IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 3 November 2025, accepted 22 November 2025, date of publication 28 November 2025, date of current version 10 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3638839

RESEARCH ARTICLE

Order Matters: Permutation-Based Prompt Optimization for Personalized Image Generation

WOOSEOK SONG AND CHANG WOOK AHN, (Member, IEEE)
AI Graduate School, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea

Corresponding author: Chang Wook Ahn (cwan@gist.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by Korean Government, Ministry of Science and ICT (MSIT), under Grant RS-2024-00347902; and in part by the Future-Leading Specialized Research Project grant funded by Gwangju Institute of Science and Technology (GIST), in 2025.

ABSTRACT Since AI-assisted content generation has received due attention of late, prompt reordering has emerged as a promising method for prompt tuning. This paper presents a framework named Interactive Prompt Permutation Optimization (IPPO) for text-to-image generation, which optimizes the order of prompt words according to user preferences. IPPO integrates user feedback into the optimization process in order to enhance personalization as well as reduce costs. We design a surrogate model in concert with an ordering matrix, which efficiently estimates user preferences from sparse feedback from the user. Furthermore, we develop a Permutation Genetic Algorithm (PermGA) equipped with a novel repairing operator derived from the surrogate model's dominance information in order to further enhance the efficiency of optimization. A set of experiments on the Linear Ordering Problem benchmark demonstrated the fundamental effectiveness of our approach. We also conducted experiments on image generation in order to verify the framework's validity in reflecting user preferences, achieving competitive results in terms of image quality and relevance compared to SOTA methods.

INDEX TERMS Genetic algorithm, interactive evolutionary computation, personalized image generation, prompt engineering, surrogate model.

I. INTRODUCTION

With the rapid advancement of diffusion-based generative technologies [1], the importance of prompt optimization for personalized content creation has become increasingly evident. Prompt optimization is a crucial field of research due to the high sensitivity of text-based generative models to input prompts, which significantly affects the resulting output. This sensitivity is even more pronounced in image generation, where the black-box nature of diffusion models poses a substantial challenge for predicting outcomes based solely on prompts.

The sensitivity arises not only from the keywords of the prompts but also significantly from the order in which these keywords are arranged. Text-to-Image (T2I) models are influenced by the position and order of words during the text embedding process [2], [3], which can lead to subtle yet meaningful differences in the emphasis, composition, and style of the final image. For instance, permutations of the same keyword set {beautiful, woman, red, dress} might generate images focusing differently on the person, the attire, or the overall mood, depending on the word order. Addressing the challenge of optimizing prompt order is therefore a promising research avenue for fine-grained control over image generation.

Previous studies [4], [5] have attempted to automate prompt optimization by leveraging large prompt-image datasets and training models using CLIP embeddings. However, these approaches face significant limitations in personalized content creation. The high computational costs associated with extensive datasets and large-scale training make them expensive. Additionally, these models are often

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos.

trained to maximize objective metrics like Aesthetic Scores or CLIP similarity, which may not align with a user's subjective aesthetic preferences.

More recently, approaches using Large Language Models (LLMs) to refine prompts have been proposed [6], [7]. While powerful, these methods incur API call costs and often alter the prompt's core content by adding or replacing words (e.g., adding "masterpiece, 4k"). While LLMs excel at content generation and rewriting, they may not be ideal when the goal is subtle nuance control over a fixed set of concepts, and they introduce external computational dependencies and costs. In contrast, prompt permutation offers a cost-effective mechanism for fine-tuning the nuance and emphasis of a given set of keywords, which is often crucial for aligning generation with subjective user aesthetics without altering the core semantic content.

This context reveals three key research gaps hindering effective personalized image generation through prompt engineering:

1) **High Cost and Dependency:** State-of-the-art methods rely on costly large-scale training or external LLM API calls.
2) **Lack of Subjective Personalization:** Optimization based on objective scores fails to adequately capture individual user preferences.
3) **Absence of Automated Order Optimization:** Reordering keywords offers a powerful yet underexplored avenue for fine-tuning, but lacks systematic and automated exploration strategies beyond manual trial-and-error.

To overcome these limitations and address the identified gaps, we propose a novel approach using prompt permutation, focusing on optimizing keyword order to align with user preferences. In this paper, we introduce the **I**nteractive **P**rompt **P**ermutation **O**ptimization (IPPO) framework. Central to IPPO is a novel surrogate model based on an ordering matrix, designed to rapidly learn pairwise keyword preferences from minimal user feedback, thereby enabling efficient heuristic search within the vast permutation space. IPPO leverages direct user interactions on generated images to gather preference scores. These scores dynamically update the surrogate model, which guides a specialized heuristic algorithm, Permutation Genetic Algorithm (PermGA), towards optimal permutations. The IPPO framework operates self-contained without requiring additional large datasets or external model calls, relying solely on a limited number of user interactions.

The main contributions of this research are summarized as follows:

1) **Formulating** prompt reordering as an interactively learned Linear Ordering Problem (LOP).
2) **Proposing** the IPPO framework, integrating user feedback with efficient surrogate-based optimization for personalized T2I.
3) **Developing** an ordering matrix-based surrogate model for effective preference estimation from sparse subjective data.
4) **Designing** a specialized PermGA featuring a novel surrogate-guided repairing operator to enhance search efficiency.
5) **Validating** IPPO's effectiveness and competitive performance through objective (LOP) and criteria-based (T2I) experiments.

II. BACKGROUND AND RELATED WORKS

A. PERSONALIZED IMAGE GENERATION AND PROMPT OPTIMIZATION

Personalized image generation aims to customize content to meet individual user preferences, a technique increasingly used in creative industries [8], [9]. Diffusion models have emerged as the state-of-the-art method for this task [9], [10], [11], [12]. These models generate images by iteratively refining random noise through a denoising process [13]. While this process yields high-fidelity images, the black-box nature of diffusion models makes precise control over outputs challenging without effective prompt engineering [14]. Prompt optimization is a key methodology within prompt engineering that seeks to find the best possible prompt to maximize a desired objective.

Existing approaches to automate prompt optimization can be broadly categorized.

1) RL/GRADIENT-BASED PROMPT OPTIMIZATION

Some methods leverage large prompt-image datasets for training. For instance, Hao's Promptist [4] employs reinforcement learning, fine-tuning a model on collected data and using aesthetic rewards. Wen's PEZ [5] utilizes gradient-based discrete optimization to directly enhance text-image similarity in the CLIP embedding space.

2) LLM-BASED PROMPT REFINEMENT

Recent work has explored using powerful LLMs to refine user prompts. These methods typically take a simple user input and expand or rewrite it into a more detailed prompt designed to produce higher-quality images [6], [7].

However, these state-of-the-art approaches present significant limitations, particularly for cost-effective personalization. Both RL/Gradient-based methods, requiring extensive training on large datasets, and LLM-based methods, incurring API call costs, involve substantial computational expense. Furthermore, methods optimizing objective scores like CLIP similarity (e.g., PEZ) or general aesthetics (e.g., Promptist) often fail to capture the nuanced, subjective preferences of individual users. This highlights a critical research gap: the need for low-cost prompt optimization techniques capable of effectively incorporating subjective user feedback for true personalization.

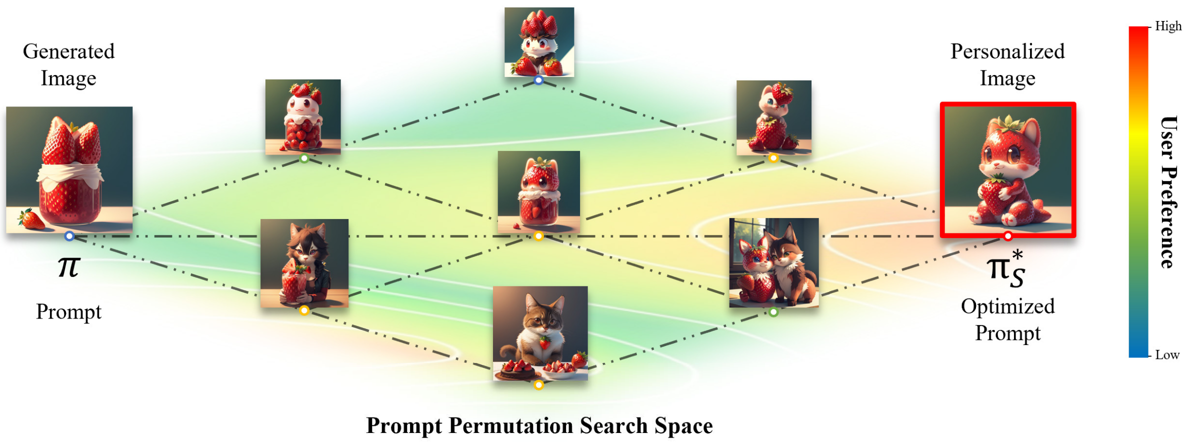

FIGURE 1. Visualization of the prompt permutation search space (Π_n) for Personalized Image Generation. Each point π represents a specific keyword permutation that generates a unique image. The underlying heatmap conceptually illustrates the user preference landscape $Score(\pi)$ approximated by our surrogate model (S). IPPO aims to efficiently navigate this vast, high-dimensional space to identify the optimal permutation (π_S^*) that maximizes user satisfaction, bypassing the need for exhaustive evaluation.

B. PROMPT PERMUTATION AS A COST-EFFECTIVE ALTERNATIVE

An alternative, cost-effective strategy for influencing T2I output is prompt permutation (or reordering). T2I models exhibit sensitivity not just to the keywords present but also to their order, likely due to mechanisms like positional encoding influencing attention during the embedding process [3]. Rearranging the words within a fixed set can thus significantly alter the generated image's emphasis or style without changing the core semantic content. This approach requires no additional data or external model calls, making it highly practical and applicable across various prompt-based models [15]. Tools like Midjourney's permutation feature allow users to explore this effect manually.

Despite its potential and cost-effectiveness, prompt permutation currently suffers from a major drawback: the reliance on manual exploration. Given that n keywords yield $n!$ possible permutations, the search space grows factorially, rendering manual trial-and-error inefficient and unlikely to discover the optimal ordering for a user's specific preference. This limitation points to a clear research gap: the need for methods that can automate and intelligently guide the search within the permutation space, leveraging user feedback effectively.

C. INTERACTIVE GENETIC ALGORITHM (IGA) FOR SUBJECTIVE OPTIMIZATION

To address the challenge of subjective optimization, we turn to surrogate model-based optimization, which uses computationally efficient models to approximate complex evaluation functions [17], [18]. Given the discrete nature of the permutation problem, we employ a Genetic Algorithm (GA), a subset of Evolutionary Algorithms (EAs) known for effectively handling discrete search spaces, unlike methods such as Particle Swarm Optimization (PSO) or Differential Evolution (DE) which are primarily designed for continuous optimization.

Interactive GA (IGA) [21] is a specific variant of GA designed to refine solutions based on subjective user feedback. IGAs iteratively update a surrogate model using user preference scores, allowing the optimization to evolve towards solutions that better match user preferences. While other surrogate approaches like Gaussian Processes or simple regression models exist, the ordering matrix is particularly well-suited for permutation problems as it directly models pairwise relationships and can learn effectively from the sparse and potentially noisy feedback inherent in interactive systems, unlike data-hungry reinforcement learning approaches. This makes IGA a powerful tool for complex, user-driven optimization problems [24].

This paper connects the research gaps identified in the preceding sections. Our work addresses the following central research question:

> Can we automate the cost-effective search for prompt permutations by using an IGA framework with an efficient surrogate model to learn from a user's subjective feedback?

To our knowledge, the application of IGA specifically to the prompt permutation problem in T2I models using an ordering matrix surrogate is novel. We bridge this gap by proposing IPPO, which combines an IGA framework utilizing a GA with an efficient LOP-inspired surrogate model (S) to solve the prompt permutation optimization problem formulated in Section III.

III. PROBLEM FORMULATION

In T2I models, the order of words in a prompt can significantly influence the resulting image's nuance and emphasis. This work aims to find the optimal word permutation that best satisfies a user's subjective preference for a given set of keywords.

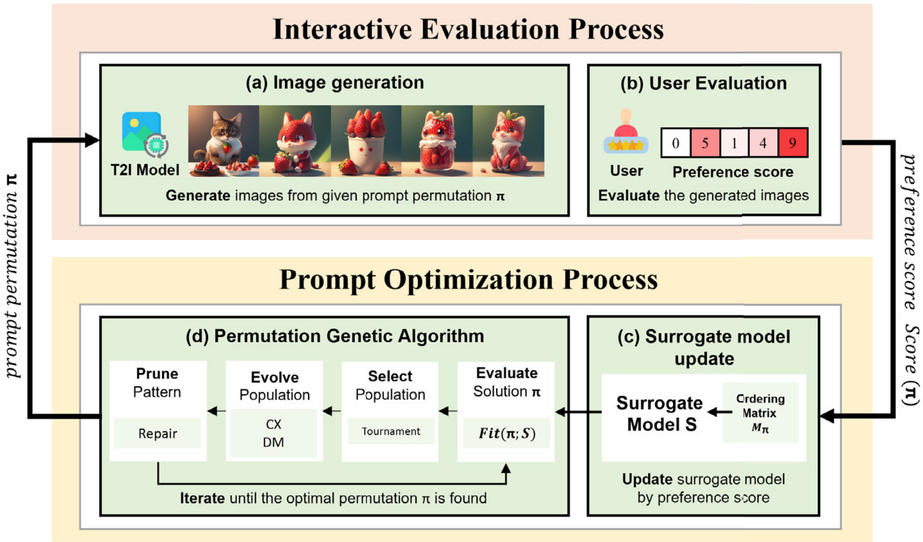

FIGURE 2. **The workflow of the Interactive Prompt Permutation Optimization (IPPO) framework. The Interactive Evaluation Process is the front end that interacts with the user, while the Prompt Optimization Process is the back end that conducts the actual optimization. The framework iteratively repeats the steps of (a), (b), (c), (d) until the user is satisfied with the generated image.**

We start with a fixed set of n unique keywords, $W = \{w_1, w_2, \cdots, w_n\}$. A solution $\pi = (p_1, p_2, \cdots, p_n)$ is a permutation of the set W, where $p_i \in W$ denotes the keyword at the i-th position in the sequence π. The set of all possible permutations, Π_n, constitutes the search space, with a size of $|\Pi_n| = n!$. A user provides a subjective preference score $Score(\pi)$ for an image generated from a given permutation π. The function *Score* is a black-box, as its internal logic is unknown and dependent on both the T2I model and the user.

The objective of this work is to find the optimal permutation π^* that maximizes the user's preference score within the entire set of permutations Π_n. This can be formally defined as:

$$\pi^* = \arg\max_{\pi \in \Pi_n} Score(\pi)$$

Solving this optimization problem faces two major challenges:

- **Combinatorial Explosion:** The search space size, $n!$, grows factorially with the number of keywords n, making an exhaustive search infeasible even for moderate n.
- **Costly and Noisy Evaluation:** Obtaining a value for $Score(\pi)$ is expensive, requiring both computational time for image generation by the T2I model and human time for subjective evaluation. Furthermore, this evaluation is inherently noisy and can be inconsistent due to its subjective nature.

Therefore, the core challenge is to efficiently approximate π^* while minimizing the number of calls to the expensive and noisy evaluation function $Score(\pi)$.

Our approach tackles this challenge by reframing the intractable optimization task as a tractable learning task. We propose to learn a surrogate model, S, that approximates the user's underlying preference structure from the holistic $Score(\pi)$ scores. Figure 1 provides a conceptual visualization of this approach, showing how the surrogate model S approximates the underlying, costly-to-evaluate preference landscape. S is defined as an $n \times n$ matrix designed to capture the underlying pairwise preferences between keywords.

The goal of our approach is twofold.

1) To learn the $O(n^2)$ parameters of S using only the limited and noisy, holistic $Score(\pi)$ scores.
2) To search for the optimal permutation on the learned, low-cost surrogate model S.

This reframing effectively transforms the original $n!$ search problem into a more manageable $O(n^2)$ parameter estimation problem. Section IV details our IPPO framework designed to learn S and subsequently find the optimal permutation.

IV. IPPO FRAMEWORK

This section introduces the **I**nteractive **P**rompt **P**ermutation **O**ptimization (IPPO) framework, a human-in-the-loop system designed to solve the intractable search problem formulated in Section III. As illustrated in Figure 2, IPPO is composed of two primary modules:

- The **Interactive Evaluation** module manages the human-in-the-loop component, querying the user to obtain the expensive and noisy $Score(\pi)$ feedback.
- The **Prompt Optimization** module is the core algorithmic engine. It receives the user's feedback, learns the surrogate preference model S, and then uses a Genetic Algorithm (GA) to search for the next optimal permutation π based on S.

The following sections detail the mechanisms of each component.

The overall workflow of IPPO, as outlined in Algorithm 1, involves the following iterative steps:

1) **Initialization:** The framework begins by initializing the surrogate model S to a zero matrix (line 1) and creating an initial random population (*Population*) of N permutations (line 2). A subset ($K = 20$) of these permutations is selected as the initial set of samples (*#samples*) to present to the user (line 3).
2) **Interactive Evaluation (Image Generation & User Evaluation)**: The selected samples (*#samples*) are used to generate images via the T2I model. These images are presented to the user, who provides subjective preference scores (UserScores) for each (line 5). This step terminates early if the user assigns a maximum score (+10), indicating satisfaction.
3) **Surrogate Model Update:** The collected user scores are used to update the surrogate model S. For each evaluated pair $(\pi, Score(\pi))$, a corresponding pairwise preference matrix s_π is created (line 7) and accumulated into S (line 8).
4) **Permutation Optimization (PermGA):** The specialized genetic algorithm, PermGA (detailed in Algorithm 2), takes the current population and the updated surrogate model S as input and evolves the population to find permutations with higher estimated fitness (*Fit*) according to S (line 10).
5) **Sample Selection for Next Iteration:** A new set of top K samples (*#samples*) is selected from the evolved population based on their fitness values according to S (line 11). These samples will be presented to the user in the next iteration.
6) **Termination and Output:** The main loop (steps 2-5) repeats until the user indicates satisfaction (e.g., by giving a +10 score). Once terminated, the best solution (π_S^*) found in the final population according to the final surrogate model S is identified (line 13) and returned as the optimized prompt permutation (line 14).

The key idea is that as more user feedback ($Score(\pi)$) is incorporated into S through repeated iterations, S becomes a more accurate model of the user's underlying preferences. Consequently, the solution π_S^* found by optimizing S (using *Fit*) will gradually approach the true optimal solution π^*. IPPO aims to find this approximation $\pi_S^* \approx \pi^*$ with minimal calls to $Score(\pi)$.

Algorithm 1 The IPPO Framework Algorithm

Input: $W = \{w_1, w_2, \cdots, w_n\}$ (A set of n keywords)
Output: Optimized permutation π_S^*

```
1:  S ← [0]_{n×n}
2:  Population ← RandomPermutations(N)
3:  #samples ← SelectTopK(Population, K)
4:  while NOT UserSatisfaction do
5:      UserScores ← GetUserScores(#samples)
6:      for all (π, Score(π)) ∈ UserScores do
7:          s_π ← CreateMatrix(π, Score(π))
8:          S += s_π
9:      end for
10:     Population ← PermGA(Population, S)
11:     #samples ← SelectTopK(Population, K)
12: end while
13: π*_S ← GetBestSolution(Population, S)
14: return π*_S
```

A. INTERACTIVE EVALUATION PROCESS

This process (lines 5-9 of Alg. 1) collects preference scores $Score(\pi)$ from the user.

1) IMAGE GENERATION & USER EVALUATION

In each iteration, up to K candidate images (K = 20) generated from selected permutations are presented to the user via a GUI. The user assigns an integer score $Score(\pi) \in [-10, +10]$ to each image based on their subjective preference. Crucially, the user does not necessarily need to evaluate all K images. If the user finds a satisfactory image among the candidates and assigns the maximum score (+10), the evaluation process for the current iteration terminates immediately, and the framework proceeds to the next step. This design aims to minimize user fatigue by allowing early termination while still presenting a reasonable number of options (up to 20) for exploration. All collected $(\pi, Score(\pi))$ pairs from each iteration are recorded.

B. PROMPT OPTIMIZATION PROCESS

This process (lines 6-11 of Alg. 1) uses the collected user feedback to update the surrogate model S and then employs a specialized Genetic Algorithm (PermGA) to find the permutation that maximizes the fitness $Fit(\pi; S)$ according to S.

1) SURROGATE MODEL UPDATE

In this step, the surrogate model S is updated using the scores recorded in the database DB, which contains all $(\pi, Score(\pi))$ pairs evaluated so far. For each evaluated pair, a mathematically designed ordering matrix s_π is constructed. This matrix reflects the pairwise relationships implied by the $Score(\pi)$ for that specific order. The surrogate model S is the accumulation of these individual matrices, as shown in Equation (1):

$$S = \sum_{\pi \in DB} s_\pi$$

$$s_\pi = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ \vdots \\ p_n \end{array} \begin{array}{c} \begin{array}{cccccc} p_1 & p_2 & p_3 & \cdots & p_n \end{array} \\ \begin{pmatrix} 0 & \text{Score}(\pi) & \text{Score}(\pi) & \dots & \text{Score}(\pi) \\ 0 & 0 & \text{Score}(\pi) & \dots & \text{Score}(\pi) \\ 0 & 0 & 0 & \dots & \text{Score}(\pi) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix} \end{array} \quad (1)$$

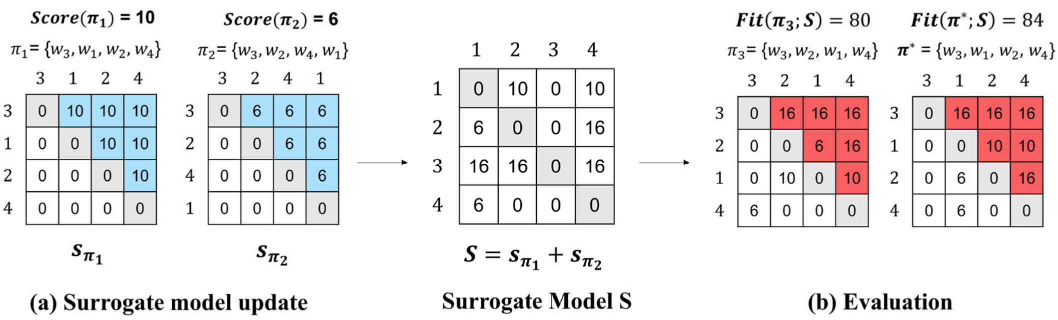

FIGURE 3. The Surrogate Model S: Update and Evaluation Mechanisms. (a) Surrogate model update: The surrogate model S is cumulatively updated. An ordering matrix s_π (blue entries) is constructed from a given π and its $Score(\pi)$, and then accumulated into S. (b) Evaluation: To get $Fit(\pi; S)$, S is reordered by π, and its upper triangular entries (red) are summed (see Eq. 2).

The matrix s_π is an $n \times n$ upper triangular matrix where for a permutation $\pi = (p_1, \ldots, p_n)$, the entry (p_i, p_j) for all $i < j$ is set to $Score(\pi)$. This process essentially distributes the overall score $Score(\pi)$ across all pairwise precedence relationships present in π. Figure 3 (a) illustrates this update.

2) PERMUTATION OPTIMIZATION (PermGA)

Once the surrogate model S is updated, a specialized Genetic Algorithm named PermGA is used to efficiently search for the permutation π_S^* that maximizes the fitness function $Fit(\pi; S)$. PermGA is designed to handle the discrete permutation domain effectively and can rapidly optimize based on real-time user preference updates without requiring retraining. The overall process of PermGA is outlined in Algorithm 2. It iterates through evaluation and evolution steps until convergence.

The key steps within each generation are detailed below.

a: EVALUATION

All solutions π in the population are evaluated by the surrogate model at every evolutionary step. Each $\pi = (p_1, \ldots, p_n)$ is assigned a fitness value $Fit(\pi; S)$ using the surrogate model S. To clarify this process, Figure 3 (b) shows a simple example of how S calculates the fitness for a given π. The fitness value is calculated by summing the relevant pairwise preference entries from S according to the permutation π, as shown in Equation (2).

$$Fit(\pi; S) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} S_{p_i, p_j} \tag{2}$$

b: SELECTION

Parents for the next generation are selected using tournament selection. In each tournament, two individuals are randomly chosen, and the one with the higher fitness value proceeds.

c: CROSSOVER

The cycle-crossover step involves selecting two arbitrary points in the parent individual, transferring the genes between them into the offspring, completing the remaining genes in sequence, and omitting those already included.

Algorithm 2 PermGA Process

Input: Current Population P_{in}, Surrogate Model S
Output: Evolved Population P_{out}

```
P_out ← ∅
gen ← 0, best_fitness ← −∞, stagnation_count ← 0
while gen < 100 AND stagnation_count < 20 do
    P_eval ← Evaluate(P_in, S)                ▷ Using Eq. 2
    current_best ← max_{π∈P_eval} Fit(π; S)
    if current_best > best_fitness then
        best_fitness ← current_best
        stagnation_count ← 0
    else
        stagnation_count ← stagnation_count + 1
    end if
    Parents ← Select(P_eval, N)
    Offspring ← Crossover(Parents)
    Offspring ← Mutate(Offspring)
    Offspring ← Repair(Offspring, S)
    P_in ← Offspring
    gen ← gen + 1
end while
P_out ← P_in
return P_out
```

d: MUTATION

The displacement-mutation step involves selecting a gene segment at random, removing it from its initial position, and reintegrating it at another point in the sequence.

e: REPAIRING

As the final stage in the evolving step, the repair step traverses individuals in the population to check for already found unnecessary patterns. It is designed to leverage the learned preferences in S and prune the search space.

- **Dominated Pattern Definition:** For any two keywords $w_i, w_j \in W$, if $S_{w_i,w_j} > S_{w_j,w_i}$, the pairwise order (w_i, w_j) dominates (w_j, w_i).
- **Repairing Operation:** Each offspring π' generated after crossover and mutation is scanned. If it contains any pairwise order (w_j, w_i) that is dominated by (w_i, w_j) according to S, the operator swaps their positions in π'. This forces solutions towards orderings consistent with learned preferences, effectively repairing dominated patterns into locally optimal ones based on the surrogate model. This repairing process significantly accelerates convergence by focusing the search on promising regions of the permutation space.

f: STOPPING CRITERIA AND OUTPUT

The internal PermGA loop terminates when the convergence criteria are satisfied, defined as a maximum of 100 generations or 20 stagnant generations. The resulting evolved population (P_{out}) is then passed back to the main IPPO loop (Algorithm 1). From this final population, the top K (K = 20) solutions based on *Fit* are selected and presented to the user in the next interactive evaluation round.

V. EXPERIMENTS AND RESULTS

This section presents a comprehensive two-part experimental evaluation designed to validate the proposed IPPO framework. The first part, detailed in Section V-A, assesses the fundamental efficacy of the core algorithmic components. Specifically, we evaluate the surrogate model and the PermGA enhanced with the repairing operator using the objective LOP benchmark. The second part, described in Section V-B, evaluates the performance of the complete IPPO framework within its target application domain of personalized text-to-image (T2I) generation involving preferences. All experiments were conducted on a system equipped with an Intel Xeon Gold 6130 CPU and an NVIDIA RTX 4090 GPU.

A. EXPERIMENTS ON THE SURROGATE MODEL AND PermGA

Two experiments were conducted on the LOP benchmark to analyze the key components of our approach:

1) **Surrogate Model Fidelity:** This experiment verifies the accuracy with which our surrogate model (S) approximates the ground-truth LOP objective function when learned from sparse samples.
2) **Ablation Study:** This experiment quantifies the individual contributions of the surrogate model and the repairing operator to the overall optimization performance of PermGA.

1) EXPERIMENTAL SETTING

The performance evaluation utilized 49 instances sourced from the established LOLIB benchmark library for the LOP [25]. The number of items to be ordered (n) varied across

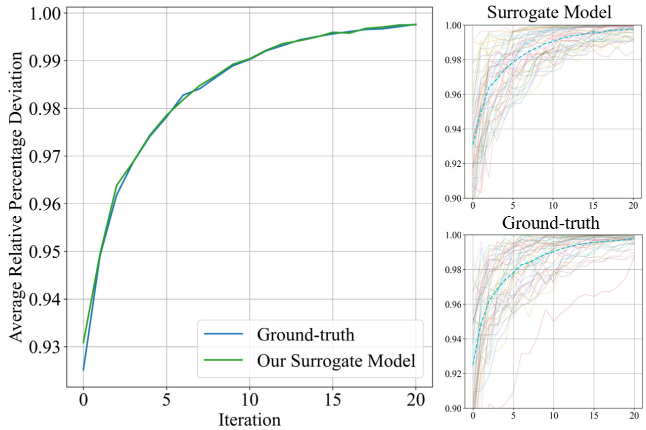

FIGURE 4. Surrogate Model Fidelity Validation on LOLIB (n=49). This figure compares the optimization performance (ARPD vs Iterations) using the ground-truth matrix (S_{true}) versus our learned surrogate model (S). The average trajectories (dashed lines) are nearly identical, demonstrating that our surrogate, learned from sparse samples, is a high-fidelity approximation that can effectively guide the optimization.

these instances, reflecting a range of problem sizes. Each LOLIB instance provides a ground-truth pairwise preference matrix S_{true}, enabling objective performance measurement. We employed the Average Relative Percentage Deviation (ARPD) as the primary performance metric:

$$ARPD = 1 - \frac{|f_{Best} - f_{true}(\pi_S^*; S_{true})|}{f_{Best}} \tag{3}$$

where $f_{true}(\pi_S^*; S_{true})$ represents the true objective score (computed using S_{true}) of the best solution π_S^* found by optimizing our surrogate S, and f_{Best} denotes the known optimal score for the instance. An ARPD value of 1 signifies that the optimal solution was recovered.

2) EXPERIMENTAL RESULTS

a: SURROGATE MODEL FIDELITY

Figure 4 illustrates the convergence behavior when optimizing using the learned surrogate model (S) compared to optimizing directly using the ground-truth matrix (S_{true}). The performance trajectories closely align. This indicates that S, despite being constructed from limited samples, effectively captures the underlying preference structure necessary to guide the optimization process. The result substantiates the viability of our surrogate-based approach for efficiently finding high-quality solutions in a large search space.

b: ABLATION STUDY RESULTS

We conduct an ablation study to quantify the contribution of our framework's two core components: the surrogate model S and the surrogate-guided repairing operator. The results are presented in Figure 5.

- **Effectiveness of the Surrogate Model**: The 'IPPO without surrogate' condition, which acts as a standard GA baseline, performs significantly worse than the full IPPO framework. This demonstrates the critical value of the surrogate-guided search strategy.

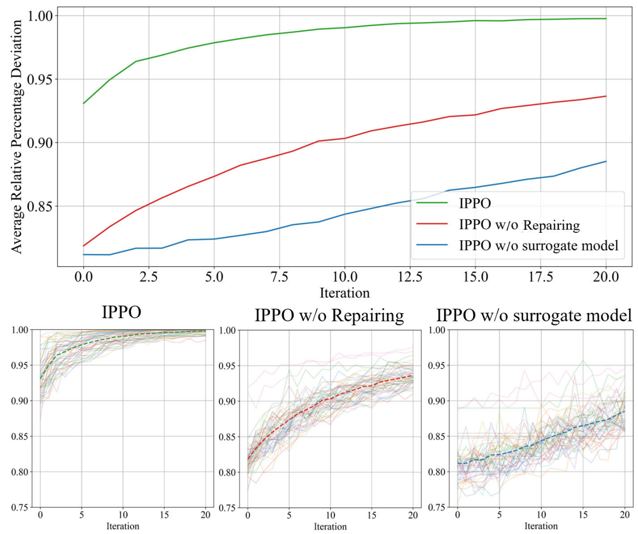

FIGURE 5. Ablation study on the components of IPPO. Average ARPD (y-axis) versus optimization iterations (x-axis). We compare three configurations: (1) IPPO, (2) IPPO without repairing operator, and (3) IPPO without surrogate (standard GA baseline). The faint lines show individual results from 49 instances, while the dashed lines represent the average. The full IPPO's 11% average performance gain over the baseline confirms the synergistic contribution of both components.

- **Effectiveness of the Repairing Operator**: Disabling the operator ('IPPO without repairing') also results in a marked performance decrease, confirming its crucial role in enhancing search efficiency.

The full IPPO framework achieves an 11% average ARPD improvement over the standard GA baseline. This highlights the synergistic benefit of combining the surrogate model with the repairing operator.

The results of the ablation study, presented in Figure 5, quantify the contributions of the core components. The 'IPPO without surrogate model' condition represents a standard GA baseline operating on the true objective function. The full IPPO framework significantly outperforms this baseline, demonstrating the effectiveness of the surrogate-guided search strategy. Furthermore, disabling the repairing operator ('IPPO without pruning') results in a marked decrease in performance compared to the full framework, confirming the operator's crucial role in enhancing search efficiency. The average ARPD improvement of approximately 11% achieved by the full IPPO framework over the standard GA baseline highlights the synergistic benefit derived from combining the ordering matrix-based surrogate model with the surrogate-guided repairing operator. This suggests an enhanced efficiency in reaching near-optimal solutions, implying a potential reduction in the number of evaluations needed compared to standard evolutionary approaches.

B. EXPERIMENTS ON PERSONALIZED IMAGE GENERATION

This section evaluates the practical applicability and effectiveness of the complete IPPO framework in the target task of personalized T2I generation based on pre-defined preferences.

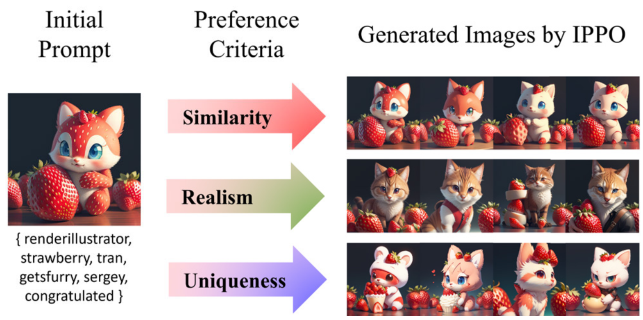

FIGURE 6. Preference reflection examples of images generated with varying preference criteria, including similarity, realism, and uniqueness. Each image is derived from the same initial prompt and then created using a prompt optimized for specific preference criteria.

1) EXPERIMENTAL SETTING

The personalization experiments relied on subjective evaluations provided by criteria-based scorers. These automated scorers were designed to assign integer scores within the range $[-10, +10]$. For the preference reflection experiment (corresponding to Fig. 6), distinct scorers targeted specific criteria: Similarity (rewarding minimal deviation from an initial image), Realism (rewarding photorealistic appearance), and Uniqueness (evaluating based on a distinctiveness metric). In the quantitative evaluation involving target image reproduction (Table 1), a scorer evaluated the perceived similarity to the target image. The interactive optimization loop (Algorithm 1) for each experiment was terminated upon a scorer assigning the maximum possible score (+10) to a generated image, signifying that the optimization target was met.

Image generation was performed using a pre-trained Stable Diffusion-v2 model [1]. All generation parameters (guidance scale = 7, inference steps = 20, sampler = DPM-Solver++, seed) were held constant throughout the experiments and adopted from the setup reported by PEZ study [5] to facilitate a fair comparison focused exclusively on the impact of prompt permutation. IPPO's performance was compared against PEZ [5] and Promptist [4] using prompts generated by these methods for specific target image reproduction tasks, with source material from Lexica.art and Civitai.

2) EXPERIMENTAL RESULTS

a: PREFERENCE REFLECTION

Figure 6 provides qualitative evidence of IPPO's effectiveness in aligning image generation with varying predefined criteria. By optimizing only the keyword order, the framework successfully produced visually distinct outputs corresponding to the targeted preferences (similarity, realism, uniqueness). This demonstrates the potential of prompt permutation as a mechanism for personalized fine-tuning.

TABLE 1. Quantitative comparison with SOTA methods on target image reproduction.

Algorithm	CLIP Score (Relevance)	Aesthetic Score (Quality)	Inception Score (Diversity)
Ours (IPPO)	84%	6.90	2.2
PEZ	83%	6.67	2.4
Promptist	80%	6.77	2.3

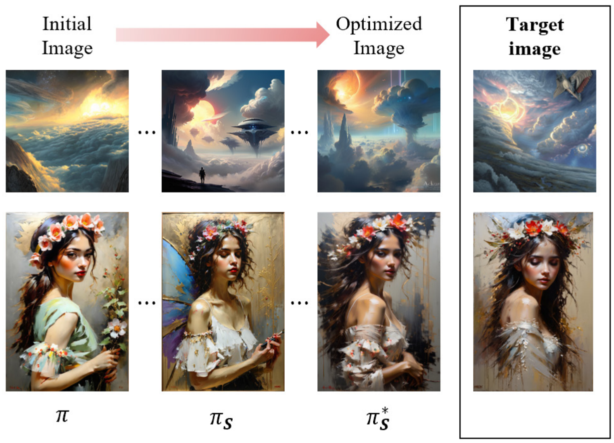

FIGURE 7. Example of the generating preference image process with the IPPO framework. The images on the left shows progressive optimization of image generation. The image on the right is the target image.

It should be noted, however, that these experiments utilized relatively simple, single-criterion preference targets.

b: QUANTITATIVE RESULTS

The quantitative comparison results are summarized in Table 1. IPPO achieved competitive performance in terms of CLIP Score and Aesthetic Score relative to the PEZ and Promptist methods. An example of the iterative image reproduction process is shown in Figure 7. The slightly lower Inception Score observed for IPPO is consistent with the framework's design philosophy. IPPO aims to converge towards a specific user preference within the permutation space defined by a fixed keyword set, and the repairing operator actively prunes the search towards learned optima. This represents an inherent trade-off between maximizing alignment with the target preference and maximizing the diversity of generated outputs.

VI. DISCUSSION

While IPPO demonstrates potential, its practical application warrants consideration of several limitations.

A. HUMAN FEEDBACK DEPENDENCY

The framework's performance inherently relies on the quality and consistency of user feedback. Subjectivity, potential noise, and user fatigue resulting from evaluating multiple images can impact the surrogate model's accuracy and the overall optimization process [21]. IPPO mitigates these challenges by requiring feedback on only a subset of generated images to reduce the user's cognitive load.

B. SCALABILITY

The $O(N \cdot n^2)$ GA complexity per generation restricts applicability to prompts with a very large number of keywords (n). However, IPPO involves a trade-off, substituting the costs of large-scale training or API calls with GA computation and user interaction time.

VII. CONCLUSION

This paper introduced IPPO, an interactive framework leveraging user feedback, an efficient ordering matrix-based surrogate model, and a specialized PermGA with a repairing operator to optimize prompt permutations for personalized T2I generation. The core advantage lies in efficiently estimating user preferences from sparse evaluations, enabling personalization without extensive data or external model dependencies.

Experiments confirmed the algorithmic components' effectiveness on LOP benchmarks and demonstrated competitive performance against SOTA methods in T2I relevance and quality. This work validates interactive prompt permutation as a cost-effective strategy for fine-tuning T2I models according to subjective nuances. Despite limitations concerning user interaction burden, model assumptions, and scope, IPPO offers a lightweight path towards user-centric generative AI. Its principles may extend to other generative domains. Future work includes exploring active learning to reduce user effort, adapting to multi-objective preferences, and investigating hybrid optimization incorporating word weighting.

REFERENCES

[1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10674–10685.

[2] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 8748–8763.

[3] J. Wang, K. C. K. Chan, and C. C. Loy, "Exploring CLIP for assessing the look and feel of images," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 2, pp. 2555–2563.

[4] Y. Hao, Z. Chi, L. Dong, and F. Wei, "Optimizing prompts for text-to-image generation," 2022, *arXiv:2212.09611*.

[5] Y. Wen, N. Jain, J. Kirchenbauer, M. Goldblum, J. Geiping, and T. Goldstein, "Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery," 2023, *arXiv:2302.03668*.

[6] L. Lian, B. Li, A. Yala, and T. Darrell, "LLM-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models," 2023, *arXiv:2305.13655*.

[7] Z. Chen, L. Zhang, F. Weng, L. Pan, and Z. Lan, "Tailored visions: Enhancing text-to-image generation with personalized prompt rewriting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 7727–7736.

[8] Y. Xu, W. Wang, Y. Zhang, B. Tang, P. Yan, F. Feng, and X. He, "Personalized image generation with large multimodal models," in *Proc. ACM Web Conf.*, Apr. 2025, pp. 264–274, doi: 10.1145/3696410.3714843.

[9] N. Kumari, B. Zhang, R. Zhang, E. Shechtman, and J.-Y. Zhu, "Multi-concept customization of text-to-image diffusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 1931–1941.

[10] J. Ma, J. Liang, C. Chen, and H. Lu, "Subject-diffusion: Open domain personalized text-to-image generation without test-time fine-tuning," in *Proc. ACM SIGGRAPH Conf. Papers*, Jul. 2024, pp. 1–12.
[11] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or, "An image is worth one word: Personalizing text-to-image generation using textual inversion," 2022, *arXiv:2208.01618*.
[12] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 22500–22510.
[13] J. Ho, A. N. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2024, pp. 6840–6851.
[14] S. Witteveen and M. Andrews, "Investigating prompt engineering in diffusion models," 2022, *arXiv:2211.15462*.
[15] C. Li, M. Zhang, Q. Mei, W. Kong, and M. Bendersky, "Automatic prompt rewriting for personalized text generation," 2023, *arXiv:2310.00152*.
[16] MultiMedia LLC. (2023). *Permutation Prompts*. [Online]. Available: https://docs.midjourney.com/docs/permutations
[17] Y. Mack, T. Goel, W. Shyy, and R. Haftka, "Surrogate model-based optimization framework: A case study in aerospace design," in *Evolutionary Computation in Dynamic and Uncertain Environments*. Cham, Switzerland: Springer, 2007, pp. 323–342.
[18] R. Tromble and J. Eisner, "Learning linear ordering problems for better translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, vol. 2, 2009, pp. 1007–1016.
[19] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw. (ICNN)*, Nov. 2002, pp. 1942–1948.
[20] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.
[21] H. Takagi, "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation," *Proc. IEEE*, vol. 89, no. 9, pp. 1275–1296, Sep. 2001.
[22] M. Grötschel, M. Jünger, and G. Reinelt, "A cutting plane algorithm for the linear ordering problem," *Operations Res.*, vol. 32, no. 6, pp. 1195–1220, Dec. 1984.
[23] J. Ceberio, A. Mendiburu, and J. A. Lozano, "The linear ordering problem revisited," *Eur. J. Oper. Res.*, vol. 241, no. 3, pp. 686–696, Mar. 2015.
[24] D. N. Duc, L. Nguyen, and K. T. Trung, "An interactive method for surrogate-assisted multi-objective evolutionary algorithms," in *Proc. 12th Int. Conf. Knowl. Syst. Eng. (KSE)*, Nov. 2020, pp. 195–200.
[25] R. Martí and G. Reinelt, *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization*, vol. 175. Cham, Switzerland: Springer, 2011.
[26] J. Hessel, A. Holtzman, M. Forbes, R. Le Bras, and Y. Choi, "CLIPScore: A reference-free evaluation metric for image captioning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 7514–7528.
[27] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. H. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, "LAION-5B: An open large-scale dataset for training next generation image-text models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022, pp. 25278–25294.
[28] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 29, 2016, pp. 2234–2242.
[29] G. Santana. (2022). *Gustavosta/Stable-Diffusion-Prompts · Datasets At Hugging Face*. [Online]. Available: https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts
[30] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models," 2022, *arXiv:2211.01095*.
[31] P. von Platen. (2022). *Diffusers: State-of-the-Art Diffusion Models*. [Online]. Available: https://huggingface.co/docs/diffusers/using-diffusers/weighted_prompts
[32] S. Baek, H. Im, J. Ryu, J. Park, and T. Lee, "PromptCrafter: Crafting text-to-image prompt through mixed-initiative dialogue with LLM," 2023, *arXiv:2307.08985*.
[33] Y. Wang, S. Shen, and B. Y. Lim, "RePrompt: Automatic prompt editing to refine AI-generative art towards precise expressions," in *Proc. CHI Conf. Human Factors Comput. Syst.*, Apr. 2023, pp. 1–29.
[34] Y. Feng, X. Wang, K. K. Wong, S. Wang, Y. Lu, M. Zhu, B. Wang, and W. Chen, "PromptMagician: Interactive prompt engineering for text-to-image creation," *IEEE Trans. Vis. Comput. Graphics*, vol. 30, no. 1, pp. 295–305, Jan. 2023.
[35] J. Gu, Z. Han, S. Chen, A. Beirami, B. He, G. Zhang, R. Liao, Y. Qin, V. Tresp, and P. Torr, "A systematic survey of prompt engineering on vision-language foundation models," 2023, *arXiv:2307.12980*.
[36] Q. Zhao, J. Li, L. Li, Z. Zhou, and J. Liu, "Word order's impacts: Insights from reordering and generation analysis," 2024, *arXiv:2403.11473*.
[37] S. Kumar and P. Talukdar, "Reordering examples helps during priming-based few-shot learning," 2021, *arXiv:2106.01751*.
[38] K. Chen, R. Wang, M. Utiyama, and E. Sumita, "Explicit reordering for neural machine translation," 2020, *arXiv:2004.03818*.

WOOSEOK SONG received the B.S. degree in electrical engineering and computer science from Gwangju Institute of Science and Technology (GIST), Republic of Korea, in 2020, where he is currently pursuing the Ph.D. degree in artificial intelligence with the AI Graduate School. His research interests include generative artificial intelligence and evolutionary optimization, focusing on personalization and model compression.

CHANG WOOK AHN (Member, IEEE) received the Ph.D. degree from the Department of Information and Communications, Gwangju Institute of Science and Technology (GIST), Republic of Korea, in 2005. From 2005 to 2007, he was with the Samsung Advanced Institute of Technology, South Korea. From 2007 to 2008, he was a Research Professor with GIST. From 2008 to 2016, he was an Assistant/Associate Professor with the Department of Computer Engineering, Sungkyunkwan University (SKKU), Republic of Korea. He is currently a Professor with the AI Graduate School, GIST. His research interests include genetic algorithms/programming, multi-objective optimization, neural networks, and quantum machine learning.