	0.0683	0.0278	0.0321

Table 8: CID prediction under alternative text-embedding backbones with fixed settings.

Embedding	Method	Games				Product				Short Videos			
		R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
text-v4	OneRec	0.0063	0.0099	0.0029	0.0038	0.0003	0.0020	0.0001	0.0005	0.0543	0.0793	0.0295	0.0343
text-v4	NaviGen	0.0502	0.0756	0.0242	0.0306	0.0116	0.0283	0.0057	0.0099	0.0580	0.0838	0.0314	0.0379
Qwen3-Emb	OneRec	0.0402	0.0545	0.0217	0.0288	0.0068	0.0137	0.0033	0.0050	0.0047	0.0111	0.0034	0.0053
Qwen3-Emb	NaviGen	0.0431	0.0621	0.0247	0.0294	0.0126	0.0211	0.0066	0.0088	0.0155	0.0239	0.0070	0.0092

A.1 Scale and Identifier Robustness

Table 7 reports the complete CID prediction comparison used in the scale analysis. The 1.7B rows reproduce the default setting; the 8B rows use the same training protocol with the larger base model.

Table 8 compares the two text-embedding backbones used for CID construction, with the base model fixed to Qwen3-1.7B and the maximum TID length fixed to 10. Across both backbones and all three domains, NaviGen outperforms OneRec on every metric, confirming that its advantage does not hinge on a particular text encoder. Here, 'text-v4' denotes text-embedding-v4 and 'Qwen3-Emb' denotes Qwen3-8B-Embedding. Tables 9 and 10 vary the maximum TID length and CID codebook size, respectively, under the default base model and fixed training budget. For each study, all methods use the same data split, seeds, decoding settings, and number of training steps; only the named component varies.

Games across scales. On the small Games catalogue, NaviGen leads OneRec on all four metrics at both 1.7B and 8B, and its lead is wider still on Product and Short Videos. Games has by far the smallest catalogue (35K items), where a compact codebook already yields well-populated codes and exact top-10 retrieval is close to saturated, so the absolute differences there are small; the larger and more long-tailed Product (74K) and Short Videos (153K) catalogues are precisely where collaborative identifier modeling pays off. The distribution of the gains also follows from the training objective: the hierarchical CID reward of NaviGen assigns coarse-to-fine partial credit and is co-trained with TID grounding, reasoning, and instruction generation, so it favors covering the correct behavioral family (top-20) over reproducing a single exact top-ranked variant—the regime that matters for downstream personalized generation.

CID-space quality under identifier configurations. Tables 9 and 10 serve as a diagnostic of the CID space itself, since both the semantic granularity of TIDs and the quantization capacity of the codebook determine how well collaborative trajectories can be resolved. A maximum TID length of 10 is consistently best across all three domains. Shorter TIDs (5) under-specify item semantics, so distinct items collapse onto near-identical term sets and the CID–TID alignment tasks provide weak discriminative signal; longer TIDs (15) admit low-salience or redundant terms that dilute the dominant semantic dimensions and lengthen the target sequence, which increases exposure to accumulated decoding errors in TID prediction. The preferred codebook size is domain dependent: Games favors 2,048 codes, whereas Product and Short Videos favor 8,192. Games has the smallest item space (35K items), so a compact codebook already yields well-populated codes and a larger one fragments collaborative signal across sparsely used entries; Product (74K) and Short Videos (153K) have denser and more heterogeneous item spaces, where higher quantization capacity reduces code collisions among semantically distinct items. The absolute recall values are low for all methods because prediction is evaluated over these large item spaces; NaviGen uses CID prediction as an intermediate behavioral signal whose quality is ultimately reflected in the generation-side results of Section 3.3.

Table 9: CID prediction across maximum TID lengths and datasets in three evaluation domains.

Max TID	Games				Product				Short Videos			
	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
5	0.0277	0.0411	0.0152	0.0186	0.0041	0.0072	0.0019	0.0027	0.0019	0.0031	0.0010	0.0013
10	0.0431	0.0621	0.0247	0.0294	0.0126	0.0211	0.0066	0.0088	0.0155	0.0239	0.0070	0.0092
15	0.0320	0.0475	0.0175	0.0214	0.0031	0.0075	0.0016	0.0027	0.0017	0.0025	0.0007	0.0009

Table 10: CID prediction across codebook sizes and datasets under fixed settings.

Size	Games				Product				Short Videos			
	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
2,048	0.0494	0.0653	0.0280	0.0320	0.0095	0.0140	0.0054	0.0065	0.0138	0.0217	0.0062	0.0081
4,096	0.0415	0.0637	0.0238	0.0293	0.0089	0.0116	0.0044	0.0051	0.0127	0.0202	0.0057	0.0076
8,192	0.0431	0.0621	0.0247	0.0294	0.0126	0.0211	0.0066	0.0088	0.0155	0.0239	0.0070	0.0092

A.2 Baseline Methods

To ensure a comprehensive study, we compare NaviGen against a broad set of baselines covering personalized generation, collaborative filtering and broader behavioral preference modeling, and prompting-based references.

Personalized Generation Methods

- **PMG** Shen et al. (2024): LLM-extracted user cues condition multimodal generators for personalized content synthesis, guided by behavior-aware prompts.
- **PIGEON** Xu et al. (2025): Retrieved preference evidence steers frozen generation agents without model fine-tuning, preserving efficient deployment workflows.
- **RAGAR** Ling et al. (2026): Semantic retrieval weights relevant histories, while ranking feedback balances personalization and fidelity across interaction-rich scenarios.
- **CIPHER** (Gao et al., 2024): Historical user edits are retrieved to infer preferences and align generated outputs with user intent through edit-aware context matching.
- **PROSE** (Aroca-Ouellette et al., 2025): Iterative refinement and consistency checks infer user preferences from demonstrations, using structured self-verification loops.
- **TRIPLE** (Noh et al., 2026): A theory-driven LLM-based personalization framework integrating planned and habitual behavior modeling for preference reasoning.

Behavioral Preference Modeling Methods

- **SASRec** (Kang & McAuley, 2018): Transformer-based sequential preference model learning user behavior from item ID sequences via self-attention.
- **TIGER** Rajput et al. (2023): A generative item-ID modeling framework that converts item IDs into tokens and models sequential preference prediction as next-token generation.
- **LC-Rec** (Zheng et al., 2024): An ID-based sequential preference modeling method that enhances representation learning with additional latent/contextual signals.
- **OpenOneRec** (Zhou et al., 2025): It integrates item-text alignment into an end-to-end generative preference modeling framework for scalable preference prediction and reasoning.

Prompting and Reference Baselines

- **NPC** : No-preference conditioning removes user evidence, such as reference images and similar historical items, using a generic prompt for generation to isolate personalization effects.
- **Oracle** : Ground-truth target semantics, such as the target caption or TID, are used as a non-deployable upper-bound reference for diagnostic comparison only.

A.3 Implementation Details

All baselines are implemented following their original papers. NaviGen uses Qwen3-1.7B as the default base language model. Unless otherwise specified, we use Qwen3.5-Flash for auxiliary method components, including the instruction-aware reward, TID construction, and reasoning generation. For cold-start embedding initialization and full-parameter supervised tuning, we set the learning rate to 5×10^{-4}, the warmup ratio to 0.03, and train for 3 epochs with AdamW, cosine scheduling, a maximum sequence length of 2048, and packed examples. For the reward-alignment stage, we use group size 8, maximum sequence length 2048, weight decay 0.01, learning rate 3×10^{-4}, 600 update steps, and batch size 480; LoRA adapters use rank 16, $\alpha = 32$, and dropout 0.05 on all attention and MLP projections.

Evolutionary search. Supervision for CID2INS is synthesized from three founder instructions (conservative, balanced, exploratory), followed by $R = 1$ round of selection, crossover, and mutation, yielding a final population of three candidates. Each candidate is scored by the LLM judge on consistency, novelty, aesthetic, and executability in $[0, 10]$, aggregated as $0.35 \cdot \text{consistency} + 0.20 \cdot \text{novelty} + 0.25 \cdot \text{aesthetic} + 0.20 \cdot \text{executability}$ with a penalty of 1.0 when consistency or executability falls below 5; the highest-scoring candidate is retained as the training target.

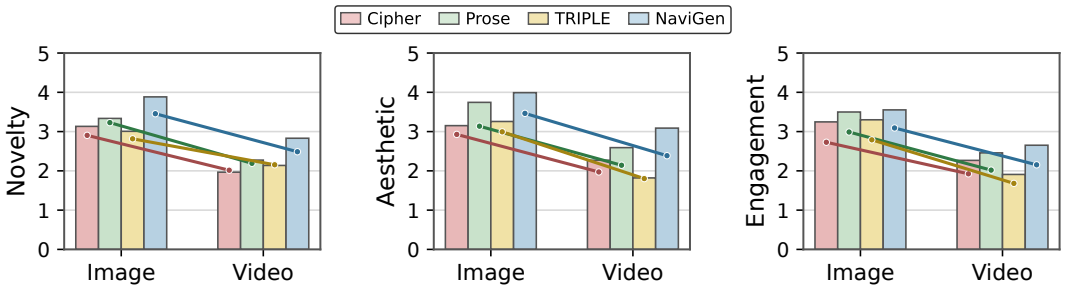
Reward weights. For CID2CID, the hierarchical weights are $(w_1, w_2, w_3) = (0.5, 0.3, 0.2)$, so an exact three-level match yields $R_{\text{cid}} = 1$; an in-vocabulary prediction that matches no level receives $R_{\text{bonus}} = 0.15$, and out-of-vocabulary predictions receive 0. For CID2INS, the four judged dimensions are averaged with equal weights 0.25 into R_{qual}, and R_{ins} averages R_{qual} and the three triangular alignment terms with equal weights 0.25 (*i.e.*, $\lambda_1 = 0.25$, $\lambda_2 = 0.75$, $\gamma_1 = \gamma_2 = \gamma_3 = 1/3$). The task rewards use $w_{\text{cid}} = w_{\text{ins}} = 0.9$ with R_{format} weighted by 0.1. In implementation, R_{penalty} is realized as multiplicative gating: an invalid or empty `<think>` block scales the task reward by a gate factor with floor 0.2, and raw-format violations scale the total reward by a factor with floor 0.3, so malformed completions retain a weak but non-zero learning signal. The KL coefficient is $\kappa = 0.01$ with the standard clipping range $\epsilon = 0.2$. We employ ViT-B/32 for image encoding, text-embedding-v4 for Relevance embedding, and Qwen3-8B-Embedding as the default CID construction encoder. We use GLM-5V-Turbo as the VLM judge, Z-Image-Turbo for 512×512 image generation, and Open-Sora 1.3 for 720p video generation at 24 fps with a total of 81 frames. The default CID codebook size is $K_{\text{cb}} = 8{,}192$ and the maximum TID length is $L_{\max} = 10$. Unless otherwise noted, results use the same held-out samples and fixed generation settings for all methods.

A.4 Human Evaluation on Content

To complement automatic VLM-based evaluation, we conduct a human study over Product, Short Videos, and Games. The expanded study uses 46 volunteers recruited from a university participant pool, anonymized outputs, and random presentation order. From each dataset, we randomly sample 20 image cases and 5 video cases; all compared methods (CIPHER, PROSE, TRIPLE, and NaviGen) are shown in randomized order. For each sampled case, the four generated outputs are presented side by side without method labels, and the method-to-position mapping is independently shuffled for every case. Participants use a 5-point Likert scale, where 1–5 indicate very poor, poor, fair, good, and excellent. Novelty measures creativity and non-triviality, Aesthetic measures visual appeal and polish, and Engagement measures willingness to interact with the content. For Engagement, the target item is withheld: participants see only the recent history and the anonymized generated outputs, so the judgment does not reward target memorization. Scores are averaged across domains and participants. As shown in Figure 8, NaviGen obtains the highest mean ratings on all three dimensions for both image and video settings.

The video Engagement mean is 7.9% higher than the strongest baseline. Because the study is evaluator-independent and target-blind, it provides a complementary check that the gains measured by automatic judges also translate into human-perceived novelty, aesthetic quality, and willingness to interact.

Additional validation. Table 11 summarizes the 46-participant ratings. Its left block gives each mean with an approximate 95% confidence interval and the inter-rater reliability. NaviGen shows moderate-to-high agreement, and its mean advantage over each baseline remains positive for every modality–dimension pair. Holm-adjusted paired tests are significant in all comparisons; the largest



(a) Novelty (b) Aesthetic (c) Engagement

Figure 8: Human evaluation on created content. Bars are third-party means; lines are owner self-ratings.

Table 11: Third-party reliability of NaviGen and its owner self-rating confidence interval. Left of the rule: the 46-participant study. Right: mean and approximate 95% bootstrap confidence interval when the history owner scores NaviGen (18 image cases, 11 video cases).

Modality	Dimension	Mean [95% CI]	ICC(2,1)	ICC(2,k)	Self-rating [95% CI]
Image	Novelty	3.8838 [3.8685, 3.8989]	0.6315	0.9875	3.455 [3.254, 3.661]
Image	Aesthetic	3.9905 [3.9757, 4.0047]	0.6657	0.9892	3.464 [3.075, 3.823]
Image	Engagement	3.5526 [3.5333, 3.5710]	0.6817	0.9900	3.092 [2.719, 3.448]
Video	Novelty	2.8309 [2.7899, 2.8681]	0.7992	0.9946	2.487 [2.002, 2.982]
Video	Aesthetic	3.0878 [3.0522, 3.1247]	0.6356	0.9877	2.386 [2.006, 2.765]
Video	Engagement	2.6515 [2.6159, 2.6913]	0.7416	0.9925	2.154 [1.791, 2.501]

adjusted p-value is 4.27×10^{-7}. The right block reports the owner self-rating mean and 95% confidence interval for NaviGen.

Owner self-ratings. The comparison above is third-party: volunteers score anonymized outputs from public benchmarks, so the rater is not the person whose history produced the item, and the original users of those benchmarks cannot be recontacted. Of the same 46 volunteers, 29 consented to share their own Taobao purchase histories. We run Cipher, Prose, TRIPLE, and NaviGen on each history and ask that volunteer to score all four outputs on the same three 1–5 scales. Eighteen histories are rendered as images and eleven as videos; each output is scored only by its owner. The lines in Figure 8 are these means, and the right side of Table 11 reports the 95% confidence interval for NaviGen. A single owner per item leaves inter-rater ICC undefined for these ratings.

Self-ratings of NaviGen lie below its 46-participant means on every modality–dimension pair. The gaps (self minus third-party) are -0.43, -0.53, and -0.46 for image Novelty, Aesthetic, and Engagement, and -0.34, -0.70, and -0.50 for the same video dimensions. On images the order is unchanged: Aesthetic (3.46) stays just above Novelty (3.45), and both stay above Engagement and above the scale midpoint. Video self-ratings fall below 3 on all three dimensions. Aesthetic drops the most and slips behind Novelty, reversing the third-party order in which video Aesthetic led. Owners are therefore stricter than outside judges, most of all on video polish, while first-person image scores still sit in the fair-to-good range. On the same owner scores, NaviGen has the highest mean on every line in Figure 8. The margin over the next method is smallest on image Engagement, where NaviGen (3.09) leads Prose (2.99) by 0.10, and largest on image Novelty against TRIPLE ($+0.64$). Video intervals are wide ($n = 11$) and overlap the next method, so the owner panel reproduces the third-party ordering without a tight separation on video.

A.5 Evaluator and Reproducibility Details

The training-time instruction judge is Qwen3.5-Flash and operates on text instructions only. The final visual judge is GLM-5V-Turbo; it receives either a static image or sampled video frames and returns JSON scores for Aesthetic and Novelty. The two judge roles therefore use different model families and modalities. Consistency is computed with ViT-B/32 CLIPScore and Relevance with

Table 12: Diversity and discriminability diagnostics.

Modality	Matched pairs	Outputs	Inter-user Div.	Intra-user Div.	Discriminability (95% CI)
Image	312	3,120	0.3795	0.0632	57.50% [52.56%,62.12%]
Video	120	1,200	0.3965	0.0692	64.17% [55.83%,72.50%]

text-embedding-v4 cosine similarity, both on fixed outputs. Prompt templates for TID construction, evolutionary selection, full finetuning, reward alignment, and visual judging are included in Figures 9–14. Data instances use an 8:1:1 train/validation/test split; preprocessing and filtering follow the original dataset conventions, and all methods share the same held-out generation cases, random seeds, and generator settings. The default training configuration, hardware, and offline cost are specified in Sections 3.1 and A.7; the released code provides the corresponding preprocessing and evaluation entry points. For the per-user inference-cost reproduction, both TRIPLE and PROSE are driven by one fixed frontier LLM (Qwen3.5-Flash) at temperature 0 with reasoning disabled, on the same user streams as the matched-1.7B measurement, and calls, prompt/completion tokens, and latency are logged per user.

The instruction-side reward is not a single scalar. It combines an LLM-judged quality term (R_{qual}, over specificity, creativity, content quality, and visual generatability) with three triangular alignment terms and structural constraints: hierarchical CID matching, CID vocabulary validity, JSON/reasoning completeness, format checks, and the KL penalty toward the reference policy. The scalar weights for R_{cid}, R_{align}, R_{ins}, and the reward-alignment objective are fixed by a small validation sweep over the generation and CID-prediction targets and are listed in Appendix A.3; all reward terms are normalized to $[0, 1]$ before weighting. Because the reward is used only for offline training and the evaluation protocol separates the text-side training judge from the visual-side evaluation judge, the model does not optimize the evaluation VLM directly. Nevertheless, we treat model-based evaluators as proxy signals rather than ground-truth human preferences. We did not perform a full judge-calibration study with repeated-call variance, prompt-sensitivity, model-replacement, or Spearman correlation to the human ratings; such evidence remains a useful future extension and is not used to claim that automatic scores are superior to human judgment.

A.6 Diversity and Mode-Collapse Diagnostics

Matched-user generation shows no evidence of severe mode collapse, and discriminability is above chance for both images and videos. We construct matched-user pairs by finding users whose recent histories are semantically similar but whose held-out targets differ, then generate five outputs per user with a fixed generator and decoding setting (Section A.5). This yields 312 image pairs (624 users, 3,120 outputs) and 120 video pairs (240 users, 1,200 outputs). We define *inter-user diversity* as the mean embedding distance between paired users' generated outputs, *intra-user diversity* as the mean distance among one user's five outputs, and *target-blind discriminability* as the accuracy of a nearest-neighbor user-identification task in which the target item is withheld. Embeddings are CLIP features for images and frame-pooled CLIP features for videos; distances are cosine distances normalized to $[0, 1]$. The 95% discriminability interval is obtained by pair-level bootstrap over matched pairs.

For both image and video outputs, discriminability is above the 50% random baseline, and inter-user diversity is substantially larger than intra-user diversity. We additionally estimate domain-level mode-collapse statistics by clustering generated embeddings into 50 clusters for images and 16 for videos. *Cluster coverage* is the fraction of non-empty clusters, *normalized entropy* is the Shannon entropy of the cluster histogram divided by $\log(\text{number of clusters})$, *effective clusters* is the inverse participation ratio of the histogram, and *effective rank* is the number of singular values needed to explain 95% of the variance. Image outputs yield coverage 0.8933, normalized entropy 0.8469, 27.48 effective clusters, and effective rank 73.92; video outputs yield 0.8333, 0.8763, 11.49 effective clusters, and effective rank 19.73. These statistics confirm that NaviGen produces well-spread outputs without severe mode collapse.

Table 13: Per-user inference cost under a frontier instruction LLM (a) and a matched Qwen3-1.7B backbone (b).

Method	Model	Calls	Prompt tok.	Gen. tok.	Latency (s)
(a) Frontier instruction LLM					
TRIPLE	Qwen3.5-Flash	16	7,432	4,157	37.01
PROSE	Qwen3.5-Flash	$\sim$213	28,511	3,731	172.00
RAGAR	Qwen3.5-Flash	11	1,580	281	9.08
(b) Matched Qwen3-1.7B backbone					
TRIPLE	Qwen3-1.7B	16	6,934	3,981	4.72
PROSE	Qwen3-1.7B	$\sim$372	55,079	7,006	12.63
RAGAR	Qwen3-1.7B	15	2,198	565	1.10
NaviGen	Qwen3-1.7B	1 pass (2 stages)	764	221	0.51

A.7 Cost and Inference Overhead

Beyond the training-time breakdown in Table 6, Table 13 reports per-user inference cost. The downstream image/video generator is identical across methods, so the table isolates the instruction-generation (or profile/preference-memory construction) cost. We report two settings. In panel (a) we reproduce TRIPLE, PROSE, and RAGAR at their intended operating point, where all three methods rely on a strong instruction LLM: we drive them with the same model, Qwen3.5-Flash, so the baselines differ only in pipeline structure and none is penalized by a weak backbone. In panel (b) we remove the LLM as a confounder and re-implement every method on the same Qwen3-1.7B backbone served by vLLM on one NVIDIA H20, which is also the deployment setting of NaviGen.

The reproduction makes the cost gap explicit. With a shared frontier LLM, TRIPLE consumes 16 calls and about 7.4k prompt plus 4.2k completion tokens per user and takes 37.0 s, while PROSE consumes roughly 213 calls and about 28.5k prompt plus 3.7k completion tokens and takes 172.0 s, and RAGAR sits in between at 11 calls and about 1.6k prompt plus 0.3k completion tokens in 9.1 s. NaviGen instead answers a user with a single two-stage policy pass of 764 prompt and 221 generated tokens in 0.51 s on one H20. In both panels, *Calls* counts instruction-LLM invocations per user and *latency* is the wall-clock time to answer one user with a single baseline request in flight. All baselines are reproduced with reasoning disabled at temperature 0, and token counts are prompt/completion totals.

The per-user call counts decompose as follows: NaviGen issues a single two-stage policy pass (reasoning then constrained JSON decoding); TRIPLE's 16 calls span one habitual-profile extraction, five TPB-intention extractions, four incremental profile updates, five behavioral-rationale extractions, and one final prediction; PROSE's calls arise, per history item, from preference aggregation, up to five refinement rounds, component breakdown, and a component-by-demonstration consistency verification whose quadratic scaling dominates the total. RAGAR instead issues five batched cue extractions, one retrieval-weighting call, one evidence-aggregation call, and then a generation-plus-ranking round (one generation call and three batched ranking calls), repeating the round only while the best candidate stays below the personalization threshold; the frontier LLM clears that threshold in the first round (11 calls), whereas the 1.7B policy needs the second round (15 calls). The PROSE count is likewise not fixed by design: its refinement and consistency loops terminate only when the check passes, so a weaker backbone needs more rounds and more calls, which is why the count rises from 213 with the shared frontier LLM to 372 on the matched 1.7B backbone, whereas TRIPLE always runs its fixed 16-call pipeline. Because the shared model's absolute latency drifts with load, PROSE's latency is pooled over single-request runs; we therefore read the latency columns as an order-of-magnitude comparison and rely on the call and token counts for the stable per-user cost.

A.8 PROMPT TEMPLATES

The templates below cover the auxiliary generation, search, reasoning, and evaluation stages of NaviGen, including:

- **TID Generation:** Fig. 10 converts item captions into compact semantic TIDs.
- **Evolutionary Search:** Fig. 9 searches for the best target-aligned AIGC prompt.
- **Oneshot Distillation:** Fig. 11 distills the user history and target reasoning into a final first-person reasoning trace.
- **Full-Finetuning Prompts:** Fig. 12 (top) defines the supervised tasks for identifier mapping, collaborative next-item prediction, and AIGC instruction generation.
- **Reward-Alignment Prompts:** Fig. 12 (bottom) defines the preference-prediction and instruction-generation tasks used by the reward-alignment stage.
- **Reasoning Cases and Thinking Generation:** Fig. 13 shows the CID2CID/CID2INS reasoning cases and the prompt that produces the thinking traces from historical TIDs to the target TID.
- **Novelty/Aesthetic Judging:** Fig. 14 illustrates the evaluation of novelty and aesthetics for image and video generation.

Evolutionary Search

SYSTEM_PROMPT = """You are an AIGC creative search teacher model.
Your task is to search for the best AIGC creative instruction for the target item.

Search principles:
1. The target item is the primary objective. Historical TIDs are only auxiliary clues.
2. Each candidate must implicitly point to the target item, but the creative instruction must not explicitly mention, copy, or quote the target item text or tokens.
3. Each candidate must also output a concise reasoning explaining why the prompt is suitable and what information it used.
4. Use multiple rounds of selection, crossover, and mutation to refine the candidate set.
5. Output only one JSON object with fields "prompt" and "reasoning"."""

SCORER_SYSTEM_PROMPT = """You are a strict AIGC prompt evaluation model.
Given the user's historical TIDs, the target TID, and a candidate prompt,
score the candidate independently on the following four dimensions:
1. consistency: whether it is centered on the target item semantics, uses history only as supporting context, and avoids drifting or explicitly naming the target.
2. novelty: whether it introduces reasonable innovation without losing the intended direction.
3. aesthetic: whether it shows clear and layered visual composition ability, including subject, scene, style, camera language, lighting, mood, and key details.
4. executability: whether it is specific, clear, contradiction-free, and directly usable for AIGC generation.

Scoring requirements:
1. Output a floating-point score from 0 to 10 for each dimension.
2. The four dimensions must be judged independently; do not give all high scores just because you like the candidate overall.
3. The reasoning should briefly explain the main strengths and weaknesses and stay within 200 characters.
4. If the prompt explicitly mentions the target item, lower consistency and executability.
5. Output only one JSON object. Do not output Markdown and do not output any extra explanation. """

Figure 9: Evolutionary search prompts that produce target-aligned AIGC candidates and score them.

Tid Generation

SYSTEM_PROMPT = f"""You are a cross-domain item identifier generator.
Your task is to compress the input caption into a Term ID (TID): a set of structured, standardized, semantically dense terms.

Goals:
1. The terms should cover the core semantics of the item.
2. Preserve key information while staying concise and avoiding redundant modifiers.
3. Prefer stable cross-domain signals such as:
   - subject or category - core scene or use case
   - key attributes or selling points
   - style, atmosphere, or audience only when truly important
4. Use standard, general, reusable wording. Avoid synonym stacking and avoid sentence-style expressions.
5. You decide how many terms are needed, but the result must be expressive enough and must not exceed {MAX_TERMS} terms.
6. Each term may be a single word or a very short phrase, but the overall result must stay compact.
7. Order the terms by importance, with the most essential term first.

Strict output requirements:
1. Output JSON only.
2. The format must be {{"terms": ["term1", "term2"]}}.
3. Do not output explanations, reasoning process, Markdown, or extra text.
4. All terms must be in English.
5. Do not use Chinese or any other non-English language in the terms."""

Figure 10: Prompt template that converts item captions into compact, structured Term IDs (TIDs).

Oneshot Distillation

SYSTEM_PROMPT = """You are an AIGC creative-planning teacher model. From the user's TID history, the step1 reasoning, the target TID, and the step2 evolution summary, distill a single final reasoning paragraph.

Requirements:
1. Output one coherent paragraph of first-person internal thinking, with "I" as the subject, written as the student model's own think content — never as a teacher note, analysis report, or explanation to someone else.
2. Open by inferring, from the history and the step1 reasoning, what kind of next item the user is likely to engage with; do not name the final target too early.
3. Then trace how the prompt-writing route closed in on the right answer: what the early drafts got right, which parts were kept, which were rewritten, and how the wording was refined.
4. Use plain language only — no biological-evolution jargon such as selection, crossover, mutation, or founder.
5. Near the end, state the final target TID explicitly and explain why the route converges there, showing how the inferred target direction, the history clues, and the prompt-revision route jointly support the prompt-writing idea.
6. Output plain prose: no XML/HTML tags, Markdown code blocks, bullet points, titles, or template fields."""

USER_PROMPT_TEMPLATE = """User historical TIDs: {history_tids}
Step 1 — Reasoning from history to the next target: {step1_reasoning}
Target TID: {target_tid}
Step 2 — Prompt revision summary: {evolution_summary}

Requirements:
1. Write entirely in first person, using "I" throughout.
2. Condense the Step 1 reasoning as needed, but preserve its core judgment about where the next item likely points.
3. Use the Step 2 summary to explain plainly how the drafts converged on the right final prompt.
4. Avoid technical or biological evolution vocabulary (e.g. selection, crossover, mutation, founder, elite, mate).
5. Near the end, name the target TID explicitly and explain why the route ultimately leads there.
6. Close by describing how I would shape the final prompt around that target direction, treating history as supporting evidence rather than the goal itself.
7. Output exactly one paragraph."""

Figure 11: Distills user history, target reasoning, and prompt refinement into final first-person reasoning.

Full Finetuning Task

"**cid2tid**": """You are an ID mapping assistant.
Map the input cid to its corresponding tid, where tid is the item's metadata.
Respond with a single JSON object containing only the field target_tid."""

"**tid2cid**": """You are an ID mapping assistant.
Map the input tid (the item's metadata) to its corresponding cid.
Respond with a single JSON object containing only the field target_cid.""",

"**cid2cid**": """You are a personalized recommendation assistant.
Predict the target cid interaction from the user's history of cid interactions.
Respond with a single JSON object containing only the field target_cid.""",

"**cid2ins**": """You are a personalized recommendation and AIGC instruction generation assistant.
From the user's history of cid interactions, output the predicted target tid interaction together with the AIGC instruction.
Respond with a single JSON object containing only the fields target_tid and target_ins.""",

GRPO Task

"**cid2cid**": """You are a personalized recommendation assistant.
Predict the target cid interaction from the user's history of cid interactions.
Respond with a single JSON object containing only the field target_cid."""

"**cid2ins**": """You are a personalized recommendation and AIGC instruction generation assistant.
From the user's history of cid interactions, output the predicted target tid interaction together with the AIGC instruction.
Respond with a single JSON object containing only the fields target_tid and target_ins.""",

user = """Task: Predict the target_cid / Generate the target recommendation result based on the user's historical interactions in hist_cid.
hist_cid: {format_hist_cid(hist_cid)} Think first, then output JSON only."""

Figure 12: Prompts for full-finetuning (top) and reward-alignment (bottom) task construction.

Reasoning cases

"**cid2cid**": "I start by analyzing the user's history, which is heavily dominated by video game titles and specific console ecosystems like PlayStation 4 and Xbox, alongside family-friendly franchises such as... Therefore, based on this progression from software consumption to necessary hardware maintenance for the identified console ecosystem, I conclude with the target_tid: Xbox One, Controller Charging Station, Dual Charger, Rechargeable Battery Packs, PDP Gaming, Energizer, Gaming Accessory.",

"**cid2ins**": "By reviewing my past interactions, I notice a consistent thread of interest in female subjects within romantic or social settings, ranging from... where a couple engages in a lively, humorous exchange, using dynamic lighting and handheld camera movements to evoke the festive yet intimate spirit of the occasion, ensuring that the historical preference for romance and lifestyle content serves as a supportive backdrop rather than the sole focus.",

Thinking Generation

SYSTEM_PROMPT = """You are a recommendation reasoning teacher model. Your task is to write a student-model-style reasoning paragraph that starts from the user's historical TIDs and finally points to the given target_tid.

Strict requirements:
1. The output must be one complete, natural paragraph of reasoning text.
2. The reasoning must be written from the perspective of a student model (use first-person thinking style)
3. The first part of the paragraph should explain how to extract semantic clues from history and narrow down candidates toward the correct target_tid.
4. The final part of the paragraph should explicitly conclude with the provided target_tid as the final decision.
5. Do not output any extra explanation, list, or metadata; output only the single reasoning paragraph."""

Figure 13: CID2CID/CID2INS reasoning cases (top) and the thinking-generation prompt (bottom).

Judging: Image

You are AIM-judge for AIGC image generation evaluation.
Evaluate the generated media conditioned on the creative text instruction (ins).
Return JSON only with fields: "novelty_score", "aesthetic_score", "reasoning". Use a 0-1 scale.

NOVELTY (0-1): How creatively does the image interpret the instruction?
A concise instruction leaves more room for creative.
Do NOT equate detail-count in the instruction with novelty; a short but imaginative prompt that yields a distinctive image deserves a high score. Reward visual creativity, unexpected yet fitting compositions, and artistic interpretation.

AESTHETIC (0-1): Evaluate purely on visual appeal — clarity, color harmony, composition, texture, lighting, and overall polish.
The subject matter (product vs. landscape) should NOT bias your score; a well-executed product shot deserves the same score as a dramatic scene of equal technical quality.

Instruction (ins): {ins}

Judging: Video

You are AIM-judge for AIGC video generation evaluation.
Evaluate the generated video conditioned on the creative text instruction (ins).
Return JSON only with fields: "novelty_score", "aesthetic_score", "reasoning". Use a 0-1 scale.

NOVELTY (0-1): How creatively does the video interpret the instruction?
A concise instruction leaves more room for creative interpretation — reward videos that make clever, inventive choices beyond the literal words.
Do NOT equate detail-count in the instruction with novelty; a short but imaginative prompt that yields a distinctive video deserves a high score. Reward visual creativity, unexpected yet fitting compositions, artistic interpretation, and coherent use of multiple frames over time.

AESTHETIC (0-1): Evaluate purely on visual appeal across the sampled frames — clarity, color harmony, composition, texture, lighting, temporal consistency, motion/camera coherence, and overall polish.
The subject matter (product vs. landscape) should NOT bias your score; a well-executed product shot deserves the same score as a dramatic scene of equal technical quality.

Instruction (ins): {ins}

Figure 14: AIM-judge prompts for novelty and aesthetic evaluation of images (top) and videos (bottom).