A Novel Hybrid Inference Scheme for Diffusion-Based AIGC Services in MEC Networks

Xinyi Zhuang, Jiaqi Wu, Yuan Luo, *Member, IEEE*, Ming Tang, *Member, IEEE*, Huaizhe Liu, Hongjia Wu, Lin Gao, *Senior Member, IEEE*, and Qinyu Zhang, *Senior Member, IEEE*

***Abstract*—AI-Generated Content (AIGC) based on Generative Diffusion Models (GDMs) is revolutionizing content creation and promoting substantial advancements in various domains. With the advancement of Mobile Edge Computing (MEC) and model compression techniques, GDMs are increasingly being deployed on Edge Servers (ESs) and end User Equipments (UEs), offering more adaptive and flexible AIGC services. In MEC scenarios, however, designing an efficient inference scheme for GDMs remains a significant challenge, due to the diverse intentions of AIGC service users, as well as the resource limitations of ESs and UEs. In this work, we propose a novel *Hybrid Inference Scheme* (HIS) for MEC-enabled AIGC services, consisting of two phases: (i) a *common inference phase* which generates common intermediate results for all users with similar intentions using public prompts (derived from all user-provided personal prompts), and (ii) a *personalized inference phase* which generates final results (contents) for each individual user based on their personal prompts. To quantitatively measure the service quality of the HIS, we conduct extensive experiments and derive a novel Hybrid Inference Quality Metric (HIQM), which measures the semantic alignment of generated contents under different semantic similarities and common inference steps. Building upon this, we formulate a joint user clustering, model inferencing, and resource allocation problem, aiming to jointly optimize service quality, delay, and energy consumption. We further propose a semantic-based clustering approach for user clustering, and an integrated deep reinforcement learning and optimization approach for joint model inferencing and resource allocation. Simulation results show that our proposed approach significantly outperforms existing benchmarks, with performance improvements ranging from 26.8% to 94.3%.**

***Index Terms*—AI-generated content, generative diffusion model, mobile edge computing, deep reinforcement learning**

I. Introduction

A. Background and Motivations

AI-Generated Content (AIGC) has recently attracted significant attention for its transformative impact on controllable content generation. A key driver behind this revolution is the development of large AI models, which have shown exceptional capability in synthesizing high-quality images, videos, and 3D representations. Among these models, Generative Diffusion Models (GDMs) have emerged as one of the most influential, playing a pivotal role in a wide range of interactive tools and creative applications, such as point cloud reconstruction [2], autonomous driving [3], robotics [4], and artistic expression [5].

The success of GDMs lies in their innovative step-wise denoising process, which progressively transforms simple noise distribution into complex data distributions. This process involves a series of small and reversible *inference steps*, which can be viewed as discretely solving a Stochastic Differential Equation (SDE) [6]. This groundbreaking capability have led to the development and deployment of advanced commercial models such as Stable Diffusion, Sora, and GPT-4o [7], demonstrating the great potential in both research and industry. As a result, the continuous evolution of GDMs is expected to drive further innovations across various application domains, particularly in wireless networks with the explosion of AI-enabled network applications.

In wireless network scenarios, GDMs are typically deployed on *cloud servers* to deliver lagre-scale AIGC services, showcasing impressive capabilities in generating diverse and coherent visual content. However, accessing AIGC services from remote cloud servers poses several challenges, such as server vulnerability, service security, and latency issue, which limit their suitability for next-generation network applications. To overcome these limitations, *Mobile Edge Computing (MEC)* has emerged as a crucial enabling technology for AIGC services. MEC mitigates these challenges by pushing computational workload down to Edge Servers (ESs) and User Equipment (UEs) that approximate to end users, thereby reducing service latency and enhancing service efficiency [5]. While model compression techniques have facilitated the deployment of GDMs on both ESs and UEs, it remains a significant challenge to design an efficient inference scheme for GDM-based AIGC services in MEC networks, due to the diverse intentions of AIGC service requesters, as well as the resource limitations of ESs and UEs.

Some recent works (e.g., [8]–[12]) have proposed to improve inference efficiency through optimizing model inferencing and task offloading. However, these studies primarily focus on *independent inference* schemes, where the inference processes of different users are optimized separately and independently, without considering the intention similarities of different users and the potential collaboration among different inference processes. In practice, given the unique characteristic of GDMs (i.e., evolving from simple noise distribution to intended data distributions), inference processes of users with similar intentions can potentially collaborate (e.g., by sharing the intermediate result), to further enhance inference efficiency. Nevertheless, existing *collaborative inference* schemes in [13]–[15] usually select a random user to execute the inference process, and share the same intermediate result with other users. Obviously, these approaches do not fully explore and exploit the similarities among user intentions. This motivates us to explore a more efficient collaborative inference

Xinyi Zhuang and Jiaqi Wu contributed equally to this work. Authors are with the School of Electronics and Information Engineering, Harbin Institute of Technology, Shenzhen, China. Email: gaol@hit.edu.cn. *(Corresponding Author: Lin Gao)*

Part of the results have been accepted by IEEE ICC 2025 [1].

scheme that leverages the intention similarities, leading to the following challenge:

Challenge 1. *How to improve the inference efficiency by leveraging the diverse similarities among user intentions to fully exploit the potential of collaborative inference?*

Furthermore, due to the resource limitations on ESs and UEs, the joint optimization of model inferencing and resource allocation becomes essential for enhancing inference efficiency. Some prior works (e.g., [16], [17]) primarily focused on optimizing the number of inference steps to balance service quality, delay, and energy consumption. Other works (e.g., [18], [19]) focused on fine-grained resource management, primarily using CPU-centric frequency scaling approaches for resource allocation. However, these works are based on independent inference schemes, overlooking the potential benefits of collaboration among different inference processes. In addition, in the context of collaborative inference, the similarities among user intentions play a ctirical role in the inference results. Specifically, for users with similar intentions, collaborative inference can significantly improve the inference efficiency, while preserving the quality of the generated contents. For users with diverse intentions, however, collaborative inference may lead to severe quality degradation. Therefore, it is crucial to design an effective user clustering algorithm that groups users with similar intentions together. This motivates us to investigate user clustering, together with model inferencing and resource allocation, in the context of collaborative inference, leading to the following challenge:

Challenage 2. *How to further improve the inference efficiency by jointly optimizing user clustering, model inferencing, and resource allocation for collaborative inference?*

To address these challenges, we first develop an efficient collaborative inference scheme that leverages the similarities among user intentions, and then propose effective algorithms for user clustering, model inferencing, and resource allocation.

B. Solution and Contributions

For **challenge 1**, we propose a novel *Hybrid Inference Scheme (HIS)* which consists of two phases: (i) a *common inference phase*, generating common intermediate results for all users with similar intentions using public prompts (constructed to capture the general characteristics of all user-provided personal prompts); (ii) a *personalized inference phase*, generating final results (contents) for each individual user based on their personal prompts. Building upon this, we further develop a Semantic Intensity Modulator (SIM) for the common inference phase, and a Negative Prompt Injector (NPI) for the personalized inference phase to collectively improve the performance of the HIS. For clarity, we illustrate the traditional independent inference scheme and our proposed HIS in Fig. 1. In the former scheme (Part A), two inference processes work independently using their respective prompts. In our proposed HIS (Part B), a common inference process generates the common intermediate results using the public prompt, after which two personalized inference processes generate their final results using their respective personal prompts.

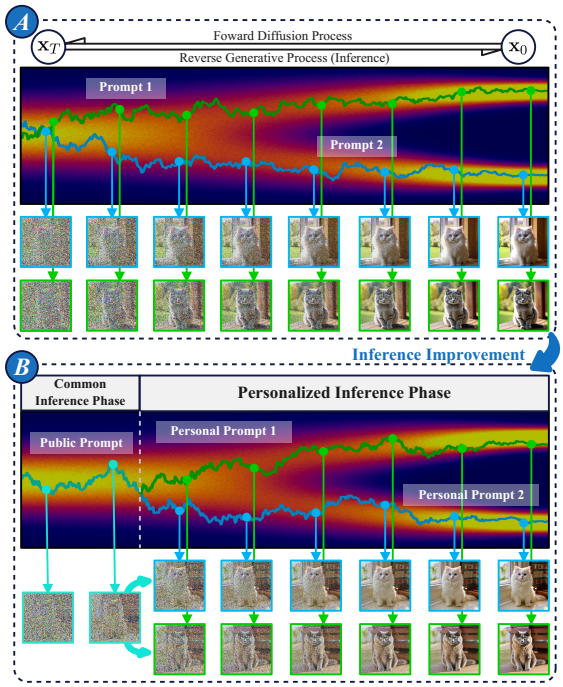

Fig. 1. A synthetic overview of the traditional independent inference scheme (**Part A**) and our proposed HIS (**Part B**).

To quantitatively measure the service quality of the HIS, we conduct extensive real-world experiments and propose a novel Hybrid Inference Quality Metric (HIQM). This new metric measures the semantic alignment of the generated contents with the corresponding prompts, under different semantic similarities (between public and personal prompts) and common inference steps.

For **challenge 2**, we consider a resource-constrained MEC network consisting of one ES and multiple UEs, where both the ES and UEs are equipped with GDMs to collaboratively deliver AIGC services. In such a scenario, we focus on the joint optimization of user clustering, model inferencing, and resource allocation, aiming to optimize service quality, service delay, and energy consumption. For the optimization of user clustering, we propose a semantic-based clustering approach by refining a Hierarchical Agglomerative Clustering (HAC) algorithm, which groups semantically similar users into the same cluster. For the joint optimization of model inferencing and resource allocation, we propose a two-tier solution that integrates both learning and optimization techniques. In the first tier, we design a Deep Reinforcement Learning (DRL)-based Independent Proximal Policy Optimization (IPPO) approach to optimize the common and personalized inference steps. In the second tier, we derive the closed-form solution for the optimal resource allocation based on convex optimization theory. The key contributions are summarized as follows:

- ***Novel Hybrid Inference Scheme:*** We propose a novel hybrid inference scheme, HIS, which leverages the similarities among user intentions to fully exploit the potential of collaborative inference. We derive a new performance metric, HIQM, which quantitatively measures the semantic alignment of the generated contents.

- ***Optimization of User Clustering:*** We propose a semantic-based clustering approach, which improves overall alignment between public and personal prompts in each cluster, thus enhancing inference efficiency by enabling larger common inference steps.
- ***Joint Optimization of Model Inferencing and Resource Allocation:*** We propose a two-tier solution that combines both learning and optimization techniques for the joint model inferencing and resource allocation problem. The approach exhibits exceptional flexibility and adaptability, making it well-suited for handling large-scale problems.
- ***Performance Evaluation:*** Simulation results demonstrate that our proposed approach significantly outperforms existing benchmarks, with performance improvements ranging from 36.8% to 94.3%, highlighting its superiority in terms of efficiency and scalability.

II. Related Work

A. Model Acceleration

GDMs have achieved state-of-the-art performance across various domains. Current best-performing GDMs mainly rely on U-Net or transformer-based architectures, which involve iterative evaluations of large-scale neural networks, resulting in significant computational overhead.

Low-resource inference for efficient GDMs has recently attracted substantial interest. One major direction focuses on reducing the total number of inference steps. For instance, DDIM [20] and DPM solver [21] have introduced first-order and high-order solvers to accelerate the inference process. Parallel efforts explore architectural compression: knowledge distillation [22] trains lightweight student models to mimic pre-trained teachers, quantization techniques [23] reduce numerical precision of weights or activations, and pruning [9] eliminates redundant network parameters. Additionally, latent diffusion framework [24] significantly reduces computational costs by operating in a compressed latent feature space rather than pixel domain. However, most existing approaches necessitate computationally intensive retraining or fine-tuning to sustain satisfactory performance, which imposes practical limitations in resource-constrained scenarios.

In our work, we tackle the computational challenge from a unique angle. We aim to curtail the overall computation of GDMs by sharing a part of the inference process. It is crucial to emphasize that our proposed scheme is *orthogonal* to the aforementioned methods and can seamlessly integrate with them without inducing extra cost.

B. Collaborative Inference

In the context of well-trained GDMs with determined SDE solvers, some researchers have explored model scheduling to alleviate the computational overhead. As one of the pioneering works, NVIDIA Corp. [8] introduced the eDiff-I framework, which decomposes the inference process into multiple stages, each handled by a distinct model to process hierarchical information. Following this, Yang *et al.* [9] and Liu *et al.* [10] further advanced this approach by scheduling varying-scale GDMs across different inference steps to accelerate inference. However, these works focus on pre-defined computational graphs and can not adapt to dynamic network conditions. In contrast, ByteDance Inc. [11] proposed the edge-cloud framework, which splits the entire inference process into two phases: the first is handled in the cloud, while the second is executed at the edge. Yang *et al.* [12] further optimized the split point in an end-edge framework by considering latency constraints and computational resources. Nonetheless, the above works focus on independent inference, where inference processes of different users are independent with each other.

By leveraging user intentions, Du *et al.* [13] first proposed the collaborative inference scheme in which multiple users share the same inference process during the initial phase to reduce redundant computation. Their subsequent work [14] introduced a two-phase architecture where the edge server first performs inference by leveraging a randomly selected prompt, followed by on-device refinement for personalization. Xie *et al.* [15] further extended this architecture to vehicular networks by introducing an entity-based clustering strategy. However, these schemes oversimplify the diversity of user intentions by assuming a homogeneous intermediate result for personalization. This stringent setting significantly shortens the common inference phase, ultimately constraining the potential efficiency for MEC-enabled AIGC services.

In our study, we *re-design* the collaborative inference scheme and propose the HIS, considering heterogeneous user intentions. Moreover, we investigate the user clustering problem, together with model inferencing and resource allocation, to improve the inference efficiency of the HIS.

C. Model Inferencing and Resource Allocation

For MEC-enabled AIGC services, existing works typically focused on optimizing either model inferencing or resource allocation independently. Regarding model inferencing, several works (e.g., [18]) primarily addressed the deep neural network partitioning problem. By leveraging the new features of GDMs, Liu *et al.* [16] and Wang *et al.* [17] proposed optimizations that reduce the total number of inference steps to enhance network utility. However, these approaches mainly concentrate on vanilla GDMs and often overlook the impact of user intention heterogeneity.

As for resource allocation, some works (e.g., [25]) have explored resource management through indirect service scheduling mechanisms. While these approaches demonstrate some initial effectiveness, there remains significantly potential for further optimization. For example, Liu *et al.* [18] and Li *et al.* [19] have proposed fine-grained resource allocation techniques through computing frequency assignment. However, these approaches are limited by the computational intensity of most GDMs, which require GPU-accelerated inference, rendering CPU-centric frequency scaling approaches less applicable.

In our study, we develop the *joint optimization* framework for model inferencing and resource allocation, with the goal of achieving an optimal trade-off among service quality, delay, and energy consumption.

III. Hybrid Inference Scheme

In this section, we first outline the image generation process, which serves as the foundation of model inferencing. Then, we introduce the hybrid inference scheme, HIS, together with two key modules (i.e., SIM and NPI), and the new performance metric, i.e., HIQM, for the HIS.

A. Image Generation

GDMs synthesize images by learning to estimate a score function, which is defined as the gradient of the log-density of image distribution [6]. For more clarity, we can interpret GDMs through the framework of SDE. From this perspective, GDMs initiate a forward diffusion process that gradually corrupts an original image distribution $p(\mathbf{x}_0)$ by adding Gaussian noise over time. This progressive corruption eventually transforms the data into a noisy distribution $p(\mathbf{x}_T)$. Mathematically, this forward diffusion process can be expressed as:

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)dt + g(t)d\mathbf{w}_t, \tag{1}$$

where $\mathbf{f}(\mathbf{x}_t, t)$ and $g(t)$ denote drift and diffusion coefficients, respectively, and $d\mathbf{w}_t$ represents the Wiener process [6].

As shown in Fig. 1 (Part A), GDMs generate images by *reversing* the forward process, which starts from a Gaussian prior and progressively refines the distribution through the estimated score function. This reverse generative process, i.e., the inference process, can be mathematically expressed as:

$$d\mathbf{x}_t = \left[\mathbf{f}(\mathbf{x}_t, t) - g(t)^2 \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{p})\right] dt + g(t)d\bar{\mathbf{w}}_t, \tag{2}$$

where $\mathbf{p}$ denotes the prompt embeddings, $d\bar{\mathbf{w}}_t$ denotes the Wiener process, and $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{p})$ represents the conditional score function approximated by neural networks.

To enhance conditional generation performance, GDMs often adopt the classifier-free guidance [26] technique. This leads to a reformulated score function:

$$\begin{aligned} \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{p}) := w\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{p}) \\ + (1-w)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t), \end{aligned} \tag{3}$$

where w denotes the guidance scale. It significantly improves generation quality by guiding the distribution towards a higher likelihood of the condition $\mathbf{p}$. In practice, the inference process is implemented by discretizing the reverse SDE into finite steps. At each step, the model evaluates both conditional and unconditional scores, then combines them using the guidance scale to iteratively refine the distribution. This iterative guidance ensures that the generated output is not only coherent but also aligned with the semantic intent of the prompt.

B. Hybrid Inference Scheme (HIS)

As discussed in Section III-A, traditional GDM inference treats each prompt independently. For each individual prompt, GDM initializes sampling from a Gaussian distribution and then employs a score function to guide the distribution toward a prompt-specific target. This approach may result in redundant computation, particularly when processing multiple prompts that exhibit semantic similarities [13]. In such cases, these semantic similarities can be leveraged at the early phase of inference to generate a common intermediate result, which can then be further refined through the independent inference. Clearly, identifying and exploiting these semantic similarities can substantially reduce computational cost. Motivated by this, we propose the HIS[1] shown in Fig. 1 (Part B), a novel scheme that decomposes the inference process into two phases:

- *Common Inference Phase:* A public prompt $\mathbf{p}^{\text{pub}}$ is first constructed to capture the general semantic characteristics common across multiple user-provided prompts. Then, the GDM utilizes the score function defined in Eq. (3) with $w := w_1$ and $\mathbf{p} := \mathbf{p}^{\text{pub}}$ to generate a common intermediate result, where w_1 is the guidance scale for the common inference phase.
- *Personalized Inference Phase:* The GDM injects the common intermediate result along with each personal prompt[2] $\mathbf{p}^{\text{per}}$ and continues inference by using the score function defined in Eq. (3) with $w := w_2$ and $\mathbf{p} := \mathbf{p}^{\text{per}}$, where w_2 is the guidance scale for the personalized inference phase. This process generates the final output that matches the specific intent of each personal prompt.

C. Semantic Intensity Modulator (SIM)

In the common inference phase, a key challenge is to ensure that the common intermediate result remains sufficiently compatible with a wide range of personal prompts. As shown in the rightmost column of Fig. 2, the default configuration (i.e., $w_1 = 7.0$), which is widely used in prior works such as [11], [12], [15], often yields poor performance. The underlying reason is that the public prompt primarily offers high-level semantic information, while lacking explicit details about subjects or background elements. As a result, during inference, GDMs often compensate for this ambiguity by autonomously "fill in" unspecified details. These unintended additions in the common intermediate result introduce noise that complicates the subsequent personalized inference phase, which must first remove extraneous elements before accurately synthesizing the specified details.

To address this challenge, we introduce the SIM, a mechanism designed to control the *semantic intensity* embedded in the common intermediate result. The key idea behind the SIM is to regulate the guidance scale w_1 in the common inference phase. On one hand, as shown in Eq. (3), lowering the guidance scale increases the randomness and naturalness in the intermediate distribution, effectively shifting it away from sharp peaks toward more central and generalized regions [27]. This transformation is particularly beneficial for subsequent personalized inference, as transiting from a central position in the distribution space is much easier than navigating between well-separated peaks. On the other hand, if the guidance scale is too small, the resulting intermediate distribution may diverge too far from the intended semantics. This excessive randomness can obscure the key features of the public prompt, ultimately degrading the fidelity of the generated images. Therefore, the SIM plays a crucial balancing role in ensur-

[1]Code is available at: https://github.com/iimxinyi/HIS

[2]We denote the user-provided prompt as the personal prompt hereafter to distinguish it from the public prompt.

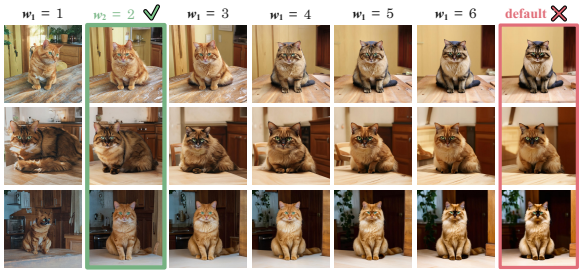

Fig. 2. Generated images under different guidance scale settings. All non cherry picked images are based on the same public prompt ("*A graceful cat sitting in a warm and story-rich environment, highlighting its silky fur.*") and the same personal prompt ("*A brown cat with green eyes sits calmly on a rustic wooden table in a sunlit kitchen.*")

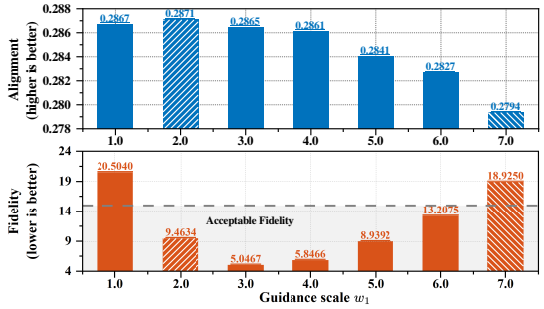

Fig. 3. The alignment (**Top**) and fidelity (**Bottom**) scores under different guidance scale settings.

ing semantic flexibility while preserving the integrity of the general intent.

To empirically investigate the above trade-off, we conduct extensive experiments using a diverse set of real-world prompts. Our goal is to quantify two critical aspects, i.e., *alignment* and *fidelity*, under different guidance scales. Specifically, the alignment reflects the semantic correspondence between the final generated image and its corresponding personal prompt. We evaluate it by using Contrastive Language-Image Pre-Training (CLIP) model [28]. The fidelity reflects both visual naturalness and adherence to realistic image distributions. We evaluate it by using Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [29].

Fig. 3 illustrates the experiment results under different guidance scales, which reveals that a guidance scale of $w_1 = 2.0$ achieves the best trade-off, i.e., reaching the highest alignment score while maintaining an acceptable fidelity score. In contrast, a higher guidance scale $w_1 = 7.0$ leads to semantic overfitting, generating overly deterministic but less relevant content, whereas a lower guidance scale $w_1 = 1.0$ introduces excessive randomness, which diminishes semantic coherence. Based on these empirical observations, we select $w_1 = 2.0$ as the optimal guidance scale. This setting strikes an effective balance between alignment and fidelity, thereby significantly enhancing the overall performance of the HIS.

D. Negative Prompt Injector (NPI)

In the personalized inference phase, a key challenge is to effectively guide the intermediate distribution toward the target distribution outlined by the personal prompt. A seemingly straightforward approach is to increase the guidance scale w_2, with the expectation that stronger conditioning will better enforce prompt-specific features. However, this approach often results in oversaturated images and the emergence of visual artifacts, ultimately degrading the overall image fidelity [30].

Fig. 4. Generated images with (**Top**) and without (**Bottom**) the NPI.

TABLE I
ALIGNMENT AND FIDELITY SCORES WITH AND WITHOUT THE NPI

	Alignment (CLIP)	Fidelity (BRISQUE)	Latency
w/ NPI	**0.287**	**9.463**	6.379 steps/s
w/o NPI	0.284	9.978	**6.377** steps/s

To address this challenge, we introduce the NPI[3], a strategy that utilizes *negative prompts* to indirectly guide the intermediate distribution without causing oversaturation. Unlike directly increasing the guidance scale w_2, negative prompts discourage the present of undesirable semantic characteristics, thereby reducing the risk of extreme distribution shifts while preserving the naturalness and diversity of the generated content [31]. To construct the negative prompt $\mathbf{p}^{\text{per-}}$ of a personal prompt $\mathbf{p}^{\text{per}}$, we retain the core subject and replace descriptive adjectives with their semantic antonyms, which embodies the undesirable characteristics that the model should avoid. By integrating both prompts in the personalized inference phase, the score function in Eq. (3) can be reformulated as follows:

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{p}^{\text{per}}) := w_2 \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{p}^{\text{per}}) \\ + (1 - w_2) \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{p}^{\text{per-}}).\end{aligned} \tag{4}$$

Fig. 4 illustrates the generated images with (top) and without (bottom) the NPI, which demonstrates a clear visual enhancement in alignment when the NPI is applied. Compared to results without the NPI, the outputs more accurately reflect the semantic intent of the personal prompt. To further support this visual evidence, we provide quantitative results in Table I, which shows that the NPI consistently improves the alignment score across diverse prompts without compromising fidelity. Additionally, we find that the computational overhead introduced by incorporating the NPI is minimal, making it a practical enhancement rather than a costly trade-off. As a result, we incorporate the NPI into the personalized inference phase of the HIS, contributing significantly to the overall performance improvement.

E. Hybrid Inference Quality Metric (HIQM)

It is easy to see that in the proposed HIS, the quality of the generated images is closely related to two key factors: (1) the semantic similarity between the public and personal prompts, and (2) the number of common inference steps. Both factors significantly influence the semantic consistency of the outputs.

[3]Note that both the SIM and NPI are training-free and require no modification to the underlying structure of GDMs.

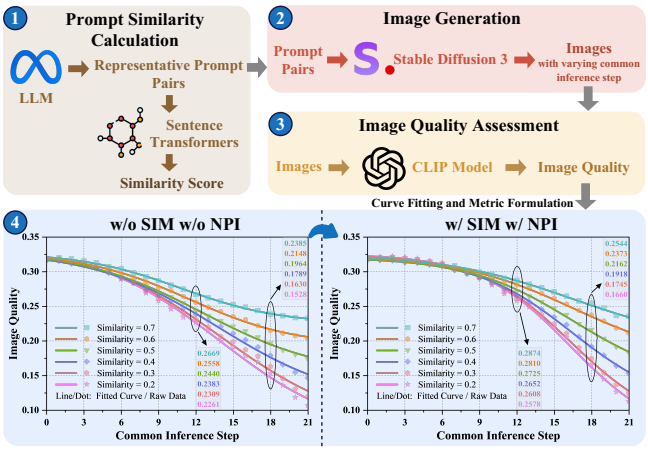

Fig. 5. An overview of the experimental process and the corresponding fitted curve based on real data.

To systematically analyze these influences, we conduct a series of empirical experiments using Ubuntu 20.04 system equipped with an Intel Xeon Gold 6248R CPU and an NVIDIA A100 GPU. Based on the experiment results, we formulate a novel metric, the HIQM, to quantify the quality of generated images under different conditions. Fig. 5 presents the experimental procedure, which consists of four steps:

1) Prompt Similarity Calculation (Step 1): We first use advanced Large Language Models (LLMs) (e.g., Llama [32]) to generate representative prompts, which are then randomly organized into public-personal prompt pairs. The semantic similarity of each prompt pair is computed using a lightweight Sentence-Transformer model [35]:

$$\phi = \mathrm{Sim}\left(\mathrm{emb}_{\text{S-T}}\left(\mathbf{p}^{\mathrm{pub}}\right), \mathrm{emb}_{\text{S-T}}\left(\mathbf{p}^{\mathrm{per}}\right)\right), \tag{5}$$

where $\mathrm{Sim}(\cdot,\cdot)$ and $\mathrm{emb}_{\text{S-T}}(\cdot)$ are the cosine similarity and prompt embedding functions, respectively.

2) Image Generation (Step 2): We adopt Stable Diffusion 3 Medium [34] as a representative GDM to generate images via the HIS with different common inference steps. We generate a total of $27,104$ images, each with a resolution of 1024×1024, allowing for robust statistical analysis. From the generated images, we find that the common inference steps primarily affect the alignment of the generated images (with the personal prompt), and the extent of this impact is closely related to the similarity of the public and personal prompts. More specifically, (i) as the number of common inference steps increases, the alignment of the generated images gradually decreases, while the fidelity of the generated images remains consistently satisfactory; (ii) with a higher prompt similarity, the alignment decreases at a slower rate with the increasing of common inference steps. Based on the above, we define the image quality as the alignment of the generated image.

3) Image Quality Assessment (Step 3): We utilize the pre-trained ViT-L/14@336px CLIP model [28] to quantify the image quality (i.e., the alignment between the generated image and the personal prompt), which is expressed as:

$$S_{\mathrm{CLIP}} = \mathrm{Sim}\left(\mathrm{emb}_{\mathrm{CLIP}}\left(\mathbf{p}^{\mathrm{per}}\right), \mathrm{emb}_{\mathrm{CLIP}}\left(\mathbf{I}\right)\right), \tag{6}$$

where $\mathrm{emb}_{\mathrm{CLIP}}(\cdot)$ is the CLIP embedding funciton, and $\mathbf{I}$ is the generated image with certain common inference steps.

4) Curve Fitting and Metric Formulation (Step 4): Fig. 5 (Part 4) illustrates the image quality under different common inference steps and prompt similarities. We can see that as the common inference steps increase, image quality initially declines gently, then deteriorates rapidly, and eventually stabilizes. Moreover, with the SIM and NPI, image quality attains a higher value at the same common inference steps compared to cases without them. To characterize the relationship between the image quality and common inference steps, we use a modified sigmoid function to fit the quality-step curve. That is, the HIQM function can be defined as follows:

$$\hat{Q}(x) = \frac{L^{\phi,[\iota]}}{1+\exp\left(K^{\phi,[\iota]}\cdot\left(x-B^{\phi,[\iota]}\right)\right)} + G^{\phi,[\iota]}, \tag{7}$$

where x is the number of common inference steps, and $L^{\phi,[\iota]}$, $K^{\phi,[\iota]}$, $B^{\phi,[\iota]}$, $G^{\phi,[\iota]}$ are hyper-coefficients related to prompt similarity ϕ and module indicator ι ($\iota=1$ if the SIM and NPI are enabled, $\iota=0$ otherwise).

From Fig. 5 (Part 4), we can find when increasing x, the image quality gradually deteriorates, with the decline becoming more pronounced at lower prompt similarity. That is, as the prompt similarity increases, the fitted curve becomes flatter and less abrupt, implying that images at higher similarity maintains better quality. Furthermore, when the SIM and NPI are enabled, the entire curve shifts upward and flattens further, implying that these modules can effectively enhance the inference process by maintaining or improving image quality throughout the inference steps.

In the following, we will present the MEC-enabled AIGC network scenario, and formulate the joint user clustering, model inferencing, and resource allocation problem.

IV. System Model

In this section, we illustrate the detailed workflow of our proposed HIS within an MEC-enabled AIGC network.

A. MEC-Enabled AIGC Network Model

As shown in Fig. 6 (Part A), we consider an MEC-enabled AIGC network, which consists of a set $\mathcal{N} \triangleq \{1,\cdots,N\}$ of N UEs (i.e., AIGC service requesters), each with limited computational resources, and one BS equipped with an ES that offers powerful computational resources to support AIGC services. UEs request AIGC services to generate desired contents. With the advancement of model compression techniques, GDMs can be deployed on both the ES and UEs.

Fig. 6 (Part B) illustrates the detailed workflow of the HIS in the MEC-enabled AIGC network. Specifically, at the user layer, UEs initiate their personal prompts, which are uploaded to the ES at the edge layer for further processing. At the edge layer, the ES groups UEs with similar prompts into a *cluster*, after which the LLM generates a corresponding public prompt for each cluster, as well as a negative prompt for each UE. Subsequently, the HIS performs the following inference works:

- *Common Inference Phase:* The GDM (on the ES) generates common intermediate results for each cluster based on the public prompt. These results will be shared by all UEs in the same cluster for later personalized inference.

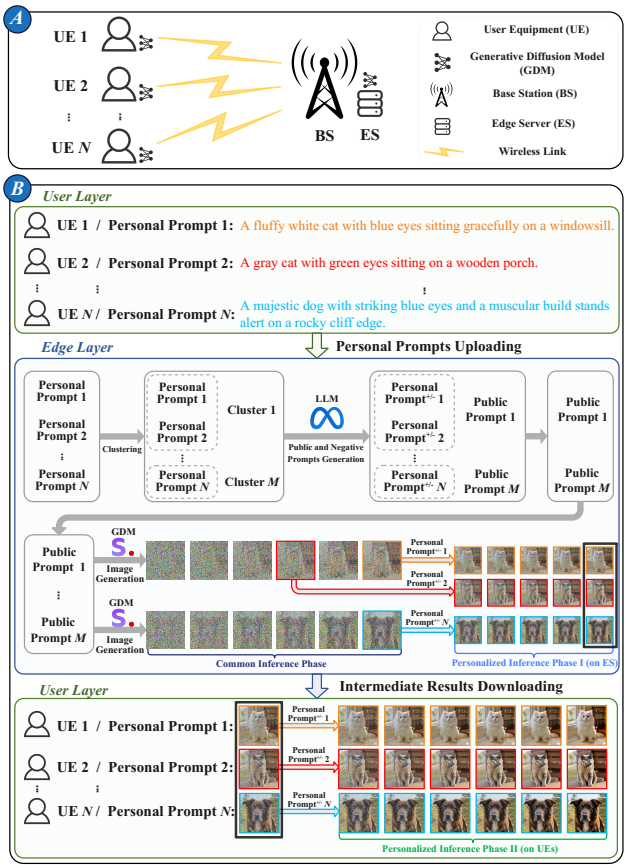

Fig. 6. **Part A:** Illustration of an MEC-enabled AIGC network. **Part B:** Workflow of the HIS within the MEC-enabled AIGC network.

- *Personalized Inference Phase I (on ES):* The GDM (on the ES) selects a appropriate common intermediate result for each UE, and injects it (along with the UE's personal and negative prompts) into the model to generate the personalized intermeidate result for each UE. This result is then sent to the UE for subsequent inference.
- *Personalized Inference Phase II (on UEs):* The GDM (on the UE) injects the received intermeidate result into the model to generate the final result.

B. Hybrid Inference Scheme

1) User Clustering: As illustrated in Fig. 6 (Part B), each UE $n \in \mathcal{N}$ is associated with a personal prompt $\mathbf{p}_n^{\text{per}}$. Let $\mathbf{P}^{\text{per}} = \{\mathbf{p}_n^{\text{per}} | n \in \mathcal{N}\}$ denote the set of all UEs' personal prompts. The ES organizes all UEs into M distinct clusters, denoted as $\mathcal{M} \triangleq \{1, \cdots, M\}$. We introduce a binary clustering decision $\gamma_{n,m} \in \{0, 1\}$, with $\gamma_{n,m} = 1$ indicating that UE n is assigned to cluster m, and $\gamma_{n,m} = 0$ otherwise. For clarity, we use $\mathcal{N}_m$ to denote the set of UEs in cluster m.

After user clustering, we adopt a lightweight fine-tuned LLM[4] (i.e., Llama-3-8B [32]) to extract the general semantic characteristics of prompts in each cluster, and generate a high-dimensional, generalized *public prompt* $\mathbf{p}_m^{\text{pub}}$ for each cluster m. Meanwhile, the LLM generates a negative prompt for each personal prompt by retaining its key subject and substituting descriptive adjectives with their semantic antonyms. Let $\mathbf{p}_n^{\text{per-}}$ denote the negative prompt of UE n's personal prompt. Thus, all personal prompts and their corresponding negative prompts can be denoted as $\mathbf{P}^{\text{per+/-}} = \{(\mathbf{p}_n^{\text{per}}, \mathbf{p}_n^{\text{per-}}) | n \in \mathcal{N}\}$.

[4]Note that Llama-3-8B is a lightweight, fine-tuned LLM with a very low computational burden, and can be executed in batches at a very high speed.

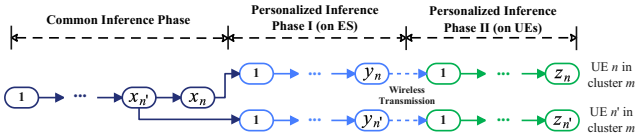

Fig. 7. Illustration of the workflow of the HIS, where UE n and n' belong to the same cluster m.

2) Hybrid Inferencing: Fig. 7 illustrates the detailed workflow of the HIS for each cluster m, where x_n denotes the common inference steps for UE n, y_n and z_n denote the personalized inference steps on the ES and on UE n, respectively. Specifically, in the common inference phase, the GDM uses the public prompt $\mathbf{p}_m^{\text{pub}}$ to perform $x_m^{\max} \triangleq \max\{x_n, n \in \mathcal{N}_m\}$ inference steps, and generate a series of common intermediate results. In personalized inference phase I (on ES), the GDM selects the corresponding intermediate result for each UE n, and perform y_n inference steps to generate the personalized intermediate result for UE n. In the personalized inference phase II (on UEs), the GDM on UE n performs the remaining z_n inference steps to generate the final result.

C. Service Delay and Energy Consumption

Now we calculate the service delay and energy consumption in each inference phase.

In the common inference phase, the GDM on the ES performs $x_m^{\max}$ inference steps for each cluster m. Thus, the computation delay for cluster m can be calculated as:

$$T_m^{\text{C}} = \frac{x_m^{\max} \xi^{\text{ES}}}{f_m^{\text{ES}}}, \tag{8}$$

where ξ^{ES} is the number of Floating Point Operations (FLOPs) required for each inference step on the ES, and f_m^{ES} is the number of FLOPs per Second (FLOPS) allocated to cluster m on the GPU at the ES.

We adopt a novel GPU SM partitioning strategy that enables precise control over parallel computational resources. According to [35], multi-task GPU scheduling can be effectively managed using the spatial sharing technique, which allows multiple processes to share a GPU while maintaining high utilization and performance. Specifically, the spatial sharing technique optimizes GPU usage and performance by enabling the assignment of SMs to distinct partitions. Based on this, f_m^{ES} can be calculated as:

$$f_m^{\text{ES}} = v_m^{\text{ES}} F^{\text{ES}} D^{\text{ES}} \rho^{\text{ES}}, \tag{9}$$

where v_m^{ES} is the number of SMs allocated to cluster m, F^{ES} is the computing frequency, D^{ES} is the number of FLOPs per cycle per SM depending on the data and GPU type, and ρ^{ES} is a discount factor reflecting non-computational bottlenecks.

Let P_m^{ES} denote the operation power of the GPU at the ES for cluster m. According to [36], the operation power increases linearly with the number of active SMs and the cube of computing frequency. Thus, P_m^{ES} can be calculated as:

$$P_m^{\text{ES}} = \kappa^{\text{ES}} \left(\mu_1^{\text{ES}} v_m^{\text{ES}} + \mu_2^{\text{ES}}\right) \left(F^{\text{ES}}\right)^3, \tag{10}$$

where κ^{ES} is an intrinsic coefficient related to the chip architecture, μ_1^{ES} is a factor reflecting the degree of power increase for each SM, and μ_2^{ES} is a factor reflecting the baseline power consumption. Based on the above, the energy consumption in this phase can be calculated as:

$$E_m^{\text{C}} = P_m^{\text{ES}} T_m^{\text{C}} = \frac{\kappa^{\text{ES}} x_m^{\max} \xi^{\text{ES}} (F^{\text{ES}})^2}{D^{\text{ES}} \rho^{\text{ES}}} \left(\mu_1^{\text{ES}} + \frac{\mu_2^{\text{ES}}}{v_m^{\text{ES}}} \right). \tag{11}$$

In the personalized inference phase I (on ES), the GDM on the ES performs y_n inference steps for each UE n. Thus, the total computation delay for all UEs in cluster m is:[5]

$$T_m^{\text{P-E}} = \sum_{n \in \mathcal{N}_m} \frac{y_n \xi^{\text{ES}}}{f_m^{\text{ES}}} = \sum_{n \in \mathcal{N}_m} \frac{y_n \xi^{\text{ES}}}{v_m^{\text{ES}} F^{\text{ES}} D^{\text{ES}} \rho^{\text{ES}}}. \tag{12}$$

Similar as (11), the energy consumption can be calculated as:

$$E_m^{\text{P-E}} = \sum_{n \in \mathcal{N}_m} \frac{\kappa^{\text{ES}} y_n \xi^{\text{ES}} (F^{\text{ES}})^2}{D^{\text{ES}} \rho^{\text{ES}}} \left(\mu_1^{\text{ES}} + \frac{\mu_2^{\text{ES}}}{v_m^{\text{ES}}} \right). \tag{13}$$

In the personalized inference phase II (on UEs), UE n downloads the personalized intermediate result from the ES via wireless links, after which the GDM on UE n performs the remaining z_n inference steps to generate the final result. The wireless transmission delay and energy consumption can be calculated as:

$$T_n^{\text{T}} = \frac{n_{\text{h}} n_{\text{w}} n_{\text{c}} \varrho}{r_n \tau}, \quad E_n^{\text{T}} = \frac{p_n n_{\text{h}} n_{\text{w}} n_{\text{c}} \varrho}{r_n \tau}, \tag{14}$$

where $(n_{\text{h}} n_{\text{w}})$ is the total number of pixels in the latent space [24], n_{c} is the number of channels for each pixel, ϱ is the number of bits required to represent the information per pixel per channel, τ is the compression ratio, and r_n is the downlink transmission rate, which can be defined as:

$$r_n = b_n \log_2 \left(1 + \frac{p_n g_n}{\sigma_0^2 b_n} \right), \tag{15}$$

where b_n and p_n are the bandwidth and downlink transmission power, respectively, g_n is the channel gain, and σ_0^2 is the noise power spectral density. The computation delay on UE n can be calculated as:

$$T_n^{\text{P-U}} = \frac{z_n \xi^{\text{UE}}}{f_n^{\text{UE}}}, \tag{16}$$

where ξ^{UE} is the number of FLOPs required for each inference step on the UE, and f_n^{UE} is the available FLOPS on UE n. Similar as Eq. (9), f_n^{UE} can be calculated as:

$$f_n^{\text{UE}} = v_n^{\text{UE}} F_n^{\text{UE}} D_n^{\text{UE}} \rho^{\text{UE}}, \tag{17}$$

where v_n^{UE} is the available SMs on UE n, F_n^{UE} is the computing frequency, D_n^{UE} is the number of FLOPs per cycle per SM, and ρ^{UE} is the discount factor. Similar as Eq. (11), the energy consumption in this phase can be calculated as:

$$E_n^{\text{P-U}} = \frac{\kappa^{\text{UE}} z_n \xi^{\text{UE}} (F^{\text{UE}})^2}{D^{\text{UE}} \rho^{\text{UE}}} \left(\mu_1^{\text{UE}} + \frac{\mu_2^{\text{UE}}}{v_n^{\text{UE}}} \right), \tag{18}$$

where κ^{UE}, μ_1^{UE}, and μ_2^{UE} are defined analogously.

Based on the above, the service delay of each UE n, denoted by T_n, can be calculated as:

$$T_n = \sum_{m=1}^{M} \gamma_{n,m} \left(T_m^{\text{C}} + T_m^{\text{P-E}} \right) + T_n^{\text{T}} + T_n^{\text{P-U}}. \tag{19}$$

The energy consumption of the ES for each cluster m, denoted by E_m, and the energy consumption of each UE n, denoted by E_n, can be calculated as:

$$E_m = E_m^{\text{C}} + E_m^{\text{P-E}}, \quad E_n = E_n^{\text{T}} + E_n^{\text{P-U}}. \tag{20}$$

[5]Here, we define the computation delay in this phase based on the worst-case estimation, i.e., the total time required for completing all personalized inference tasks in the cluster.

As shown in Section III, the service quality of each UE n, denoted by Q_n, can be evaluated by the HIQM function defined in Eq. (7). Then, the total service quality Q, service delay T, and energy consumption E can be calculated as:

$$Q = \sum_{n=1}^{N} Q_n, \quad T = \sum_{n=1}^{N} T_n, \tag{21}$$

$$E = \sum_{m=1}^{M} E_m + \sum_{n=1}^{N} E_n. \tag{22}$$

To balance the trade-off among service quality, service delay, and energy consumption, we define the *network utility* as a weighted sum of these factors:

$$U \triangleq \varpi_1 Q - \varpi_2 T - \varpi_3 E, \tag{23}$$

where ϖ_1, ϖ_2, and ϖ_3 are weight factors.

D. Problem Formulation

Based on the above, we can find that the service quality, service delay, and energy consumption are all closely related to user clustering decisions (i.e., $\gamma_{n,m}$), model inferencing decisions (i.e., x_n, y_n, z_n), and resource allocation decisions (i.e., v_m). Therefore, in this work, we focus on the joint optimization of user clustering, model inferencing, and resource allocation problem to optimize the network utility in the HIS. In what follows, we will first study the user clustering problem in Section V, and then study the joint model inferencing and resource allocation problem in Section VI.

V. Optimization of User Clustering

In this section, we study the user clustering problem. As shown in Fig. 5, the quality of the generated image increases with the prompt similarity. Motivated by this, our goal is to group UEs with similar prompts into the same cluster, such that the personal prompt of each UE is well aligned with the public prompt of the corresponding cluster.

Formally, we denote the clustering decisions with a matrix $\mathbf{\Gamma} \triangleq [\gamma_{n,m}]_{n \in \mathcal{N}, m \in \mathcal{M}}$, where $\gamma_{n,m} \in \{0, 1\}$ denotes whether UE n is assigned to cluster m. Let $\mathbf{p}_n^{\text{per}}$ denote the personal prompt of UE n, and $\mathbf{p}_m^{\text{pub}}$ denote the public prompt extracted from all personal prompts in cluster m. Let $\phi_{n,m}$ denote the public-personal prompt similarity between the public prompt $\mathbf{p}_m^{\text{pub}}$ of cluster m and the personal prompt $\mathbf{p}_n^{\text{per}}$ of UE n, which is derived from Eq. (5). It is notable that both the public prompt $\mathbf{p}_m^{\text{pub}}$ and the public-personal prompt similarity $\phi_{n,m}$ are dependent on the clustering decisions $\mathbf{\Gamma}$, and thus can be expressed as $\mathbf{p}_m^{\text{pub}}(\mathbf{\Gamma})$ and $\phi_{n,m}(\mathbf{\Gamma})$, respectively.

Based on the above, we can formulate the user clustering optimization problem as follows:

$$\mathbb{P}_{\text{I}}^{(1)}: \quad \max_{\mathbf{\Gamma}} \quad \sum_{n=1}^{N} \sum_{m=1}^{M} \gamma_{n,m} \phi_{n,m}(\mathbf{\Gamma}) \tag{24a}$$

$$\text{s.t.} \quad \gamma_{n,m} \in \{0, 1\}, \quad \forall n \in \mathcal{N}, \ \forall m \in \mathcal{M} \tag{24b}$$

$$\sum_{m=1}^{M} \gamma_{n,m} = 1, \quad \forall n \in \mathcal{N}. \tag{24c}$$

A. Problem Transformation

It is important to note that directly solving $\mathbb{P}_{\text{I}}^{(1)}$ is very challenging, due to the strong coupling between the public-personal prompt similarity $\phi_{n,m}$ and the clustering decisions $\mathbf{\Gamma}$, as well as the intractability in strictly characterizing this

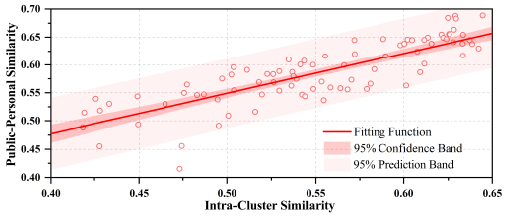

Fig. 8. Relationship between two similarity metrics.

coupling. To deal with this issue, we conduct empirical experiments to analyze the relationship between the public-personal similarity and the intra-cluster similarity.

1) *Intra-Cluster Similarity:* The average similarity among personal prompts within the same cluster.
2) *Public-Personal Similarity:* The average similarity between each personal prompt and its corresponding public prompt (i.e., the average value of $\phi_{n,m}$ in each cluster.).

Fig. 8 presents the experiment result, which reveals a high linear correlation between these two similarity metrics. That is, as the intra-cluster similarity increases, the public-personal similarity also increases linearly, and vice versa.

Let $\hat{\phi}_{n,n'}$ denote the similarity between personal prompts of UEs n and n', i.e., $\mathbf{p}_n^{\text{per}}$ and $\mathbf{p}_{n'}^{\text{per}}$. Clearly, $\hat{\phi}_{n,n'}$ is independent of the clustering decisions $\mathbf{\Gamma}$. Based on the above observation, we can transform $\mathbb{P}_{\text{I}}^{(1)}$ into the following new problem $\mathbb{P}_{\text{I}}^{(2)}$, which aims to maximize the intra-cluster similarity.

$$\mathbb{P}_{\text{I}}^{(2)}: \max_{\mathbf{\Gamma}} \quad \sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}\sum_{n'\in\mathcal{N}}\gamma_{n,m}\gamma_{n',m}\hat{\phi}_{n,n'} \tag{25a}$$
$$\text{s.t.} \quad \text{(24b)} - \text{(24c)},$$

where $\gamma_{n,m}\gamma_{n',m}=1$ indicates that UEs n and n' are in the same cluster m. Note that $\mathbb{P}_{\text{I}}^{(2)}$ is *non-convex* and challenging due to the product of variables $\gamma_{n,m}$ and $\gamma_{n',m}$. To this end, we introduce auxiliary variables $\mathbf{\Lambda} \triangleq [\lambda_{n,n',m}]_{n,n'\in\mathcal{N},m\in\mathcal{M}}$ and transform $\mathbb{P}_{\text{I}}^{(2)}$ into an equivalent problem $\mathbb{P}_{\text{I}}^{(3)}$:

$$\mathbb{P}_{\text{I}}^{(3)}: \max_{\mathbf{\Gamma},\mathbf{\Lambda}} \sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}\sum_{n'\in\mathcal{N}}\lambda_{n,n',m}\hat{\phi}_{n,n'} \tag{26a}$$
$$\text{s.t. (24b)} - \text{(24c)}$$
$$\lambda_{n,n',m}\in\{0,1\}, \quad \forall n,\forall n',\forall m \tag{26b}$$
$$\lambda_{n,n',m}\le\gamma_{n,m}, \quad \forall n,\forall n',\forall m \tag{26c}$$
$$\lambda_{n,n',m}\le\gamma_{n',m}, \quad \forall n,\forall n',\forall m \tag{26d}$$
$$\lambda_{n,n',m}\ge\gamma_{n,m}+\gamma_{n',m}-1, \forall n,\forall n',\forall m \tag{26e}$$

Clearly, $\mathbb{P}_{\text{I}}^{(3)}$ is an integer linear programming problem, widely known to be NP-Hard. While exact solutions can be obtained using algorithms such as branch-and-bound and cutting-plane methods, their computational complexity $\mathcal{O}(2^{N^2M})$ grows exponentially with the problem size, making them unsuitable for large-scale MEC networks.

B. Algorithm Design

To effectively solve $\mathbb{P}_{\text{I}}^{(3)}$, we design an efficient semantic-based clustering algorithm based on the HAC [37], as shown in Algorithm 1. Specifically, we first compute the similarity matric $\hat{\mathbf{\Phi}} \triangleq [\hat{\phi}_{n,n'}]_{n,n'\in\mathcal{N}}$ using the lightweight Sentence-Transformer model. Then, the algorithm begins by treating each UE (personal prompt) as an individual cluster. In each

Algorithm 1: Semantic-Based Clustering Algorithm

```
Initialize each personal prompt as a cluster;
Calculate the similarity matric Φ̂;
for each iteration k = 1, ··· , N − M do
    Calculate all cluster-wise similarities based on (27);
    Merge two clusters with the highest similarity;
```

iteration, the algorithm merges the two clusters with the highest cluster-wise similarity according to:

$$\ell(m,m') = \frac{1}{|\mathcal{N}_m||\mathcal{N}_{m'}|}\sum_{n\in\mathcal{N}_m}\sum_{n'\in\mathcal{N}_{m'}}\phi_{n,n'}, \tag{27}$$

This process continues until a predefined cluster number M is reached. Intuitively, this algorithm enhances intra-cluster similarity by iteratively merging the most similar clusters. With a complexity of $\mathcal{O}(N^2\log N)$, it provides a powerful solution to $\mathbb{P}_{\text{I}}^{(3)}$ that can be executed efficiently on the ES.

VI. Optimization of Model Inferencing and Resource Allocation

Based on the user clustering results obtained from Algorithm 1, we study the joint model inferencing and resource allocation problem for all clusters in this section. For notational convenience, we introduce the following notations: $\mathbf{X} \triangleq [x_n]_{n\in\mathcal{N}}$, $\mathbf{Y} \triangleq [y_n]_{n\in\mathcal{N}}$, $\mathbf{Z} \triangleq [z_n]_{n\in\mathcal{N}}$, and $\mathbf{V} \triangleq [v_m]_{m\in\mathcal{M}}$. Then, the joint model inferencing and resource allocation optimization problem can be formulated as:

$$\mathbb{P}_{\text{II}}: \max_{\mathbf{X},\mathbf{Y},\mathbf{Z},\mathbf{V}} \quad U \triangleq \varpi_1 Q - \varpi_2 T - \varpi_3 E \tag{28a}$$
$$\text{s.t.} \quad x_n\in\{0,\cdots,x_{\max}\}, \quad \forall n\in\mathcal{N} \tag{28b}$$
$$y_n\in\{y_{\min},\cdots,y_{\max}\}, \quad \forall n\in\mathcal{N} \tag{28c}$$
$$z_n\in\{z_{\min},\cdots,z_{\max}\}, \quad \forall n\in\mathcal{N} \tag{28d}$$
$$\sum_{m=1}^{M} v_m^{\text{ES}} \le v_{\max}^{\text{ES}}, \tag{28e}$$
$$x_n+y_n+z_n = T_{\max}, \quad \forall n\in\mathcal{N}. \tag{28f}$$

Note that constraint (28f) sets the total number of inference steps as a pre-determined constant. The value of $T_{\max}$ is scheduler-specific and represents the inference steps required to achieve satisfactory results [21].

A. Problem Decomposition

Directly optimizing $\mathbb{P}_{\text{II}}$ is challenging due to the heterogeneous decision variables and the asynchronous decision-making process. To streamline the optimization process, we decompose $\mathbb{P}_{\text{II}}$ into a *user-wise* model inferencing problem $\mathbb{P}_{\text{II}}^{\text{A}}$ and a *cluster-wise* resource allocation problem $\mathbb{P}_{\text{II}}^{\text{B}}$:

$$\mathbb{P}_{\text{II}}^{\text{A}}: \max_{\mathbf{X},\mathbf{Y},\mathbf{Z}} \quad U \qquad \text{s.t. (28b), (28c), (28d), (28f).}$$

$$\mathbb{P}_{\text{II}}^{\text{B}}: \max_{\mathbf{V}} \quad U \qquad \text{s.t. (28e).}$$

To solve the above two sub-problems, we propose a two-tier Model Inferencing and Resource Allocation (MIRA) algorithm. In the first tier, we design a *learning-based* approach to make real-time model inferencing decisions for UEs. In the seconde tier, we employ *optimization-based* techniques to make optimal resource allocation decisions for clusters. Next, we will provide the detailed algorithm designs.

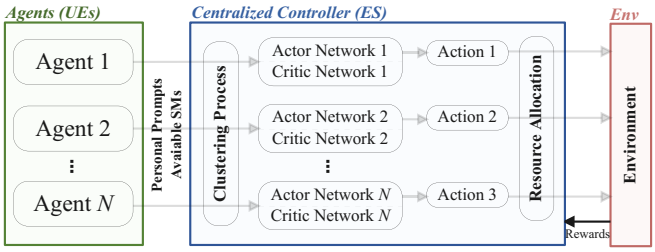

Fig. 9. Decision process of the learning-based model inferencing algorithm.

B. Learning-Based Model Inferencing

In the first tier, we utilize the DRL algorithm to solve sub-problem $\mathbb{P}_{\text{II}}^{\text{A}}$. Note that the state-action space grows exponentially with the number of agents (i.e., UEs). This curse of dimensionality poses challenges to the centralized execution scheme. On the other hand, since parts of the inference process are performed on the ES, the decision of each agent needs to be uploaded to the ES. This interaction process introduces unnecessary communication overhead, posing challenges to the traditional decentralized execution scheme.

To address the above challenges, we employ the IPPO approach [38] and introduce the centralized control for the multi-agent system. Specifically, the ES acts as a centralized controller, hosting the actor and critic networks of all agents (UEs). The actor network is responsible for selecting the action to be performed, while the critic network is used to evaluate the action by estimating the expected future rewards. Due to space limit, we put the detailed algorithm in [41].

Next, we define the observation, action, and reward.

1) Observation: The observation consists of two parts: resource status and prompt similarity. Here, we define the observation of UE n at decision point i as:

$$\mathbf{o}_n^{(i)} = (\phi_{n,m}^{(i)}, v_{\max}^{\text{ES},(i)}, v_n^{\text{UE},(i)}). \tag{31}$$

2) Action: The action consists of two parts: the number of common and personalized inference steps. Here, we define the action of UE n at decision point i as:

$$\mathbf{a}_n^{(i)} = (x_n^{(i)}, y_n^{(i)}). \tag{32}$$

3) Reward: To precisely capture the reward improvement resulting from different actions, the reward of UE n at decision point i is defined as:

$$\begin{aligned} r_n^{(i)} &= Q_n + T_n + E_n \\ &+ \textstyle\sum_{m=1}^{M} \gamma_{n,m} \left(\frac{x_n}{\sum_{n\in\mathcal{N}_m} x_n} E_m^{\text{C}} + \frac{y_n}{\sum_{n\in\mathcal{N}_m} y_n} E_m^{\text{P-E}} \right). \end{aligned} \tag{33}$$

Thus, we have $U^{(i)} = \sum_{n=1}^{N} r_n^{(i)}$. To improve stability and accelerate convergence, we introduce reward normalization to adjust the scale of rewards. Specifically, reward normalization dynamically maintains the mean and standard deviation of all observed rewards for each agent, normalizing the current reward by subtracting the mean and dividing by the standard deviation. This ensures that rewards are centered around zero with unit variance, mitigating the negative effects of large or small rewards on the training process.

As shown in Fig. 9, at each decision point, the ES obtains individual observation $\mathbf{o}_n^{(i)}$ and chooses corresponding action $\mathbf{a}_n^{(i)}$ for all UEs in a parallel way. Once all content has been generated, the ES obtains reward and next individual observations, and stores the experiences in corresponding replay buffers. The actor network can be updated according to the following loss function:

$$\begin{aligned} \mathcal{L}_n^{\text{A}_1} = \arg\max_{\boldsymbol{\Theta}_n} \Bigg\{ \tfrac{1}{DI} \textstyle\sum_{d=1}^{D} \sum_{i=1}^{I} \min \Bigg\{ \frac{\pi_{\boldsymbol{\Theta}_n}(\mathbf{a}_n^{(d,i)}|\mathbf{o}_n^{(d,i)})}{\pi_{\boldsymbol{\Theta}_n^{\text{old}}}(\mathbf{a}_n^{(d,i)}|\mathbf{o}_n^{(d,i)})} \hat{A}_n^{(d,i)}, \text{trun}\left(\frac{\pi_{\boldsymbol{\Theta}_n}(\mathbf{a}_n^{(d,i)}|\mathbf{o}_n^{(d,i)})}{\pi_{\boldsymbol{\Theta}_n^{\text{old}}}(\mathbf{a}_n^{(d,i)}|\mathbf{o}_n^{(d,i)})} \hat{A}_n^{(d,i)}, 1-\varepsilon, 1+\varepsilon \right) \hat{A}_n^{(d,i)} \Bigg\} \Bigg\}, \end{aligned} \tag{34}$$

where $\pi_{\boldsymbol{\Theta}_n}$ is the actor network for UE n, D is the batch size, ε is the coefficient of the truncation function $\text{trun}(\cdot)$, and $\hat{A}_n^{(d,i)}$ is the generalized advantage estimation [39].

To improve the algorithm performance, we further introduce batch advantage normalization [40] to normalize the advantages by subtracting the mean and dividing by the standard deviation of the entire batch. Furthermore, to balance exploration and exploitation, we introduce the policy entropy term for the loss function, which can be defined as:

$$\mathcal{L}_n^{\text{A}_2} = \arg\max_{\boldsymbol{\Theta}_n} \left\{ \tfrac{1}{DI} \textstyle\sum_{d=1}^{D} \sum_{i=1}^{I} \mathbb{E}_{\mathbf{a}} \left[-\log \pi_{\boldsymbol{\Theta}_n}(\mathbf{a}|\mathbf{o}_n^{(d,i)}) \right] \right\}. \tag{35}$$

Thus, the loss function of the actor network is as follows:

$$\mathcal{L}_n^{\text{A}} = \mathcal{L}_n^{\text{A}_1} + \eta \mathcal{L}_n^{\text{A}_2}, \tag{36}$$

where η is a coefficient controling the strength of the entropy regularization. Finally, the critic network can be updated according to the following loss function:

$$\begin{aligned} \mathcal{L}_n^{\text{C}} = \arg\min_{\boldsymbol{\Phi}_n} \Bigg\{ \tfrac{1}{DI} \textstyle\sum_{d=1}^{D} \sum_{i=1}^{I} \max \Bigg\{ \left(V_{\boldsymbol{\Phi}_n}(\mathbf{o}_n^{(d,i)}) - \hat{R}_n^{(d,i)} \right)^2, \left(\text{trun}\left(V_{\boldsymbol{\Phi}_n}(\mathbf{o}_n^{(d,i)}), V_{\boldsymbol{\Phi}_n^{\text{old}}}(\mathbf{o}_n^{(d,i)}) - \epsilon, \right. \right. \\ \left. \left. V_{\boldsymbol{\Phi}_n^{\text{old}}}(\mathbf{o}_n^{(d,i)}) + \epsilon \right) - \hat{R}_n^{(d,i)} \right)^2 \Bigg\} \Bigg\}, \end{aligned} \tag{37}$$

where $V_{\boldsymbol{\Phi}_n}$ is the critic network for UE n, ϵ is the coefficient of the truncation function, and $\hat{R}_n^{(d,i)}$ is the reward-to-go.

C. Optimization-Based Resource Allocation

Once the model inferencing decisions (i.e., $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$) are determined, the sub-problem $\mathbb{P}_{\text{II}}^{\text{B}}$ can be written as:

$$\mathbb{P}_{\text{II}}^{\text{B}}: \quad \max_{\mathbf{v}} \quad \textstyle\sum_{m=1}^{M} \left(\frac{\alpha_m^{(1)} + \alpha_m^{(2)}}{v_m^{\text{ES}}} \right) \quad \text{s.t.} \quad (28\text{e}),$$

where $\alpha_m^{(1)} = \sum_{n=1}^{N} \frac{\left(x_m^{\max} + \sum_{n\in\mathcal{N}_m} y_n\right)\gamma_{n,m}\varpi_2\xi^{\text{ES}}}{F^{\text{ES}} D^{\text{ES}} \rho^{\text{ES}}}$ and $\alpha_m^{(2)} = \frac{\left(x_m^{\max} + \sum_{n\in\mathcal{N}_m} y_n\right)\varpi_3\kappa^{\text{ES}}\xi^{\text{ES}}(F^{\text{ES}})^2\mu_2^{\text{ES}}}{D^{\text{ES}}\rho^{\text{ES}}}$.

Next, we provide the closed-form solution to $\mathbb{P}_{\text{II}}^{\text{B}}$ by relaxing the decision variables.

Theorem 1. *The relaxed sub-problem $\mathbb{P}_{II}^{B}$ is convex, and its optimal solution $(v_m^{ES})^\star$ is given by*[6]*:*

$$(v_m^{ES})^\star = v_{\max}^{ES} \frac{\sqrt{\alpha_m^{(1)} + \alpha_m^{(2)}}}{\sum_{m\in\mathcal{M}} \sqrt{\alpha_m^{(1)} + \alpha_m^{(2)}}}, \quad \forall m \in \mathcal{M}. \tag{39}$$

Proof. The detailed proof is provided in [41]. □

[6]In practice, $(v_m^{\text{ES}})^\star$ can be rounded down with minimal performance loss.

VII. Performance Evaluation

A. Experiments Setting

In the simulations, we consider a wireless network consisting of one ES and 50 UEs randomly distributed within a radius of 50 to 300 meters from the ES. To simulate diverse generation requirements, the UEs are grouped into three distinct clusters. To support advanced AIGC services, both the ES and UEs deploy the state-of-the-art Stable Diffusion 3 Medium model [34], which is configured with the Flow-Match-Euler-Discrete-Scheduler to ensure high-quality image generation. The total number of inference steps is fixed at 28 inference steps (i.e., $T_{\max} = 28$) as suggested in [34]. For the hardware setting, the ES is equipped with an industrial-grade NVIDIA A100 GPU cluster, with each GPU featuring 108 SMs for intensive processing. Meanwhile, each UE is equipped with a consumer-grade RTX 3090 Ti GPU, each with 84 SMs for real-time processing. Table II lists the key parameters.

TABLE II
Default Parameter Settings

Parameter	Value
N, M, $x_{\max}$, $y_{\min}$, $y_{\max}$	50, 3, 16, 0, 11
ξ^{ES}, $v_{\max}^{\mathrm{ES}}$, F^{ES}	12 TFLOPs, $[1026, 1674]$, 1410 MHz
D^{ES}, κ^{ES}, μ_1^{ES}, μ_2^{ES}	2048, 10^{-27}, 4, 135
ξ^{UE}, v_n^{UE}, F^{UE}	12 TFLOPs, $[64, 84]$, 1860 MHz
D^{UE}, κ^{UE}, μ_1^{UE}, μ_2^{UE}	1024, 10^{-27}, 4, 135
n_{h}, n_{w}, n_{c}, ϱ, τ	128, 128, 16, 16, 2
learning rate, hidden layers	0.005, $[256, 256, 256]$
optimizer, D, ε, ϵ, η	Adam, 128, 0.2, 0.2, 0.002

B. The Effectiveness and Robustness of the HIS

In this subsection, we present simulation results to demonstrate the effectiveness and robustness of our proposed HIS. For comparisons, we introduce the following three inference schemes from existing literatures as benchmarks:

- *Public Prompt Generation-Free Inference Scheme (PPGFIS)* [15]: This scheme employs the knowledge graph for clustering. In each cluster, a randomly selected personal prompt can be designated as the public prompt.
- *Clustering-Free Inference Scheme (CFIS)* [12], [14]: This scheme treats all UEs as a single cluster, from which a personal prompt is randomly selected as the public one.
- *Independent Inference Scheme (IIS)* [25]: This scheme does not introduce the common inference phase.

1) Case Study: Sample images generated by three collaborative inference schemes (i.e., HIS, PPGFIS, and CFIS) are shown in the supplementary [41]. Among them, the HIS shows superior performance by maintaining high perceptual quality even with large common inference steps. This effectively overcomes a key limitation reported in prior work [15], where the number of common inference steps is typically restricted to one-third of total number of inference steps.

2) HIS Performance: A detailed numerical comparison highlighting the performance of the HIS is presented in Fig. 10. Compared to the PPGFIS, CFIS, and IIS, the HIS achieves a significantly better trade-off among service quality, service delay, and energy consumption. These improvements translate into network utility improvements of 36.8%, 51.0%, and 94.3%, respectively. Notably, the HIS achieves superior service quality while simultaneously reducing both service delay and energy consumption when compared to the PPGFIS and CFIS. This advantage stems from its efficient utilization of the common inference phase, which allows larger common inference steps without degrading the quality of the generated content. As a result, the overall inference workload is reduced, enabling faster response times and lower power consumption across the network.

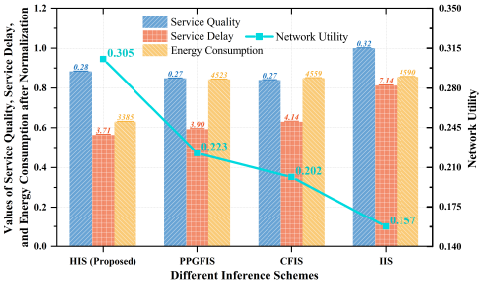

Fig. 10. Values of service quality, service delay, energy consumption, and network utility for several inference schemes.

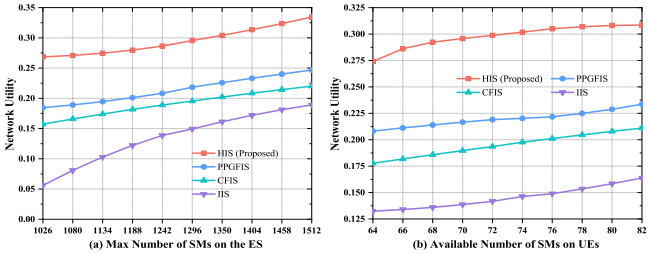

Fig. 11. Service robustness of several inference schemes to computational resources. (a) Network utility under varying maximum number of SMs on the ES. (b) Network utility under varying available number of SMs on UEs.

3) Service Robustness: The service robustness of various inference schemes is evaluated by analyzing their network utility under varying levels of computation resources. As shown in Fig. 11, the HIS consistently delivers higher network utility than all benchmark schemes across a broad range of available SMs on both the ES and UEs. When the computational capacity on the ES fluctuates, the HIS exhibits only a moderate utility degradation of 18.7%, in contrast to the more substantial drops observed in the PPGFIS (23.1%), CFIS (29.0%), and IIS (74.1%). A similar trend is observed when UE resources vary: the HIS maintains a low utility reduction of 9.0%, outperforming the PPGFIS (11.6%), CFIS (15.3%), and IIS (14.9%). Notably, the HIS also requires fewer computational resources to achieve the same utility level as its counterparts. These results highlight strong adaptability of the HIS to dynamic environments, allowing it to sustain high-quality services even under resource constraints.

4) Ablation Study: Fig. 12 illustrates the quantitative impact of two key modules in the HIS: the SIM and NPI. The empirical results reveal that enabling both modules leads to a 19.1% improvement in overall network utility. This performance gain can be better understood by referring to the quality transition curves shown in Fig. 5. Specifically, the SIM and NPI work in tandem to flatten the quality degradation curve, effectively reducing the sensitivity of service quality to variations in the common inference steps. As a result, the HIS is able to

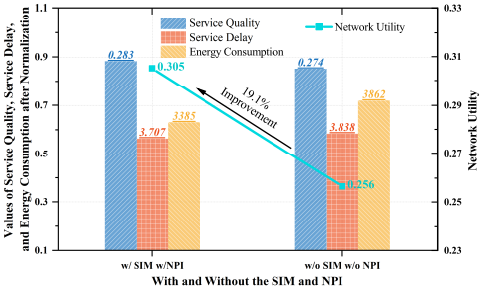

Fig. 12. Values of service quality, service delay, energy consumption, and network utility with and without the SIM and NPI in the HIM.

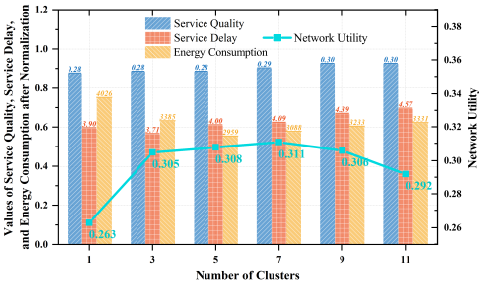

Fig. 13. Value of service quality, service delay, energy consumption, and network utility under different number of clusters.

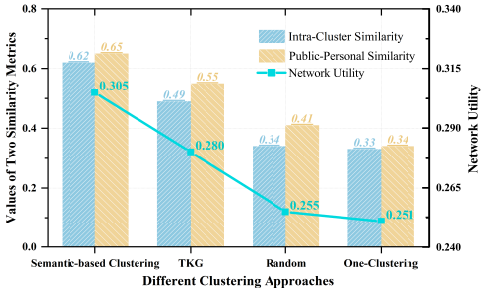

Fig. 14. Values of intra-cluster similarity, public-personal similarity, and network utility using different clustering strategies.

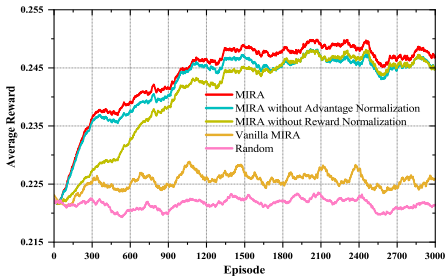

Fig. 15. Reward curves of different model inferencing algorithms. All algorithms are based on the optimal resource allocation.

complete personalized inference with fewer total steps, leading to notable reductions in resource consumption.

5) Network Setting: Fig. 13 presents the impact of the number of clusters. With fewer clusters, lower public-personal similarity necessitates a smaller number of common inference steps to maintain service quality. As a result, the total number of executed steps increases, thus exacerbating service delay and energy consumption. As the number of clusters increases, more similar personal prompts are grouped together, leading to an increase in public-personal similarity. This allows for a larger number of common inference steps, thereby improving network utility. However, excessive number of clusters lead to diminishing returns in public-personal similarity and excessive computational overhead, ultimately degrading network utility.

C. The Performance and Generalization of the Algorithms

In this subsection, we demonstrate the performance and generalization of our proposed algorithms.

1) Performance of Clustering: Fig. 14 compares the performance of various clustering approaches using two key evaluation metrics: intra-cluster similarity and public-personal similarity. Among them, our proposed semantic-based clustering approach demonstrates the best overall performance, achieving a public-personal similarity score of 0.65. In contrast, the Task Knowledge Graph (TKG)-based strategy used in the PPGFIS prioritizes entity-level consistency but fails to account for deeper semantic relationships across prompts, resulting in notable degradation in network utility. Furthermore, baseline methods such as random clustering and one-clustering lack a principled clustering mechanism. As a result, the public prompts generated under these methods fail to represent the semantic diversity of user inputs, severely limiting their effectiveness in guiding the common inference.

2) Performance of Model Inferencing: Fig. 15 presents the impact of various techniques incorporated into the model inferencing algorithm. The absence of advantage normalization leads to higher variance in advantage estimates, which amplifies noise and results in degrading performance. Similarly, without reward normalization, the algorithm struggles to balance reward values, leading to slower convergence. Vanilla MIRA, which does not incorporate normalization and loss truncation, tends to converge to suboptimal solutions. This occurs as the lack of these techniques leads the algorithm to overfit noisy gradients, thereby diminishing its ability to generalize effectively and converge to an optimal solution.

3) Performance of Resource Allocation: We evaluate the effectiveness of our proposed resource allocation strategy in MIRA under scenarios characterized by highly imbalanced user intentions, where the number of UEs varies substantially across clusters. To understand the performance impact of allocation granularity, we compare our method against two baseline strategies. In the user-wise equal allocation scheme, the computational resources on the ES are evenly distributed across all UEs. Conversely, the cluster-wise equal allocation scheme assigns equal resources to each cluster, without accounting for the number of UEs per cluster. As shown in Fig. 16, our proposed optimal allocation strategy dynamically adapts to cluster-level demands, significantly improving network performance. Specifically, it achieves the lowest service delay and energy consumption, leading to the highest network utility value of 0.26.

4) Generalization: As illustrated in Fig. 17, as the weight factor of service quality increases, the values of service quality, service delay, and energy consumption also rise, indicating that our proposed MIRA algorithm tends to achieve higher service quality with smaller common inference steps. This trends reveals that MIRA algorithm is adaptable to both quality-sensitive and delay-energy-sensitive MEC networks, exhibiting strong generalization capabilities.

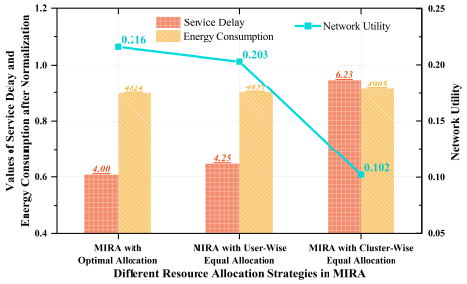

Fig. 16. Values of service delay, energy consumption, and network utility using different resource allocation strategies.

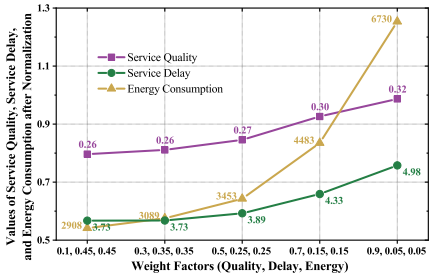

Fig. 17. Values of service quality, service delay, and energy consumption under different weight factors.

VIII. Conclusion

In this work, we designed an efficient inference scheme, the HIS, for MEC-enabled AIGC services. First, in such a scheme, we developed two advanced modules, the SIM and NPI, to improve the inference efficiency, and introduced a performance metric, the HIQM, to quantify the service quality. Second, we formulated the joint optimization problem of user clustering, model inferencing, and resource allocation. We further proposed the semantic-based clustering along with the integrated learning and optimization approach for effective problem solving. Finally, simulation results show that our proposed approach significantly outperforms existing benchmarks, with the performance improvements ranging from 36.8% to 94.3%.

References

[1] X. Zhuang, J. Wu, H. Wu, M. Tang, and L. Gao, "QoS-driven hybrid inference scheme for generative diffusion models in MEC-enabled AI-generated content networks," in *Proc. IEEE ICC*, Canada, Jun. 2025.

[2] F. A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10850-10869, Sep. 2023.

[3] A. Hu *et al.*, "GAIA-1: A generative world model for autonomous driving," 2023, *arXiv:2309.17080*.

[4] I. Kapelyukh, V. Vosylius, and E. Johns, "Dall-e-bot: Introducing web-scale diffusion models to robotics," *IEEE Robot. Autom. Lett.*, vol. 8, no. 7, pp. 3956-3963, Jul. 2023.

[5] M. Xu *et al.*, "Unleashing the power of edge-cloud generative AI in mobile networks: A survey of AIGC services," *IEEE Commun. Surv. Tutor.*, vol. 26, no. 2, pp. 1127-1170, Secondquarter 2024.

[6] Y. Song *et al.*, "Score-based generative modeling through stochastic differential equations," in *Proc. ICLR*, Online, May 2021.

[7] Z. Yan *et al.*, "GPT-ImgEval: A comprehensive benchmark for diagnosing GPT4o in image generation," 2025, *arXiv:2504.02782*.

[8] Y. Balaji *et al.*, "eDiff-I: Text-to-image diffusion models with an ensemble of expert denoisers," 2023, *arXiv:2211.01324*.

[9] S. Yang *et al.*, "Denoising diffusion step-aware models," in *Proc. ICLR*, Austria, May 2024.

[10] E. Liu *et al.*, "OMS-DPM: Optimizing the model schedule for diffusion probabilistic models," in *Proc. ICML*, USA, Jul. 2023.

[11] C. Yan *et al.*, "Hybrid SD: Edge-cloud collaborative inference for stable diffusion models," 2024, *arXiv:2408.06646*.

[12] W. Yang *et al.*, "Efficient multi-user offloading of personalized diffusion models: A DRL-convex hybrid solution," 2024, *arXiv:2411.15781*.

[13] H. Du *et al.*, "Exploring collaborative distributed diffusion-based AI-generated content (AIGC) in wireless networks," *IEEE Network*, vol. 38, no. 3, pp. 178-186, May 2024.

[14] H. Du *et al.*, "User-centric interactive AI for distributed diffusion model-based AI-generated content," 2023, *arXiv:2311.11094*.

[15] G. Xie *et al.*, "GAI-IoV: Bridging generative AI and vehicular networks for ubiquitous edge intelligence," *IEEE Trans. Wireless Commun.*, vol. 23, no. 10, pp. 12799-12814, Oct. 2024.

[16] H. Liu *et al.*, "Joint communication and computation scheduling for MEC-enabled AIGC services based on generative diffusion model," in *Proc. WiOpt*, Korea, Oct. 2024.

[17] Y. Wang, C. Liu, and J. Zhao, "Offloading and quality control for AI generated content services in 6G mobile edge computing networks," in *Proc. IEEE VTC*, Singapore, Jun. 2024.

[18] Z. Liu *et al.*, "DNN partitioning, task offloading, and resource allocation in dynamic vehicular networks: A Lyapunov-guided diffusion-based reinforcement learning approach," *IEEE Trans. Mobile Comput.*, vol. 24, no. 3, pp. 1945-1962, Mar. 2025.

[19] Y. Li *et al.*, "Diffusion-enabled digital twin synchronization for AIGC services in space-air-ground integrated networks," *IEEE Internet Things J.*, Early Access, 2024.

[20] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. ICLR*, Online, May 2021.

[21] C. Li *et al.*, "Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps," in *Proc. NeurIPS*, USA, Nov. 2022.

[22] T. Salimans and J. Ho, "Progressive distillation for fast sampling of diffusion models," in *Proc. ICLR*, Online, Apr. 2022.

[23] X. Li *et al.*, "Q-diffusion: Quantizing diffusion models," in *Proc. IEEE ICCV*, France, Oct. 2023.

[24] R. Rombach *et al.*, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE CVPR*, USA, Jun. 2022.

[25] H. Du *et al.*, "Diffusion-based reinforcement learning for edge-enabled AI-generated content services," *IEEE Trans. Mobile Comput.*, vol. 23, no. 9, pp. 8902-8918, Sep. 2024.

[26] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *Proc. NeurIPS Workshop*, Online, Dec. 2021.

[27] S. Sadat *et al.*, "CADS: Unleashing the diversity of diffusion models through condition-annealed sampling," in *Proc. ICLR*, Austria, May 2024.

[28] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. ICML*, Online, 2021.

[29] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4695-4708, Dec. 2012.

[30] S. Sadat, O. Hilliges, and R. M. Weber, "Eliminating oversaturation and atrifacts of high guidance scales in diffusion models," 2024, *arXiv:2410.02416*.

[31] Y. Ban *et al.*, "Understanding the impact of negative prompts: When and how do they take effect?" 2024, *arXiv: 2406.02965*.

[32] A. Grattafiori *et al.*, "The Llama 3 herd of models," 2024, *arXiv:2407.21783*.

[33] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proc. EMNLP*, China, Nov. 2019.

[34] Stability AI. "stable-diffusion-3-medium." huggingface.io. Accessed: Feb. 1, 2025. [Online.] Available: https://huggingface.co/stabilityai/stable-diffusion-3-medium/tree/main

[35] J. T. Adriaens, K. Compton, N. S. Kim, and M. J. Schulte, "The case for GPGPU spatial multitasking," in *Proc. IEEE HPCA*, USA, Feb. 2012.

[36] V. Kandiah *et al.*, "AccelWattch: A power modeling framework for modern GPUs," in *Proc. IEEE/ACM MICRO*, Online, 2021.

[37] Y. Shen *et al.*, "Improving medical short text classification with semantic expansion using word-cluster embedding," in *Proc. ICISA*, China, Jun. 2018.

[38] C. S. D. Witt *et al.*, "Is independent learning all you need in the StarCraft multi-agent challenge?," 2020, *arXiv:2011.09533*.

[39] J. Schulman *et al.*, "High-dimensional continuous control using generalized advantage estimation," 2015, *arXiv:1506.02438*.

[40] G. Tucker *et al.*, "The mirage of action-dependent baselines in reinforcement learning," in *Proc. ICML*, Sweden, Jul. 2018.

[41] https://github.com/iimxinyi/HIS/blob/main/Supplementary/Supplementary.pdf

SUPPLEMENTARY

A. MIRA Algorithm

Based on the above, we summarize the MIRA algorithm in Algorithm 2. At each decision point, each UE uploads its personal prompt to the ES, after which the ES groups UEs into clusters according to Algorithm 1. Then, the ES collects individual observations and chooses corresponding actions for all UEs (lines 6-7). Next, the ES decides computational resource allocation based on Eq. (39) and performs the common inference phase and personalized inference phase I based on all actions (lines 8-9). After wireless transmission, the personalized inference phase II is performed on the UE based on Eq. (28f). Once all content has generated, the ES stores the experiences in corresponding replay buffers (lines 10-11). Finally, the actor and critic networks can be updated by Eq. (36) and Eq. (37) (lines 14-15).

Algorithm 2: MIRA Algorithm

1 Initialize actor networks $\{\pi_{\mathbf{\Theta}_n}\}_{n\in\mathcal{N}}$, critic networks $\{V_{\mathbf{\Phi}_n}\}_{n\in\mathcal{N}}$, replay buffers $\{\mathcal{D}_n\}_{n\in\mathcal{N}}$, and learning parameters;
2 **for** each episode $e = 1, \cdots, E$ **do**
3 **for** each decision point $i = 1, \cdots, I$ **do**
4 Each UE n uploads its personal prompt and resource status to the ES;
5 The ES groups UEs into clusters according to Algorithm 1;
6 The ES collects observations $\{\mathbf{o}_n^{(i)}\}_{n\in\mathcal{N}}$ for UEs;
7 The ES selects actions $\{\mathbf{a}_n^{(i)}\}_{n\in\mathcal{N}}$ for UEs based on actor networks $\{\pi_{\mathbf{\Theta}_n}(\mathbf{a}_n^{(i)} \mid \mathbf{o}_n^{(i)})\}_{n\in\mathcal{N}}$;
8 The ES optimizes computational resource allocation $\{v_m^{\text{ES}}\}_{m\in\mathcal{M}}$ by (39);
9 Execute actions for all UEs simultaneously;
10 The ES obtains rewards $\{r_n^{(i)}\}_{n\in\mathcal{N}}$ and collects new obervations $\{\mathbf{o}_n^{(i+1)}\}_{n\in\mathcal{N}}$ for UEs;
11 The ES stores experiences $\{(\mathbf{o}_n^{(i)}, \mathbf{a}_n^{(i)}, r^{(i)}, \mathbf{o}_n^{(i+1)})\}_{n\in\mathcal{N}}$ in replay buffers $\{\mathcal{D}_n\}_{n\in\mathcal{N}}$;
12 **for** each training step **do**
13 **for** each UE $n = 1, \cdots, N$ **do**
14 Update actor network $\pi_{\mathbf{\Theta}_n}$ by maximizing (36);
15 Update critic network $V_{\mathbf{\Phi}_n}$ by minimizing (37).

B. Proof of Theorem 1

The Lagrange function of sub-problem $\mathbb{P}_{\text{II}}^{\text{B}}$ can be defined as:

$$\mathcal{L}(\mathbf{V}, \beta) = \sum_{m=1}^{M} \left(\frac{\alpha_m^{(1)} + \alpha_m^{(2)}}{v_m^{\text{ES}}} \right) + \beta \left(\sum_{m=1}^{M} v_m^{\text{ES}} - v_{\max}^{\text{ES}} \right), \tag{40}$$

where β is the Lagrange multiplier.

The Karush-Kuhn-Tucker (KKT) conditions are as follows:

· Stationarity:

$$\frac{\partial \mathcal{L}}{\partial v_m^{\text{ES}}} = -\frac{\alpha_m^{(1)} + \alpha_m^{(2)}}{(v_m^{\text{ES}})^2} + \beta = 0, \quad \forall m \in \mathcal{M}. \tag{41}$$

· Complementary Slackness:

$$\beta \left(\sum_{m=1}^{M} v_m^{\text{ES}} - v_{\max}^{\text{ES}} \right) = 0. \tag{42}$$

· Primal Feasibility:

$$\sum_{m=1}^{M} v_m^{\text{ES}} \leq v_{\max}^{\text{ES}}. \tag{43}$$

· Dual Feasibility: $\beta \geq 0$.

According to Eq. (41), we have:

$$v_m^{\text{ES}} = \sqrt{\frac{\alpha_m^{(1)} + \alpha_m^{(2)}}{\beta}}, \quad \forall m \in \mathcal{M}. \tag{44}$$

Considering Eq. (43), we have:

$$\beta = \left(\frac{\sum_{m=1}^{M} \sqrt{\alpha_m^{(1)} + \alpha_m^{(2)}}}{v_{\max}^{\mathrm{ES}}} \right)^2. \tag{45}$$

Based on Eq. (44) and Eq. (45), we have:

$$(v_m^{\mathrm{ES}})^\star = v_{\max}^{\mathrm{ES}} \frac{\sqrt{\alpha_m^{(1)} + \alpha_m^{(2)}}}{\sum_{m \in \mathcal{M}} \sqrt{\alpha_m^{(1)} + \alpha_m^{(2)}}}, \quad \forall m \in \mathcal{M}. \tag{46}$$

C. Performace Evaluation (Case Study)

Fig. 18 presents sample images generated by three collaborative inference schemes: the HIS, PPGFIS, and CFIS. Among them, the HIS demonstrates superior performance by maintaining high perceptual quality even when the number of common inference steps is set to a large value. This effectively overcomes a key limitation reported in prior work [15], where the common inference steps is typically restricted to one-third of total number of inference steps (i.e., 8 out of 28). In contrast, both the PPGFIS and CFIS suffer from noticeable semantic drift, with significant content mismatches arising at step 8 and step 6, respectively. These inconsistencies suggest an inability to maintain semantic alignment across phases of the inference process.

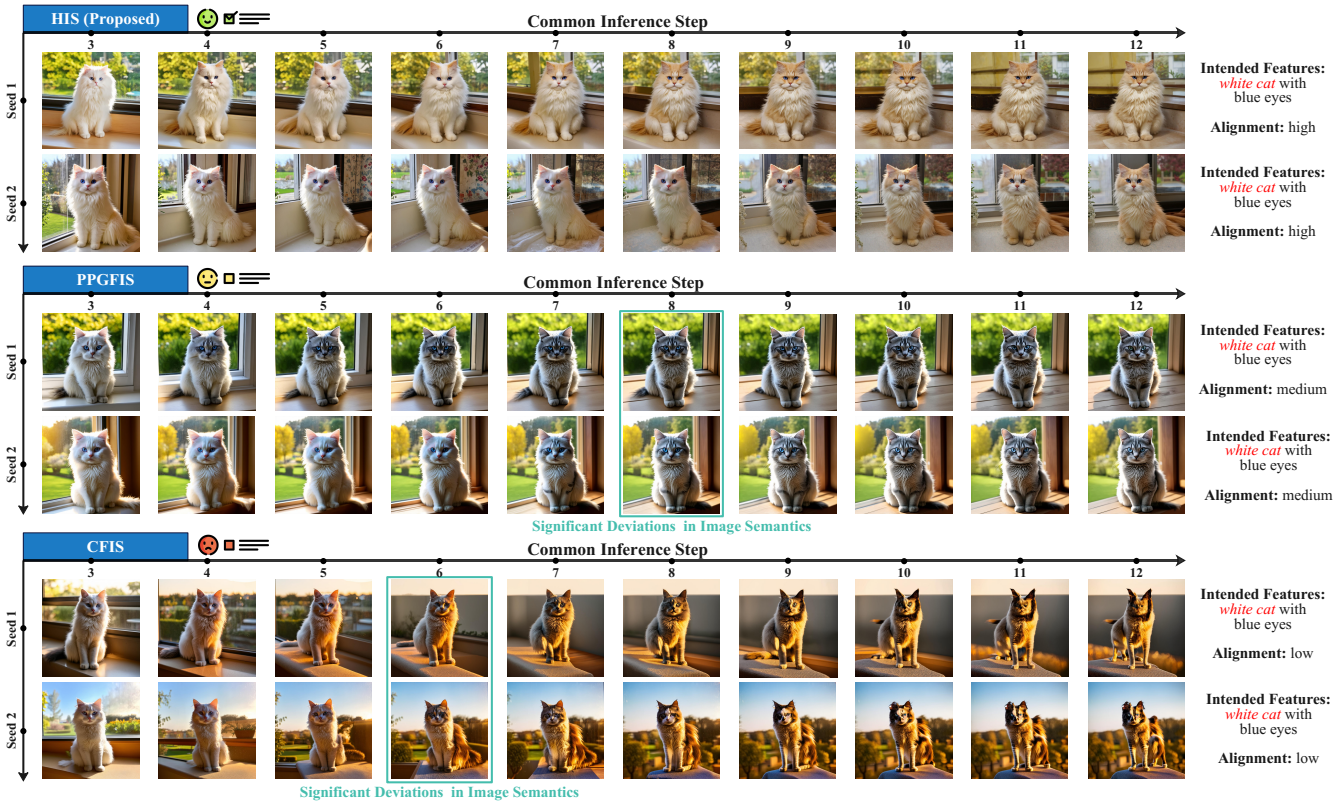

Fig. 18. Images generated by the HIS (**Top**), PPGFIS (**Middle**), and GFIS (**Bottom**) under different common inference steps. All images are based on the same personal prompt: *A fluffy white cat with blue eyes sitting gracefully on a windowsill.*