Appendix

In this supplementary material, Section A gives more implementation details about architecture and optimization. Section B presents more results on different types of low-level perceptual variations. Section C studies the contribution of each component in our approach. Section D analyzes the influence of hyper-parameters. Section E provides the unsupervised version of L_{reg} and the mathematical derivation of L_w.

A. More Implementation Details

The **network architectures** of content encoder E_c and degradation encoder E_d are respectively given in Table 1 and Table 2, while the structure of degradation attacker A_d is illustrated in Figure 1. The network implementation of the generator G and the discriminator D follows [46]. Note that the architecture of content encoder E_c is similar to the structure encoder in [46] without using 1-channel inputs.

The **optimization process** of our proposed approach is described in Algorithm 1.

Algorithm 1 Attack-Guided Perceptual Data Generation.

```
Train the disentangled generative model which consist
  of $E_c$, $E_d$ and $G$.
Estimate the real-world degradation distribution $\mathcal{D}_d$.
Train the degradation attacker $A_d$ when the generative
  model is fixed.
For each sample $I$ from the training set do:
    a. Produce $N_s$ augmented samples $I'$ based on $\mathcal{D}_d$;
    a. Produce an adversarial sample $I''$ based on $A_d$;
    c. Extract identity features of all samples;
    d. Compute losses $L_{cls}$, $L_{sc}$ and $L_w$;
    e. Update identity encoder $E_{id}$.
```

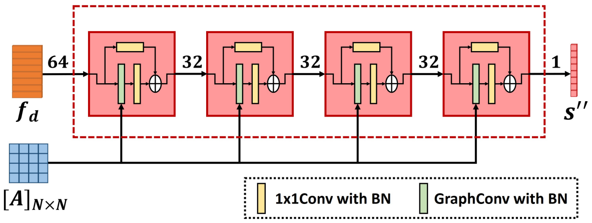

Figure 1. The network structure of the degradation attacker A_d. All activation functions (**Tanh** for the last layer and **ReLU** for the rest) are not shown for simplification.

Layer	Parameters	Output Size
Input	-	$3 \times 256 \times 128$
Conv1	$[3 \times 3, 16]$	$16 \times 128 \times 64$
Conv2	$[3 \times 3, 32]$	$32 \times 128 \times 64$
Conv3	$[3 \times 3, 32]$	$32 \times 128 \times 64$
Conv4	$[3 \times 3, 64]$	$64 \times 64 \times 32$
ResBlocks	$\begin{bmatrix}3 \times 3, 64 \\ 3 \times 3, 64\end{bmatrix} \times 4$	$64 \times 64 \times 32$
ASPP	$[1 \times 1, 32]$ $\begin{bmatrix}1 \times 1, 32 \\ 3 \times 3, 32\end{bmatrix} \times 3$	$128 \times 64 \times 32$
Conv5	$[1 \times 1, 128]$	$128 \times 64 \times 32$

Table 1. Architecture of the content encoder E_c.

Layer	Parameters	Output Size
Input	-	$3 \times 256 \times 128$
Conv1	$[3 \times 3, 16]$	$16 \times 128 \times 64$
Conv2	$[3 \times 3, 16]$	$16 \times 128 \times 64$
Conv3	$[3 \times 3, 32]$	$32 \times 64 \times 32$
Conv4	$[3 \times 3, 64]$	$64 \times 32 \times 16$
Conv5	$[3 \times 3, 64]$	$64 \times 16 \times 8$
Conv6	$[3 \times 3, 64]$	$64 \times 8 \times 4$
AvgPool	-	$64 \times 1 \times 1$

Table 2. Architecture of the degradation encoder E_d.

B. More Types of Perceptual Variations

As shown in Table 3, we provide more results with new types of low-level variations based on synthetic low-quality Market-1501 datasets. Our proposed method is able to consistently improve the baseline performance against different types of low-level perceptual variations. In fact, if we assume that each variation is independent of each other, our method can further handle the entangled case, *e.g.*, *Res.* + *Illu.*, by defining a hybrid degraded function which is used to synthesize normal-degraded image pairs.

C. More Ablation Studies

To study the contribution of each component in our approach, we further conduct comprehensive ablation analysis on the MLR-CUHK03 and MLR-VIPeR datasets, as reported in Table 4. The 'w/o $hard$' configuration denotes that the max operation (*i.e.*, hard sample mining) in the self-center loss is replaced by an $average$ operation.

It can be observed that all the components consistently result in improvements, where the self-center loss achieves the most significant performance gains, *e.g.*, 7.2% and 14.0% at Rank-1 on the MLR-CUHK03 and MLR-VIPeR dataset, respectively. This is because the self-center loss directly leverages augmented samples to regularize the feature manifold, resulting in robust identity representation learning.

D. Hyper-parameter Analysis

Here we show how the hyper-parameter N_s affects the Re-ID performance, as illustrated in Figure 2. It can be observed that the Rank-1 scores on the three datasets can be consistently improved with the increase of N_s at first, and then reaches a steady state with slight fluctuations. We also find that even with a small N_s, satisfactory results can be achieved. This improvement benefits from the hard sample mining for self-center loss as well as perceptual resampling based on the estimated real-world degradation distribution. The other hyper-parameters, such as balancing weights λ_{sc} and λ_w, are determined by grid search on the val set of the MLR-CUHK03 dataset.

Method	Low-level Variation Types			
	Noise	*Motion*	*Illu.*	*Res.+Illu.*
Baseline	74.2	68.0	73.0	63.6
Ours	79.7	76.8	80.3	70.9

Table 3. Rank-1 score (%) on low-quality Market-1501 datasets, where *Res.* denotes resolution and *Illu.* denotes illumination.

Method	MLR-CUHK03		MLR-VIPeR	
	Rank-1	Rank-5	Rank-1	Rank-5
w/o L_{adv}^{tri}	83.3	95.1	50.3	76.6
w/o L_{adv}^{att}	80.8	95.4	48.4	76.9
w/o $hard$	85.5	96.9	51.3	77.8
w/o L_{sc}	80.4	93.5	38.2	67.8
w/o L_w	86.1	97.1	48.5	78.1
Ours	**87.6**	**97.5**	**52.2**	**79.7**

Table 4. Ablation analysis on the MLR-CUHK03 and MLR-VIPeR datasets.

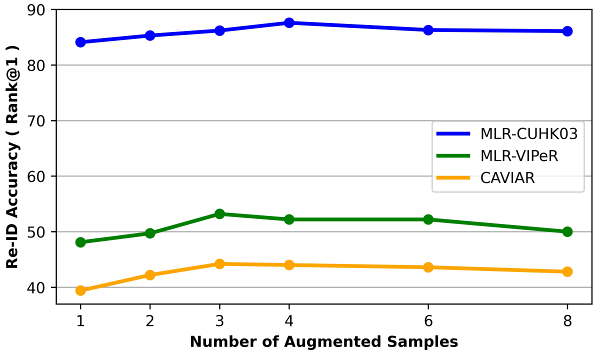

Figure 2. Analysis of the hyper-parameter N_s which controls the number of augmented samples for each input sample.

E. Details of Loss Functions

The supervised **score regression loss $\boldsymbol{L_{reg}}$** requires the domain division (*e.g.*, LR/HR) of training data to provide labels. For MSMT17 dataset, however, no such a domain division can be used, hence we further introduce an unsupervised score regression loss based on degradation ranking and pseudo-labels.

Considering an input image pair (I, I^{de}), where I^{de} is the degraded version of I produced by a non-differentiable degraded function, their corresponding perceptual quality scores (s, s^{de}) should satisfy $s > s^{de}$, which leads to a score ranking loss:

$$L_{reg}^{rank} = max(-1 \times (s - s^{de}) + \Delta_s, 0), \tag{1}$$

where Δ_s is set to 1.0 empirically.

In order to make full use of the non-synthetic images I, pseudo-labels are assigned so that the supervised MSE regression loss is available:

$$L_{reg}^{mse} = \|s - s^{pse}\|_2, \tag{2}$$

where the estimated pseudo-labels:

$$s^{pse} = \begin{cases} 1, & s \geq 0 \\ -1, & s < 0 \end{cases}. \tag{3}$$

As a result, the unsupervised score regression loss can be defined as:

$$L_{reg} = \lambda_{reg}^{rank} L_{reg}^{rank} + \lambda_{reg}^{mse} L_{reg}^{mse}, \tag{4}$$

where λ_{reg}^{rank} is set to 1.0, while λ_{reg}^{mse} is initialized to 0 then linearly increases to 1.0 for training stability.

The **Wasserstein loss $\boldsymbol{L_w}$**. Given identity embeddings $\tilde{e} \sim \mathcal{N}(\mu, \Sigma)$ and $\tilde{e}^* \sim \mathcal{N}(\mu^*, \Sigma^*)$, we employ the standard 2-Wasserstein distance to measure the similarity of these two Gaussian distributions:

$$\begin{aligned} W_2(\tilde{e}, \tilde{e}^*)^2 = ||\mu - \mu^*||_2^2 + \\ trace(\Sigma + \Sigma^* - 2(\Sigma^{\frac{1}{2}} \Sigma^* \Sigma^{\frac{1}{2}})^{\frac{1}{2}}). \end{aligned} \tag{5}$$

Assuming that $\Sigma\Sigma^* = \Sigma^*\Sigma$, it can be further simplified and derive the Wasserstein loss we used:

$$L_w \triangleq W_2(\tilde{e}, \tilde{e}^*)^2 = ||\mu - \mu^*||_2^2 + ||\Sigma^{\frac{1}{2}} - \Sigma^{*\frac{1}{2}}||_F^2. \tag{6}$$