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]*Independent Researcher*

Motivation

Problem

In the classical MLaaS paradigm, the client sends the prompts (e.g., in text) to the generation services like Midjourney, which typically has large computation power. However, there remains two essential problems in the above scenario:

1. The prompts might inevitably leaks sensitive information (e.g., genders).
2. Server cost increases drastically and becomes a pain point in real-world deployment.

Challenge

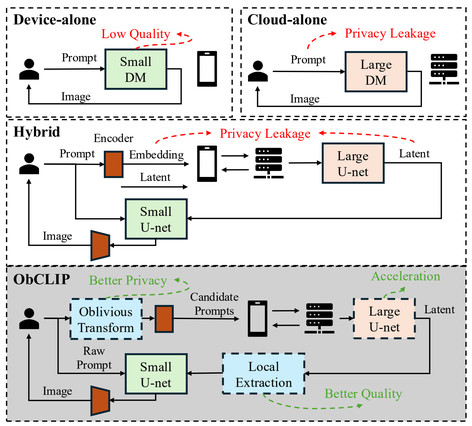
Table 1: Comparison of related work. ●, ◐ and ○ refer to high-, medium- and low-performance.

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	●	◐	◐

- **Server-Only**
  - *Zero privacy*
  - *Huge server cost*
- **Client-Only**
  - *Low utility*
- **Cryptographic**
  - *Low efficiency*
- **Differential Privacy**
  - *Hard to balance privacy and utility*

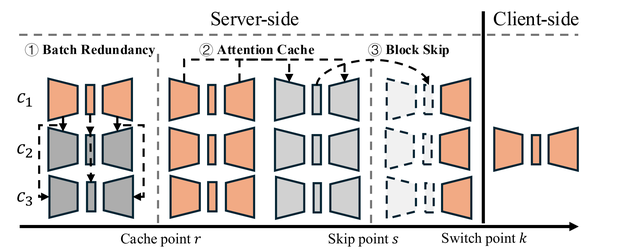
Can we perform privacy-preserving image generation with moderate image quality and lower server cost?

Framework



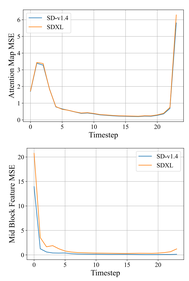
Overview

Quality Score **Quality Visualization**

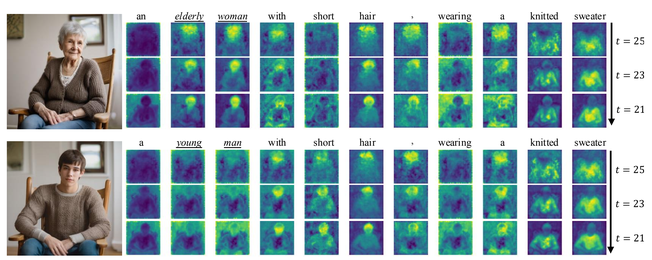


Experimental Results

Redundancy



Temporal Difference



Heatmap of cross-attention maps

Quantification

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.

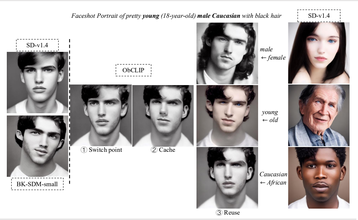
Generation Method	1-Attribute (gender, $N=2$)					2-Attribute (gender + age, $N=6$)				
	FID ↓	IS ↑	CLIP ↑	FLOPs (T)	Latency (s)	FID ↓	IS ↑	CLIP ↑	FLOPs (T)	Latency (s)
Realistic Vision v4.0	113.45	4.69	0.3322	18.53 (+0)	1.12	113.39	5.32	0.3215	18.53 (+0)	1.12
small-sd	128.87	5.04	0.3051	0 (+11.20)	0.78	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

Visualization

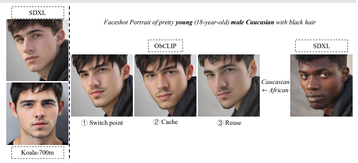
SD-v1.4

SDXL

SD-v1.4 + BK-SDM-small



SDXL + Koala-700m



Contribution

To provide rigorous privacy and comparable utility to large cloud models with slightly increased server cost, we propose

- **Oblivious Cloud-Device Hybrid Generation Scheme**: We introduce oblivious transformation to protect privacy and local extraction with hybrid generation to lower server cost.
- **Temporal- and Batch- Redundancy based Acceleration**: we incorporate cache-based acceleration, leveraging both temporal and batch redundancy, to further reduce server cost with minimal utility degradation.