CONTENTS

Table 5: Some important notations in Sec. 2

Notations	Description
μ	the distribution of the subset of the training data
μ_T	a subset of the contributors
G	generative model
G^*	well-trained generative model
$\mathcal{Z}$	noise distribution
z	a latent sample in the latent space
$d(\cdot,\cdot)$	distance function
X, x	training dataset, training sample
$\hat{X}, \hat{x}$	generated dataset, generated sample
S	a subset of the training data
S^*	the real K contributors
K	$K = \lvert S^* \rvert$
T	K contributors found by a data valuator
$\mathcal{A}$	attribute space
$\mathcal{X}$	data distribution
$\mathcal{L}$	loss function
f, f'_{S^*}	labeling function, model trained on S^*

Roadmap of Appendix In this appendix, we provide a concise summary of key notations and Algorithm 1, detailing the pipeline of GMValuator in Sec. A and Sec. B. A more comprehensive review of related work is also provided in Sec. C. Additionally, Sec. D contains detailed statistical analysis for the results depicted in Figure 1. Our main text focuses on experiments conducted on benchmark datasets such as MNIST and CIFAR-10, along with a large-scale dataset, ImageNet, to evaluate the most significant contributors found by GMVALUATOR are in the same class as the generated sample (**C1**). In Sec. E, we extend the evaluation of GMVALUATOR to other generative models for **C1**. Additionally, we expand our evaluation using **C2** by considering the ground truth of multiple combined attributes. We operate under the assumption that the most significant contributors to a generated sample should exhibit similar attributes in the images. For a more thorough validation of the effectiveness of GMVALUATOR, high-resolution datasets like AFHQ and FFHQ are also employed for **C2**. The appendix includes additional experimental insights such as ablation studies, alternative embedding approaches, alternative distance metrics, calibration and further experimental

details in Sec. F, H, I and J, respectively. The potential applications, limitations and future directions are also provided in Sec. K and Sec. L. Lastly, we present the proof of our theorem and ensure the reproducibility of our experiments.

A NOTATION

To make motivation and problem formulation clear, we list some significant notations from Sec. 2 in Table 5.

B ALGORITHM

To better introduce GMVALUTOR, the pseudocode is shown in Algorithm 1.

Algorithm 1 GMVALUATOR

Input: Training dataset $X = \{x_i\}_{i=1}^n$, a well-trained model G^*, random distribution $\mathcal{Z}$.
Output: Generated dataset $\hat{X}$, the value of training data points $\Phi = \{\phi_1, \phi_2, ..., \phi_n\}$

$\hat{X} = \{\hat{x}_j\}_{j=1}^m \leftarrow G^*(z_j)$, for $z_j \in \mathcal{Z}$ *// Generate the synthetic dataset*
for $\hat{x}_j$ in $\hat{X}$ **do**
  // Matching process (see Sec. 3.1)
  $\mathcal{P}_j = f(X, \hat{x}_j)$ *// Including two phases*
  for x_i in $\mathcal{P}_j$ **do**
    $d_{ij} \leftarrow \mathrm{DreamSim}(x_i, \hat{x}_j)$ or others
  end for
  $q_j = MANIQA(\hat{x}_j)$ *// Image Quality Assessment (see Sec. 3.2)*
  Calculate score $\mathcal{V}(x_i, \hat{x}_j, d_{ij}, q_j)$ using Eq. equation 6 *// Contribution Score Calculation (see Sec. 3.3)*
end for
// Calculation of data value and return the result Φ
for x_i in X **do**
  Calculate x_i's value ϕ_i using Eq.equation 5
end for
return $\Phi = \{\phi_1, \phi_2, ..., \phi_n\}$

C RELATED WORK

C.1 DATA VALUATION

There are three lines of methods on data valuation: *metric-based methods*, *influence-based methods* and *data-driven methods*. In terms of *metric-based methods*, the commonly-used approach is to calculate its *marginal contribution* (MC) based on performance metrics (*e.g.,* accuracy, loss). As the basic method depending on performance metrics for data valuation, LOO (Leave-One-Out) Cook (1977) is used to evaluate the value of the training sample by observational change of model performance when leaving out that data point from the training dataset. To overcome inaccuracy and strict desirability of LOO, SV Ghorbani & Zou (2019) and BI Wang & Jia (2023) originated from *Cooperative Game Theory* are widely used to measure the contribution of data Jia et al. (2019b); Ghorbani et al. (2020). Considering the joining sequence of each training data point, SV needs to calculate the marginal performance of all possible subsets in which the time complexity is exponential. Despite the introduction of techniques such as Monte-Carlo and gradient-based methods, as well as others proposed in the literature, approximating data significance value (SV) is computationally expensive and it typically requires retraining Ghorbani & Zou (2019); Jia et al. (2019a). The computational cost and need for unconventional performance metrics present difficulties in adapting the methods to generative models. As for *influence-based methods*, they evaluate the influence of data points on model parameters by computing the inverse Hessian for data valuation Jia et al. (2019a); Richardson et al. (2019); Saunshi et al. (2022). Due to the high computational cost,

Table 6: The statistic test of data values of X_{v1} versus X_{v2} using different generative models. X_{v1} is supposed to have higher value than X_{v2}, given the generated data.

$$H_0\text{: } \phi(D_i, S, \mu_i) \geq \phi(D_j, S, \mu_i)$$
$$H_1\text{: } \phi(X_i, S, \mu_i) < \phi(X_j, S, \mu_i), i \in X_{v1}, j \in X_{v2}$$

	BigGAN	**Classifier-free Guidance Diffusion**
Average value (v1)	0.319654	0.030434
Average value (v2)	1.632352	0.369565
P-value	6.937027×10^{-68}	8.053195×10^{-55}
T-statistic	17.924512	15.947860
Significance level	0.01	0.01
Result	p-value less than 0.01, reject H_0, value of v2 less than v1 averagely	p-value less than 0.01, reject H_0, value of v2 less than v1 averagely

some approximation methods have also been proposed Pruthi et al. (2020). In addition, the use of influence function for data valuation is not limited to discriminative models, but can also be applied to specific generative models such as GAN and VAE Terashita et al. (2021); Kong & Chaudhuri (2021). When it comes to data-driven methods, most of them are training-free methods that focus on the data itself Xu et al. (2021); Wu et al. (2022); Just et al. (2023).

C.2 Generative Model

Generative models are a type of unsupervised learning that can learn data distributions. Recently, there has been significant interest in combining generative models with neural networks to create *Deep Generative Models*, which are particularly useful for complex, high-dimensional data distributions. They can approximate the likelihood of each observation and generate new synthetic data by incorporating variations. Variational auto-encoders (VAEs) Rezende et al. (2014) optimize the log-likelihood of data by maximizing the evidence lower bound (ELBO), while generative adversarial networks (GANs) Goodfellow et al. (2020); Karras et al. (2020) involves a generator and discriminator that compete with each other, resulting in strong image generation. Recently proposed diffusion models Ho et al. (2020); Rombach et al. (2022) add Gaussian noise to training data and learn to recover the original data. These models use variational inference and have a fixed procedure with a high-dimensional latent space.

D Statistical Results for Figure 1

It is evident by visualization in Figure 1 that the data points in X_{v2} (used for training) are more overlapped with generated data than data points in X_{v1} (not used for training). We perform statistic testing on data values obtained by GMValuator, to examine if data points X_{v2} (used for training) have significantly higher values than those of the data points in X_{v1} (not used for training).

To this end, we use a t-test with the null hypothesis that data values in X_{v1} should not be smaller than those of X_{v12}. We compute p-value, which is the probability of getting a difference as large as we observed, or larger, under the null hypothesis. If the p-value is very low, we reject the null hypothesis and consider our approach, GMValuator, to be verified with a high level of confidence (1-p). Typically, a p-value smaller than significance level 0.01 is used as a threshold for rejecting the null hypothesis. Table 6 showcases the outcomes of X_{v1} and X_{v2} in CIFAR-10 with $p \ll 0.01$ for both BigGAN and diffusion model, indicating that the data points in X_{v2} have significantly more value than those in X_{v1}. Consequently, these findings align with the presumption that the trained dataset X_{v2} has a higher value than the untrained dataset X_{v1} and verify our approach.

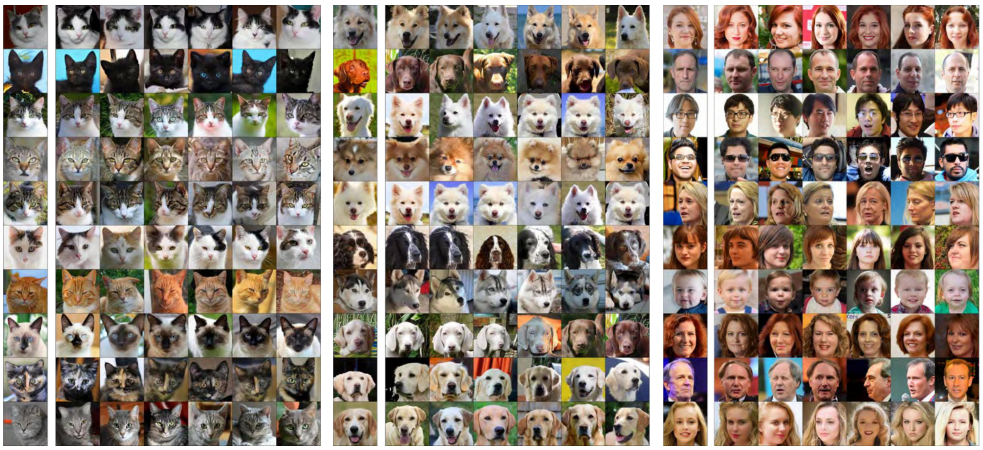

Figure 6: Visualization of Identical Attributes Test on AFHQ and FFHQ. The results shown in the first and second subfigures on the left are conducted on AFHQ-Cat and AFHQ-Dog, respectively. The subfigure on the right presents the results of FFHQ. In each subfigure, the generated samples are on the left, and the top k contributors in the training dataset are on the right.

Table 7: Performance comparison of Identical Class Test.

MNIST			
GAN (%)	k=30	k=50	k=100
GMValuator (No-Rerank)	96.27	96.26	95.86
GMValuator (l_2-distance)	97.73	97.58	96.03
GMValuator (LPIPS)	97.77	97.72	97.38
GMValuator (DreamSim)	97.43	97.44	97.40
Diffusion (%)	k=30	k=50	k=100
GMValuator (No-Rerank)	92.40	91.82	91.26
GMValuator (l_2-distance)	92.90	92.66	91.88
GMValuator (LPIPS)	93.73	97.72	92.42
GMValuator (DreamSim)	93.90	93.44	92.55
CIFAR-10			
BigGAN (%)	k=30	k=50	k=100
GMValuator (No-Rerank)	64.70	63.80	62.14
GMValuator (l_2-distance)	64.70	63.80	62.14
GMValuator (LPIPS)	63.67	62.80	61.51
GMValuator (DreamSim)	70.33	68.74	65.18
Class-free Guidance Diffusion (%)	k=30	k=50	k=100
GMValuator (No-Rerank)	72.67	72.00	71.00
GMValuator (l_2-distance)	72.67	72.00	71.00
GMValuator (LPIPS)	72.53	72.28	71.06
GMValuator (DreamSim)	79.37	78.08	74.61

E Additional Results on C1 and C2

E.1 (C1) Identical Class Test on Other Generative Models

We have presented Identical Class Test (C1) on β-VAE and MNIST LeCun et al. (1998), CIFAR-10 Krizhevsky et al. (2009) in Sec. 4.3 in our main context. Since GMValuator is model-agnostic, we further validate our method of **C1** on other generative models.

Here, we conduct the experiments using a GAN and a Diffusion Model on MNIST. The architectural details of the used generative models are described in Sec. J.3 in the appendix. We also conduct the experiment on BigGAN Brock et al. (2018) and Class-free Guidance Diffusion Ho & Salimans

(2022) with CIFAR-10. We used the same number of generated samples $m = 100$ as the experiments presented in Sec. 4.

Following the similar settings in Sec. 4.3 (**C1**), we examine the class(es) of is top k contributors for a given generated data in the training data. We calculate the number of training samples, denoted as Q, from the top k contributors that have the same class as the generated data. The identical class ratio, denoted as ρ, is calculated as $\rho = Q/k$. We report the average value of ρ across the generated datasets for different choices of k in Table 7. GMVALUATOR (DreamSim) has the highest ratio of contributors that belong to the same class as the generated sample among most of the models evaluated on MNIST and CIFAR-10 datasets for different values of k. And the ratio improves as the value of k decreases, which is consistent with the top k assumption and validates our method.

Table 8: Performance of Identical Attributes Test (C2) of multiple combined attributes.

Top K contributors:	k=5	k=10	k=15
Attribute: *Eyeglasses & Gender* (%)			
GMValuator (No-Rerank)	50.10	48.48	48.42
GMValuator (l_2-distance)	59.19	56.67	56.23
GMValuator (LPIPS)	78.18	74.24	73.87
GMValuator (DreamSim)	92.53	90.61	90.30
Attribute: *Eyeglasses & Hat* (%)			
GMValuator (No-Rerank)	78.79	78.38	85.86
GMValuator (l_2-distance)	84.44	82.93	83.23
GMValuator (LPIPS)	86.87	86.16	86.33
GMValuator (DreamSim)	89.49	87.58	87.41
Attribute: *Gender & Hat* (%)			
GMValuator (No-Rerank)	58.59	57.47	57.71
GMValuator (l_2-distance)	61.62	60.51	60.40
GMValuator (LPIPS)	63.84	63.23	62.96
GMValuator (DreamSim)	65.25	64.44	64.18

E.2 (C2) IDENTICAL ATTRIBUTES TEST ON CELEBA

We extend **C1** to focus on the attributes present in the images, treating them as ground truth rather than class labels, as discussed in Sec. 4.4. Our experiments now incorporate multiple attributes simultaneously, rather than just a single attribute, to verify the performance of GMVALUATOR. For instance, when evaluating a generated image with both a hat and eyeglasses, the most significant contributors identified should also include both of these attributes. Table 8 showcases some outcomes of identical attributes test when regarding multiple combined attributes as ground truth, which validate the effectiveness of our methods.

E.3 (C2) IDENTICAL ATTRIBUTES TEST ON OTHER DATASETS

To further validate GMVALUATOR, we also conducted experiments on high-resolution datasets: AFHQ Choi et al. (2020) and FFHQ Karras et al. (2019), following the same settings in Sec 4.4. The results are shown in Figure 6, which demonstrates the effectiveness of our methods in **C2**. The results show that the top k contributors have similar attributes with the generated sample such as fur color of cats or dogs in AFHQ. For the experiment conducted on FFHQ, human faces attributes of the most significant contributors are also similar to the attributes in generated images.

F DIFFERENT GENERATED DATA SIZES

Since our value function ϕ_i (Eq. equation 5) for training data x_i is computed by averaging over generated samples, it is expected that the sensitivity of ϕ_i is connected to the size m of the investigated generated sample. To explore the influence of generated data size m on the utilization of GMVALUATOR, we perform sensitivity testing on MNIST, CIFAR10 using generative models GAN Goodfellow et al. (2020), and Diffusion models Dhariwal & Nichol (2021) as depicted below. The dataset, model and used k are denoted under each subfigure of Figure 7. First, we generate a

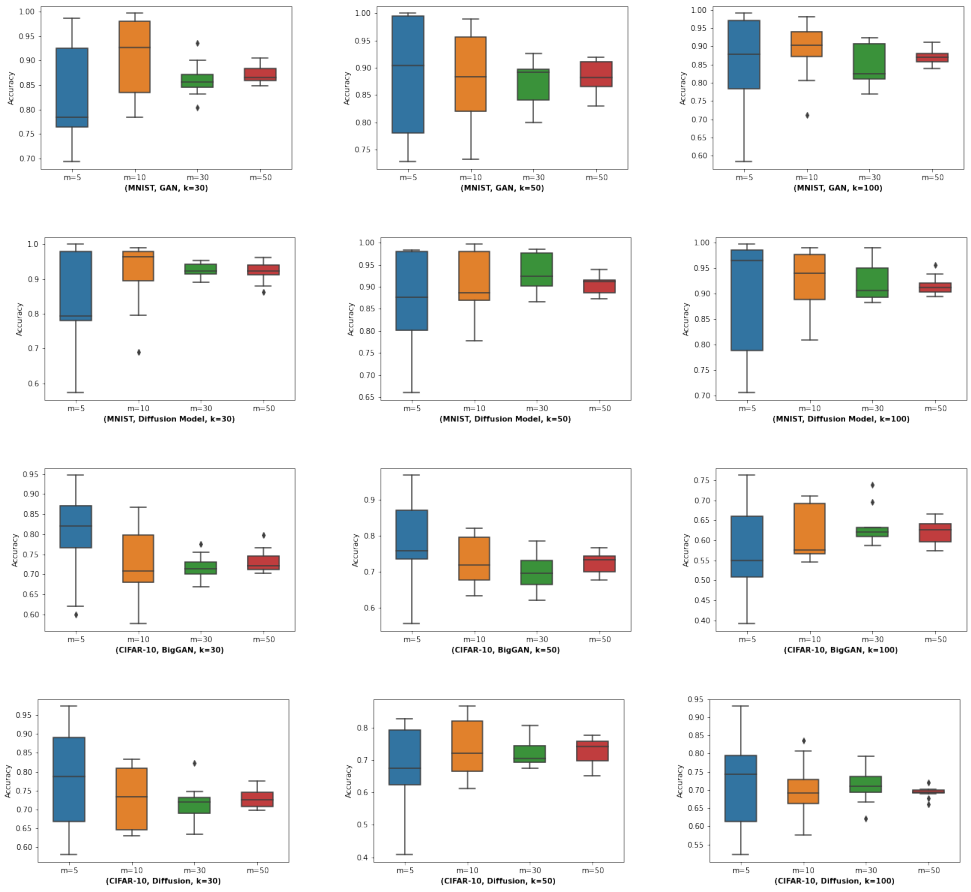

Figure 7: The change of ρ with the different number of generated samples m on MNIST and CIFAR-10 by diverse generative models.

varying number of samples from the same class. Specifically, we consider four different sample sizes, denoted by m, which are given by 1, 10, 30, and 50. Next, we evaluate the GMVALUATOR using parameter **C1**, and this evaluation is performed for each of the aforementioned values of m. Subsequently, we conduct the experiment 10 times using GMVALUATOR (No-Rerank), each time with different m-sized generated data samples from the same class. The results are presented as the mean and standard deviation (ρ) for accuracy, taken over these 10 runs. The results shown in Figure 7 imply that varying m does not yield notable differences in mean accuracy and increasing the number of generated samples m leads to more stable and consistent results.

G ALTERNATIVE EMBEDDING APPROACHES

In the step of efficient similarity matching, the embedding f_e is exclusively used in the recall phase for retaining n samples with non-minimal contributions. Apart from using CLIP for embedding, we also investigate the influence for the results when using other embedding methods. We present results in Table 9 using more embedding methods, showing similar performance with CLIP with "Rerank". The reason for this results is that we implement re-ranking that *relies on the image space rather than the embedding space*, which allows us to derive accurate top k contributors from n samples $n \gg k$. Thus, the embedding could tolerate some noise in the recall phase.

Table 9: Comparison of different embedding methods.

MNIST (%)	No-Rerank	l_2-distance	LPIPS	DreamSim
CLIP	86.41	87.76	88.78	88.78
Alexnet	79.77	84.82	86.51	88.72
Densenet	80.47	86.77	87.97	89.35

H ALTERNATIVE DISTANCE METRIC

We suggest utilizing Learned Perceptual Image Patch Similarity (LPIPS) Zhang et al. (2018) or DreamSim Fu et al. (2023) as the distance metric d during the re-ranking phase. This enables to understand the perceptual dissimilarity between generated and real data points. These metrics are applied in the image embedding space derived from pre-trained models. Alternatively, distance measurement can be based on pixel space, such as employing $l2$-distance. Notably, we find that distances calculated in the input pixel space yield comparable outcomes as shown in Table 7. This implies that our method GMVALUATOR could be flexible to the different choices of distance metrics and the selection can depend on data prior.

I NECESSITY FOR CALIBRATION

For challenges 2 and 3, we utilize image quality assessment and establish a non-zero scores rule for calibration in GMVALUATOR. To better understand the impact and necessity of calibration, we compare the distribution of the top 1, 2, and 3 contributors' scores with (w) and without (w/o) calibration conducted on MNIST in Figure 8. It is evident that scores without calibration are generally higher than with calibration. We also extend C3 and perform an ablation study in Figure 9 on scenarios where the generated samples include low-quality outputs. The results show that the OOD (out-of-distribution) training samples' value rank by GMValuator without (w/o) calibration is smaller, indicating significant bias and poor performance.

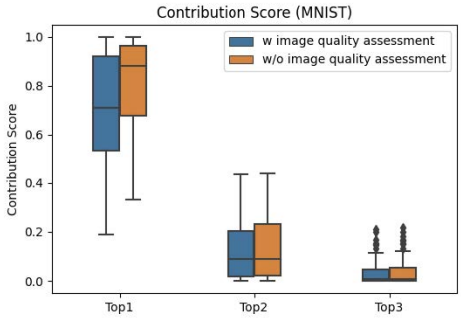

Figure 8: Sore Distribution

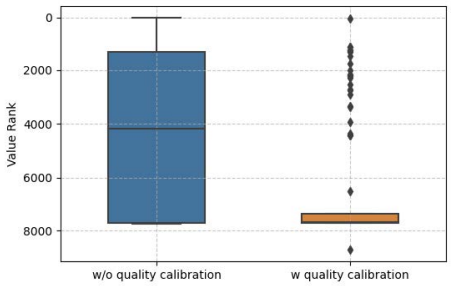

Figure 9: Value Rank

J ADDITIONAL EXPERIMENTAL DETAILS

J.1 JUSTIFICATION OF EXPERIMENT SETUP

We provide a detailed justification in Table 10 for our experiment setup from **C1** to **C4**.

- **C1.** In Identical Class Test, the baseline method is VAE-TracIn Kong & Chaudhuri (2021), which can find the most influenced instances in the training dataset. Since VAE-TracIn is the model-specific method, we only need to compare it with GMVALUATOR when VAE model is used. Besides, all the datasets used in **C1** should have the class labels. Considering the computational demands detailed in VAE-TracIn Kong & Chaudhuri (2021), the runtime complexity of VAE-TracIn correlates with the number of network parameters and the size of

the dataset. Therefore, our analysis primarily utilizes simpler benchmark datasets such as MNIST and CIFAR-10 for comparative evaluations with VAE-TracIn.

- **C2.** In extending **C1**, we use image attributes to supplant the concept of class for datasets lacking class labels. For datasets with attribute labels, quantified experiments are feasible, as demonstrated in Table 3 and Table 8. For datasets without attribute labels, we employ visualized experiments.

- **C3.** IF4GAN, a model-specific method for GAN, serves as the baseline for measuring data value. Adhering to the settings outlined in Terashita et al. (2021) and considering the impractical computational costs, we conduct experiments on the same dataset (MNIST) in Terashita et al. (2021) using DCGAN.

- **C4.** We present a comparison of the efficiency of GMVALUATOR against baseline methods. In accordance with the settings used for VAE-TracIN and IF4GAN in Kong & Chaudhuri (2021), we report the efficiency results for **C1** on the MNIST and CIFAR-10 datasets, and for **C3** on MNIST.

J.2 DATASETS

We conduct the generation tasks in the experiments on benchmark datasets (*i.e.,* MNIST LeCun et al. (1998) and CIFAR Krizhevsky et al. (2009)), face recognition dataset (*i.e.,* CelebA Liu et al. (2018)), high-resolution image dataset AFHQ Choi et al. (2020) and FFHQ Karras et al. (2019), large-scale image dataset ImageNet Deng et al. (2009).

MNIST. The MNIST dataset consists of a collection of grayscale images of handwritten digits (0-9) with a resolution of 28x28 pixels. The dataset contains 60,000 training images and 10,000 testing images.

CIFAR-10. CIFAR-10 dataset consists of 60,000 color images in 10 different classes, with 6,000 images per class. The classes include objects such as airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Each image in the CIFAR-10 dataset has a resolution of 32x32 pixels.

CelebA. The CelebA dataset is a widely used face recognition and attribute analysis dataset, which contains a large collection of celebrity images with various facial attributes and annotations. The dataset consists of more than 200,000 celebrity images, with each image labeled with 40 binary attribute annotations such as gender, age, facial hair, and presence of eyeglasses.

AFHQ. The AFHQ dataset is a high-resolution image dataset that focuses on animal faces (*e.g.,* dogs, cat), and it consists of high-resolution images with 512×512 pixels.

FFHQ. The FFHQ dataset is a high-resolution face dataset that contains high-quality images (1024x1024 pixels) of human faces.

ImageNet. ImageNet is a large-scale image dataset, which contains over 14 million images and is categorized into more than 20,000 classes.

Table 10: Justification of Experiment Setup. The selection of datasets and models was based on three critical factors that guided the process. Firstly, attribute labels were required to evaluate C2 effectively. Secondly, benchmark datasets were meticulously chosen to ensure a fair comparison with baselines while also taking into account computational costs (C3 and C4). Finally, the selected generative models are powerful enough to generate good-quality data for the datasets.

	Dataset	Model	Baseline	Label Requirements
C1	MNIST, CIFAR-10	VAE	VAE-TracIn Kong & Chaudhuri (2021)	Class labels
		Diffusion, GAN	-	Class labels
	ImageNet	Masked Diffusion Transformer Gao et al. (2023)	-	Class labels
C2	CelebA	Diffusion-StyleGAN	-	Attribute labels
	AFHQ, FFHQ	StyleGAN	-	-
C3	MNIST	DCGAN	IF4GAN Terashita et al. (2021)	Class labels
C4	MNIST, CIFAR-10	VAE	VAE-TracIn Kong & Chaudhuri (2021)	Class labels
	MNIST	DCGAN	IF4GAN Terashita et al. (2021)	Class labels

J.3 ARCHITECTURE OF GENERATIVE MODELS

In our experiments, we leverage different generative models in the class of GAN, VAE and diffusion models. We utilize β-VAE for both MNIST and CIFAR-10 datasets while a simple GAN is conducted on MNIST. BigGAN and β-VAE are also conducted on CIFAR-10. We list the architecture details for these generative models from Table 11 to Table 13. StyleGAN is used for high-resolution datasets AFHQ and FFHQ. CelebA uses Diffusion-StyleGAN Wang et al. (2022), for which we use the exact architecture in their open-sourced code. In addition, Masked Diffusion Transformer, as introduced by Gao et al. Gao et al. (2023), is applied to the ImageNet.

Table 11: The architecture of GAN for MNIST.

Generator
FC(100, 8192), BN(32), ReLU
Conv2D(128, 64, 4, 2, 1), BN(64), ReLU
Discriminator
Conv2D(1, 128, 4, 2, 1), BN(128), LeakyReLU
FC(8192, 1024), BN(1024), LeakyReLU

Table 12: The architecture of BigGAN.

Input	28×28×1 (MNIST) & 32×32×3 (CIFAR-10).
Encoder	Conv 32×4×4 (stride 2), 32×4×4 (stride 2), 64×4×4 (stride 2), 64×4×4 (stride 2), FC 256. ReLU activation.
Latents	32
Decoder	Deconv reverse of encoder. ReLu acitvation. Gaussian.

K DISCUSSION ON THE POSSIBLE APPLICATIONS

The application of data valuation within generative models offers a wide range of opportunities. A potential use case is to quantify privacy risks associated with generative model training using specific datasets, since the matching mechanism GMVALUATOR can help re-identify the training samples given the generated data. By doing so, organizations and individuals will be able to audit the usage of their data more effectively and make informed decisions regarding its use.

Table 13: The architecture of β-VAE.

β-**VAE**	
Generator	**Discriminator**
$z \in \mathbb{R}^{120} \sim \mathcal{N}(0, I)$ $\text{Embed}(y) \in \mathbb{R}^{32}$	RGB image $x \in \mathbb{R}^{32\times32\times3}$
Linear $(20 + 128) \to 4 \times 4 \times 16ch$	ResBlock down $ch \to 2ch$
ResBlock up $16ch \to 16ch$	Non-Local Block (64×64)
ResBlock up $16ch \to 8ch$	ResBlock down $2ch \to 4ch$
ResBlock up $8ch \to 4ch$	ResBlock down $4ch \to 8ch$
ResBlock up $4ch \to 2ch$	ResBlock down $8ch \to 16ch$
Non-Local Block (16×16)	ResBlock down $16ch \to 16ch$
ResBlock up $2ch \to ch$	ResBlock $16ch \to 16ch$
BN, ReLU, 3×3 Conv ch $\to 3$	ReLU, Global sum pooling
Tanh	Embed $(y) \cdot h +$ (linear $\to 1$)

Another promising application is material pricing and finding in content creation. For example, when training generative models for various purposes, such as content recommendation or personalized advertising, data evaluation can be used to measure the value of reference content.

In addition, GMVALUATOR can play an important role in the development of ensuring the responsibility of using synthetic data in safe-sensitive fields, such as healthcare or finance. By assessing the value of the data used in generative model training, researchers can ensure that the generated data are robust and reliable.

Last but not least, the applications of GMVALUATOR can promote the recognition of intellectual property rights. Determining the value of the intellectual property being generated by generative models is critical. By evaluating the data employed in training generative models, we can develop a more comprehensive understanding of copyright that may emerge from the generative models. In essence, such insights can help advance licensing agreements for the utilization of the generative model and its outputs.

L CURRENT LIMITATION AND FUTURE DIRECTIONS

The limitation of this work is that it only measures data value for vision-related generative models and conducts experiments exclusively within the field of computer vision. However, this does not mean that GMVALUATOR cannot be easily adapted to Natural Language Processing (NLP) fields, given its core idea of similarity matching. In the future, we should extend GMVALUATOR to NLP and assess the data value for language-related generative models, such as large language models (LLMs).

M OMITTED PROOFS

We follow Just et al. (2023) to prove the theorem. Firstly, we give several assumptions that will be used in later proof.

Assumption M.1 *Following Assumption 2.3, given a distance function $d(\cdot,\cdot)$ between , we defined the coupling between $\mathcal{X}_{(T|f)}$ and $\mathcal{X}_{(S^*|f)}$ as π^*:*

$$\pi^* := \underset{\pi\in\Pi(\mathcal{X}_{(T|f)},\mathcal{X}_{(S^*|f)})}{\arg\inf} \mathbb{E}_{(x_T,x_{S^*})\sim\pi} d(x_T, x_{S^*}) \tag{7}$$

It is easy to see that all joint distributions defined above are couplings between the corresponding distribution pairs. Then, following Just et al. (2023) we prove the main Theorem.

Theorem M.2 *(Restated of Theorem 2.4.) Let $f'_{S^*} : \mu \to \mathcal{A} = \{0,1\}^V$ be the model trained on the optimal contributor dataset S^*. Following Assumption 2.3, if the contributors are corresponding to the given generated data $\hat{X}$, we have:*

$$\begin{aligned}&\mathbb{E}_{x\sim\mu_T}\left[\mathcal{L}\left(f(x), f'_{S^*}(x)\right)\right] - \mathbb{E}_{x\sim\mu_S}\left[\mathcal{L}\left(f(x), f'_{S^*}(x)\right)\right]\\ &\quad\le k\epsilon\cdot\left[d_W(\mathcal{X}_{(T|f)}, \mathcal{X}_{(\hat{X}|f)}) + d_W(\mathcal{X}_{(S^*|f)}, \mathcal{X}_{(\hat{X}|f)})\right]\end{aligned} \tag{8}$$

Proof M.3

$$\begin{aligned}\mathbb{E}_{x\sim\mu_T}\left[\mathcal{L}\left(f(x), f'_{S^*}(x)\right)\right] &= \mathbb{E}_{x\sim\mu_T}\left[\mathcal{L}\left(f(x), f'_{S^*}(x)\right)\right]\\ &\quad - \mathbb{E}_{x\sim\mu_S}\left[\mathcal{L}\left(f(x), f'_{S^*}(x)\right)\right] + \mathbb{E}_{x\sim\mu_S}\left[\mathcal{L}\left(f(x), f'_{S^*}(x)\right)\right]\\ &\le \mathbb{E}_{x\sim\mu_S}\left[\mathcal{L}\left(f(x), f'_{S^*}(x)\right)\right]\\ &\quad + \left|\mathbb{E}_{x\sim\mu_S}\left[\mathcal{L}\left(f(x), f'_{S^*}(x)\right)\right] - \mathbb{E}_{x\sim\mu_T}\left[\mathcal{L}\left(f(x), f'_{S^*}(x)\right)\right]\right|\end{aligned} \tag{9}$$

We bound $\left|\mathbb{E}_{x\sim\mu_{S^*}}\left[\mathcal{L}\left(f(x), f'_{S^*}(x)\right)\right] - \mathbb{E}_{x\sim\mu_T}\left[\mathcal{L}\left(f(x), f'_{S^*}(x)\right)\right]\right|$ as follows:

$$
\begin{aligned}
&\left|\mathbb{E}_{x\sim\mu_{S^*}}\left[\mathcal{L}\left(f(x), f'_{S^*}(x)\right)\right] - \mathbb{E}_{x\sim\mu_T}\left[\mathcal{L}\left(f(x), f'_{S^*}(x)\right)\right]\right| \\
&= \left|\int_{\mathcal{X}^2} \mathcal{L}\left(f(x_{S^*}), f'_{S^*}(x_S)\right) - \mathcal{L}\left(f(x_T), f'_{S^*}(x_T)\right) d\pi^*(x_T, x_{S^*})\right| \\
&= |\int_{\mathcal{X}^2} \mathcal{L}\left(f(x_{S^*}), f'_{S^*}(x_{S^*})\right) \\
&- \mathcal{L}\left(f(x_{S^*}), f'_{S^*}(x_T)\right) + \mathcal{L}\left(f(x_{S^*}), f'_{S^*}(x_T)\right) - \mathcal{L}\left(f(x_T), f'_{S^*}(x_T)\right) d\pi^*(x_T, x_{S^*})| \\
&\leq \int_{\mathcal{X}^2} \left|\mathcal{L}\left(f(x_{S^*}), f'_{S^*}(x_{S^*})\right) - \int_{\mathcal{X}^2} \mathcal{L}\left(f(x_{S^*}), f'_{S^*}(x_T)\right)\right| d\pi^*(x_T, x_{S^*}) \\
&+ \int_{\mathcal{X}^2} \left|\mathcal{L}\left(f(x_{S^*}), f'_{S^*}(x_T)\right) - \int_{\mathcal{X}^2} \mathcal{L}\left(f(x_T), f'_{S^*}(x_T)\right)\right| d\pi^*(x_T, x_{S^*})
\end{aligned} \tag{10}
$$

Then due to k-Lipschitzness of $\mathcal{L}$ and ϵ-Lipschitzness of f, we can obtain:

$$
\begin{aligned}
\textit{RHS of Eq.equation } 10 &\leq k\int_{\mathcal{X}^2} ||f'_{S^*}(x_{S^*}) - f'_{S^*}(x_T)|| d\pi^*(x_T, x_{S^*}) \\
&+ k\int_{\mathcal{X}^2} ||f(x_{S^*}) - f(x_T)|| d\pi^*(x_T, x_{S^*}) \\
&\leq k\epsilon \int_{\mathcal{X}^2} 2d(x_T, x_{S^*}) d\pi^*(x_T, x_{S^*}) \\
&= k\epsilon d_W(\mathcal{X}_{(T|f)}, \mathcal{X}_{(S^*|f)}),
\end{aligned}
$$

where the last step is due to the definition of 1-Wasserstein distance. Then, according to the triangle inequality of Wasserstein distance Peyré et al. (2019), we can obtain:

$$
d_W(\mathcal{X}_{(T|f)}, \mathcal{X}_{(S^*|f)}) \leq d_W(\mathcal{X}_{(T|f)}, \mathcal{X}_{(\hat{X}|f)}) + d_W(\mathcal{X}_{(\hat{X}|f)}, \mathcal{X}_{(S^*|f)}) \tag{11}
$$

Combining Eq. equation 9 and Eq. equation 11 we finished the proof and obtained the Theorem 2.4. By reducing the distance term $d_W\left(\mathcal{X}_{(T|f)}, \mathcal{X}_{(\hat{X}|f)}\right)$, we have $\mathcal{X}_{(T|f)} \to \mathcal{X}_{(S^*|f)}$. As a result, the expected distance

$$
\mathbb{E}_{(S^*\sim\mathcal{X}_{(S^*|f)}, T\sim\mathcal{X}_{(T|f)})} \min_{\pi\in\Pi(T,S^*)} \mathbb{E}_{(x_T, x_{S^*})\sim\pi} d(x_T, x_{S^*}) \to 0,
$$

with randomly sampling S^* and T with K elements.

N REPRODUCIBILITY

To ensure reproducibility, we make our implementation available to reviewers through this anonymous link: https://anonymous.4open.science/r/GMValuator-V2-E0BE.