DLCR: A Generative Data Expansion Framework via Diffusion for Clothes-Changing Person Re-ID - Supplementary Material

6. Overview

We organize the supplementary material into the following sections:

- Licensing information regarding datasets used in this paper, as well as any dataset/privacy/ethics information regarding our data, is detailed in Section 7.
- Section 8 contains ablations for each component of DLCR.
- Section 9 contains additional qualitative examples of our publicly released clothes-changed generated data.
- Sections 10 and 11 provide more quantitative and qualitative CC-ReID results to exhibit the superiority of DLCR over previous works.
- Section 12 details an optional discriminator that can be added to the DLCR data generation stage to generate synthetic data closer to the original training data distribution, marginally improving results.
- Section 13 motivates our specific use of LLaVA and LLaMA.
- Section 14 provides more insights and motivation behind our progressive learning method.
- Section 15 provides an ablation on the number of generated images per training image in DLCR.
- Section 16 details the throughput and space complexity of each DLCR component.
- Lastly, we briefly detail some possible limitations and future avenues of work in Section 17.

7. Licensing/Dataset Information

In this section, we provide information regarding licensing, since we are modifying and publicly releasing data that originates from previous CC-ReID datasets.

Dataset Licenses: Most importantly to note, the **LTCC** dataset license explicitly states that no modification or redistribution of the dataset is allowed unless by Fudan University. Thus, while we show that DLCR does work on LTCC and improves results, we will not publicly release our generated data for LTCC. Researchers are free to use DLCR to re-generate data on their own for the LTCC dataset if they have requested and been granted acccess to the dataset. The **PRCC** dataset does not follow a standardized license, but simply states that the dataset may only be used for academic purposes, which our work falls under. The **CCVID** dataset explicitly states their dataset falls under a CC BYNC-SA 4.0 license, which allows for sharing, modifying and/or adapting the data in anyway as long as credit is given, the data is not used for commercial purposes. Our generated data falls under the same license, and we do not impose any additional limitations. The **VC-Clothes** dataset follows the Apache License 2.0, which also allows for redistribution and modification for academic purposes. In summary, DLCR follows all licensing requirements for every dataset in the paper, with the only note being not publicly releasing our generated LTCC data.

Release, Maintenance, and Ethical Use of DLCR-generated Data: As mentioned throughout the paper, we publicly release our generated data with full accessibility (no PI contact required, full data and code available) at this URL: https://huggingface.co/datasets/ihaveamoose/DLCR. Since we release our data on a publicly available data storage website, there is no maintenance requirements or future access restriction for our generated data. Regarding ethical use of our data, since we only modify the clothing items of human subjects in pre-existing CC-ReID, there are no additional privacy or ethical concerns that are not already addressed by these datasets when they were released. However, we do cover the face when possible in the paper to further protect privacy (we do show the face occasionally in order to exhibit certain qualities of DLCR-generated data). Regarding the license of our data, we choose CC BYNC-SA 4.0 only because CCVID requires our data to be released under that license due to our use and modification of their data. **For all intents and purposes, we allow for full and unrestricted academic use of our code and data as long as we are properly credited in the work.**

8. Additional Experiments and Ablations

8.1. Ablations

To demonstrate the utility of each proposed component of DLCR, we perform ablations on the PRCC dataset and show these results in Table A1. The first row contains results obtained with a baseline CAL model without using DLCR.

Effectiveness of ID-preserving generated data: Simply adding our generated data during Re-ID training (Table A1, row 3) leads to the most significant improvement in performance, with a 5.5% increase in top-1 accuracy and 3% increase in mAP over the baseline (Table A1, row 1). In row 2 of Table A1, where data is generated using standard image-to-image diffusion, we see marginal improvements

Table A1. Ablations on each proposed component of DLCR on the PRCC dataset. The addition of each component yields consistent improvements in performance. Baseline CAL results are given in the first row. Cumulative performance gains of each component with respect to the baseline are shown in green.

Generated Data	LLMs	Progressive Learning	Prediction Refinement	Top-1	mAP
✗	✗	✗	✗	55.2	55.8
✓ (Standard Diffusion)	✗	✗	✗	55.7 **+0.5**	55.9 **+0.1**
✓ (Ours)	✗	✗	✗	60.7 **+5.5**	58.9 **+3.1**
✓ (Ours)	✓	✗	✗	62.9 **+7.7**	60.9 **+5.1**
✓ (Ours)	✓	✓	✗	65.0 **+9.8**	62.4 **+6.6**
✓ (Ours)	✓	✓	✓	66.5 **+11.3**	63.0 **+7.2**

as opposed to our method, which uses ID-preserving masks and inpainting. This shows that increasing the variety of clothes-changing training samples, while still preserving the subject's ID-related information, is integral to improving CC-ReID performance.

Effectiveness of LLM prompts: Row 3 in Table A1 corresponds to generating data by using random clothes prompts as a text condition. In row 4 of Table A1, we show the impact of using LLMs to extract the textual clothing descriptions in a dataset for text conditioning, with a 2% boost in performance. More information regarding the use and impact of LLMs in DLCR can be found in Sec. 13.

Effectiveness of progressive learning: Since DLCR generates multiple clothes-changed samples for each training image, G_{train} is larger in size than D_{train}. To effectively utilize G_{train} during training, while also mitigating additional training time, we gradually introduce new clothing variations at the mini-batch level with our progressive learning strategy. This further increases model performance by another 2% (Table A1, row 5), highlighting the importance of elaborate procedures when training with generated data.

Effectiveness of prediction refinement: As discussed in Sec. 3.2.2, our diffusion-based inpainting method can also be used as a query augmentation at test-time. The model's predictions on these augmentations are ensembled using Alg. 1 to obtain refined similarity scores for each subject, resulting in better test-time predictions and yielding a further 1.5% improvement (Table A1, row 6).

9. Additional Generated Examples

Figure A3 showcases additional qualitative examples of our generated data spanning three datasets: PRCC, LTCC, and CCVID. These examples illustrate how our inpainted images respect the provided prompts, while also displaying the realism and diversity in the generated clothing. For example, in the top-left sample of Figure A3, we see that the diffusion inpainting model properly generates all the different combinations of pants and shirts described in the prompts. In the top-right example, the inpainting model even completes slightly more difficult tasks, such as replacing a dress with two separate clothing items (blouse and shorts), while preserving the realism of the synthesized image. Leveraging diffusion to generate additional clothes-changed images is paramount for enriching training data, as DLCR is equipped to controllably and accurately increase the clothing diversity of any given dataset.

10. Improving Existing Methods using DLCR (cont.)

In Table 3 of the main paper, we showed how training any Re-ID model simply only using generated data from stage 1 of DLCR still yields large improvements. To fully exhibit the benefits of DLCR, we provide additional results in Table A2, where we apply both stages of DLCR to these models. When only using stage 1 of DLCR to train various Re-ID models, large improvements of $\approx 7\% - 28\%$ are observed across many models (middle rows of Table A2). With the introduction of progressive learning and prediction refinement in stage 2 of DLCR, top-1 accuracy on standard Re-ID models further increases by roughly $2-3\%$ for a cumulative increase of nearly $\approx 10\% - 30\%$ over the baseline (last rows of Table A2). Similarly, top-1 accuracy on CC-ReID models improves by $\approx 1 - 4\%$ when adding stage 2 of DLCR, with the larger improvement possibly coming from the explicit clothes-invariance already instilled in these models. Thus, while stage 1 of DLCR can be applied to any model for significant performance gains, we further exhibit that using both stages yields the best results across many standard and CC-ReID models.

11. Visualizing DLCR's Improvements

11.1. Qualitative Retrieval Examples

One way to visualize DLCR's improvement over CAL is shown in Figure A4, where we visualize the query-gallery retrievals for both models during evaluation. In the top half of the figure, DLCR correctly matches a query image with multiple gallery images of the same subject, regardless of the change in the subject's clothing. In the bottom half, we show how CAL fails on the same exact samples by erroneously retrieving images from the gallery of different subjects wearing similar clothing items to the query image. Despite the fact that CAL is explicitly designed for clothes-invariance, there still appears to be some bias towards clothing during evaluation. As we mention in the main paper, solely discriminative approaches to CC-ReID are not currently sufficient and leave significant room for improvement, such as utilizing generative approaches like DLCR. For example, one explanation for CAL's limitation could be the limited number of clothes-changes in the training data which prevents the full use of CAL's clothes-invariant learning strategy. Hence, DLCR better equips CC-ReID models to learn clothes-agnostic person

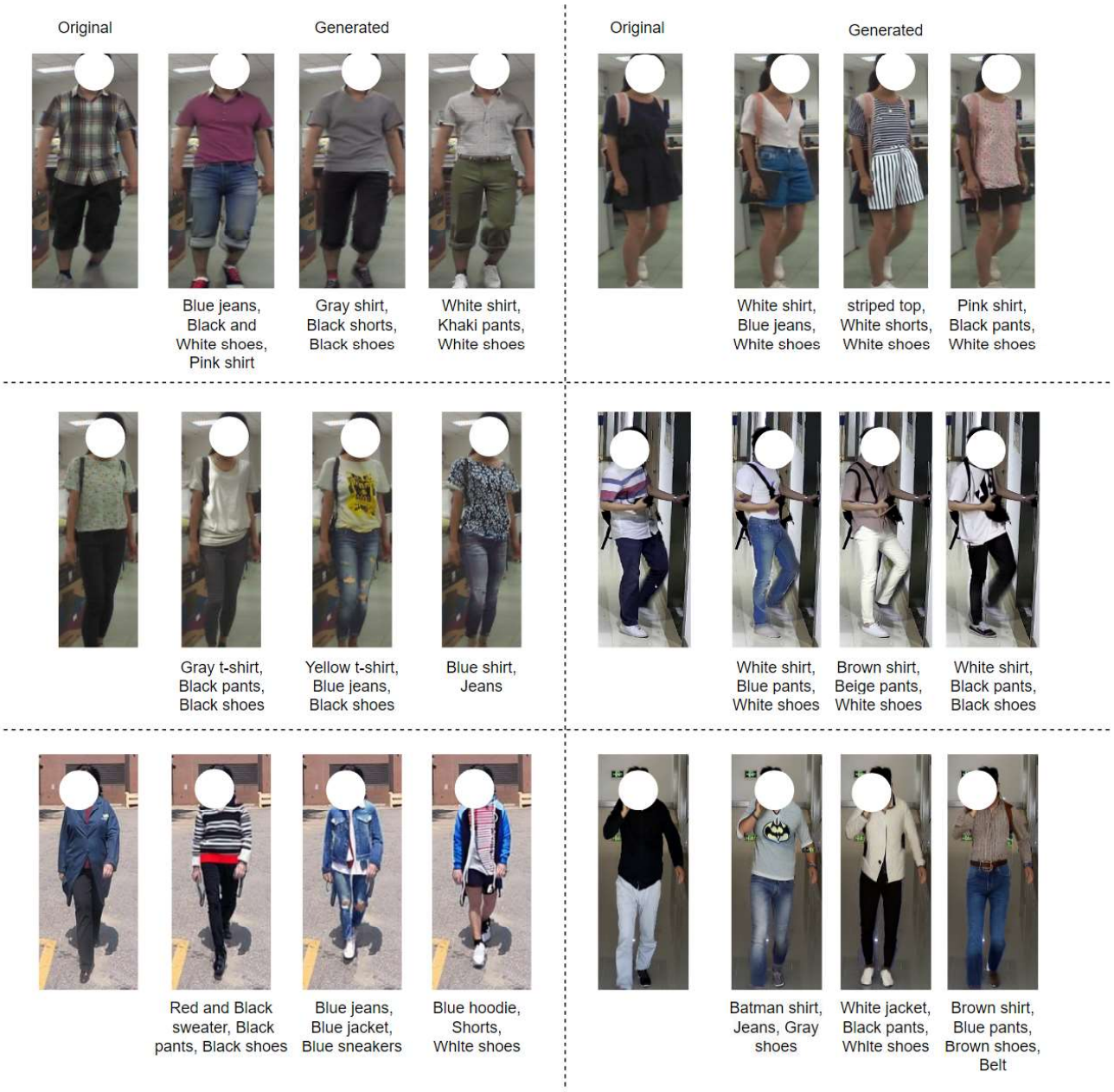

Figure A3. Qualitative examples of our generated data for PRCC (row 1), LTCC (row 2) and CCVID (row 3) datasets. For each original image, we show three inpainted versions. The prompts used to generate the inpainted samples are placed under the corresponding images. These samples depict high-quality, diverse generated data that is prompt-aligned.

features through the use of its generated data (stage 1) and training/testing strategies (stage 2).

11.2. t-SNE Feature Plots

As an additional way to visualize how DLCR improves top-1 accuracy during retrieval, Figure A5 provides t-SNE plots to compare learned person features between a baseline CAL model and DLCR. As described in Sec. 4 of the main paper, a query image is paired with a gallery image during evaluation by retrieving the gallery image with the most similar person features to the query image. An oracle model would produce identical person features for a query and gallery image if the same subject is in the image, regardless of clothing, background, occlusions, body pose, etc. In the left plot in Figure A5, we show the resulting query and gallery person features from DLCR for five randomly

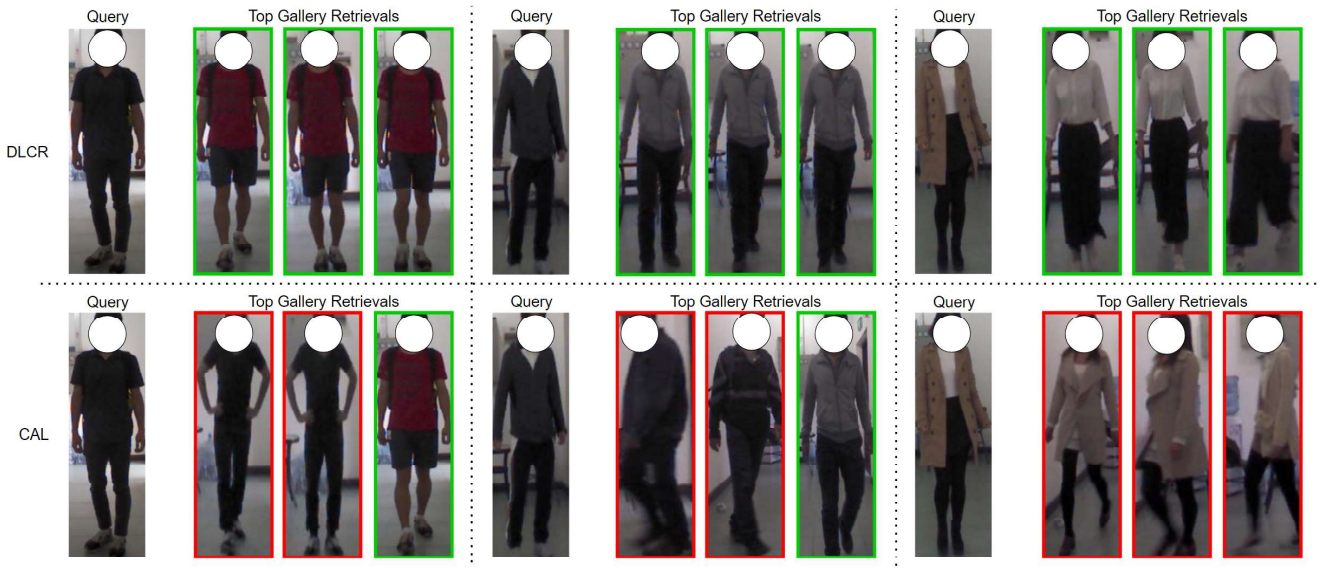

Figure A4. Qualitative retrievals of CAL+DLCR versus baseline CAL. For a given query image, the top-3 retrieved images from the gallery are shown, with correct and incorrect retrievals outlined in green and red, respectively. Despite clothing changes between the query and gallery images, CAL+DLCR retrieves the correct subject regardless of appearance. However, CAL still favors clothing items during retrieval, often retrieving incorrect subjects from the gallery that share similar clothing items to the query. This shows that discriminative approaches to clothing-invariance, such as CAL, can still be further improved using generative methods.

selected subjects in the PRCC testing set. For each query and gallery image of a particular subject, DLCR correctly produces person features that cluster in both an inter- and intra-class fashion. Not only does each gallery feature of a particular subject cluster with other gallery features of the same subject, but the same clustering behavior occurs between each query feature of a particular subject as well. More importantly, the smallest distance between a query feature cluster and gallery feature cluster produced by DLCR share the same subject ID, indicating correct test-time retrievals. However, this behavior is not seen for the same exact subjects and samples when using a baseline CAL model, as shown in the right plot of Figure A5. CAL does not produce similar person features between query and gallery images of the same subject, with the erroneous query-gallery clustering examples explicitly highlighted with multi-colored boundaries. For example, the query feature cluster for Subject 272 is closer in distance to the gallery feature cluster of Subject 4, decreasing top-1 performance since the wrong gallery image would be retrieved.

11.3. Activation Maps

In Figure A6, we compare the feature maps of CAL+DLCR with a baseline CAL model on the PRCC and LTCC datasets. Notably, CAL+DLCR exhibits a stronger focus on identity-related features. For instance, the DLCR feature map in the LTCC examples prioritizes the subject's face over the footwear when making a prediction. Furthermore, DLCR retains the ability to leverage person-specific features that are within the clothing region (*e.g.* body shape) despite its invariance towards clothing, as seen in the PRCC examples.

12. Optional: Discriminator-guided diffusion

Considering our downstream task of Re-ID training, it is important to ensure that G_{train} closely resembles the distribution from which D_{train} originated. The generated set G_{train} is obtained using a pretrained diffusion model, but it has been shown that without additional fine-tuning, there can be a moderate gap between the diffusion model's generated data and the real data distribution [25]. At the same time, fine-tuning a diffusion model can become computationally expensive and may lead to undesired results, like overfitting [25, 36]. To avoid these problems, similar to [25], we investigated using a discriminator d_ϕ to guide the pretrained diffusion model to generate data that is better aligned with the training data distribution. Following [25], we train the discriminator d_ϕ to minimize the domain gap with respect to the noisy examples at different timesteps from the real (D_{train}) and generated (G_{train}) data sets. Consequently, the employed training objective of d_ϕ (*i.e.* $\mathcal{L}_d$) is to distinguish between the real and generated examples:

$$\mathcal{L}_d = -\mathbb{E}\left[\log d_\phi(x_t, t) + \log\left(1 - d_\phi(\hat{x}_t, t)\right)\right], \quad (5)$$

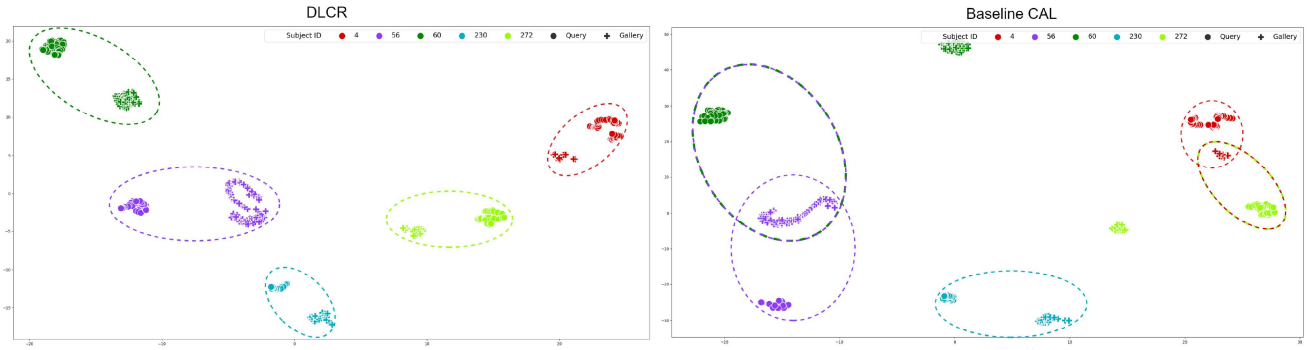

Figure A5. t-SNE visualizations of the query and gallery features produced by CAL+DLCR and CAL for 5 randomly selected test subjects in the PRCC dataset. With a baseline CAL model (right), the query feature cluster for Subject 272 are erroneously closer to the gallery feature cluster for Subject 4, with the same issue between Subject 60 and 56. The incorrect clustering behaviors are marked with multi-colored boundaries. In contrast, the gallery and query feature clusters produced by CAL+DLCR (left) for the same subjects correctly cluster together, exhibiting DLCR's direct impact in learning better discriminative features and improving top-1 accuracy.

Table A2. Results on PRCC when using both stages of DLCR on various standard Re-ID and CC-ReID models. Adding stage 2 results in better mAP and top-1 accuracy values on every model, with the green numbers in parentheses signifying cumulative improvement over the original baseline model. † denotes reproduced results using open-source code.

Standard Re-ID Models		
Model	Top-1	mAP
PCB [49]	41.8	38.7
PCB + DLCR (Stage 1)	53.3 (+11.5)	50.7 (+12.0)
PCB + DLCR (Stage 1 + 2)	**56.5** (+14.7)	**51.0** (+12.3)
MGN [53]	33.8	35.9
MGN + DLCR (Stage 1)	62.5 (+28.7)	57.6 (+21.7)
MGN + DLCR (Stage 1 + 2)	**64.8** (+31.0)	**58.0** (+22.1)
HPM [12]	40.4	37.2
HPM + DLCR (Stage 1)	56.0 (+15.6)	50.9 (+13.7)
HPM + DLCR (Stage 1 + 2)	**57.5** (+17.1)	**51.2** (+14.0)
CC-ReID Models		
Model	Top-1	mAP
CAL † [15]	55.2	55.8
CAL + DLCR (Stage 1)	62.9 (+7.7)	60.9 (+5.1)
CAL + DLCR (Stage 1 + 2)	**66.5** (+11.3)	**63.0** (+7.2)
AIM † [59]	55.7	56.3
AIM + DLCR (Stage 1)	60.2 (+4.5)	59.0 (+2.7)
AIM + DLCR (Stage 1 + 2)	**61.9** (+6.2)	**60.5** (+4.2)
GEFF [2]	83.6	64.0
GEFF + DLCR (Stage 1)	84.6 (+1.0)	66.0 (+2.0)
GEFF + DLCR (Stage 1 + 2)	**85.8** (+2.2)	**66.2** (+2.2)

where $t \sim \mathcal{U}([0,T])$, $x_t \sim q(x_t|x), x \sim \mathcal{U}(D_{train})$ and $\hat{x}_t \sim q(\hat{x}_t|\hat{x}), \hat{x} \sim \mathcal{U}(G_{train})$.

We use the trained discriminator to design a score function that guides the generative process to synthesize samples that are highly likely to be classified as real by the discriminator. The respective score function is the following:

$$h_\phi = \nabla_{x_t} \log \frac{d_\phi(x_t, t)}{1 - d_\phi(x_t, t)}. \tag{6}$$

We incorporate this score function h_ϕ into noise estimation as follows (see Sec. 12.1 for the full derivation):

$$\epsilon_\phi(x_t, t) = -\sigma_t \cdot h_\phi, \tag{7}$$

where $\sigma_t = \sqrt{1 - \bar{\alpha}_t}$ is the standard deviation of $q(x_t|x_0)$. Then, we guide the reverse denoising process of Stable Diffusion by adding the noise estimation from our discriminator, as follows:

$$\epsilon_\theta^\phi(x_t, t) = \epsilon_\theta(x_t, t) + w \cdot \epsilon_\phi(x_t, t), \tag{8}$$

where w denotes the weight of discriminator guidance.

Finally, we generate an improved version of G_{train} by modifying the mean $\boldsymbol{\mu}_\theta(x_t, t)$ of the reverse denoising pro-

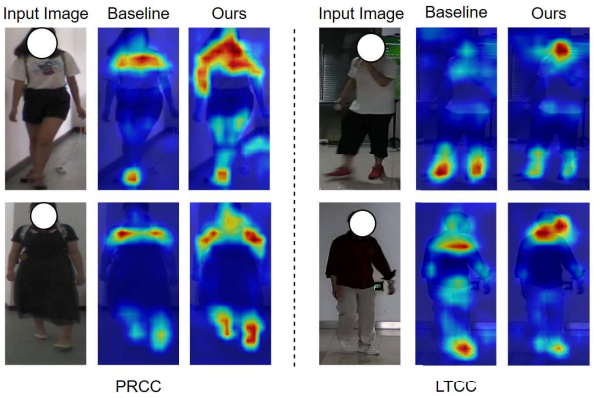

Figure A6. Feature map comparison between DLCR and a baseline CAL model. DLCR enforces better robustness against clothing variations in CC-ReID.

cess (Eq. (3)) to include the discriminator-guided noise estimation $\epsilon_\phi(x_t, t)$:

$$\boldsymbol{\mu}_\theta^\phi(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta^\phi(x_t, t)\right). \quad (9)$$

We found that in some cases, using discriminator-guided diffusion improved results on PRCC and CCVID by $1-2\%$, however the process is optional since it requires some additional training of a discriminator and re-generation of the data (following the process proposed in [25]). In certain cases where data using a pretrained diffusion model is not generating sufficiently in-domain data, we included this section as one potential solution as well as a strong area for future work.

12.1. Details Regarding Discriminator Guidance

In this section, we will present the derivations for Eq. (6) and Eq. (7), while also providing more intuitive explanations for them. Before describing these details, we remind the reader that in the continuous formulation [47] of diffusion models, the forward process is described by a stochastic differential equation (SDE):

$$\partial x_t = f(x_t, t)\partial t + g(x_t)\partial w, \quad (10)$$

and the reverse process is also a diffusion process [1, 47], given by the following SDE:

$$\partial x_t = [f(x_t, t) - g^2(x_t)\nabla_x \log q(x_t)]\partial t + g(x_t)\partial \bar{w}, \quad (11)$$

where f is called the drift coefficient, g denotes the diffusion coefficient and $\nabla_x \log q(x_t)$ is called the score function. This score function is what is estimated by diffusion models in order to solve the reverse process.

Regarding discriminator guidance [25], we emphasize that its objective is to address the situations when the diffusion model converges to a local optimum, thereby failing to provide the most accurate noise or score estimations. In a formal sense, when dealing with these cases, it becomes necessary to correct the marginal distribution $p_\theta(x_t)$ of the forward process (Eq. (10)), which originates with samples drawn from $p_\theta(x_0)$, in order to match the marginal distribution $q(x_t)$ of the forward process initiated with samples from $q(x_0)$. Dongjun *et al*. [25] introduce this correction term as an additional score function that depends on a discriminator, and the term is used in the reverse process at each denoising step. We derive this term starting with the following simple observation:

$$q(x_t) = p_\theta(x_t) \cdot \frac{q(x_t)}{p_\theta(x_t)}, \quad (12)$$

and then, if we apply the logarithm and the gradient, we get:

$$\nabla_x \log q_(x_t) = \nabla_x \log p_\theta(x_t) + \nabla_x \log \frac{q(x_t)}{p_\theta(x_t)}. \quad (13)$$

Eq. (13) implies that if we want to obtain the optimal score function ($\nabla_x \log q(x_t)$ - the one required to solve Eq. (11)), we can correct the model score estimation $\nabla_x \log p_\theta(x_t)$ using the log-gradient of the rate $\frac{q(x_t)}{p_\theta(x_t)}$. However, we cannot compute this rate in practice, thereby Dongjun *et al*. [25] propose to estimate it via a discriminator. More precisely, a discriminator, $d_\phi(x_t, t)$, is trained to distinguish between real and generated samples. After the training is completed, $d_\phi(x_t, t)$ will return the probability for the sample x_t of being a real example at every timestep. Thus, it is an estimation for $q(x_t)$ and, in a similar manner, $1 - d_\phi(x_t, t)$ will approximate $p_\theta(x_t)$. Therefore, we use:

$$h_\phi = \nabla_{x_t} \log \frac{d_\phi(x_t, t)}{1 - d_\phi(x_t, t)} \quad (14)$$

as a correction term in the reverse process, which is the same h_ϕ as defined in Eq. (6).

Moving further, Eq. (7) denotes the relation between the noise estimation and the corresponding score function. We will derive a more general form of this equation by exploiting the reparameterization trick for a Gaussian distribution and Tweedie's formula [41].

Given an arbitrary Gaussian distribution $\mathcal{N}(x; \mu, \sigma^2\mathbf{I})$ and its corresponding density function $p(x)$. The reparameterization trick applied for distribution is the following:

$$x = \mu + \sigma \cdot \epsilon \iff \mu = x - \sigma \cdot \epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I}). \quad (15)$$

In statistical literature, Tweedie's formula generally shows how to express the mean of an arbitrary Gaussian distribution (μ) given its samples (x) and the score function ($\nabla_x \log p(x)$). Specifically, we apply Tweedie's formula on the previous Gaussian distribution and we obtain the following result:

$$\mu = x + \sigma^2 \nabla_x \log p(x). \quad (16)$$

If we combine the results from Eq. (15) and Eq. (16), then we obtain the following:

$$\epsilon = -\sigma \nabla_x \log p(x), \quad (17)$$

and we can apply this result for the discriminator score function denoted by h_ϕ and $\sigma = \sqrt{1-\bar{\alpha}_t}$ to obtain Eq. (7):

$$\epsilon_\phi(x_t, t) = -\sqrt{1-\bar{\alpha}_t} h_\phi. \quad (18)$$

13. Utility of LLaVA and LLaMA

In this section, we provide reasoning behind utilizing LLaVA and LLaMA to extract clothing descriptions of the clothing IDs present in a dataset.

In Table A3, we provide the results obtained for three cases on the PRCC dataset. In the first row, we ask only

LLaMA to give us random sets of clothing items for each specific body part (top, bottom, footwear) for clothes inpainting. In the second row, we only use LLaVA to create clothing prompts from a single image of a given clothing ID (no use of LLaMA summarization). We compare these two ablations to DLCR's main results in the third row, where both LLaVA and LLaMA are used to construct prompts for clothes inpainting. Table A3 highlights that it is beneficial to generate data from clothes that are already (or close to) present in the dataset. This aspect is intuitive because during training, if two subjects are wearing relatively identical clothing items, a CC-ReID model cannot exploit the clothing information to classify a subject. Thus, the model must rely on ID-specific features to differentiate between the subjects. Moreover, we illustrate the robustness of our method to different LLMs in Table A4, where we compare LLaVA with InstructBLIP [7] for extracting textual clothing descriptions. We observe very similar results, implying that the selection of LLMs has a minimal impact on DLCR's performance. Overall, our use of LLaVA and LLaMA is beneficial for CC-ReID performance, as seen in the $\approx 2\%$ increase in both top-1 accuracy and mAP.

Table A3. Results on the PRCC dataset when generating data with and without extracted text prompts from LLaVA and LLaMA. Cumulative improvements over the baseline are shown in green.

LLaVA	LLaMA	PRCC	
		Top-1	mAP
✗	✓ (random clothing prompts)	60.7	58.9
✓	✗	61.6 (+0.9)	60.1 (+1.2)
✓	✓	**62.9 (+2.2)**	**60.9 (+2.0)**

Table A4. Stage 1 DLCR results on PRCC with different VLMs.

Visual Language Model (VLM)	Top-1	mAP
LLaVA	62.9	60.9
InstructBlip	62.9	61.3

14. Progressive Learning Intuition

As discussed in Sec. 14, one limitation of generating so much additional data would be the impact on training time. The choice to inject the generated data at the batch level was largely driven by this point, as shown in Table A5. On the other hand, Figure A7, provides a t-SNE plot to visualize and compare the principal components between our generated images and the original images for three random subjects from the PRCC dataset. It is easy to notice that the inclusion of generated samples leads to a larger variance in the data, thus making the task of differentiating between users more challenging. To alleviate this issue at the early stages of training, we use a smaller number of generated samples and gradually incorporate additional generated samples as training progresses. This strategy allows the model to more effectively adapt to the increasingly diverse distribution of the generated data, as illustrated by the performance shown in Table A1, row 5.

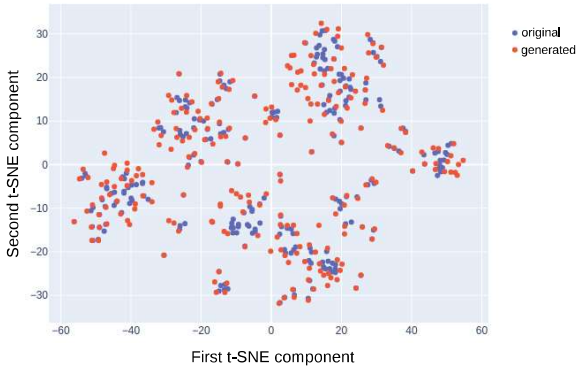

Figure A7. t-SNE plot of our generated data versus the original data for three random subjects in the PRCC dataset. As our generated data introduces a larger variance in the training distribution, our progressive learning strategy is effective in optimizing CC-ReID performance.

Table A5. Training times with and without progressive learning when using DLCR generated data. With identical experimental setups, progressive learning not only reduces training time, but also provides some performance boost (see Table A1).

Progressive Learning	PRCC Training Epoch Time (seconds) ↓	Best Top-1 ↑
✗	691s	64.0
✓	65s	65.0

15. Ablation on Number of Generated Images (K)

As detailed in the implementation details of the main paper, we set the number of inpainted versions for each original training image $K = 10$. In Figure A8, we perform an ablation on the values of K and provide the resulting top-1 accuracies on the PRCC dataset. Notably, the improvements observed when increasing K beyond 10 are minimal, hence explaining why we do not simply generate more images to increase the size of our contributed data. We set $K = 10$ as it strikes a favorable balance between the performance gain facilitated by the additional generated data, and the time consumed by the generation process. However, the DLCR generation process is open-source and can be used by others to generate more data for cases such as large-scale CC-ReID pretraining. Note that the results presented in Figure A8

are obtained by simply concatenating the generated data with the initial training set of PRCC, *i.e*. we do not perform progressive learning, discriminator guidance or prediction refinement in this study.

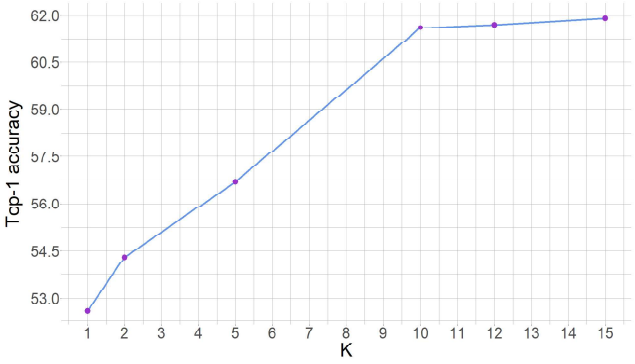

Figure A8. Top-1 accuracy rates achieved for various values of K (number of inpainted images per original image) by DLCR on PRCC dataset. The accuracy improvements for $K \geq 10$ are marginal. Therefore, we standardized the value of K to 10 for all our experiments.

16. Time and Space Complexity of DLCR

We provide explicit time and space complexities for each component of DLCR in Table A6. Specifically, we report the throughput of each of our components, measured by how many images each component can process per second, as well as the memory each component takes on the GPU. Despite our use of LLMs and diffusion, DLCR as a whole is still fairly computationally inexpensive since we only use these large models in an inference setting. During the generation of CC-ReID training data via diffusion inpainting, which is an offline process (Stage 1 of DLCR), we use 50 denoising steps at a resolution of 768×256. Then we apply stage 2 and train the CC-ReID model with the combination of original and generated data. During inference, for query inpainting in our prediction refinement strategy, we reduce the denoising steps from 50 to 10 and divide the image resolution by half, which results in a nearly $\sim 3\times$ speedup while maintaining similar performance. In summary, stage 1 of DLCR can be fully implemented on a single NVIDIA A100 80GB GPU (or a A6000 48GB GPU with some tricks to fit the LLMs).

17. Limitations and Future Work

Limitations: One limitation of DLCR is low-quality generated images on low-resolution images. On low-resolution or small-scale images, not only can the ID-preserving mask be inaccurate, but the diffusion model itself struggles to properly inpaint the clothing regions correctly. Some datasets mentioned in this paper have these types of images, and the generated data on these small corner case images can be low quality and incorrect/not prompt-aligned. Due to our use of a pretrained Stable Diffusion model, one possible solution is to fine-tune Stable Diffusion on low-resolution images for better performance. Diffusion models have also shown strong promise in image super-resolution, which could be included in the DLCR generation pipeline to deal with low-resolution images. Another limitation could be the diminishing returns of additional data (Fig. A8). While DLCR will most likely show significant performance gains in data-scarce domains, finding a method to better leverage mass amounts of data while mitigating diminishing returns would heavily impact the positive effect DLCR has on downstream performance.
Future Work: One possible direction of future work for DLCR could be investigating our test-time prediction strategy compared to other re-ranking methods. In the context of CC-ReID, we show that our prediction refinement strategy is compatible with other re-reranking methods such as GEFF [2], but further experimentation could yield even higher SOTA results. Another avenue of future use is leveraging the DLCR generation method for other vision tasks that are compatible with localized image editing. For example, our method of generating data with preserved areas using a binary mask could apply to medical imaging for data augmentation.

Table A6. Analysis of time and space complexity for each component in DLCR. Throughput is measured in images/sec, with all experiments run on a single NVIDIA A100 80GB GPU.

DLCR Component	Throughput (img/sec) ↑	Memory ↓
ID-Preserving Mask Extraction	61.6	1GB
Clothes Description Extraction	4.0	56GB
Training Inpainting	0.89	4GB
Query Inpainting	3.33	2.6GB
Prediction Refinement (Algorithm 1)	642.6	$<$ 1GB