Unified Personalized Understanding, Generating and Editing

Supplementary Material

A. Details of Dataset

We manually constructed five types of editing operations, each accounting for 1/5 of the evaluation set, which are: object manipulation (e.g., Remove <sks>), attribute modification (e.g., Make <sks> raise their hand.), spatial transformation (e.g., Show <sks> from the side view.), environment interaction (e.g., Make <sks> appear in sunset lighting.), style appearance (e.g., Make <sks> look like a pencil sketch.)

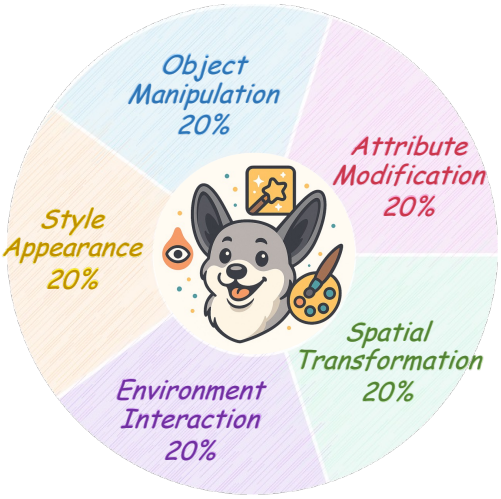

Figure 7. Distribution of five editing operation types in the OmniPBench evaluation set, each accounting for 20% of the dataset.

B. Details of Explicit Knowledge Replay

To maintain consistency in expression, we need to clarify that the knowledge-driven generation mentioned at the beginning of Section 4 of the main paper refers to Personalized Attribute-Reasoning Generation. Then, we provide a comprehensive analysis of our explicit knowledge replay mechanism (Section 3.2), addressing inference efficiency and prompt design.

Computational Overhead Analysis. Section 3.2 describes a conceptual three-stage pipeline (Intent Parser → Memory Retriever → Prompt Composer). Actually we implement this as a **single unified prompt** to minimize latency. Table 3 compares sequential versus unified implementations. Our unified single-stage implementation requires only **1.45s**, achieving a **1.29× speedup** over the sequential approach (1.87s) while providing substantial quality gains (+96.5% PARG score over baseline).

Implementation	Latency (s)	Model Calls	PARG Score
Sequential Three-Stage			
Intent Parser	0.54	1	-
Memory Retriever	0.49	1	-
Prompt Composer	0.84	1	-
Total	1.87	**3**	**0.635**
Unified Single-Stage (Ours)	**1.45**	**1**	0.613
Baseline (No Replay)	0	0	0.312

Table 3. Computational overhead comparison. Our unified prompt achieves 1.29× speedup over sequential implementation with comparable PARG performance.

C. Additional Study of Personalized Editing

We provide a detailed analysis of our personalized editing approach, addressing training data design.

C.1. Training Data Design and Rationale

Focus on Removal Instructions. Our training dataset primarily consists of concept removal instructions (e.g., "Remove <sks> from the image"). While this may appear limited, we emphasize that **training data scope differs from evaluation scope**. The removal task serves as an effective training signal because:

(1) Fine-Grained Localization: Removal forces the model to precisely identify and locate the personalized concept, encoding fine-grained spatial and semantic information in learned concept tokens $\mathbf{P}^{(\text{gen})}$.

(2) Leveraging Pretrained Capabilities: By training only concept tokens while freezing all Transformer parameters (Section E.1), we specialize the concept representation for personalized editing while preserving the backbone's broad pretrained editing skills. This avoids catastrophic forgetting of diverse editing capabilities.

(3) Data Efficiency: Edit data can be straightforwardly constructed automatically via inpainting models, enabling scalable training data generation.

C.2. Baseline Comparisons

Editing Baselines. We acknowledge the possible concern regarding limited editing-specific baselines in Table 1. This reflects a fundamental challenge: to our knowledge, **OmniPersona is the first work to address instructional editing of newly learned personalized concepts** (represented as <sks> tokens).

Personalized Generation Methods (e.g., DreamBooth, IP-Adapter): These methods learn <sks> but focus on generation from scratch, not editing existing images.

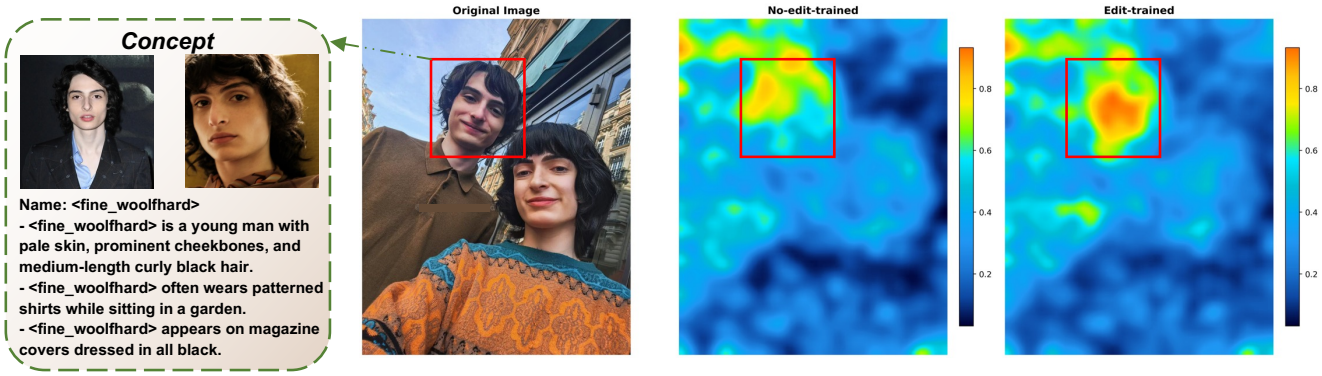

Figure 8. Attention visualization showing focused localization of <sks> tokens after incorporating edit training data.

Adapting them for editing requires both **training an inversion model to encode input images** and **developing instruction-following mechanisms**.

This adaptation is non-trivial and constitutes a research contribution on its own, beyond the scope of comparing to OmniPersona.

C.3. Detailed Analysis of Baseline Comparisons

To provide a clearer illustration of the results presented in Table 1 of the main paper, we summarize the key baseline comparisons with detailed breakdowns.

Method	SEMA-C ↑	QUAL-I ↑	Avg. ↑	Personalization Strategy
Bagel+TP	0.297	0.566	0.432	Text prompt descriptions
GPT-4o+IP	0.512	0.603	0.558	Image prompts
OmniPersona (Ours)	**0.711**	**0.605**	**0.658**	Decoupled concept tokens

Table 4. Detailed breakdown of personalized editing performance across different personalization strategies. **Bagel+TP** uses the same backbone architecture but represents concepts via long-context text descriptions. **GPT-4o+IP** leverages GPT-4o's multimodal capabilities with few-shot image prompts.

Few-Shot Prompting vs. Learned Tokens: Bagel+TP achieves 0.432 average, demonstrating the limitations of text-only concept representations for fine-grained editing tasks. The 52.3% performance gap indicates that learned concept tokens encode richer spatial and semantic information. GPT-4o+IP achieves 0.558 average with notably lower semantic consistency (SEMA-C: 0.512) despite comparable image quality (QUAL-I: 0.603). This reveals challenges in preserving personalized identity through implicit few-shot learning.

The improvements stem from our decoupled token design mechanism, which enable specialized concept representations while leveraging pretrained capabilities.

C.4. Attention Visualization

We examined whether adding edit data during training would help, and visualized the attention between <fine woolfhard> and the image. Using challenging examples with the query "Can you point out where <fine woolfhard> is?", we observe that after incorporating edit data, the model's attention on <fine woolfhard> becomes more focused and captures the concept's fine-grained features more effectively.

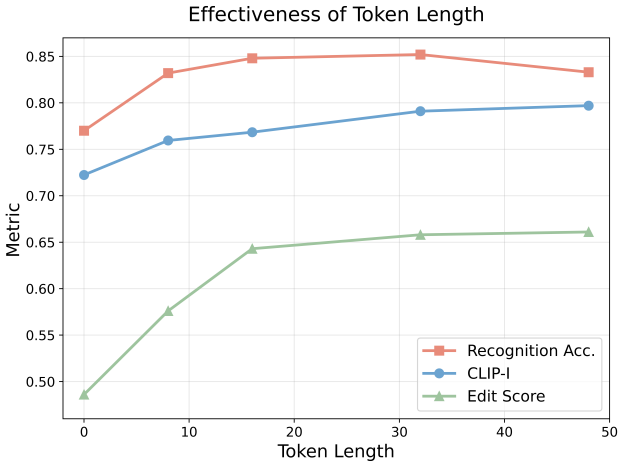

Figure 9. Effect of learnable token number on model performance.

D. In-depth Study of Token Number

We conducted an ablation study on the number of learnable tokens and observed that performance improves as the number of tokens increases, but after reaching a certain point, the performance begins to decline. This phenomenon can be attributed to several factors.

Initially, increasing token numbers enhances the model's capacity to encode fine-grained personalized attributes and spatial features. With more tokens, the model can capture richer semantic information about the concept. However, beyond an optimal point, the added tokens introduce optimization challenges: the increased parameter space becomes harder to optimize with limited training data (only a few reference images per concept), leading to overfitting and degraded generalization.

Hyperparameter	Value
Optimizer	AdamW
Learning rate	1×10^{-4} with warmup
Training steps	2,000 step per concept
Batch size	8
Weight decay	0
Gradient clipping	Max norm 1.0
Mixed precision	bfloat16

Table 5. Training hyperparameters.

Information Redundancy. Excessive tokens may encode redundant information, where multiple tokens learn similar features. This redundancy not only wastes model capacity but also dilutes the discriminative power of individual tokens, making it harder for the model to selectively attend to relevant concept features during generation and editing.

Based on our ablation results, we select 32 tokens (16 for understanding expert and 16 for generation expert) as the optimal configuration, balancing representation capacity and training efficiency.

E. Additional Implement Detail

E.1. Architecture Clarification

We provide a detailed clarification on the architectural implementation of our Understanding Expert and Generation Expert Transformers, specifically addressing the parameter organization and routing mechanisms.

Complete Parameter Decoupling. The Understanding Expert and Generation Expert maintain **fully independent parameter sets**, not LoRA adapters or routing within a shared backbone. As defined in Eq. (1), each expert branch $\mathcal{F}_{\text{und}}(\cdot)$ and $\mathcal{F}_{\text{gen}}(\cdot)$ comprises a complete Transformer with independent parameters:

- **Understanding Expert:** $W^{(\text{und})}_{q,k,v}$, $\text{MLP}^{(\text{und})}$, $\text{LN}^{(\text{und})}_{\text{pre/post}}$
- **Generation Expert:** $W^{(\text{gen})}_{q,k,v}$, $\text{MLP}^{(\text{gen})}$, $\text{LN}^{(\text{gen})}_{\text{pre/post}}$

Crucially, **only the newly injected concept token embeddings** $\mathbf{P}^{(\text{und})}$ and $\mathbf{P}^{(\text{gen})}$ are trainable during personalization, while all expert-specific Transformer parameters ($W^{(\text{und})}$, $W^{(\text{gen})}$, MLPs, LayerNorms) are **frozen**. This parameter-efficient design enables few-shot concept learning without modifying the pretrained backbone. The forward pass through each expert follows:

$$\mathbf{H}^{(\text{und})} = \mathcal{F}_{\text{und}}\big(\mathbf{P}^{(\text{und})}, \mathbf{X}^{(\text{und})}\big), \quad \mathbf{H}^{(\text{gen})} = \mathcal{F}_{\text{gen}}\big(\mathbf{P}^{(\text{gen})}, \mathbf{X}^{(\text{gen})}\big), \tag{10}$$

where $\mathbf{P}^{(\text{und})} \in \mathbb{R}^{(N_u+1)\times d}$ and $\mathbf{P}^{(\text{gen})} \in \mathbb{R}^{N_g \times d}$ are the learnable concept token embeddings (initialized to zero), and $\mathbf{X}^{(\text{und})}$, $\mathbf{X}^{(\text{gen})}$ are the corresponding input embeddings routed to each expert.

Task	Hyperparameter	Value
Understanding	Max generation length	1000 tokens
	Sampling temperature	0.3
	Do sample	False
	Think mode	Disabled
Generation	Image size	512×512
	CFG text scale	4.0
	CFG image scale	1.0
	CFG interval	[0.4, 1.0]
	Timestep shift	3.0
	Diffusion steps	50
	CFG renorm type	Global
	CFG renorm min	0.0
	Think mode	Disabled
Editing	Image size	512×512
	CFG text scale	4.0
	CFG image scale	2.0
	CFG interval	[0.0, 1.0]
	Timestep shift	3.0
	Diffusion steps	50
	CFG renorm type	Text-channel
	CFG renorm min	0.0
	Think mode	Disabled

Table 6. Inference hyperparameters for different evaluation tasks.

Token Injection and Routing. Each concept is represented by 32 learnable embeddings structured as:

"<sks> is <und_1> $\cdots$ <und_N_u> <gen_1> $\cdots$ <gen_N_g>."

where <sks> identifies the concept, {<und_i>} are routed to the Understanding Expert, and {<gen_j>} are routed to the Generation Expert. During forward propagation, each token is processed exclusively by its designated expert's parameters:

$$\mathbf{Q}, \mathbf{K}, \mathbf{V}[i] = W^{(\text{und})}_{q,k,v,o}\mathbf{X}[i], \quad \forall i \in \mathcal{I}_{\text{und}} \tag{11}$$

$$\mathbf{Q}, \mathbf{K}, \mathbf{V}[j] = W^{(\text{gen})}_{q,k,v,o}\mathbf{X}[j], \quad \forall j \in \mathcal{I}_{\text{gen}} \tag{12}$$

This static, index-based routing ensures deterministic expert assignment based on token type.

Clarification on "Shared Attention". The term "shared attention" in Fig. 3 refers to **computational flow sharing**, not parameter sharing. While $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ are computed using expert-specific projections, all tokens participate in a single attention operation:

$$\mathbf{O} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{13}$$

This allows cross-expert information flow while maintaining parameter-level decoupling.

E.2. Evaluation Metrics

For personalized understanding, generation, and Attribute-Reasoning Generation, we follow the evaluation metrics provided by UnifyBench. For the personalized editing task, we modify the GEDIT evaluation template to conduct personalized assessment.

Personalized Understanding Metrics. Recognition (Rec.): We evaluate the model's ability to identify whether a given image contains the personalized concept. For each concept, we construct a balanced set of positive (containing the concept) and negative (not containing the concept) samples. Recognition accuracy is measured as:

$$\text{Rec.} = \frac{1}{2}\left(\text{Recall}_{\text{pos}} + \text{Recall}_{\text{neg}}\right) \tag{14}$$

where $\text{Recall}_{\text{pos}}$ is the recall on positive samples and $\text{Recall}_{\text{neg}}$ is the recall on negative samples. This balanced metric ensures the model is not biased toward either predicting presence or absence.

Visual Question Answering (VQA): We assess the model's ability to answer visual questions about personalized concepts. We report two complementary metrics:

- **VQA-BLEU:** BLEU score measuring n-gram overlap between predicted and reference answers, capturing lexical accuracy.
- **VQA-GPT:** GPT-4o-based evaluation scoring responses from 0 to 1 based on semantic alignment with key points in reference answers, capturing meaning beyond exact phrasing.

Text-Only Question Answering (QA): We evaluate the model's ability to answer text-only questions about learned concept attributes without visual input (e.g., "Can you describe <sks>?"). This tests whether personalized knowledge is internalized in token representations. Metrics follow VQA: QA-BLEU and QA-GPT.

Personalized Generation Metrics.

- **CLIP-I (Image Similarity):** We measure identity preservation by computing cosine similarity between CLIP image embeddings of generated images and reference images:

$$\text{CLIP-I} = \frac{1}{N_{\text{gen}} N_{\text{ref}}} \sum_{i,j} \cos\left(\text{CLIP}_{\text{img}}(\mathbf{I}^i_{\text{gen}}), \text{CLIP}_{\text{img}}(\mathbf{I}^j_{\text{ref}})\right) \tag{15}$$

  Higher CLIP-I indicates better preservation of visual identity across generations.

- **CLIP-T (Text-Image Alignment):** We measure prompt adherence by computing cosine similarity between CLIP text embeddings of generation prompts and CLIP image embeddings of generated images:

$$\text{CLIP-T} = \frac{1}{N_{\text{gen}}} \sum_{i} \cos\left(\text{CLIP}_{\text{txt}}(\mathbf{T}_i), \text{CLIP}_{\text{img}}(\mathbf{I}^i_{\text{gen}})\right) \tag{16}$$

  This evaluates whether generated images faithfully reflect textual instructions.

- **DINO (Perceptual Similarity):** We employ DINO-v2 features to measure semantic visual similarity between generated and reference images, providing a complementary signal to CLIP-I that captures object-level semantics:

$$\text{DINO} = \frac{1}{N_{\text{gen}} N_{\text{ref}}} \sum_{i,j} \cos\left(\text{DINO}(\mathbf{I}^i_{\text{gen}}), \text{DINO}(\mathbf{I}^j_{\text{ref}})\right) \tag{17}$$

- **Face Similarity (Face-Simi):** For human subject concepts (10 out of 20 concepts in OmniPBench), we measure facial identity preservation using the pretrained ArcFace model:

$$\text{Face-Simi} = \frac{1}{N_{\text{gen}} N_{\text{ref}}} \sum_{i,j} \cos\left(\text{ArcFace}(\mathbf{I}^i_{\text{gen}}), \text{ArcFace}(\mathbf{I}^j_{\text{ref}})\right) \tag{18}$$

  This metric is only computed for person concepts and provides a specialized measure of identity fidelity critical for personalized human generation.

Personalized Attribute-Reasoning Generation (PARG) Metrics. PARG evaluates the model's ability to leverage *textual* concept attributes during generation. For example, given "Generate <sks> in his home" without specifying home features, the model must recall learned attributes (e.g., "<sks>'s home is by the sea"). We use:

- **PARG-Score:** GPT-4o-based holistic evaluation (0-1 scale) assessing whether generated images correctly incorporate learned textual attributes.
- **PARG-CLIP-I:** CLIP-I computed between PARG generations and reference images, measuring identity preservation under attribute-conditioned generation.

Personalized Image Editing Metrics.

- **Semantic Consistency (SEMA-C):** We employ GPT-4o as a judge to evaluate whether edited images faithfully follow editing instructions while preserving concept identity. The model is prompted to rate semantic alignment on a 0-1 scale based on instruction adherence.
- **Quality of Image (QUAL-I):** We assess visual naturalness quality of edited images using GPT-4o-based evaluation (0-1 scale).

Average Editing Score (Avg): We report the arithmetic mean of SEMA-C and QUAL-I as the overall editing performance:

$$\text{Avg} = \frac{1}{2}\left(\text{SEMA-C} + \text{QUAL-I}\right) \tag{19}$$

F. Additional Qualitative Results

We provide more cases in Fig. 11.

Explicit knowledge replay template

You should recall the visual attributes of <sks> and the context, then rewrite the prompt with complete details.
The thinking process must be enclosed within <think> </think> tags.
Inside <think>, ask a question about what needs to be recalled, then answer it in concise format. After </think>, rewrite the prompt naturally with complete details.

Examples:
Input:
Generate a photo of <sks>'s toy.
Output:
<think>
what is <sks>'s toy? <sks>'s toy is a black excavator.
</think>
Generate a photo of <sks> with a black excavator.
Input:
Generate a photo of <sks> outdoor.
Output:
<think>
what is <sks> wearing outdoor? <sks> wears a blue shirt outdoor.
</think>
Generate a photo of <sks> wearing a blue shirt outdoor.

Input:

Edit evaluate template

PROMPTS_SEMA-C:
From scale 0 to 10:
A score from 0 to 10 will be given based on the success of the editing. (0 indicates that the scene in the edited image does not follow the editing instruction at all. 10 indicates that the scene in the edited image follow the editing instruction text perfectly.)
<sks> is <text prompt>.
Editing instruction: <instruction>

PROMPTS_QUAL-I:
The image is an AI-generated image.
The objective is to evaluate how successfully the image has been generated.

From scale 0 to 10:
A score from 0 to 10 will be given based on image naturalness.
(
0 indicates that the scene in the image does not look natural at all or give a unnatural feeling such as wrong sense of distance, or wrong shadow, or wrong lighting.
10 indicates that the image looks natural.
)

Figure 10. Prompt template for explicit knowledge replay and editing evaluation.

G. Limitations

Due to limited personalized training data, we trained with only a small amount of edit data. As a result, editing in complex backgrounds remains challenging: the model may unintentionally alter backgrounds or fail to accurately localize the main subject. Although we used ablation studies to analyze attention maps with and without edit-data training, the improvements brought by edit data still require more comprehensive evaluation. In addition, while enhancing the similarity of the main subject, we observed a decline in instruction-following ability. This trade-off between subject preservation and instruction adherence in personalized settings requires further investigation.

Figure 11. Additional qualitative results for personalized generation and editing.