Reinforcement-Guided Synthetic Data Generation for Privacy-Sensitive Identity Recognition

Supplementary Material

A. Dynamic Sample Selection Strategy

After obtaining samples from the adapted diffusion model, distributional discrepancies between synthesized and real data remain inevitable, leading to varying levels of training utility among synthetic samples. To mitigate this issue, we introduce a lookahead-guided sample selection strategy that dynamically prioritizes high-utility samples, thereby enhancing the effectiveness of downstream model training. Let $\mathcal{X} = \bigcup_{i=1}^{m}\{(\boldsymbol{x}_i, y_i)\}$ denote the real training set, and $\mathcal{S} = \bigcup_{i=1}^{n}\{(\hat{\boldsymbol{x}}_i, y_i)\}$ denote the synthesized set, both sampled from an underlying label space $\mathcal{Y}$. At each iteration t, we construct a mixed batch $\mathcal{B} = \bigcup_{p=1}^{P}\left(\left\{(\boldsymbol{x}_i^{(p)}, y^{(p)})\right\}_{i=1}^{N-1} \cup \left\{(\hat{\boldsymbol{x}}^{(p)}, y^{(p)})\right\}\right)$, composed of P identity classes, each with $N-1$ real and 1 synthetic real and synthetic samples.

In parallel, we define a candidate batch $\mathcal{B}_c = \bigcup_{p=1}^{P}\left\{(\hat{\boldsymbol{x}}_i^{(p)}, y^{(p)})\right\}_{i=1}^{M}$, containing M candidate synthetic samples for each identity $y^{(p)} \in \mathcal{Y}$. Let the current model parameters be $\boldsymbol{w}_t$. We simulate a one-step virtual gradient update based on the original batch $\mathcal{B}$:

$$\boldsymbol{w}_t' = \boldsymbol{w}_t + \eta \cdot \nabla_{\boldsymbol{w}} L(\boldsymbol{w}_t; \mathcal{B}), \tag{13}$$

where $L(\boldsymbol{w}; \mathcal{B}) = \frac{1}{|\mathcal{B}|}\sum_{(\boldsymbol{x},y)\in\mathcal{B}} l(\boldsymbol{w}, \boldsymbol{x}, y)$ is the empirical loss over the batch, and η is the learning rate. For each candidate sample $(\hat{\boldsymbol{x}}_j^{(p)}, y^{(p)}) \in \mathcal{B}_c$, we define its utility score based on the change in identity-specific loss before and after the virtual update:

$$\Delta l_j^{(p)} = l_{\text{id}}(\boldsymbol{w}_t', \hat{\boldsymbol{x}}_j^{(p)}) - l_{\text{id}}(\boldsymbol{w}_t, \hat{\boldsymbol{x}}_j^{(p)}), \tag{14}$$

where $l_{\text{id}}(\cdot)$ is an identity-consistency loss (*e.g.*, cross-entropy loss) computed using a fixed classifier or identity head. For each identity $y^{(p)}$, we select the most compatible sample by solving:

$$\hat{\boldsymbol{x}}^{*(p)} = \arg\min_{\hat{\boldsymbol{x}}_j^{(p)} \in \mathcal{B}_c^{(p)}} \Delta l_j^{(p)}. \tag{15}$$

These selected samples $\{\hat{\boldsymbol{x}}^{*(p)}\}_{p=1}^{P}$ are then used to replace the original synthesized instances in $\mathcal{B}$, forming a refined batch $\mathcal{B}_r$. Finally, we compute the true gradient update on $\mathcal{B}_r$ and update model parameters accordingly:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \eta \cdot \nabla_{\boldsymbol{w}} L(\boldsymbol{w}_t; \mathcal{B}_r). \tag{16}$$

This lookahead-guided mechanism ensures that training steps are influenced by synthetic samples aligned with the current optimization direction, thereby improving gradient stability and promoting better generalization.

Algorithm 1 Adaptive Selection Strategy

Input: Real training set $\mathcal{X} \triangleq \bigcup_{i=1}^{m}\{(\boldsymbol{x}_i, y_i)\}$; Synthesized training set $\mathcal{S} \triangleq \bigcup_{i=1}^{n}\{(\hat{\boldsymbol{x}}_i, y_i)\}$; Overall loss function l; Identity loss function l_{id}; Number of identities P; Real/Synthesized samples per identity N; Candidate samples per identity M; Step size η
Output: Updated model parameters $\boldsymbol{w}_T$
Initialize: $\boldsymbol{w}_0$, $t = 0$

while training **do**
  Sample batch $\mathcal{B} = \bigcup_{p=1}^{P}\left(\left\{(\boldsymbol{x}_i^{(p)}, y^{(p)})\right\}_{i=1}^{N-1} \cup \left\{(\hat{\boldsymbol{x}}^{(p)}, y^{(p)})\right\}\right)$
  Sample candidate set $\mathcal{B}_c = \bigcup_{p=1}^{P}\left\{(\hat{\boldsymbol{x}}_i^{(p)}, y^{(p)})\right\}_{i=1}^{N}$
  Store current weights: $\boldsymbol{w}_t$
  Compute gradient on batch $\mathcal{B}$: $\boldsymbol{g}' = \nabla_{\boldsymbol{w}} L(\boldsymbol{w}_t; \mathcal{B})$
  Virtual update: $\boldsymbol{w}' = \boldsymbol{w}_t + \eta \cdot \boldsymbol{g}'$
  for each sample $(\hat{\boldsymbol{x}}_j^{(p)}, y^{(p)}) \in \mathcal{B}_c$ **do**
    Compute loss difference: $\Delta l_j^{(p)} = l_{\text{id}}(\boldsymbol{w}'; \hat{\boldsymbol{x}}_j^{(p)}) - l_{\text{id}}(\boldsymbol{w}_t; \hat{\boldsymbol{x}}_j^{(p)})$
  end for
  for each identity $p = 1, \ldots, P$ **do**
    Select best candidate: $\hat{\boldsymbol{x}}^{*(p)} = \arg\min_j \Delta l_j^{(p)}$
  end for
  Replace synthetic samples in $\mathcal{B}$ with $\{\hat{\boldsymbol{x}}^{*(p)}\}_{p=1}^{P}$ to form updated batch $\mathcal{B}_r$
  Compute updated gradient: $\boldsymbol{g} = \nabla_{\boldsymbol{w}} L(\boldsymbol{w}_t; \mathcal{B}_r)$
  Update model: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \eta \cdot \boldsymbol{g}$
  $t \leftarrow t + 1$
end while
return $\boldsymbol{w}_T$

B. Implementing Details

B.1. Datasets

Market1501 [58] is a large-scale re-identification benchmark comprising 32,668 images of 1,501 pedestrians, 751 for training and 750 for testing, captured by six cameras. CUHK03-NP [23] is a widely used re-identification dataset containing 14,097 images of 1,467 identities captured by six cameras. Following the new protocol [64], 767 identities are used for training, and the remaining 700 are divided

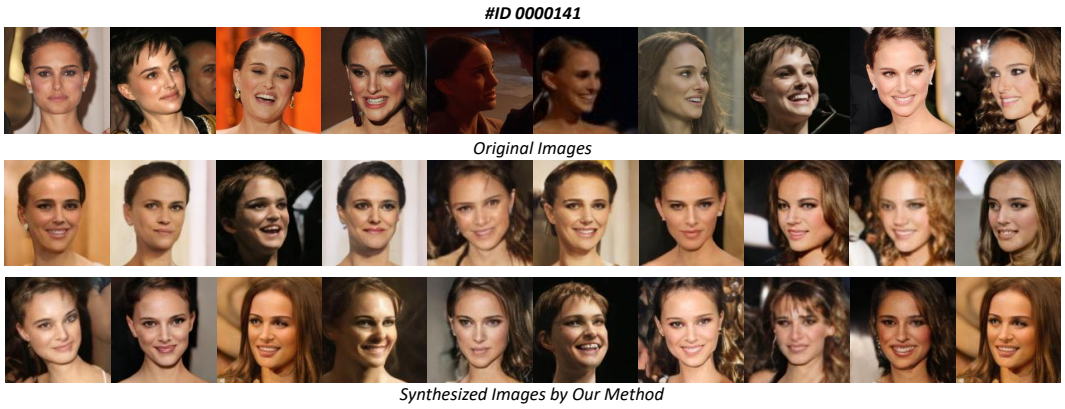

Figure 7. Samples of *ID#0000141* generated by our method (CFG scale = 2.0).

Figure 8. Samples of *ID#0000455* generated by our method (CFG scale = 2.0).

into query and gallery sets. A subset of CASIA [51] containing 16,511 face images of 1,000 identities are randomly selected in our experiments.

B.2. Hyperparameters Setup

In Distributional Coverage, the bandwidth σ of RBF kernel is at 0.25 and 0.5, and the coefficient α is set as 0.7.

In Expressive Diversity, small expansion ratio ε is set as 0.1. τ, controlling the tolerance of variance deviation, is at 0.2. A small constant $\epsilon = 1e-8$ is added for numerical stability during normalization.

Then the parameters λ_{sem}, λ_{cov}, and λ_{exp} are at 1.0, 0.75, and 0.25.

C. More Visualizations

As shown in Figure 7 and 8, both cases show a strong balance between identity fidelity and appearance diversity. For ID#0000141, core cues such as facial geometry and hair silhouette remain stable, while pose shifts from profile to near-frontal, expressions vary, illumination ranges from backlit to soft indoor, and backgrounds change widely; no mode collapse is observed, only slight highlight smoothing. For ID#0000455, eyeglasses shape, hairline, and jaw contour are consistently preserved, alongside broad variation in pose, indoor and outdoor lighting, color and grayscale renders, and mild blur or noise. Overall, the synthesized sets keep identity intact while expanding coverage, making them well suited for downstream training without overfitting to the originals.

D. Additional Evaluations

D.1. Cross-domain Generalization

Figure 9 shows consistent cross-domain improvements with our method in both directions (Market→CUHK and CUHK→Market) and for both mAP and Rank-1. Gains are evident on both backbones, with larger relative boosts on the CNN backbone and in the harder Market→CUHK transfer, while the ViT baseline is already strong and sees steady but smaller increments. Overall, the results indicate better generalization and backbone-agnostic robustness under domain shift, aligning with our method's goal of enhancing identity cues without overfitting to the source domain.

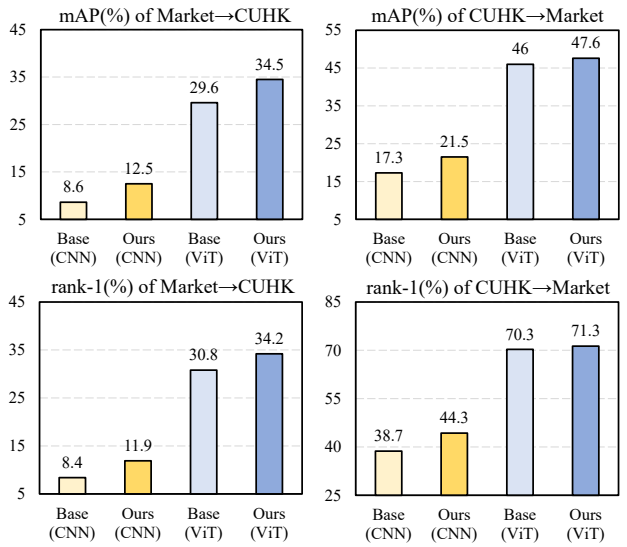

Figure 9. Cross-domain evaluation results. Source-target pair (*e.g.*, M→C) denotes training on one dataset and testing on another. Results are reported in mAP and Rank-1 accuracy.

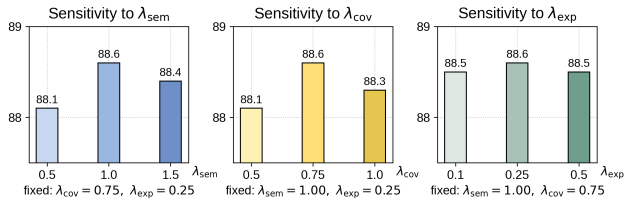

Figure 10. Sensitivity of reward function parameters. Each weight is varied while the others are fixed.

D.2. Hyperparameter Sensitivity

We further analyze the sensitivity of the reward weights λ_{sem}, λ_{cov}, and λ_{exp}. Each parameter is varied within the ranges $[0.5, 1.5]$, $[0.5, 1.0]$, and $[0.1, 0.5]$, respectively, while keeping the others fixed. As shown in Fig. 10, the resulting mAP on Market-1501 varies by less than 0.5% across all settings, indicating that the framework is largely insensitive to moderate changes in reward weighting.

This stability mainly arises from the reward normalization and $\tanh$ bounding strategy described in Sec. 4.2, which dynamically calibrate heterogeneous reward scales. As a result, the optimization process is guided by the relative quality of reward signals rather than their raw magnitudes, improving training stability and reducing the need for delicate hyperparameter tuning.

D.3. Time Consumption of DSS

We analyze the training-time overhead introduced by DSS in Table 4. Enabling DSS increases the training time per epoch from approximately 16s to 76s for CNN backbones and from 20s to 124s for ViT backbones on an H200 GPU. This overhead mainly stems from the virtual lookahead updates required by the DSS mechanism.

Table 4. Computational overhead analysis of DDS.

Backbone	CNN			ViT		
Method	Time per Epoch	mAP	R-1	Time per Epoch	mAP	R-1
w/o DSS	$\approx 16s$	85.4	94.3	$\approx 20s$	74.1	76.5
w DSS	$\approx 76s$	88.6	94.9	$\approx 124s$	76.5	79.5

Despite the increased training cost, DSS provides consistent performance improvements. In small-data regimes, the absolute training time remains modest (minutes), while DSS yields notable gains in retrieval accuracy (e.g., +3.2% mAP). Importantly, DSS introduces no additional overhead during inference.