A. Differential Privacy

Differential Privacy (DP) [12]. A randomized mechanism $\mathcal{M}$ is (ε, δ)-DP, if

$$\Pr[\mathcal{M}(X) \in O] \leq e^{\varepsilon} \cdot \Pr\left[\mathcal{M}\left(X'\right) \in O\right] + \delta \tag{9}$$

holds for all output $O \subseteq \mathrm{Range}(\mathcal{M})$ and for any adjacent datasets X and X' that differ from each other with only one record. Here, the privacy budget, ε, is the upper bound on the privacy loss corresponding to $\mathcal{M}$, and δ is the probability of violating the DP constraint. Usually, $(\varepsilon, 0)$-DP is also written as ε-DP.

In practice, Gaussian mechanism can be used to achieve (ε, δ)-DP while Laplace mechanism and randomized response can be used to achieve ε-DP. As DPGEN relies on randomized response to reach ε-DP guarantee, we only describe the randomized response below.

Randomized Response (RR). Given a value $v \in [k]$, where $[k] \triangleq \{1, \ldots, k\}$. The randomized response is defined as:

$$\forall y \in [k], \Pr[\mathbf{RR}_{k,\varepsilon}(v) = y] = \begin{cases} p = \frac{e^{\varepsilon}}{e^{\varepsilon}+k-1}, & \text{if } y = v \\ p' = \frac{1}{e^{\varepsilon}+k-1}, & \text{if } y \neq v \end{cases}$$

$\mathbf{RR}_{k,\varepsilon}(\cdot)$ achieves ε-DP since $\frac{p}{p'} = e^{\varepsilon}$. In fact, the function $H(\cdot)$ in Eq. (8) is equivalent to $\mathbf{RR}_{k,\varepsilon}(\cdot)$. DP is featured by its sequential composition, parallel composition, and post-processing properties. We have a brief description of them below.

Sequential Composition. Given $\mathcal{M}_1(\cdot)$ that satisfies ε_1-DP and $\mathcal{M}_2(\cdot)$ that satisfies ε_2-DP. Then, $\mathcal{M}_1 \circ \mathcal{M}_2$ satisfies $(\varepsilon_1 + \varepsilon_2)$-DP, where $\circ$ denote the composition operator.

Parallel Composition. Assume that each $\mathcal{M}_i$ satisfies ε-DP. Let X_i be arbitrary disjoint subsets of the domain of X. The sequence of $\mathcal{M}_i(X \cap X_i)$ satisfies ε-DP.

Post-processing. If $\mathcal{M}$ satisfies (ε, δ)-DP, then $F \circ \mathcal{M}$ will also satisfy (ε, δ)-DP for any function F independent of the sensitive dataset, where $\circ$ denote the composition operator.

B. DPGEN satisfies ε-Differential Privacy (ε-DP)

Here, we provide a rigorous proof that DPGEN satisfies satisfies ε-DP. The roadmap goes as follows.

1. We first recall the setting of DPGEN.
2. We prove that the mechanism $\mathcal{M}$ that perturbs the recovery direction required by the energy-guided network (see the rightmost network in Figure 1 of our main text) satisfies ε-DP in Lemma 4.
3. We prove that mechanism $\mathcal{M}$, even when the coverage of the RR is limited to k candidates, satisfies ε-DP in Lemma 5.
4. After all of the above procedures, we will make a conclusion that DPGEN satisfies ε-DP in Theorem 1.

The algorithmic procedure of DPGEN is shown in Algorithm 1. We notice a difference between DPGEN described in Section 4.2 and Algorithm 1. The $k-1$ candidate images for RR in $H(\cdot)$ are inherently chosen by using k nearest neighbors (kNN) approach in the main text of Section 4.2, while the $k-1$ candidate images for RR in $H(\cdot)$ are chosen by using probabilistic k nearest neighbors (kNN) approach in Line 20 of Algorithm 1. In fact, they can be proved equivalent in the context of DP.

Setting. Let $X = \{x_i \in \mathbb{R}^{H \times W \times C} : i = 1, \ldots, m\}$ be a sensitive dataset, where H is the height, W is the width and C is the channel of an image. One of the components in DPGEN is $\mathcal{M} : X \to X$, where $\mathcal{M}(x) \triangleq (H \circ f)(x)$, and let O be the output range of $\mathcal{M}$. Here, instead of the $H(\cdot)$ defined in Eq. (8), in the proofs of Lemmas 1$\sim$4, we here temporarily consider $H(\cdot)$ below:

$$\Pr\left[H(\tilde{x}_v) = x_y^r\right] = \begin{cases} \dfrac{e^{\varepsilon}}{e^{\varepsilon}+m-1}, & \text{if } y = v \\ \dfrac{1}{e^{\varepsilon}+m-1}, & \text{if } y \neq v \end{cases}. \tag{10}$$

In this sense, $\mathcal{M}$ takes as input a training image and outputs a random training image from the training dataset (consisting of our sensitive images). More precisely, we have the probability $\Pr\left[\mathcal{M}(x_i) = x_i^r\right] = \Pr\left[H(\tilde{x}_i) = x_i^r\right] = e^{\varepsilon}/(e^{\varepsilon}+m-1)$,

Algorithm 1: DPGEN

Let $X = \{x_1, \ldots, x_m\}$, $\Sigma = \{\sigma^{(0)}, \ldots, \sigma^{(\ell)}\}$ and $W = \emptyset$, $T = \emptyset$, $V = \emptyset$
for *each image* $x_i \in X$ **do**
    sample σ_i from Σ
    $\tilde{x}_i = f(x_i) + z_i$; $z_i \sim \mathcal{N}(0, \sigma_i)$
    $\tilde{X} = \tilde{X} \cup \tilde{x}_i$
while $|W| < m$ **do**
    for *each image* $x_i \in X$ **do**
        sample $\tilde{x}$ from $\tilde{X}$ uniformly
        $\nu_i = \max |\tilde{x} - x_i|/3$
        $V = V \cup \nu_i$
    for *each image* $x_i \in X$ **do**
        σ_i is sampled from $\{\sigma' : \sigma' \geq \max V; \sigma' \in \Sigma\}$
    $j \sim \{1 \ldots m\}$
    if $j \notin W$ **then**
        $W = W \cup \{j\}$
        $\tilde{x}$ is assigned to be $\tilde{x}_j$
        $T = T \cup (\tilde{x}_j, x_j, \sigma_j)$

for *each* $(\tilde{x}_i, x_i, \sigma_i) \in T$ **do**
    Define X_k^i as the set of k random samples from X with probability $p(x_i) = \frac{\exp(-d_\infty(x_i, \tilde{x}_i, \sigma_i))}{\sum_{a=1}^{m} \exp(-d_\infty(x_a, \tilde{x}_i, \sigma_a))}$ with $d_\infty(x, \tilde{x}, \sigma) \triangleq \frac{\max |\tilde{x} - x|}{\sigma}$
    Calculate $x_i^r = H(\tilde{x}_i)$ over X_k^i
    Calculate $d_i^r = \frac{\tilde{x}_i - x_i^r}{\sigma_i^2}$

and $\Pr[\mathcal{M}(x_j) = x_i^r] = \Pr[H(\tilde{x}_j) = x_i^r] = 1/(e^\varepsilon + m - 1)$. However, since DPGEN allows the recovery direction $d^r = (\tilde{x} - x^r)/\sigma^2$ to train the neural network, it requires to calculate the probability:

$$\begin{aligned} \Pr[d^r = (\tilde{x}_{ii} - x_i^r)/\sigma_i] &= \Pr[x_i^r, \sigma_i, \tilde{x}_{ii}, \tilde{x}, x_i] \\ &= \Pr[H(\tilde{x}_{ii}) = x_i^r] \cdot \Pr[\sigma_i | \tilde{x}_{ii}, x_i] \cdot \Pr[\tilde{x}_{ii} | x_i] \cdot \Pr[x_i | \tilde{x}] \cdot \Pr[\tilde{x}], \end{aligned} \tag{11}$$

and

$$\begin{aligned} \Pr[d^r = (\tilde{x}_{ij} - x_i^r)/\sigma_i] &= \Pr[x_i^r, \sigma_i, \tilde{x}_{ij}, \tilde{x}, x_j] \\ &= \Pr[H(\tilde{x}_{ij}) = x_i^r] \cdot \Pr[\sigma_i | \tilde{x}_{ij}, x_j] \cdot \Pr[\tilde{x}_{ij} | x_j] \cdot \Pr[x_j | \tilde{x}] \cdot \Pr[\tilde{x}]. \end{aligned} \tag{12}$$

Here $\tilde{x}_{ii}$ and $\tilde{x}_{ij}$ are equivalent to $\tilde{x}_i$, the second subscript is used to distinguish the input sources. For example, if the algorithm 1 uses the true image x_i to pair the noisy image $\tilde{x}$ and compute the true standard deviation σ_i, then $\tilde{x}_{ii}$ means that $\tilde{x}_i$ is paired by the true image x_i. And $\tilde{x}_{ij}$ means that the adversary thinks that $\tilde{x}_i$ is paired by another image x_j. Note that the superscript r of x_i^r is used to distinguish this x_i is responded by the RR mechanism.

Lemma 1. *Given a noisy image $\tilde{x}$, if the algorithm 1 is used to obtain the pairs $(\tilde{x}_i, x_i, \tilde{x})$, $(\tilde{x}_j, x_j, \tilde{x})$ of any two images x_i, x_j, then the joint probability* $\Pr[\tilde{x}_i, x_i, \tilde{x}] = \Pr[\tilde{x}_j, x_j, \tilde{x}] = q/m$.

Proof. According to Bayes' theorem, the joint probability $\Pr[\tilde{x}_i, x_i, \tilde{x}] = \Pr[\tilde{x}_i | x_i, \tilde{x}] \cdot \Pr[x_i | \tilde{x}] \cdot \Pr[\tilde{x}]$. In the algorithm 1 , we sample $\tilde{x}$ uniformly, and assume that the probability is q. Then given $\tilde{x}$, the x_i is also sampled uniformly, so $\Pr[x_i | \tilde{x}] = \Pr[x_i] = 1/m$. Since $\tilde{x}$ is assigned directly to x_i after x_i is sampled, thus $\Pr[\tilde{x}_i | x_i, \tilde{x}] = 1$. Likewise, $\Pr[\tilde{x}_j | x_j, \tilde{x}] = 1$, and $\Pr[x_j | \tilde{x}] = \Pr[x_j] = 1/m$. Therefore, $\Pr[\tilde{x}_i, x_i, \tilde{x}] = \Pr[\tilde{x}_j, x_j, \tilde{x}] = q/m$ according to the joint probability, thus finishing the proof. □

Lemma 2. *Given a noisy image $\tilde{x}$, if the standard deviation σ_i, σ_j of any two images x_i, x_j are sampled by the algorithm 1 , then* $\Pr[\sigma_i | \tilde{x}, x_i] = \Pr[\sigma_j | \tilde{x}, x_j] = 1/|S|$.

Proof. Since σ_i, σ_j are sampled from $S = \{\sigma' : \sigma' \geq \max V; \sigma' \in \Sigma\}$, $\Pr[\sigma_i|\tilde{x}, x_i] = \Pr[\sigma_j|\tilde{x}, x_j] = 1/|S|$. □

Lemma 3. *For any two images x_i and x_j, $x_i \neq x_j$, the mechanism $\mathcal{M}$ satisfies*

$$\Pr[\mathcal{M}(x_i) = x_i^r] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(x_j) = x_i^r]. \tag{13}$$

Proof. We know that the probability that $\mathcal{M}$ takes as input x_i and returns $x_i^r = \mathcal{M}(x_i)$ is $e^{\varepsilon}/(e^{\varepsilon} + m - 1)$. Note that the superscript r of x_i^r is used to distinguish this x_i is responded by the RR mechanism. We also know that the probability that $\mathcal{M}$ takes as input x_j but returns x_i^r is $1/(e^{\varepsilon} + m - 1)$. Hence, $\Pr[\mathcal{M}(x_i) = x_i^r] / \Pr[\mathcal{M}(x_j) = x_i^r] \leq e^{\varepsilon}$. □

Lemma 4. *The sequence of $\mathcal{M}(x_i)$, denoted as $\mathcal{M}(X)$, satisfies ε-DP.*

Proof. Our proof derives from the parallel composition of DP. Consider two adjacent datasets X, X' that differ only by one image data. Let $\mathcal{M}$ be a mechanism which accesses a disjoint subset of our training dataset. In our case, each training image constitutes a disjoint subset of the training dataset. Therefore, the probability of returning $\mathcal{M}(X)$ from sequence of $\mathcal{M}(x_i)$ is

$$\Pr[\mathcal{M}(X) \in O] = \prod_i \Pr[\mathcal{M}(x_i) = x_i^r]. \tag{14}$$

After applying Lemma 3 to Eq. (14), we can derive

$$\begin{aligned} &\prod_i \Pr[\mathcal{M}(x_i) = x_i^r] \\ \leq &\prod_i \Pr[\mathcal{M}(x_i') = x_i^r] \times \prod_i \exp(\varepsilon \cdot |x_i \oplus x_i'|) \\ = &\Pr[\mathcal{M}(X') \in O] \times \exp(\varepsilon \cdot |X \oplus X'|), \end{aligned} \tag{15}$$

where $x_i \in X, x_i' \in X'$, and the term $\oplus$ represents XOR. As X, X' only differs in one image, we have $|X \oplus X'| = 1$. After the re-arrangement, Eq. (15) can be proved to be ε-DP,

$$\Pr[\mathcal{M}(X) \in O] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(X') \in O].$$

□

An inherent assumption behind Lemma 4 is that the randomized response has to assign either probability $e^{\varepsilon}/(e^{\varepsilon} + m - 1)$ or probability $1/(e^{\varepsilon} + m - 1)$ to each training image. As the number of training images or the batch size is too large, it follows that the number of candidates in the randomized response will also be large. Thus, the probability of each candidate to be selected is nearly uniformly distributed, which leads to a significant loss of information of the training images. In the following, we alleviate this problem by selecting and considering only k candidates in the randomized response by using prior probability. Note that in Section 4.1 of the main text, we mention that $\{x_j : \max|\tilde{x}_i - x_j|/\sigma_j \leq \beta, j \in [m]\}$ with the smallest k values are selected as the k candidates for the randomized response.

Lemma 5. *Given that each data is processed by $\mathcal{M} = H \circ f$, the mechanism $\mathcal{M}$ with $H(\cdot)$ on k candidate training images satisfies ε-DP.*

Proof. Given a noisy image $\tilde{x}_i$ and a modified infinity norm $d_{\infty}(x, \tilde{x}, \sigma) = \max|\tilde{x} - x|/\sigma$ to calculate the k candidates from m training images, where the sampling probability is defined as $p(x_i) = \exp(-d_{\infty}(x_i, \tilde{x}_i, \sigma_i))/\sum_{a=1}^{m} \exp(-d_{\infty}(x_a, \tilde{x}_i, \sigma_a))$. Let X_k^i be the sampled k images from $\tilde{x}_i$, so that given the true image x_i and its paired noisy image $\tilde{x}_i$, the probability of sampled k images is $p(X_k^i|x_i) = \sum_{a=1}^{k} p(x_a)$, and $x_a \in X_k^i$. While the probability that the adversary thinks the noisy image $\tilde{x}_i$ is paired with x_j and produce X_k^i is $p(X_k^i|x_j)$, $x_i \neq x_j$. Thus, the probability that $\mathcal{M}$ takes as input x_i and returns x_i^r from the k candidates is:

$$\Pr[\mathcal{M}(x_i) = x_i^r] = \frac{e^{\varepsilon}}{e^{\varepsilon} + k - 1} \cdot p(X_k^i|x_i). \tag{16}$$

Similarly, the probability that $\mathcal{M}$ takes as input x_j but returns x_i from the k candidates is:

$$\Pr[\mathcal{M}(x_j) = x_i^r] = \frac{1}{e^{\varepsilon} + k - 1} \cdot p(X_k^i|x_j). \tag{17}$$

No matter the noisy image $\tilde{x}_i$ is paired by x_i or x_j, the distance between $\tilde{x}_i$ and any element from X_k^i is fixed, so $p(X_k^i|x_i) = p(X_k^i|x_j)$. Therefore, derived from Lemma 3, the ratio of the probability of returning x_i^r from the k candidates calculated from $\mathcal{M}$ is:

$$\Pr\left[\mathcal{M}(x_i) = x_i^r\right] \leq e^{\varepsilon} \cdot \Pr\left[\mathcal{M}(x_j) = x_i^r\right]. \tag{18}$$

Finally, we follow lemma 4 to prove the claim:

$$\begin{aligned}
\Pr[\mathcal{M}(X) \in O] &= \prod_i \Pr\left[\mathcal{M}\left(x_i\right) \in O\right] \\
&\leq \prod_i \Pr\left[\mathcal{M}\left(x_i'\right) \in O\right] \cdot \exp(\varepsilon \cdot |X \oplus X'|) \\
&= \exp(\varepsilon) \cdot \Pr[\mathcal{M}(X') \in O].
\end{aligned} \tag{19}$$

□

Theorem 1. DPGEN satisfies ε-DP.

Proof. Since DPGEN allows to train the neural network with noisy image $\tilde{x}$ and recovery direction $d^r = (\tilde{x} - x^r)/\sigma^2$, it is necessary to calculate the joint probability in order to clarify the association between all involved variables and the training images. For brevity of the proof, we let $\mathcal{K}(\tilde{x}, x) = (\tilde{x} - \mathcal{M}(x))/\sigma(\tilde{x}, x) = (\tilde{x} - H(\tilde{x}))/\sigma(\tilde{x}, x) = (\tilde{x} - x^r)/\sigma(\tilde{x}, x)$, where $\sigma(\tilde{x}, x)$ means its dependence on $\tilde{x}$ and x. In addition, let $\mathcal{R}$ be the range of the output of the $\mathcal{K}$, we have

$$\begin{aligned}
\Pr\left[\mathcal{K}(\tilde{X}, X) \in \mathcal{R}\right] &= \prod \Pr\left[\tilde{x} = \tilde{x}_{ii}, d^r = (\tilde{x}_{ii} - x_i^r)/\sigma_i\right] \\
&= \prod \Pr\left[H(\tilde{x}_{ii}) = x_i^r\right] \cdot \Pr\left[\sigma_i|\tilde{x}_{ii}, x_i\right] \cdot \Pr\left[\tilde{x}_{ii}|x_i, \tilde{x}\right] \cdot \Pr\left[x_i|\tilde{x}\right] \cdot \Pr\left[\tilde{x}\right] \\
&= \prod \Pr\left[H(\tilde{x}_{ii}) = x_i^r\right] \cdot 1/|S| \cdot q/m \\
&\leq \prod \Pr\left[H(\tilde{x}_{ij}) = x_i^r\right] \cdot \Pr\left[\sigma_i|\tilde{x}_{ij}, x_j\right] \cdot \Pr\left[\tilde{x}_{ij}|x_j, \tilde{x}\right] \cdot \Pr\left[x_j|\tilde{x}\right] \cdot \Pr\left[\tilde{x}\right] \cdot \exp\left(\varepsilon \cdot |X \oplus X'|\right) \\
&= \prod \Pr\left[H(\tilde{x}_{ij}) = x_i^r\right] \cdot 1/|S| \cdot q/m \cdot \exp\left(\varepsilon \cdot |X \oplus X'|\right) \\
&= \exp(\varepsilon) \cdot \Pr\left[\mathcal{K}(\tilde{X}, X') \in \mathcal{R}\right].
\end{aligned}$$

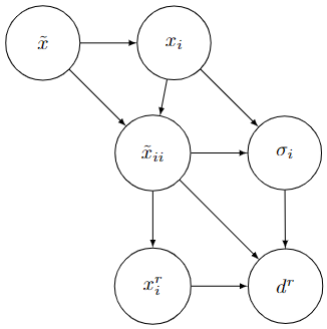

Figure 9. Bayesian Network representation of our method.

In the above proof, the second line can reference to Figure 9 for better understanding. The probability $\Pr\left[\tilde{x}_{ii}|x_i, \tilde{x}\right] \cdot \Pr\left[x_i|\tilde{x}\right] \cdot \Pr\left[\tilde{x}\right]$ and $\Pr\left[\tilde{x}_{ij}|x_j, \tilde{x}\right] \cdot \Pr\left[x_j|\tilde{x}\right] \cdot \Pr\left[\tilde{x}\right]$ are both equal to q/m is from Lemma 1. Furthermore, the probability $\Pr\left[\sigma_i|\tilde{x}_{ii}, x_i\right]$ and $\Pr\left[\sigma_i|\tilde{x}_{ij}, x_j\right]$ are both equal to $1/|S|$ is from Lemma 2. Finally, from Lemma 4, we can arrive at a conclusion that DPGEN satisfies ε-DP, which is the statement of Theorem 1.

□

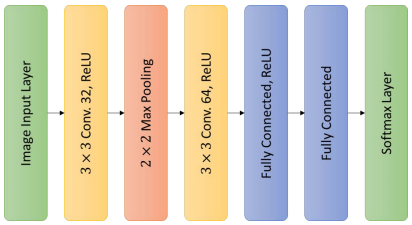

Figure 10. The architecture of classifier used in Section 5.

C. The Architecture of the Classifier

Figure 10 shows the architecture of the classifier used in our experiments for the downstream classification task.

D. Backbone NN

The NN in **(S3)** of Figure 1 is RefineNet [31]. The architecture of RefineNet used in our setting is shown in Figure 6.

3×3 Conv2D, 128
ResBlock, 128
ResBlock, 128
ResBlock down, 256
ResBlock, 256
ResBlock down, 256 dilation 2
ResBlock, 256 dilation 2
ResBlock down, 256 dilation 4
ResBlock, 256 dilation 4
RefineBlock, 256
RefineBlock, 256
RefineBlock, 128
RefineBlock, 128
3×3 Conv2D, 3

Table 6. The architecture of RefineNet [31] used in our experiments.

E. Technical Explanation

The connection among RR, MCMC, and NN. The NN provides the direction of movement required by MCMC (also shown in the caption of Figure 2b). In particular, the connection between NN and MCMC is shown in Figure 2, where the gradient of $x^{(t)}$ over log-likelihood, $\nabla_x \log q_\theta \left(x^{(t)}\right)$, is parameterized by an NN with the parameter θ. Here, the input of NN is $x^{(t)}$ and the NN is trained to generate the gradient that points to the position with the largest log-likelihood (this position is likely to be the true image), with q_θ being an approximation of true data distribution. Notably, the whole process in Figure 2b is MCMC.

On the other hand, one needs the labeled dataset (gradients as the label) to ensure that the NN outputs the direction of movement required by MCMC. The RR used in $\mathcal{M}$ is aimed to ensure that the dataset satisfies DP (stated in Theorem 1, and the proof is provided in Supplementary Material). Once the dataset satisfies DP, the NN training can be seen as post-processing of DP, and thus the NN output also meets DP. In conclusion, the NN output satisfying DP (achieved by RR) makes MCMC differentially private.

How MCMC is approximated by NN. MCMC can be regarded as a series of movements specified by gradients. The loss function in Eq. (7) used in the NN training describes how NN (especially, the gradient parameterized by the NN with

parameter θ) approximates the actual gradient $(\tilde{x} - x)/\sigma^2$ that points to the true input image x from noisy image $\tilde{x}$. In other words, the loss function in Eq. (7) is designed such that the output of NN, $\nabla_x \log q_\theta \left(x^{(t)}\right)$, approximates the true gradient, $(\tilde{x} - x)/\sigma^2$. Therefore, the gradient outputted from the NN can be used by MCMC to point to the true input image x.