CipherDM: Secure Three-Party Inference for Diffusion Model Sampling

Xin Zhao[1,2,3], Xiaojun Chen[1,2,3,*], Xudong Chen[1,2,3],

He Li[1,2,3],Tingyu Fan [1,2,3] and Zhendong Zhao[1,2,3]

[1]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

[2]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

[3]Key Laboratory of Cyberspace Security Defense, Beijing, China

Explain the mechanism of diffusion models.

Diffusion models are a type of generative model that generate data (e.g., images, audio, etc.) by reversing a gradual, noisy process. The general idea is to learn to denoise data progressively. These models are popular for their ability to generate high-quality images and other types of complex data.

Basic Mechanism of Diffusion Models

1. **Forward Process (Diffusion Process):**
   - This process starts with clean data (e.g., an image) and progressively adds noise to it over several time steps, making the data increasingly noisy until it turns into pure noise (e.g., Gaussian noise).
   - Let x_0 be the original data (e.g., an image). At each step t, noise is added according to a pre-defined noise schedule to produce a sequence of data $x_1, x_2, \ldots, x_T$, where x_T is almost pure noise. The forward process typically follows a Markov chain, and the transformation at each step is simple, often modeled as:

$$x_t = \sqrt{1-\beta_t} \cdot x_{t-1} + \sqrt{\beta_t} \cdot \epsilon,$$

I’ m a student. I need help for my course paper: how to …

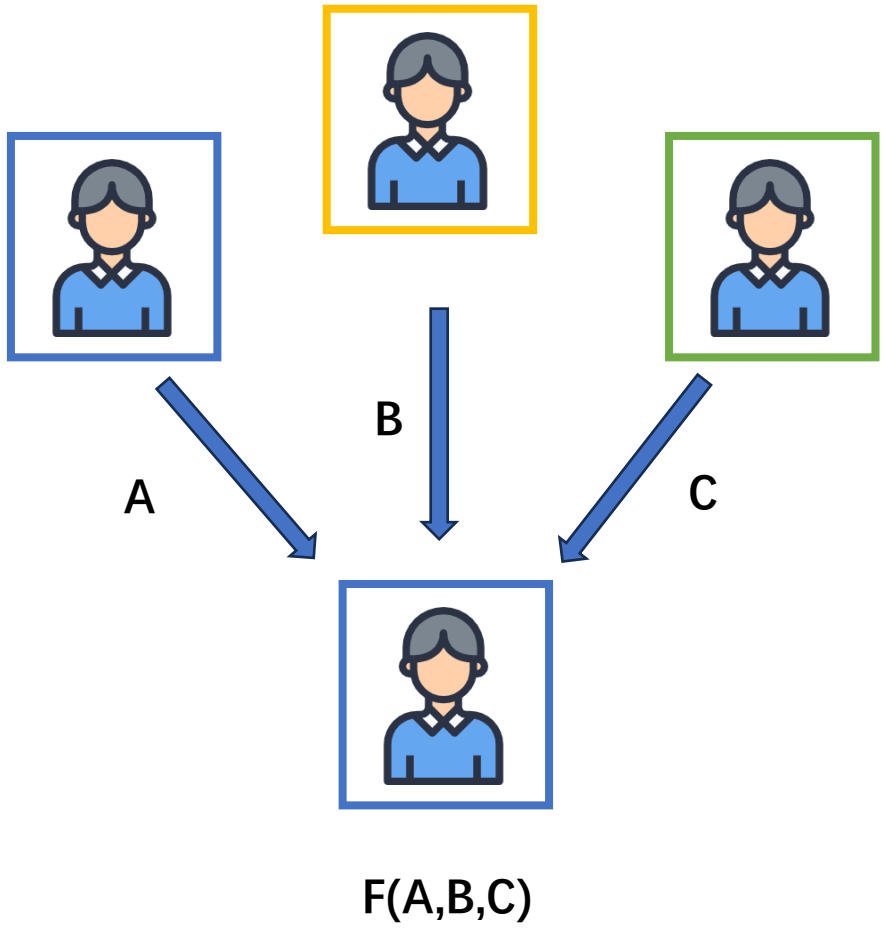
!

Please change the background color of my ID photo.

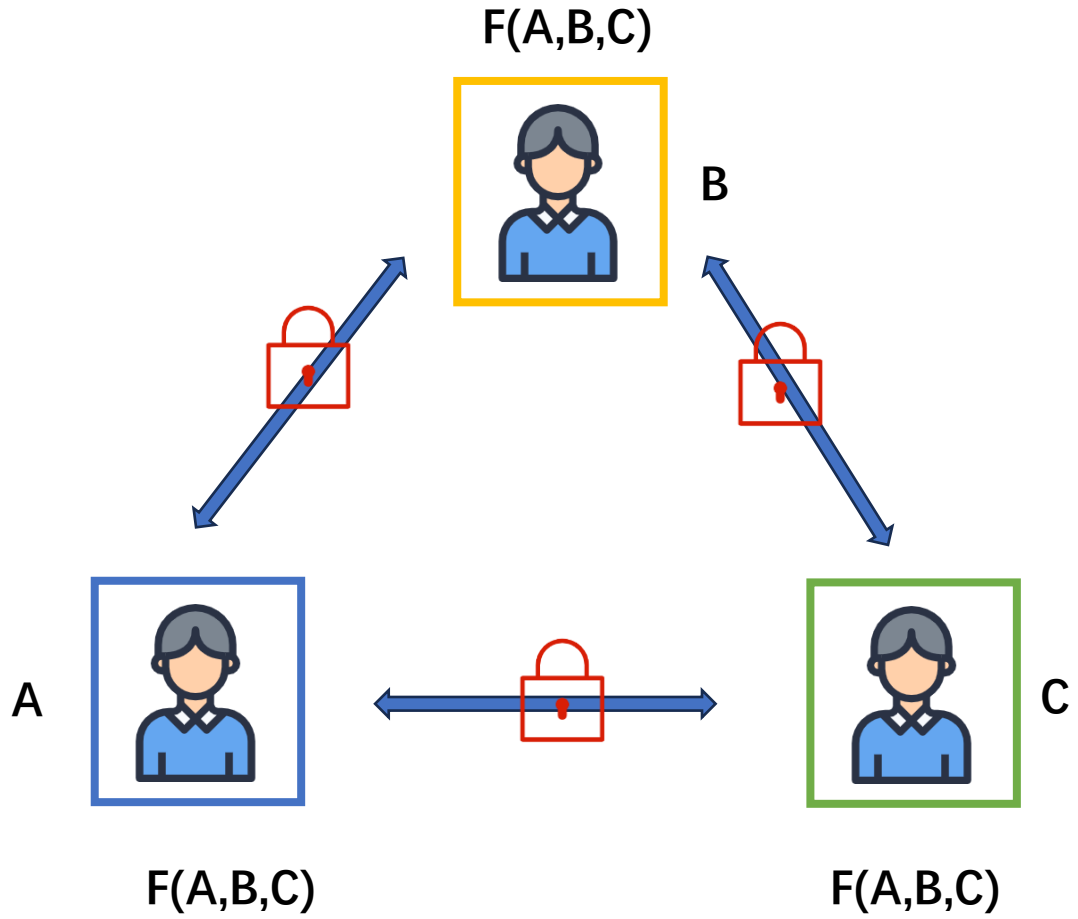
!

!

Train and upload the model to the websites.



Central Trusted Authority



Secure Multiparty Machine Learning

Protected Data Fields

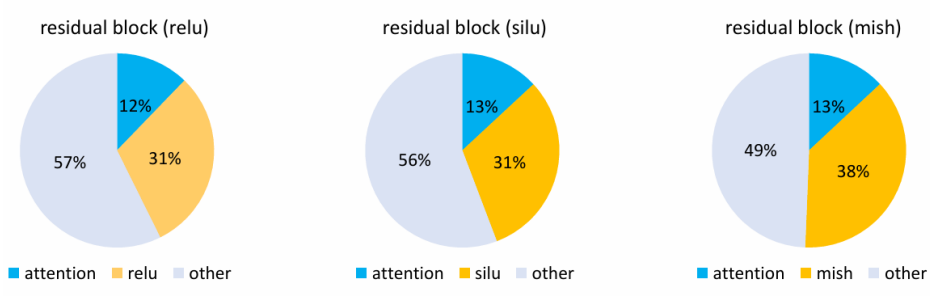


Fig. 1: Module running time percentage of residual block in plaintext.

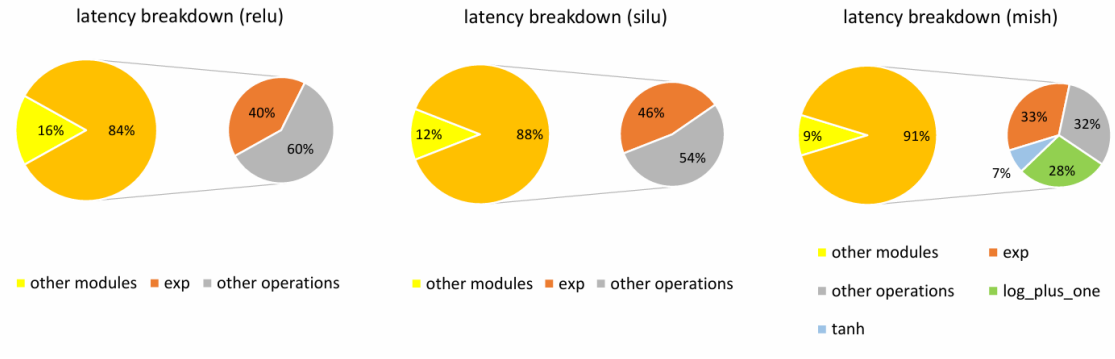


Fig. 2: Latency breakdown of total workflow in ciphertext.

Contributions:

- **Secure DM Sampling**: We first leverage MPC technology to ensure confidentiality and privacy of the sampling phase.
- **Unified Secure DM**: It's confirmed that MPC can be effectively implemented on commonly used DMs, including DDPM, DDIM and SD.
- **Optimized Secure Nonlinear Operators**: Efficient activation protocols (SoftMax, SiLU and Mish) are specifically designed for DMs.

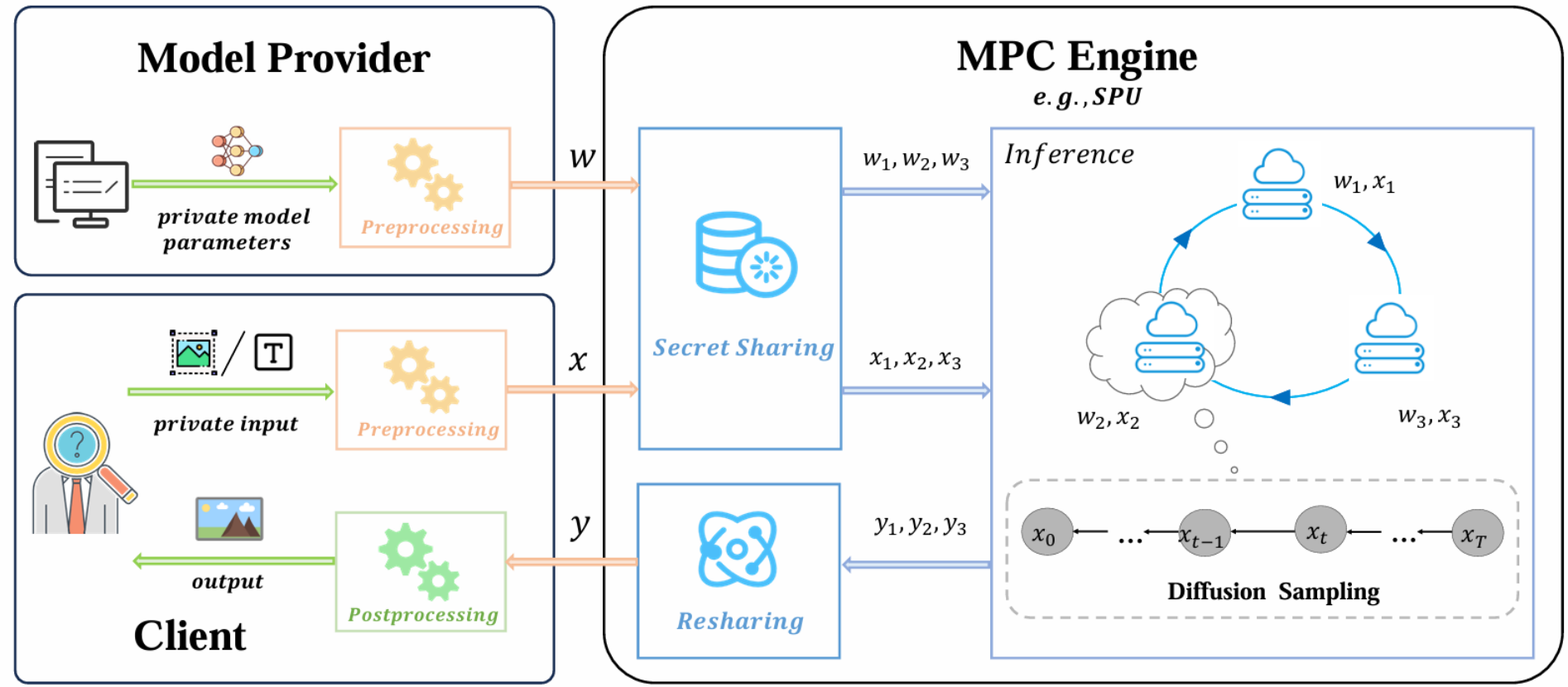


Fig. 3: An illustration of our proposed CipherDM framework. CipherDM takes model parameters and images/texts as two private inputs, preprocesses them locally, secretly shares them to a three-party MPC Engine, and receives the final sampling result from it. MPC systems such as SPU involve the joint computation.

Secure SoftMax

$$negExp(x) = \begin{cases} 0, x < T_{exp} \\ Chebyshev(x), x \in [T_{exp}, 0] \end{cases}$$

$$Chebyshev(x) = C_0 T_0(x_t) + C_1 T_1(x_t) + \cdots + C_7 T_7(x_t)$$

Table 3: The coefficients and Chebyshev polynomials of exponential function.

i	0	1	2	3	4	5	6	7
$C_i(0.)$	14021878	27541278	22122865	14934221	09077360	04369614	02087868	00996535
$T_i(x)$	1	x	$2x^2 - 1$	$4x^3 - 3x$	$8x^4 - 8x^2 + 1$	$16x^5 - 20x^3 + 5x$	$32x^6 - 48x^4 + 18x^2 - 1$	$64x^7 - 112x^5 + 56x^3 - 7x$

Algorithm 1 Secure SoftMax Protocol $\prod_{\text{SoftMax}}$

Input: P_i holds the 2-out-of-3 replicate secret share $[\![\mathbf{x}]\!]$ for $i \in \{0, 1, 2\}$, and $\mathbf{x}$ is a vector of size n.

Output: P_i gets the 2-out-of-3 replicate secret share $[\![\mathbf{y}]\!]$ for $i \in \{0, 1, 2\}$, where $\mathbf{y} = \text{SoftMax}(\mathbf{x})$.

1: P_0, P_1, P_2 jointly compute $[\![\mathbf{b}]\!]^{\text{B}} = \prod_{\text{LT}}(\text{T}_{\text{exp}}, [\![\text{x}]\!])$ and the maximum $[\![\bar{x}]\!] = \prod_{\text{Max}}([\![\mathbf{x}]\!])$.
2: Locally compute $[\![\hat{\mathbf{x}}]\!] = [\![\mathbf{x}]\!] - [\![\bar{\text{x}}]\!] - \epsilon$ and $[\![\mathbf{t}]\!] = -2 * ([\![\hat{\mathbf{x}}]\!] - \text{T}_{\text{exp}}) * \text{T}_{\text{exp}}^{-1} - 1$.
3: **for** i = 2, 3, ..., 7 **do**
4: Jointly compute $[\![\mathbf{t}^{\text{i}}]\!] = \prod_{\text{Mul}}([\![\mathbf{t}^{\text{i}-1}]\!], [\![\mathbf{t}]\!])$ and $[\![T_i(\mathbf{t})]\!]$ based on $[\![\mathbf{t}^{\text{i}}]\!]$ as Tab. 3.
5: **end for**
6: Locally compute $[\![\mathbf{z}]\!] = \sum_{\text{j}=0}^{7} \text{C}_{\text{j}} [\![\text{T}_{\text{j}}(\mathbf{t})]\!]$ and $[\![z]\!] = \sum_{k=1}^{n} [\![\mathbf{z}[k]]\!]$.
7: Jointly compute $[\![1/z]\!] = \prod_{\text{Recip}}([\![z]\!])$ and $[\![\mathbf{z}/z]\!] = \prod_{\text{Mul}}([\![\mathbf{z}]\!], [\![1/z]\!])$.
8: **return** $[\![\mathbf{y}]\!] = \prod_{\text{Mul}_{\text{BA}}}([\![\mathbf{b}]\!]^{\text{B}}, [\![\mathbf{z}/z]\!])$.

Secure Activations

$$Activation(x) = \begin{cases} 0, x < -6 \\ F_0(x), -6 \le x < -2 \\ F_1(x), -2 \le x \le 6 \\ x, x > 6 \end{cases}$$

SiLU
$$\begin{cases} F_0(x) = -0.01420163x^2 - 0.16910363x - 0.52212664 \\ F_1(x) = 0.00008032x^6 - 0.00602401x^4 + 0.19784596x^2 + 0.49379432x \\ \qquad +0.03453821 \end{cases} \quad (6)$$

Mish
$$\begin{cases} F_0(x) = -0.01572019x^2 - 0.18375535x - 0.55684445 \\ F_1(x) = 0.00010786x^6 - 0.00735309x^4 + 0.20152583x^2 + 0.54902050x \\ \qquad +0.07559242 \end{cases} \quad (7)$$

Algorithm 2 Secure SiLU and Mish Protocol $\prod_{\text{SiLU}}/\prod_{\text{Mish}}$

Input: P_i holds the 2-out-of-3 replicate secret share $[\![x]\!]$ for $i \in \{0,1,2\}$.

Output: P_i gets the 2-out-of-3 replicate secret share $[\![y]\!]$ for $i \in \{0,1,2\}$, where $y = \text{SiLU(x)/Mish(x)}$.

1: P_0, P_1, P_2 jointly compute $[\![b_0]\!]^B, [\![b_1]\!]^B, [\![b_2]\!]^B$ and $[\![z_0]\!]^B, [\![z_1]\!]^B, [\![z_2]\!]^B$ where

$$\begin{array}{lll} [\![b_0]\!]^B = \prod_{\text{LT}}([\![x]\!], -6), & [\![b_1]\!]^B = \prod_{\text{LT}}([\![x]\!], -2), & [\![b_2]\!]^B = \prod_{\text{LT}}(6, [\![x]\!]), \\ [\![z_0]\!]^B = [\![b_0]\!]^B \oplus [\![b_1]\!]^B, & [\![z_1]\!]^B = [\![b_1]\!]^B \oplus [\![b_2]\!]^B \oplus 1, & [\![z_2]\!]^B = [\![b_2]\!]^B. \end{array}$$

Note that $z_0 = 1\{-6 \le x < -2\}, z_1 = 1\{-2 \le x \le 6\}$ and $z_2 = 1\{x > 6\}$.

2: Jointly compute $[\![x^2]\!] = \prod_{\text{Square}}([\![x]\!])$, $[\![x^4]\!] = \prod_{\text{Square}}([\![x^2]\!])$, and $[\![x^6]\!] = \prod_{\text{Mul}}([\![x^2]\!], [\![x^4]\!])$.

3: Locally compute polynomials $[\![F_0(x)]\!]$ and $[\![F_1(x)]\!]$ based on $[\![x^i]\!]$ as Eq. (6) / Eq. (7).

4: **return**
$[\![y]\!] = \prod_{\text{Mul}_{\text{BA}}}([\![z_0]\!]^B, [\![F_0(x)]\!]) + \prod_{\text{Mul}_{\text{BA}}}([\![z_1]\!]^B, [\![F_1(x)]\!]) + \prod_{\text{Mul}_{\text{BA}}}([\![z_2]\!]^B, [\![x]\!])$.

Table1: Total time costs of sampling one single image.

Framework		ReLU		SiLU		Mish	
		Local	LAN	Local	LAN	Local	LAN
DDPM	CPU	373	377	370	369	372	376
	SPU	9879	17265	13054	24042	17698	31002
	CipherDM	9473	16669	9688	17915	9520	16686
	Improv.	1.043×	1.037×	1.347×	1.342×	1.859×	1.858×
DDIM	CPU	28	27	28	27	28	28
	SPU	587	949	808	1250	1047	1781
	CipherDM	534	910	648	1088	576	979
	Improv.	1.099×	1.043×	1.247×	1.241×	1.818×	1.819×

Table2: MPC time and communication costs of sampling one single image.

Framework		ReLU		SiLU		Mish	
		Time	Comm.	Time	Comm.	Time	Comm.
Local (DDPM)	SPU	7076	186.18	9794	268.04	14295	391.11
	CipherDM	6069	198.61	7278	198.72	6405	230.33
	Improv.	1.166×	0.937×	1.346×	1.349×	2.232×	1.698×
LAN (DDPM)	SPU	14816	186.10	20679	268.04	27586	391.07
	CipherDM	12174	198.40	14583	198.84	12660	230.31
	Improv.	1.217×	0.938×	1.418×	1.348×	2.179×	1.698×
Local (DDIM)	SPU	492	9.13	712	13.42	957	19.84
	CipherDM	413	10.00	474	11.07	411	11.07
	Improv.	1.191×	0.915×	1.502×	1.212×	2.328×	1.791×
LAN (DDIM)	SPU	855	9.13	1260	13.44	1678	19.85
	CipherDM	789	9.56	907	11.08	795	11.09
	Improv.	1.084×	0.917×	1.389×	1.213×	2.111×	1.790×

Table3: Time costs of U-Net in Flax Stable Diffusion from Diffusers.

Numsteps	CPU	SPU	CipherDM	Improv.
1	44	8332	7711	1.081×
5	81	8856	8208	1.079×

Table4: FID of 10k images generated by CPU and CipherDM in plaintext.

FID	ReLU	SiLU	Mish
CPU	110.13	79.46	86.24
CipherDM	272.26	252.12	202.64

Fig4: Images generated by CPU and CipherDM.

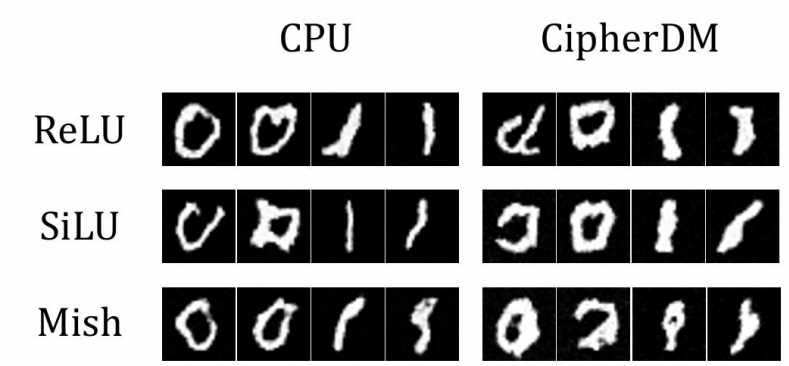


Fig5: The impact of each module on total time improvement.