ObCLIP: Oblivious CLoud-Device Hybrid Image Generation with Privacy Preservation

Haoqi Wu[1,*], **Wei Dai**[1], **Ming Xu**[2], **Li Wang**[1], **Qiang Yan**[1]
[1]TikTok Inc., [2]National University of Singapore

Abstract

Diffusion Models have gained significant popularity due to their remarkable capabilities in image generation, albeit at the cost of intensive computation requirement. Meanwhile, despite their widespread deployment in inference services such as Midjourney, concerns about the potential leakage of sensitive information in uploaded user prompts have arisen. Existing solutions either lack rigorous privacy guarantees or fail to strike an effective balance between utility and efficiency. To bridge this gap, we propose ObCLIP, a plug-and-play safeguard that enables *oblivious* cloud-device hybrid generation. By *oblivious*, each input prompt is transformed into a set of semantically similar candidate prompts that differ only in sensitive attributes (e.g., gender, ethnicity). The cloud server processes all candidate prompts without knowing which one is the real one, thus preventing any prompt leakage. To mitigate server cost, only a small portion of denoising steps is performed upon the large cloud model. The intermediate latents are then sent back to the client, which selects the targeted latent and completes the remaining denoising using a small device model. Additionally, we analyze and incorporate several cache-based accelerations that leverage temporal and batch redundancy, effectively reducing computation cost with minimal utility degradation. Extensive experiments across multiple datasets demonstrate that ObCLIP provides rigorous privacy and comparable utility to cloud models with slightly increased server cost.

1 Introduction

Stable diffusion models [35, 34] have emerged as a de-facto standard technique in text-to-image (T2I) generation due to their superior capability to generate high-quality images. This drives the widespread application of T2I in inference services hosted by cloud servers. As depicted in Figure 1, the client uploads text prompts to the cloud, which generates images and sends them back to the client. This paradigm is widely-adopted since the generation typically requires huge computation cost, which is unaffordable for clients, especially for devices with limited computation power.

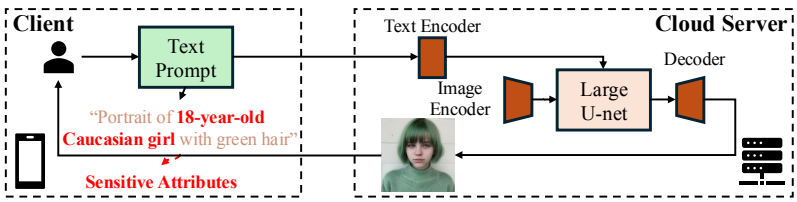


Figure 1: Illustration of existing server-only text-to-image generation services.

However, despite its growing popularity, there remain two essential problems: ① ***Prompt privacy*** remains a critical concern [43, 48]: In image generation services like Midjourney [30] and DALL·E [32],

*Email at haoqi.1997@tiktok.com

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

the server deploys their models on the cloud and provides APIs that take client prompts as inputs, which might contain sensitive attributes like *gender, ethnicity*, etc. ② ***Server cost*** increases drastically. According to the scaling law [16, 20], the superior model capacity comes at the expense of larger model size, leading to considerable hardware requirement and computation cost on the cloud servers.

Existing solutions typically only address one of the aforementioned issues. Cryptographic-based approaches [47, 3] offer provable security to the sensitive prompts. However, the huge computation overhead and efficiency decay hinders their application in real world scenarios. Meanwhile, some prior works [46, 4] provide a client-side input filter that detects and perturbs sensitive information before sent to the server. However, such kind of random perturbation incurs inevitable semantic and utility loss, leading to mismatch between the user intent and generation output. Another line of works employ on-device models [22, 49] to allow efficient image generation, avoiding data transmission to the server. However, despite significant improvement that has been made, the image quality inevitably decreases, failing to meet the users' needs. Recently, Hybrid SD [44] incorporates a hybrid image generation pipeline to lower the server-side computation cost, which however fails to preserve the prompt privacy, the embedding of which is directly sent to the server. Such kind of information is vulnerable to extraction attacks [33, 31]. Therefore, here arises the question:

Can we perform privacy-preserving image generation with better image quality and lower server cost?

As an attempt to answer this question, we propose `ObCLIP`, an oblivious cloud-device hybrid image generation scheme that provides rigorous privacy and comparable utility to large cloud models with slightly increased server cost. Specifically, our contributions are summarized as follows:

- **Oblivious Cloud-Device Hybrid Generation Scheme**. `ObCLIP` consists of two main components to address the aforementioned challenges: 1) *Oblivious transformation*: each input prompt is transformed into a set of semantically similar candidate prompts that differ only in sensitive attributes (e.g., gender ,age, ethnicity). The cloud server processes all candidate prompts without knowing which one is the real one, thus preventing any prompt leakage. 2) *Local extraction*: the client selects the targeted output corresponding to the real prompt. One straightforward drawback of vanilla oblivious generation is heavy server cost. Therefore, we devise a hybrid generation pipeline, where only partial denoising steps are performed by the server. Besides, we analyze and incorporate several cache-based acceleration methods, leveraging both temporal and batch redundancy, to further reduce server cost with minimal utility degradation.
- **Temporal- and Batch- Redundancy based Acceleration**. We analyze and incorporate several cache-based acceleration techniques to exploit temporal redundancy in server-side generation. Additionally, inspired by batch-level redundancy, we propose reusing attention maps across the batch of candidate prompts. Together, these two acceleration strategies effectively reduce computation costs with minimal utility degradation.
- **Empirical Evaluations.** We conduct extensive text-to-image generation experiments on several stable diffusion models across three datasets. The experiments confirm that `ObCLIP` provides rigorous privacy and comparable utility to large cloud models with slightly increased server computation costs, which is about $4.4 \sim 7.6\times$ lower than vanilla oblivious generation baseline and orders of magnitude lower than cryptographic approach.

2 Related Work

Existing works typically employ various privacy enhancing technologies. We provide a comprehensive comparison of related works in Table 1, focusing on application domain and trade-off among prompt privacy, server cost and image utility.

Cryptographic methods like secure multi-party computation (MPC) that support computation over encrypted data and models are widely used to enable secure machine learning inference [27, 6, 42, 47]. MPCViT [47] employed MPC and proposed a search algorithm for MPC-friendly neural architecture. HE-Diffusion [3] leveraged homomorphic encryption (HE) to perform partial image encryption and protected the diffusion process. However, these works impose significant overhead compared to plaintext baselines, limiting their practicality for real-world deployment. Other works employ lightweight privacy techniques like differential privacy (DP) [7] to perturb prompts by adding random noises in text generation. SANTEXT [46] designed a Exponential mechanism based word-level

Table 1: Comparison of related work. ●, ◐ and ○ refer to high-, medium- and low-performance.

Method			Domain	Privacy	Server Cost	Utility
Non-private	Standalone	Server-Only [34]	Text-to-Image	○	○	●
		Client-Only [22, 49]	Text-to-Image	●	●	○
	Hybrid	Hybrid SD [44]	Text-to-Image	○	◐	◐
Private	MPC	MPCViT [47]	Text-to-Image	●	○	◐
		HE-Diffusion [3]	Text-to-Image	●	○	◐
	DP	SANTEXT [46]	Text Generation	◐	●	○
		CAPE [41]	Text Generation	◐	●	○
	Ours	ObCLIP	Text-to-Image	●	◐	◐

perturbation mechanism to hide sensitive attributes within a prompt. CAPE [41] further proposed an optimized perturbation mechanism by incorporating contextual information that achieves a better trade-off between privacy and utility. However, privacy comes at the cost of semantics distortion and utility degradation. Besides, the scalability to text-to-image generation domain is uncertain. Another line of works manage to offload the generation to user devices, avoiding the transmission of prompts to server. SnapFusion [22] introduced efficient network architecture and improved step distillation process to enable generation within 2 seconds. MobileDiffusion [49] further studied one-step sampling technique, which decreased the runtime to less than 1 second. However, its scalability to larger models like SDXL [34] is not yet explored.

Recently, a new paradigm of cloud-device hybrid generation scheme is proposed to lower the server-side computation cost. Hybrid SD [44] proposed an cloud-device collaborative stable diffusion pipeline. By offloading a great portion of denoising steps to client devices, the server-side costs can be optimized. However, it only reduces server costs, failing to effectively protect the privacy of user prompts, the embedding of which is directly sent to the server. Such kind of information is vulnerable to extraction attacks [5, 31, 33], which could reconstruct the original prompt or infer partial sensitive attributes. To make matters worse, even if embedding inversion fails, the server can still perform full image generation based on the received embeddings, inevitably revealing sensitive visual patterns. Built upon such paradigm, we manage to preserve the privacy in an oblivious way, providing rigorous privacy. To hedge against the additional server-side cost introduced by our scheme, we incorporate several acceleration methods based on temporal and batch redundancy.

3 Preliminary

Diffusion Model. Diffusion models [13] have attracted significant attentions due to their ability to generate high-quality images. Its forward process adds Gaussian noise to the data over a fixed number of timesteps as $q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t} \cdot x_{t-1}, \beta_t \mathbf{I})$, where x_0 refers to the original image, $t \in [1, ..., T]$ denotes the total diffusion timestep, $\{x_1, ..., x_T\}$ denote the sequence of noisy latents, $\beta_t \in (0, 1)$ determines the amount of noise added at each timestep, $\mathbf{I}$ is the identity matrix and $\mathcal{N}(x; \mu, \sigma)$ denotes the Gaussian distribution with mean μ and covariance σ. During image generation, the reverse (denoising) process aims to recover x_{t-1} from x_t using a neural network as the noise predictor (typically, U-net [36]) $\epsilon_\theta(x_t, t)$ that predicts the noise in each timestep. Concretely,

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) \tag{1}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{T} \alpha_i$. This process is iteratively applied starting from a random sample drawn from the noise prior and finally yields a denoised sample. For stable diffusion models [35], the core idea is to perform the diffusion process in a lower-dimensional latent space—obtained via a pretrained variational autoencoder (VAE)—rather than directly in pixel space. This approach significantly reduces computation overhead during denoising process. Recently, the MMDiT architecture [8, 10], which jointly processes image and text tokens to model cross-modal relationships, has also been gaining increasing attention.

Training-free Diffusion Acceleration. To mitigate the expensive computation cost of diffusion generation, prior works either resort to training-based model distillation [38, 26] and model compression [9, 17] approaches, or training-free caching [28, 24, 21] methods. In this paper, we mainly focus on training-free acceleration methods that leverages different kinds of feature redundancy throughout

the diffusion process. As observed by prior works, adjacent steps exhibits temporal redundancy, happening in layer outputs or even block outputs. In U-net, the attention module is computed as:

$$M = \text{Softmax}(\frac{Q \cdot K^T}{\sqrt{d}}), O = M \cdot V \tag{2}$$

where Q refers to projected features from latent, K, V refers to latent (self-attention) or text embedding (cross-attention). Recently, Faster Diffusion [21] explored the feasibility of skip U-net encoder computation with a delicate skip strategy. Applying these acceleration methods to the new paradigm of hybrid inference presents unique challenges that necessitate a reexamination of previous strategies.

4 ObCLIP: Design

Inspired by prior works [2, 45], which highlight the feasibility of employing a mixture of diffusion models at different stages of the denoising process, we devise ObCLIP for an optimal cloud-device hybrid generation scheme with privacy preservation. As illustrated in Figure 2, the two key components in ObCLIP are oblivious transformation and local extraction. By transforming the private prompt into a set of candidate prompts, server only acting as guidance of several initial steps and offloading most of later diffusion steps to device, we achieve the following features: 1) enhanced image quality by leveraging the on-cloud large-capacity models; 2) strengthened prompt privacy through on-device computation. To further lower the server computation cost, we devise several server-side acceleration methods tailored to our scenario.

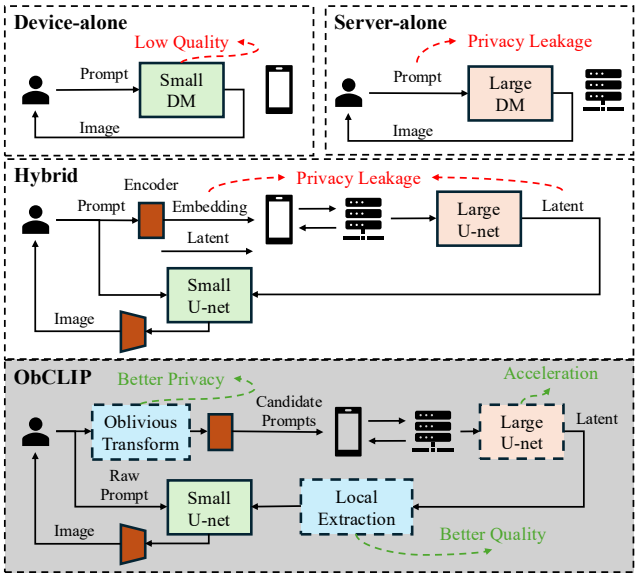


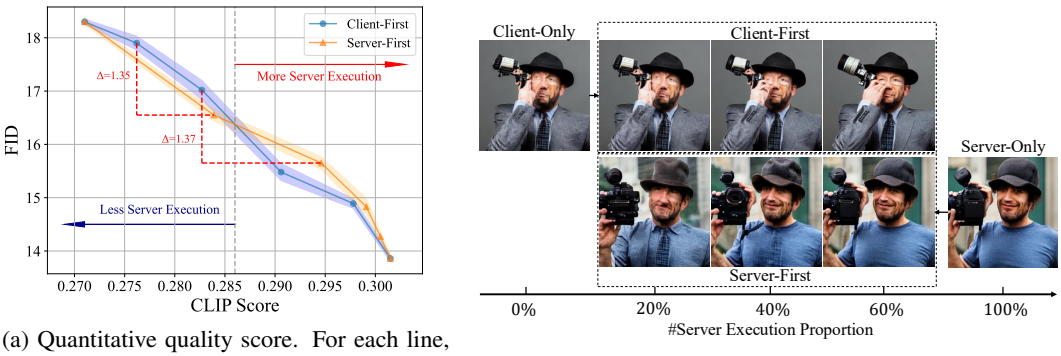
Figure 2: Holistic comparison of existing schemes.

4.1 Oblivious Cloud-Device Hybrid Image Generation

Before delving into the detailed design of ObCLIP, we first answer the two essential research questions.

- **RQ 1**: How to decide the *diffusion allocation strategy* to achieve better utility?
- **RQ 2**: How to *hide the sensitive attributes in a prompt* from the server?

Diffusion Allocation Strategy. As an answer to **RQ 1**, we empirically investigate the impact of different allocation strategies, focusing on two key factors: 1) whether the initial denoising steps are performed by a large server-side model or a small client-side model; and 2) the proportion of diffusion steps distributed between the server and the client. We employ a 25-step DPM-Solver [25] on the SD-v1.4 model and its compressed variant BK-SDM-Small [17], evaluated on the MS-COCO dataset [23]. We run both quantitative and qualitative analysis by varying server execution proportion. As shown in Figure 3a, when server involvement is limited (0%$\sim$40%), allowing the server-side



(a) Quantitative quality score. For each line, from left to right, the server execution proportion starts from 0% to 100% with interval 20%.

(b) Qualitative quality visualization. Prompt = 'A man holding a camera up over his left shoulder.'

Figure 3: Quantitative and qualitative comparison result of using different hybrid execution strategies.

model to handle the initial diffusion steps yields better quantitative performance (i.e., lower FID) compared to the client-first strategy (the performance gap is highlighted with a red dashed line). This observation is consistent with the qualitative results in Figure 3b. As revealed in [24], the initial steps—referred to as the semantics-planning stage—are crucial for determining global semantic information. Consequently, server-guided semantics planning leads to better image quality. Based on these findings, we adopt the server-first strategy in our approach. Besides, we also control the proportion of server-side execution using the hyper-parameter *switch point* k, which serves as a trade-off between utility and efficiency. As k increases, the FID consistently decreases, albeit at the cost of higher server cost. More comprehensive evaluation are provided in Section 5.2.

Oblivious Generation Scheme. Regarding **RQ2**, one straightforward solution is to replace the actual prompt (e.g., "*portrait of young African woman*") to a random candidate prompt (e.g., "*portrait of elderly Caucasian woman*", semantically-close yet differ in sensitive attributes) to the server, who sends back intermediate denoised latent. The client then use the actual prompt for remaining denoising steps, aiming to *rectify the semantics deviation*. We examine the impact of different denoising proportions where the initial steps are conditioned on a candidate prompt, and the remaining steps use the actual prompt as the text condition. We conduct the analysis using SDXL [34] model with a 25-step DPM-Solver. As illustrated in Figure 4, the intended semantics are accurately captured only when the actual prompt governs more than 80% of the entire diffusion process. This observation suggests that initial steps are critical for establishing semantic information, making it difficult to correct semantic deviations introduced early on. Hence, to enhance the text-image semantic alignment, we propose the oblivious hybrid generation scheme, where the client transforms the actual prompt into a set of candidate prompts (including the actual prompt, security analysis in Section 4.3), serving as text conditions during the server-side guidance of intermediate latents. The intended intermediate latents are then retrieved by client for subsequent denoising. To construct the candidate prompt set $\mathcal{P}$, we identify sensitive attributes and traverses their value space as the algorithm in Appendix B.1.

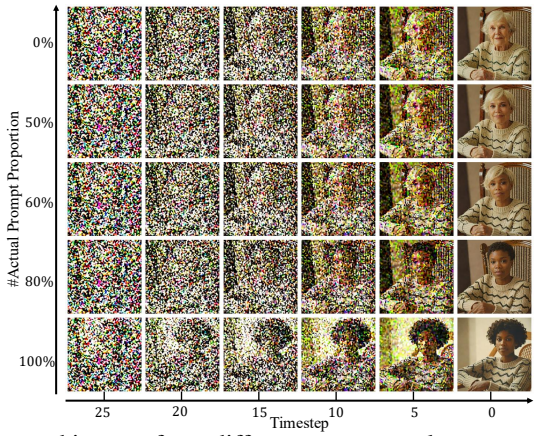


Figure 4: Generated images from different prompt replacement configurations.

Notably, vanilla oblivious generation—where the server performs all the denoising and sends generation images to client—incurs an $N\times$ increase in total computation cost ($N = |\mathcal{P}|$). While partial denoising in `ObCLIP` effectively reduces server-side computation, the additional overhead introduced by redundant denoising of multiple candidate prompts remains non-negligible. To mitigate this, we introduce server-side acceleration techniques aimed at minimizing such overhead.

4.2 Server-side Acceleration

Conceptually, in `ObCLIP`, we optimize server-side generation efficiency from two perspectives: 1) **Batch Redundancy**: In oblivious generation, a set of candidate prompts is processed, which differ only in the values of sensitive attributes while sharing most tokens, theoretically leading to similar global semantics, with minor changes on local details. 2) **Temporal Redundancy**: Due to the inherently sequential nature of the denoising process, intermediate features—such as outputs from attention modules and down/mid blocks—across adjacent timesteps can be cached and reused. High-level overview of the acceleration scheme is illustrated in Figure 5.

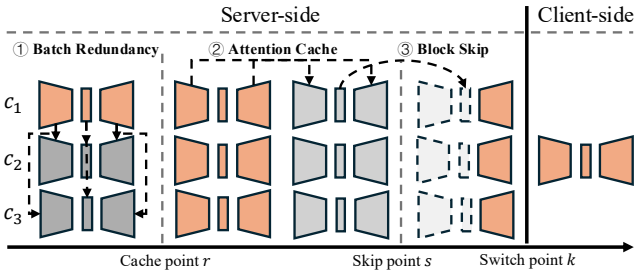


Figure 5: Server-side acceleration. The temporal- and batch- redundancy based caching are controlled by three hyper-parameters: cache point r, skip point s and switch point k.

4.2.1 Batch Redundancy

As empirically validated by the visualization of both cross-attention maps (Figure 6b) and self-attention maps (deferred to Figure 9 in Appendix D.1) that implicitly reflect semantic information for two candidate prompts by varying *gender* and *age* attributes, the global features like background, gesture, etc. are similar, while those sensitive attributes share similar focus areas. In this regard, we propose to reuse these attention maps across these candidate prompts. Specifically, we only compute the attention map for pivot prompt (with index i^*, e.g., the first prompt with $i^* = 0$), and reuse these attention maps before *cache point* to lower server-side attention computation as follows:

$$q^*, k^*, V = \texttt{to\_q}(\mathcal{Q}[i^*]), \texttt{to\_k}(\mathcal{K}[i^*]), \texttt{to\_v}(\mathcal{V}) \tag{3}$$

$$m^* = \texttt{get\_attention\_map}(q^*, k^*) \tag{4}$$

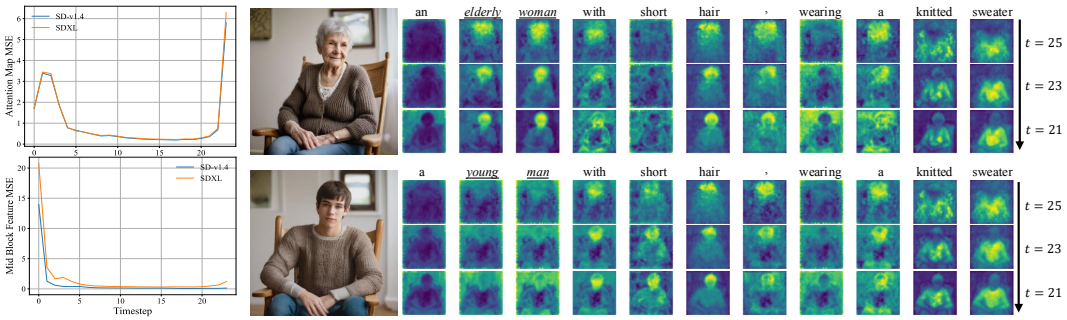
$$O = M \cdot V\{M \leftarrow \texttt{broadcast}(m^*)\} \tag{5}$$

In this case, the computation bottleneck of attention module (i.e., `to_q`, `to_k`, and Softmax in `get_attention_map`) can be greatly reduced. The detailed algorithm is deferred to Appendix B.2.

4.2.2 Temporal Redundancy

Attention Cache. We motivate this optimization by T-Gate [24], which observes that in the early phase (i.e., semantic-planning) conducted on the server side, self-attention makes limited contributions. We thus bypass the self-attention computations in subsequent diffusion steps after certain initial steps (denoted as *cache point* r), making use of such temporal redundancy [24]. Furthermore, we investigate the evolution of cross-attention map differences across adjacent timesteps. As shown in Figure 6a (top), these differences are substantial during the first 2∼3 steps, but drop significantly and stabilize thereafter. In addition, the cross-attention heatmap visualization in Figure 6b shows that the distribution at the *3rd* step is already similar to that at the *5th* step. We thus propose to cache the cross-attention maps after *cache point* as well. Notably, we follow T-Gate to refresh the cache every 5 steps to prevent significant deviations. The attention module is computed as follows:

$$O_t = \begin{cases} \texttt{Attn}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) & \text{if } t \leq r \text{ or } (t \mod 5) = 0 \\ O^* & \text{if } t > r \text{ and } (t \mod 5)! = 0 \end{cases} \tag{6}$$



(a) Temporal difference. (b) Heatmap of cross-attention maps in two candidate prompts varying {*gender*, *age*}.

Figure 6: Temporal and batch redundancy analysis on SD-v1.4 and SDXL models.

Block Skip. Previous observations [28, 21] have revealed that features from the down-block and mid-block exhibit relatively subtle variations across adjacent timesteps, especially when compared to those from the subsequent up-blocks. As illustrated in Figure 6a (bottom), the mid-block outputs show minimal changes during the denoising process—dropping significantly within the first 2∼3 steps and stabilizing after approximately 20% of total denoising steps. Motivated by this temporal feature similarity, we propose to skip server-side computation of the down-block and mid-block after a certain timestep (denoted as *skip point* s). We follow an intuitive way to select the skip point and defer the parameter configurations to Section 5. The computation is formulated as follows, where f_{mid} denotes the cached output of mid-block and $\mathcal{P}$ refers to the set of candidate prompts.

$$z_t = \begin{cases} (\texttt{Downblock} \circ \texttt{MidBlock} \circ \texttt{Upblock})(z_{t-1}, \mathcal{P}, t) & \text{if } t < s \\ \texttt{Upblock}(z_{t-1}, f_{mid}, \mathcal{P}, t) & \text{if } t \geq s \end{cases} \tag{7}$$

4.3 Security Analysis

In ObCLIP, the client $\mathcal{C}$ transforms the actual prompt p^* into a set of candidate prompts $\mathcal{P}$, with $|\mathcal{P}| = N$. We consider the LLM inference service provider as the potential adversary $\mathcal{A}$. We do not consider man-in-the-middle attacks and assume the communication channel is secure. $\mathcal{A}$ is assumed to be *semi-honest*, meaning the adversary follows the hybrid generation scheme, which is publicly known, but may attempt to extract sensitive information by collecting and analyzing the messages (i.e., $\mathcal{P}$) from the client. Our scheme shows that $\mathcal{A}$ cannot distinguish the real prompt p^* from N candidates with probability better than $1/N + \lambda$, where λ is negligible given no information other than $\mathcal{P}$. We defer the detailed proof of Theorem 1 to Appendix C.

Theorem 1 (Prompt In-distinguishability)**.** *The oblivious generation scheme is λ-oblivious if for any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$:*

$$|\Pr[A(\mathcal{P} = p^*)] - \frac{1}{N}| \leq \lambda$$

5 Experiments

5.1 Experiment Settings

Models. We consider several combinations for hybrid generation. We consider SD-v1.4 [35], and its compressed versions BK-SDM-small and BK-SDM-tiny [17]. We also test finetuned Realistic Vision v4.0 [40] and compressed small-sd model [39]. For high-resolution setting, we consider SDXL [34] and Koala-700m [18], along with the scalability to step-distilled server model LCM-SDXL [26].

Datasets & Metrics. To evaluate the performance of ObCLIP, we adopt two commonly-used datasets: 1) MS-COCO 2014 dataset [23] with a resolution of 512×512. We use 30k prompts from its validation split. 2) MJHQ [19] with a resolution of 1024×1024. For more comprehensive evaluation on oblivious generation, we construct a candidate prompt dataset using 10 templates, like "*High-quality, face portrait photo of a <age> <ethnicity> <gender>*" with random fill on these sensitive attributes. The detailed construction is provided in Appendix B.3. Regarding image quality, we follow prior works to evaluate the visual quality using Frechet Inception Distance (FID) [12] and Inception

Score (IS) [37]. We assess text-image alignment using CLIP score [11] with CLIP-ViT-g/14 model. Regarding efficiency, we use Floating-point Operations (FLOPs) and average running time.

Baselines. We compare ObCLIP with three lines of works: 1) ***Standalone Generation***: The commonly used paradigm where cloud-only or device-only generation is applied. This exhibits either high privacy risk and server cost or low utility; 2) ***Hybrid SD*** [44]: This is the first work that proposed cloud-device collaborative generation paradigm. 3) ***HE-Diffusion*** [3]: Last, we compare with cryptographic approach HE-Diffusion in terms of efficiency. We take the runtime of the SD-v1.4 model as reported in their paper. We opt for 25-step DPM scheduler (8-step for LCM-SDXL) for all evaluated works. For ObCLIP, we mainly adopt two acceleration configurations: 1) switch point $k = 5$, cache point $r = 3$ and skip point $s = 3$; 2) $k = 10$, $r = 4$ and $s = 6$. We use N to denote the cardinality of candidate prompt set, i.e., those edited prompts after oblivious transformation. To do so, we use rule-based method and a finetuned distilbert model [15] to detect sensitive attributes. We additionally evaluate vanilla oblivious generation (OG), where the cloud alone generates N images without any accelerations. All the experiments are conducted on one Ubuntu machine equipped with one Intel Xeon Platinum 8260 CPU, 16GB of RAM and 1 NVIDIA Tesla-V100-SXM2-32GB GPU.

5.2 Quantitative Results

Results on Candidate Prompt Dataset. To begin, we run experiments on Realistic Vision v4.0 and small-sd models on the candidate prompt dataset to evaluate oblivious generation. For gender alone, $N = 2$. While for multi-attribute combinations, i.e., gender and age, we have $N = 2 \times 3 = 6$. We present the results for 1-attribute and 2-attribute oblivious generation in Table 2. The results for 3-attribute are deferred to Appendix D.4. We here mainly focus on cloud-side latency. In general, ObCLIP shows better performance in the trade-off between generation performance and computation cost, offering flexibility through the acceleration strategies, e.g., k. In terms of generation performance, ObCLIP achieves comparable and even better FID and IS compared to Realistic Vision v4.0 when $k = 10$ (e.g., FID drops from 113.39 to 109.76 when $N = 6$). Even when we use a more aggressive $k = 5$, we still achieve comparable FID and IS while much lower latency (e.g., FID increases from 113.39 to 113.92 with latency decreases from 1.12s to 0.98s). In terms of latency, compared to Hybrid SD, ObCLIP incurs about $N\times$ FLOPs due to the oblivious generation. However, with the cache and reuse accelerations enabled, the latency of ObCLIP is nearly reduced by 50%, which is comparable to Hybrid SD when $N = 2$ (e.g., latency increases from 0.55s to 0.57s when $k = 10$) and about $3\times$ slower when $N = 6$. Given the strong privacy protection brought by ObCLIP, such computation cost is significantly reduced. Especially,ObCLIP is orders of magnitude faster than HE-Diffusion and about $4.4 \sim 7.6\times$ faster than vanilla OG. The results validate the effectiveness of ObCLIP, which offers rigorous privacy and better generation performance (measured by FID and CLIP scores), with only a marginal increase in latency.

Table 2: Multi-Attribute ObCLIP on candidate prompt dataset. For FLOPs, we use $a(+b)$, where a and b refer to cloud/device computations. For latency, we only measure the cloud-side runtime.

Generation Method	1-Attribute (gender, $N = 2$)					2-Attribute (gender + age, $N = 6$)				
	FID ↓	IS ↑	CLIP ↑	FLOPs (T)	Latency (s)	FID ↓	IS ↑	CLIP ↑	FLOPs (T)	Latency (s)
Realistic Vision v4.0	113.45	4.69	0.3322	18.53 (+0)	1.12	113.39	5.32	0.3215	18.53 (+0)	1.12
small-sd	128.87	5.04	0.3051	0 (+11.20)	0.78	118.19	5.11	0.2980	0 (+11.20)	0.78
Vanilla OG	113.45	4.69	0.3322	37.06 (+0)	2.51	113.39	5.32	0.3215	111.18 (+0)	7.47
HE-Diffusion	-				>106	-				>106
Hybrid SD ($k = 10$)	117.18	4.96	0.3215	7.41 (+6.54)	0.55	114.05	5.02	0.3226	7.41 (+6.54)	0.55
ObCLIP($k = 10$)	117.18	4.96	0.3215	14.82 (+6.54)	0.97	114.05	5.02	0.3226	44.46 (+6.54)	2.90
+ cache	118.59	4.99	0.3168	12.26 (+6.54)	0.62	115.65	5.02	0.3174	36.76 (+6.54)	1.85
+ reuse	114.26	4.82	0.3167	11.48 (+6.54)	0.57	109.76	4.94	0.3152	33.28 (+6.54)	1.55
Hybrid SD ($k = 5$)	119.31	4.99	0.3107	3.71 (+8.96)	0.28	116.15	5.05	0.3117	3.71 (+8.96)	0.28
ObCLIP($k = 5$)	119.31	4.99	0.3107	7.41 (+8.96)	0.49	116.15	5.05	0.3117	22.23 (+8.96)	1.48
+ cache	120.44	4.88	0.3079	6.13 (+8.96)	0.38	117.29	5.00	0.3091	18.38 (+8.96)	1.12
+ reuse	118.36	4.98	0.3077	5.74 (+8.96)	0.33	113.92	4.87	0.3076	16.64 (+8.96)	0.98

Results on Real-world Datasets. Table 3 and Table 4 (upper part) present the results for MS-COCO and MJHQ datasets. For numbers marked with *, the total FLOPs should be multiplied by N. Compared to distilled models, ObCLIP considerably reduces FLOPs while achieving better image fidelity and stronger semantic alignment between image and prompt across various configurations. When compared to base models, ObCLIP offers substantial FLOPs reduction at the cost of a slightly

higher FID. For instance, on SD and BK-SDM-small models, the FID of `ObCLIP` ($k = 10$, with cache) increases from 13.86 to 15.73 with a reduction of FLOPs from 18.53T to 5.84^*T. Similarly, on SDXL and Koala-700m models, `ObCLIP` ($k = 10$, with cache) achieves a FID of 30.79, which is comparable to SDXL's 30.67, while reducing the FLOPs from 159.35T to 45.11^*T — even lower when $N < 4$. The trade-off between performance and latency can also be controlled via k: a larger k yields better image fidelity at the expense of higher FLOPs.

Table 3: SD-v1.4 and BK-SDM-{small, tiny} on 30k MS-COCO dataset.

Generation Method	FID ↓	IS ↑	CLIP ↑	FLOPs (T)
SD-v1.4	13.86	37.75	0.3015	18.53
BK-SDM-small	18.30	31.73	0.2710	10.90
`ObCLIP`($k = 10$)	15.65	36.72	0.2946	7.41^*
+ cache	15.73	33.62	0.2865	5.84^*
`ObCLIP`($k = 5$)	16.55	33.95	0.2839	3.71^*
+ cache	16.45	33.36	0.2833	3.06^*
BK-SDM-tiny	18.30	29.94	0.2681	10.25
`ObCLIP`($k = 10$)	15.86	35.54	0.2936	7.41^*
+ cache	16.44	32.80	0.2887	5.84^*
`ObCLIP`($k = 5$)	16.87	32.73	0.2812	3.71^*
+ cache	17.14	31.84	0.2854	3.06^*

Table 4: {SDXL, LCM-SDXL} and Koala-700m on 5k MJHQ dataset. $\triangle t$ denotes timestep shift.

Generation Method	FID ↓	IS ↑	CLIP ↑	FLOPs (T)
SDXL	30.67	26.31	0.3464	159.35
koala-700m	36.11	22.06	0.3263	58.85
`ObCLIP`($k = 10$)	31.92	24.56	0.3389	63.74^*
+ cache	30.79	24.52	0.3320	45.11^*
`ObCLIP`($k = 5$)	32.42	23.70	0.3337	31.87^*
+ cache	31.97	23.61	0.3317	24.41^*
LCM-SDXL	33.25	28.22	0.3296	50.99
`ObCLIP` (k=4,$\triangle t$=8)	33.92	27.55	0.3315	25.50^*
+ cache ($\triangle t = 8$)	34.12	27.15	0.3311	21.78^*
+ cache ($\triangle t = 10$)	45.51	22.64	0.3293	
+ cache ($\triangle t = 4$)	39.07	22.73	0.3166	
+ cache ($\triangle t = 0$)	51.85	16.94	0.2936	

Scalability to Distilled Models. We hereby further explore the scalability of `ObCLIP` to step-distilled cloud models. Note that the LCM-SDXL uses a 8-step scheduler, while Koala-700m uses standard 25-step scheduler. We adopt $k = 4, r = 2$ for acceleration. Besides, to align the denoising timesteps between cloud and device, we propose to apply a timestep shift as $t_{device} = t_{cloud} + \triangle t$. A smaller or larger $\triangle t$ introduces incompatible denoising scales. As depicted in the bottom of Table 4, $\triangle t = 8$ yields the best performance (marked in gray). We provide visual examples in Appendix D.7. Compared to the standalone LCM-SDXL, `ObCLIP` (without cache) achieves a 0.67 drop in FID, a slightly better CLIP score, and reduces FLOPs from 50.99 to 25.50^*. With cache enabled, the FLOPs are further reduced by approximately 15%, with only a marginal FID drop of 0.2.

Additional Overhead. In `ObCLIP`, we introduce two additional operations: 1) device-side oblivious transformation; 2) cloud-to-device latent transmission. The runtime for detecting sensitive attributes using the fine-tuned DistilBERT model on a single V100 GPU is about 6.37ms. According to the report [2], the latency on iPhone 13 Pro is below 10ms. For cloud-to-device transmission, the noise latent is sized at $4 \times \mathsf{res} \times \mathsf{res} \times N$. Taking SD-v1.4 as an example, where $\mathsf{res} = 64$, the total data in FP16 precision is $\sim 32N$ KB. Considering the average WiFi bandwidth of 18.88Mbps [14], the total transmission time is around $0.013N$ seconds. Even for $N = 30$, the transmission time is approximately 0.39 seconds, demonstrating an acceptable overhead. We note that private information retrieval (PIR) [1, 29] can also be employed for this transmission, achieving a communication size of $\mathcal{O}(\log^2 N)$ or $\mathcal{O}(\sqrt{N})$, albeit at the expense of $\mathcal{O}(N)$ server computation. This approach may be preferable when the bandwidth is limited. Due to page limitation, we provide a more comprehensive efficiency evaluation and detailed efficiency improvement breakdown in Appendix D.5∼D.6.

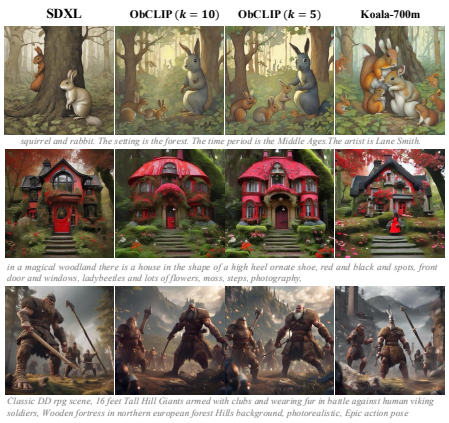
5.3 Qualitative Results

Effect of k. Figure 7a and Figure 7b show the samples generated using base models and `ObCLIP` (with cache) with different k. One observation is that with the guidance of large cloud models, `ObCLIP` achieves better semantic alignment and finer local details. For example, Koala-700m mistakenly generates a girl for the second prompt, while small-sd generates three overlapping teddy bears. Besides, a larger k exhibits better visual quality—closer to that of large cloud models—which is consistent with our previous quantitative findings.

Effect of batch reuse. Furthermore, we present the samples generated with different acceleration strategies in Figure 8. Specifically, the reuse across different gender, age, and ethnicity attributes results in strong semantic alignment, even when the attributes are contradictory (e.g., from female to male). Global structural information unrelated to the sensitive attributes—such as hairstyle and gesture—are well preserved, while the transformation to the target attribute is effectively accomplished.

[2] `https://machinelearning.apple.com/research/neural-engine-transformers`

(a) Results on Realistic Vision v4.0 and small-sd.



(b) Results on SDXL and Koala-700m.

Figure 7: Images generated by cloud (left), device (right) and ObCLIP (middle) with different k.

For instance, reusing female-associated attention maps in male image generation yields a male with long hair, illustrating both structural consistency and effective attribute modification.

(a) SD-v1.4 and BK-SDM-small

(b) SDXL and Koala-700m

Figure 8: Qualitative visual results when using different server-side acceleration strategies.

6 Conclusion

In this paper, we propose an oblivious cloud-device hybrid image generation scheme, acting as a plug-and-play safeguard to ML inference services, to provide rigorous prompt privacy, with better utility against on-device generation and only slightly increased server computation cost. Extensive experiments across multiple datasets demonstrate that ObCLIP provides comparable utility to cloud models with slightly increased server cost compared to non-private baselines.

Limitations. Despite the significant efficiency improvement over private baselines, a limitation lies in the inherent nature of oblivious generation, which leads to a sub-linear increase in computation relative to the number of candidate prompts. A potential mitigation is to achieve statistical indistinguishability, thereby reducing the effective size of candidate prompts. This can be achieved by differential privacy-based top-k selection of these candidate prompts. By choosing appropriate privacy budget, we can achieve measurable privacy, while providing a much better efficiency. In the future, we also plan to extend our approach to the image-to-image generation domain, where inputs include not only text prompts but also real reference images—such as human faces—that carry more sensitive information.

References

[1] Sebastian Angel, Hao Chen, Kim Laine, and Srinath T. V. Setty. PIR with compressed queries and amortized query processing. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*, pages 962–979. IEEE Computer Society, 2018.

[2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *CoRR*, abs/2211.01324, 2022.

[3] Yaojian Chen and Qiben Yan. Privacy-preserving diffusion model using homomorphic encryption. *CoRR*, abs/2403.05794, 2024.

[4] Chun Jie Chong, Chenxi Hou, Zhihao Zephyr Yao, and Seyed Mohammadjavad Seyed Talebi. Casper: Prompt sanitization for protecting user privacy in web-based large language models. *CoRR*, abs/2408.07004, 2024.

[5] Zhang Collin, Morris John, X., and Shmatikov Vitaly. Universal zero-shot embedding inversion. *arXiv preprint arXiv:2504.00147v1*, 2025.

[6] Ye Dong, Wen-jie Lu, Yancheng Zheng, Haoqi Wu, Derun Zhao, Jin Tan, Zhicong Huang, Cheng Hong, Tao Wei, and Wenguang Chen. PUMA: secure inference of llama-7b in five minutes. *CoRR*, abs/2307.12533, 2023.

[7] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[8] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024.

[9] Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[10] Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, Wei Liu, Yichun Shi, Shiqi Sun, Yu Tian, Zhi Tian, Peng Wang, Rui Wang, Xuanda Wang, Xun Wang, Ye Wang, Guofeng Wu, Jie Wu, Xin Xia, Xuefeng Xiao, Zhonghua Zhai, Xinyu Zhang, Qi Zhang, Yuwei Zhang, Shijia Zhao, Jianchao Yang, and Weilin Huang. Seedream 3.0 technical report, 2025.

[11] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: a reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6626–6637, 2017.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[14] Chuang Hu, Wei Bao, Dan Wang, and Fengming Liu. Dynamic adaptive DNN surgery for inference acceleration on the edge. In *2019 IEEE Conference on Computer Communications, INFOCOM 2019, Paris, France, April 29 - May 2, 2019*, pages 1423–1431. IEEE, 2019.

[15] Isotonic. Distilbert fine-tuned for ai4privacy task, 2025.

[16] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020.

[17] Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. BK-SDM: A lightweight, fast, and cheap version of stable diffusion. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LIV*, volume 15112 of *Lecture Notes in Computer Science*, pages 381–399. Springer, 2024.

[18] Youngwan Lee, Kwanyong Park, Yoorhim Cho, Yong-Ju Lee, and Sung Ju Hwang. KOALA: empirical lessons toward memory-efficient and fast diffusion models for text-to-image synthesis. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[19] Daiqing Li, Aleks Kamko, Ali Sabet, Ehsan Akhgari, Linmiao Xu, and Suhail Doshi. Playground v2.

[20] Hao Li, Yang Zou, Ying Wang, Orchid Majumder, Yusheng Xie, R. Manmatha, Ashwin Swaminathan, Zhuowen Tu, Stefano Ermon, and Stefano Soatto. On the scalability of diffusion-based text-to-image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 9400–9409. IEEE, 2024.

[21] Senmao Li, Taihang Hu, Joost van de Weijer, Fahad Shahbaz Khan, Tao Liu, Linxuan Li, Shiqi Yang, Yaxing Wang, Ming-Ming Cheng, and Jian Yang. Faster diffusion: Rethinking the role of the encoder for diffusion model inference. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[22] Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[23] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014.

[24] Haozhe Liu, Wentian Zhang, Jinheng Xie, Francesco Faccio, Mengmeng Xu, Tao Xiang, Mike Zheng Shou, Juan-Manuel Perez-Rua, and Jürgen Schmidhuber. Faster diffusion via temporal attention decomposition. *Transactions on Machine Learning Research*, 2025.

[25] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *CoRR*, abs/2211.01095, 2022.

[26] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *CoRR*, abs/2310.04378, 2023.

[27] Junming Ma, Yancheng Zheng, Jun Feng, Derun Zhao, Haoqi Wu, Wenjing Fang, Jin Tan, Chaofan Yu, Benyu Zhang, and Lei Wang. Secretflow-spu: A performant and user-friendly framework for privacy-preserving machine learning. In Julia Lawall and Dan Williams, editors, *Proceedings of the 2023 USENIX Annual Technical Conference, USENIX ATC 2023, Boston, MA, USA, July 10-12, 2023*, pages 17–33. USENIX Association, 2023.

[28] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 15762–15772. IEEE, 2024.

[29] Samir Jordan Menon and David J. Wu. YPIR: high-throughput single-server PIR with silent preprocessing. In Davide Balzarotti and Wenyuan Xu, editors, *33rd USENIX Security Symposium, USENIX Security 2024, Philadelphia, PA, USA, August 14-16, 2024*. USENIX Association, 2024.

[30] MidJourney. Midjourney, 2025. Accessed: 2025-03-31.

[31] John X. Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M. Rush. Text embeddings reveal (almost) as much as text. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12448–12460. Association for Computational Linguistics, 2023.

[32] OpenAI. Dall·e: Creating images from text, 2025. Accessed: 2025-03-31.

[33] Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020*, pages 1314–1331. IEEE, 2020.

[34] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE, 2022.

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells III, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, 2015.

[37] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2226–2234, 2016.

[38] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[39] Segmind. Small-sd: A distilled stable diffusion model. `https://huggingface.co/segmind/small-sd`, 2023. Accessed: 2025-04-16.

[40] SG161222. Realistic vision v4.0 (novae). `https://huggingface.co/SG161222/Realistic_Vision_V4.0_noVAE`, 2023. Accessed: 2025-04-14.

[41] Haoqi Wu, Wei Dai, Li Wang, and Qiang Yan. Cape: Context-aware prompt perturbation mechanism with differential privacy. *CoRR*, abs/2505.05922, 2025.

[42] Haoqi Wu, Wenjing Fang, Yancheng Zheng, Junming Ma, Jin Tan, and Lei Wang. Ditto: Quantization-aware secure inference of transformers upon MPC. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[43] Chejian Xu, Jiawei Zhang, Zhaorun Chen, Chulin Xie, Mintong Kang, Zhuowen Yuan, Zidi Xiong, Chenhui Zhang, Lingzhi Yuan, Yi Zeng, Peiyang Xu, Chengquan Guo, Andy Zhou, Jeffrey Ziwei Tan, Zhun Wang, Alexander Xiong, Xuandong Zhao, Yu Gai, Francesco Pinto, Yujin Potter, Zhen Xiang, Zinan Lin, Dan Hendrycks, Dawn Song, and Bo Li. Mmdt: Decoding the trustworthiness and safety of multimodal foundation models. In *ICLR 2025*, February 2025.

[44] Chenqian Yan, Songwei Liu, Hongjian Liu, Xurui Peng, Xiaojian Wang, Fangming Chen, Lean Fu, and Xing Mei. Hybrid SD: edge-cloud collaborative inference for stable diffusion models. *CoRR*, abs/2408.06646, 2024.

[45] Xingyi Yang, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Diffusion probabilistic model made slim. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 22552–22562. IEEE, 2023.

[46] Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman S. M. Chow. Differential privacy for text analytics via natural text sanitization. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3853–3866. Association for Computational Linguistics, 2021.

[47] Wenxuan Zeng, Meng Li, Wenjie Xiong, Tong Tong, Wen-Jie Lu, Jin Tan, Runsheng Wang, and Ru Huang. Mpcvit: Searching for accurate and efficient mpc-friendly vision transformer with heterogeneous attention. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 5029–5040. IEEE, 2023.

[48] Qingjie Zhang, Han Qiu, Di Wang, Yiming Li, Tianwei Zhang, Wenyu Zhu, Haiqin Weng, Liu Yan, and Chao Zhang. A benchmark for semantic sensitive information in LLMs outputs. In *The Thirteenth International Conference on Learning Representations*, 2025.

[49] Yang Zhao, Yanwu Xu, Zhisheng Xiao, Haolin Jia, and Tingbo Hou. Mobilediffusion: Instant text-to-image generation on mobile devices. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXII*, volume 15120 of *Lecture Notes in Computer Science*, pages 225–242. Springer, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We provide detailed application scenario and main contributions in the abstract and Section 1. The experiments in Section 5 also exhibit the effectiveness of our work.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the primary limitations of this work in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide a detailed security analysis in Section 4.3 and proof in the Appendix C.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experiments are conducted on open-source models and three datasets—two of which are publicly available, and one is a synthetic dataset, with detailed construction described in Appendix B.3. The full experimental configurations are provided in Section 5.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We do not have the time to refactor the code, which is of poor readability. We promise to open-source the code to reproduce the experimental results on GitHub once accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We follow the standard method of prior work as we have mentioned in Section 5.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We conduct the experiments many times and report average results. In Figure 3, we show the one standard deviation as a shaded region.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We mention that we run the experiments on 1 NVIDIA V100 GPU. The specific configurations are included in the Section 5.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential positive societal impacts of using `ObCLIP` for protecting the prompt privacy in the Introduction Section 1. More detailed elaboration is provided in Appendix A.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The answer NA means that the paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original paper that produced the code, dataset and models.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We only use the LLM for paper writing, editing, or formatting purposes.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Broad Impacts

This paper presents work whose goal is to advance the field of Machine Learning. Our work aims to address critical privacy concerns surrounding user prompts in widely adopted stable diffusion based text-to-image generation services. To safeguard sensitive user information during inference, we propose an oblivious cloud-device hybrid image generation scheme, acting as a plug-and-play safeguard to ML inference services, to provide rigorous prompt privacy. We also acknowledge the redundant computational overhead introduced by generating intermediate latents for all candidate prompts in oblivious generation. Beyond our proposed acceleration strategies, further improvements could be achieved by constraining the sampling space of candidate prompts. Ultimately, this reflects a fundamental trade-off between privacy and efficiency. These advancements have the potential to make private LLM inference more practical and scalable for real-world applications.

B Implementation Details

B.1 Oblivious Cloud-Device Hybrid Generation Algorithm

The detailed design is presented in Algorithm 1. Note that in line 2, `all_combination` refers to enumerating all possible combinations by traversing every value of each attribute $F_i \in \mathcal{F}_{\text{occur}}$, where $F_i = \{f_1, \ldots, f_q\}$.

Algorithm 1: Oblivious Hybrid Generation

Input: User prompt $p = \{w_1, ..., w_i, ..., w_n\}$, w_i corresponds to some sensitive attribute $F_i \in \mathcal{F} = \{F_1, ..., F_m\}$, with a value space of size q as $F_i = \{f_1, ..., f_q\}$, a client-side small diffusion model $\mathcal{M}_C$ and a small language model $\mathcal{M}_{LM}$, a server-side large model $\mathcal{M}_S$, denoising steps $T = \{t_{T-1}, ..., t_0\}$, switch point k, random latent $z_T \sim \mathcal{N}(0, \mathbf{I})$, VAE decoder $\mathcal{D}(\cdot)$

Output: Generated image y

`/* Oblivious Transform: construct candidate prompts based on` p `*/`

1 $\mathcal{F}_{occur} = \mathcal{M}_{LM}(p)$, with $F_i \in \mathcal{F}_{occur} = \{f_1, ..., f_q\}$

2 $\mathcal{P} \leftarrow$ `all_combination`$(p, \mathcal{F}_{occur})$

3 $\hat{z}_T = \text{repeat}(z_T, N)$, with $N = |\mathcal{P}|$

`/* Run initial` k `steps on cloud-side */`

4 **for** $t \in T[:k]$ **do**

5 $\quad \hat{z}_{t-1} = \mathcal{M}_S(\hat{z}_t, \mathcal{P}, t)$

`/* Extraction:`$\hat{z}_{T-k}$ `are sent to device, who chooses the actual latent` z_{T-k} `*/`

6 $z_{T-k} = \hat{z}_{T-k,j}$, s.t. $p_j \in \mathcal{P}$ and $p_j = p$

`/* Run remaining` $T-k$ `steps on device-side */`

7 **for** $t \in T[k:]$ **do**

8 $\quad z_{t-1} = \mathcal{M}_C(z_t, p, t)$

9 $y = \mathcal{D}(z_0)$

10 **return** y

B.2 Batch Reuse Algorithm

In this section, we provide the batch-reuse in attention modules in Algorithm 2.

B.3 Candidate Prompt Dataset Construction

```
template:
    - "headshots portrait with a $age $ethnicity $gender covered in religious tattoos."
    - "$age $ethnicity $gender in hat Fashion portrait photo"
    - "Smiling $age $ethnicity $gender sitting on flower field, Outdoor portrait photo"
    - "$age red haired $gender $ethnicity urban portrait photo"
    - "Faceshot Portrait of pretty $age $ethnicity $gender wearing a high neck sweater"
    - Closeup portrait photo of a $age $ethnicity $gender, wearing a rugged leather
        jacket, with a five o'clock shadow and prominent laugh lines around his eyes,
        captured in soft, golden hour lighting."
```

Algorithm 2: Batch-reused Attention Module

Input: Hidden states $\mathcal{Q} = \{q_i\}_{i=1}^N$, Encoder hidden states $\mathcal{K} = \{k_i\}_{i=1}^N, \mathcal{V} = \{v_i\}_{i=1}^N$ for N candidate prompts, Pivot sample index i^* for pivot prompt p

Output: Attention outputs $O = \{o_i\}_{i=1}^N$

/* Compute query q^*, key k^* for pivot prompt p (e.g., $i^* = 0$) */

$q^* = \texttt{to\_q}(\mathcal{Q}[i^*])$

$k^* = \texttt{to\_k}(\mathcal{K}[i^*])$

/* Compute attention map for pivot prompt */

$m^* = \texttt{get\_attention\_map}(q^*, k^*)$

/* Compute value v for all candidate prompts */

$V = \texttt{to\_v}(\mathcal{V})$

/* Compute attention outputs for all candidate prompts */

$M = \texttt{broadcast}(m^*, N)$

$O = M \cdot V$

return O

```
    - "RAW photo, (closeup:1.2), portrait of a $age $ethnicity $gender, wearing minimal
       makeup, showcasing the freckles, with a serene expression in a lush botanical
       garden, illuminated by gentle dappled sunlight."
    - "High-quality, face portrait photo of a $age $ethnicity $gender, wearing glasses,
       revealing the fine lines and character on the forehead."
    - "B&W photo of a $age $ethnicity $gender, shot from the side, highlighting elegant
       profile and the delicate lines etched across cheeks."
    - "High-quality, closeup portrait photo of a $age $ethnicity $gender, wearing
       traditional clothing."
age:
    - young
    - middle-aged
    - old
gender:
    - male
    - female
ethnicity:
    - caucasian
    - african
    - asian
    - indian
    - european
```

Listing 1: Prompt templates and sensitive attributes taxonomy.

We list the 10 templates used in constructing a small prompt dataset, tailored for candidate prompts, with replacement on sensitive attributes age, gender and ethnicity. We consider *age* $\in$ {young, middle-aged, old}, *gender* $\in$ {male, female} and *ethnicity* $\in$ {caucasian, african, asian, indian, european}.

C Security Analysis

Proof of Theorem 1. Consider a user prompt p. We obtain its candidate prompts as $\mathcal{P} = \texttt{candidate\_prompt}(p)$ by traversing the entire value space for each sensitive attribute in p. Then, we randomly select another sensitive prompt $p' \in \mathcal{P}$. Since we traverse the sensitive attributes to get $\mathcal{P}$, we have $\mathcal{P} = \texttt{candidate\_prompt}(p) = \texttt{candidate\_prompt}(p')$. That is,

$$\textbf{View}(p) = \textbf{View}(p'), \forall p' \in \mathcal{P}$$

We thus have, for any two sensitive prompts $p, p' \in \mathcal{P}$, the server's observable view is identically distributed. That is, no efficient adversary can guess the correct p with significantly better probability than random guessing $(1/N)$ as:

$$|\Pr[A(\mathcal{P} = p^*)] - \frac{1}{N}| \leq \lambda$$

where λ is negligible. The proof shows that the server learns nothing about the true input p beyond what is leaked by the candidate prompts (i.e., the value space for sensitive attributes) and those non-sensitive tokens, which are independent of the specific sensitive tokens. □

D Additional Experiments

D.1 Self-Attention Visualization

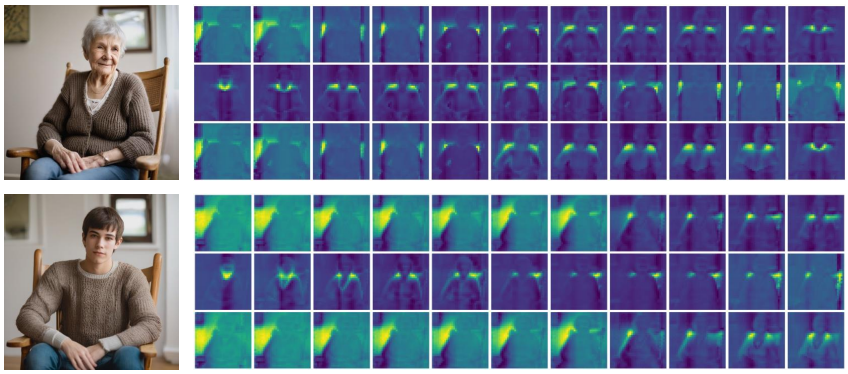


Figure 9: The top components obtained using SVD of self-attention maps for two candidate prompts.

Figure 9 illustrates the self-attention maps for two candidate prompts by varying *gender* and *age* attributes. Specifically, from "An elderly woman with short hair, wearing a knitted sweater, sitting in a rocking chair." to "A young man with short hair, wearing a knitted sweater, sitting in a rocking chair.". We capture the self-attention maps in middle layers and run SVD to obtain the top components.

D.2 Model Statistics

Table 5 presents the quantitative statistics for standalone cloud and device models, including utility scores, parameter size and FLOPs consumption. We adopt 25-step DPM scheduler for all these models by default. For LCM-SDXL, we use 8-step LCM scheduler instead. The FLOPs and latency are measured for a batch size of 4. We run each experiment ten times and report the average results.

Table 5: Model statistics. By default, we use 25-step DPM scheduler. $B = 4$.

Generation Method	FID ↓	IS ↑	CLIP ↑	#Params (M)	FLOPs (T)	Latency (s)
MS-COCO						
SD-v1.4	13.86	37.75	0.3015	859.40	74.10	5.01
BK-SDM-small	18.30	31.73	0.2710	482.28	43.60	3.01
BK-SDM-tiny	18.30	29.94	0.2681	323.34	41.00	2.87
Realistic Vision v4.0	16.21	37.39	0.3033	859.40	74.10	4.42
small-sd	14.59	35.06	0.3046	579.30	44.80	2.96
MJHQ						
SDXL	30.67	26.31	0.3464	2562.13	637.40	29.33
koala-700m	34.11	22.06	0.3263	2562.13	235.40	12.00
LCM-SDXL (8-step)	33.25	28.22	0.3296	777.53	203.97	8.35

D.3 Visualization of Hybrid Generation

In this section, we provide the visualization of generated images for large server model (top), and `ObCLIP` with/without server-side acceleration (middle and bottom) in Figure 10 and Figure 11. The used prompt is "*Faceshot Portrait of pretty young (18-year-old) female Caucasian wearing a high neck sweater*". We mark the two images with an optimal balance between image quality and server computation cost, i.e., when $k \in \{5, 10\}$. The better semantic from large server model are well preserved with minimal server denoising cost.

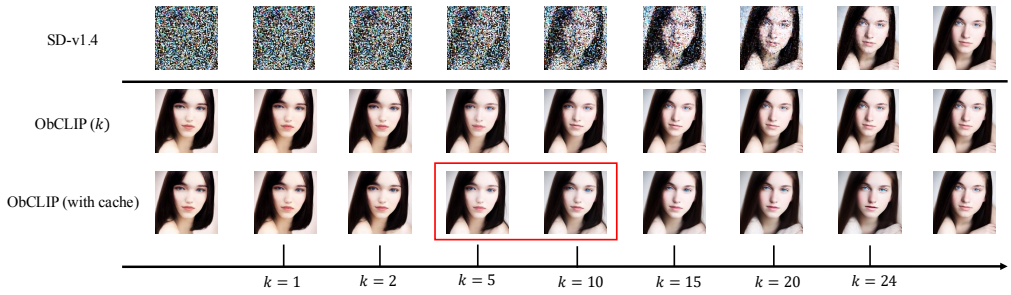


Figure 10: Visualization of SD-v1.4 + BK-SDM-small

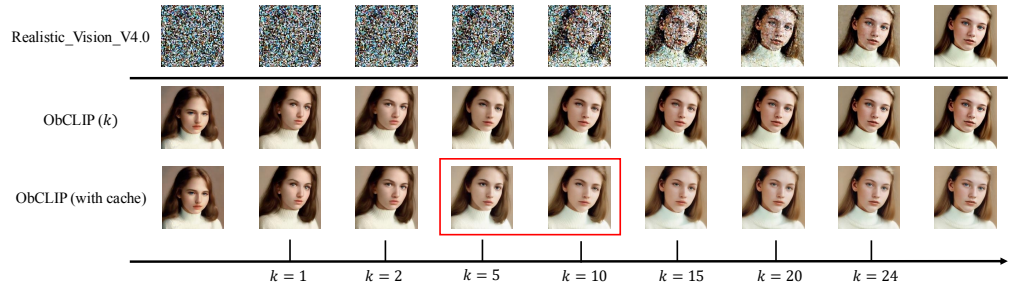


Figure 11: Visualization of Realisti-Vision-V4.0 + small-sd

D.4 3-Attribute Oblivious Hybrid Generation

The detailed utility scores for 3-attribute oblivious hybrid image generation are presented in Table 6. Compared to the large server-side model, there is only a marginal drop in image quality and text-image alignment, while achieving significantly better utility than the on-device small model.

Table 6: Evaluation results for 3-attribute oblivious generation.

Generation Method	**FID** ↓	**IS** ↑	**CLIP** ↑
Realistic	111.87	4.78	0.3322
small-sd	115.96	5.02	0.3034
Vanilla OG (Realistic)	111.87	4.78	0.3322
OG (k = 10)	112.75	5.04	0.3214
OG (k = 10) w cache	113.31	5.03	0.3171
OG (k = 10) w cache + reuse	110.22	4.62	0.3138
OG (k = 5)	113.57	5.02	0.3108
OG (k = 5) (w cache)	114.07	5.04	0.3083
OG (k = 5) (w cache) + reuse	113.30	4.66	0.3077

D.5 Comprehensive Efficiency Evaluation

We here present a more comprehensive efficiency evaluation across different candidate prompt size N to validate the effectiveness of `ObCLIP`. As shown in Table 7, in most cases, `ObCLIP` achieves server-side efficiency comparable to that of server-only image generation on both SD-v1.4 and SDXL models. When accounting for device-side computation, the total latency remains of the same order of magnitude as the baseline. These results demonstrate the superior performance of `ObCLIP`, which offers rigorous privacy with only a slight increase in overall computation cost.

Table 7: Comprehensive efficiency evaluation.

Generation Method		FLOPs				Latency (s)			
		$k=5$		$k=10$		$k=5$		$k=10$	
		$N=4$	$N=6$	$N=4$	$N=6$	$N=4$	$N=6$	$N=4$	$N=6$
SD (SD-v1.4 + BK-SDM-small)									
Non-Private	SD-v1.4	18.53				1.28			
	Hybrid SD	3.71		7.41		0.30		0.51	
Private	HE-Diffusion	-				>106			
	Vanilla OG	74.10	111.15	74.10	111.15	5.01	7.47	5.01	7.47
	ObCLIP	14.82	22.23	29.64	44.46	1.11	1.48	1.93	2.90
	+ cache	12.25	18.38	23.36	35.04	0.67	1.12	1.33	1.85
	+ reuse	11.13	16.64	21.86	32.72	0.61	0.98	1.12	1.55
	Total (+ device)	19.85	25.36	28.40	39.26	1.25	1.62	1.65	2.08
SDXL (SDXL + koala-700m)									
Non-Private	SDXL	159.35				7.45			
	Hybrid SD	31.87		63.74		1.45		2.84	
Private	Vanilla OG	637.40	956.10	637.40	956.10	29.33	43.16	29.33	43.16
	ObCLIP	127.48	191.22	254.96	382.44	5.92	8.43	11.35	16.90
	+ cache	97.62	146.44	180.42	270.62	4.08	6.81	7.98	11.28
	+ reuse	86.32	128.38	165.34	246.55	3.51	5.05	6.81	9.92
	Total (+ device)	133.40	175.46	200.65	281.86	6.07	7.61	8.88	11.99

D.6 Efficiency Improvement Breakdown

Take SD-v1.4 and BK-SDM-small hybrid generation with switch point $k = 10$, cache point $r = 4$, skip point $s = 6$ as an example. We illustrate the per-step FLOPs of server-side denoising in Figure 12, where the reduced computation FLOPs are highlighted in yellow. Prior to the cache point, only batch reuse of attention maps is enabled, resulting in a reduction of approximately 10%. Subsequently, with the addition of temporal attention reuse and block skipping, reductions of about 20% and nearly 50% are achieved, demonstrating the effectiveness of the overall acceleration strategy.

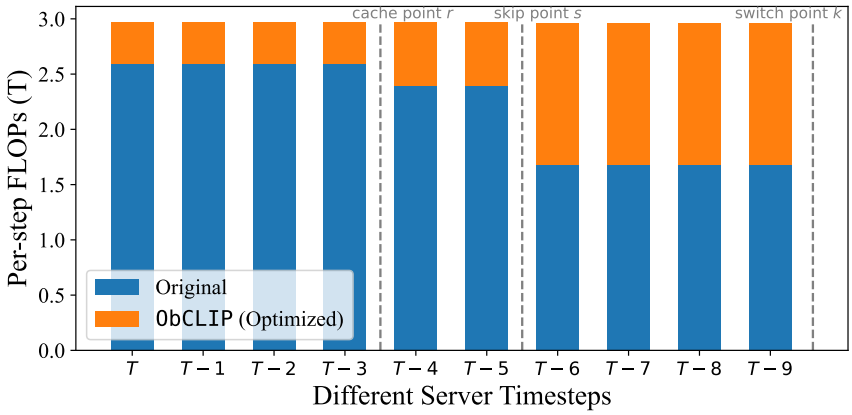


Figure 12: Per-step FLOPs (T) for SD-v1.4 with $k = 10, r = 4, s = 6$.

D.7 Visualization of Step-distilled Model Generation

In this section, we present visualizations of images generated with the server-side LCM-SDXL (optimized using step distillation) and client-side Koala-700m models. Note that we use an 8-step LCM scheduler for LCM-SDXL and a 25-step DPM scheduler for Koala-700m. Recall that as mentioned in Section 5.2, we apply a timestep shift $\triangle t$ to align the timesteps between cloud and device as $t_{device} = t_{cloud} + \triangle t$. We here vary $\triangle t \in \{0, 2, 4, 6, 8\}$. The input prompt is "Faceshot Portrait of a pretty young (18-year-old) male *African* with black hair." To demonstrate the effect of

batch reuse, we choose a candidate prompt: "Faceshot Portrait of a pretty young (18-year-old) male *Caucasian* with black hair." As shown in Figure 13, the final generated images show poor fidelity when $\triangle < 6$, with $\triangle = 8$ yielding the best visual quality. Additionally, the text-image semantics are well aligned.

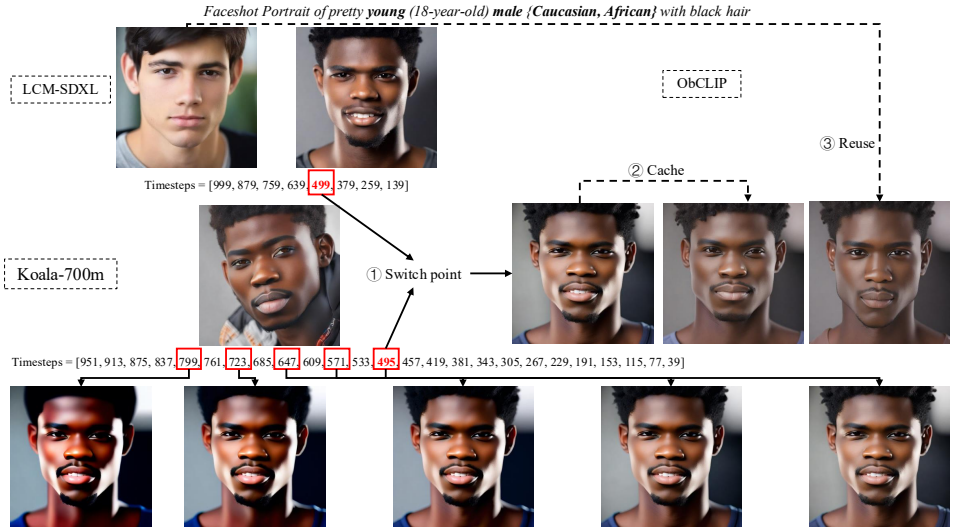


Figure 13: Visualization of LCM-SDXL + Koala-700m.