LANIT :

Language-Driven Image-to-Image Translation for Unlabeled Data

Youngjae Kim

2023.11.03

Medical Imaging & Intelligent Reality Lab (MI2RL)
Convergence Medicine/Radiology,
University of Ulsan College of Medicine
Asan Medical Center
South Korea

1. About the paper

LANIT: Language-Driven Image-to-Image Translation for Unlabeled Data

Jihye Park *[1], Sunwoo Kim *[1], Soohyun Kim *[1], Seokju Cho [1],
Jaejun Yoo [2], Youngjung Uh [3], Seungryong Kim †[1]

[1] Korea University, Seoul, Korea [2] UNIST, Ulsan, Korea [3] Yonsei University, Seoul, Korea
[1]{ghp1112, sw-kim, shkim1211, seokju_cho, seungryong_kim}@korea.ac.kr
[2]jaejun.yoo@unist.ac.kr [3]yj.uh@yonsei.ac.kr

1. About the paper

openaccess.thecvf.com/CVPR2023?day=all

An Image Quality Assessment Dataset for Portraits
Nicolas Chahine, Stefania Calarasanu, Davide Garcia-Civiero, Théo Cayla, Sira Ferradans, Jean Ponce
[pdf] [supp] [arXiv] [bibtex]

MSeg3D: Multi-Modal 3D Semantic Segmentation for Autonomous Driving
Jiale Li, Hang Dai, Hao Han, Yong Ding
[pdf] [supp] [arXiv] [bibtex]

Robust Outlier Rejection for 3D Registration With Variational Bayes
Haobo Jiang, Zheng Dang, Zhen Wei, Jin Xie, Jian Yang, Mathieu Salzmann
[pdf] [supp] [arXiv] [bibtex]

Dynamically Instance-Guided Adaptation: A Backward-Free Approach for Test-Time Domain Adaptive Semar
Wei Wang, Zhun Zhong, Weijie Wang, Xi Chen, Charles Ling, Boyu Wang, Nicu Sebe
[pdf] [supp] [bibtex]

Painting 3D Nature in 2D: View Synthesis of Natural Scenes From a Single Semantic Mask
Shangzhan Zhang, Sida Peng, Tianrun Chen, Linzhan Mou, Haotong Lin, Kaicheng Yu, Yiyi Liao, Xiaowei Zhou
[pdf] [arXiv] [bibtex]

LANIT: Language-Driven Image-to-Image Translation for Unlabeled Data
Jihye Park, Sunwoo Kim, Soohyun Kim, Seokju Cho, Jaejun Yoo, Youngjung Uh, Seungryong Kim
[pdf] [supp] [arXiv] [bibtex]

MoLo: Motion-Augmented Long-Short Contrastive Learning for Few-Shot Action Recognition
Xiang Wang, Shiwei Zhang, Zhiwu Qing, Changxin Gao, Yingya Zhang, Deli Zhao, Nong Sang
[pdf] [arXiv] [bibtex]

Fast Point Cloud Generation With Straight Flows
Lemeng Wu, Dilin Wang, Chengyue Gong, Xingchao Liu, Yunyang Xiong, Rakesh Ranjan, Raghuraman Krishnamoo
[pdf] [supp] [arXiv] [bibtex]

Text-Guided Unsupervised Latent Transformation for Multi-Attribute Image Manipulation
Xiwen Wei, Zhen Xu, Cheng Liu, Si Wu, Zhiwen Yu, Hau San Wong
[pdf] [bibtex]

Achieving a Better Stability-Plasticity Trade-Off via Auxiliary Networks in Continual Learning
Sanghwan Kim, Lorenzo Noci, Antonio Orvieto, Thomas Hofmann
[pdf] [supp] [arXiv] [bibtex]

1. Introduction

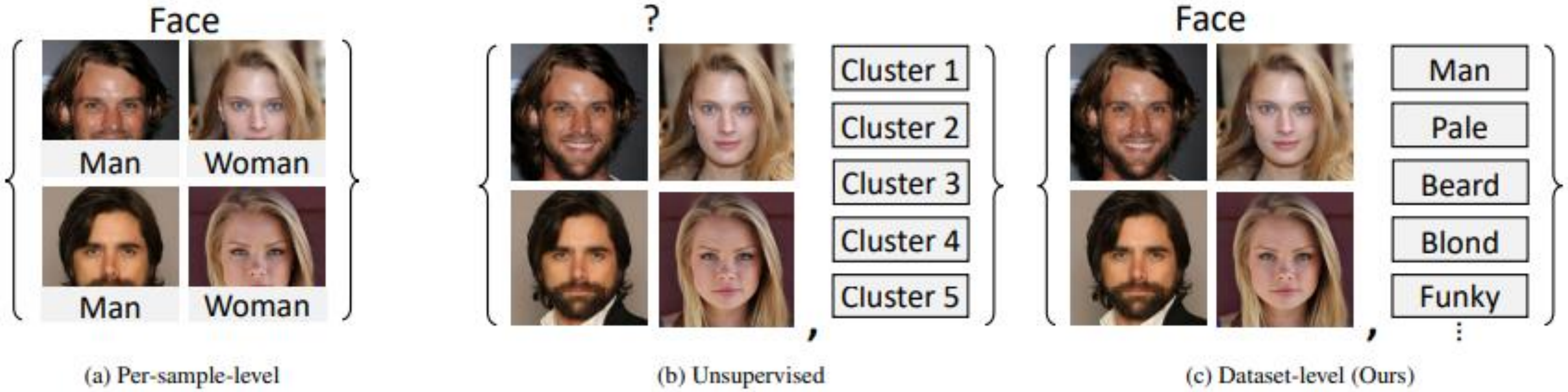


Figure 1. **Levels of supervision.** For unpaired image-to-image translation, (a) conventional methods [10, 21, 35, 55] require at least *per-sample*-level domain supervision, which is often hard to collect. To overcome this, (b) unsupervised learning methods [3, 29] learn image translation model using a dataset itself without any supervision, but it shows limited performance and lacks the semantic understanding of each cluster, limiting its applicability. Unlike them, (c) we present a novel framework that requires a dataset with possible textual domain descriptions (i.e., *dataset-level* annotation), which achieves comparable or even better performance than previous methods.

1. Introduction

- Unsupervised learning
  - \+ Ease the burden of per-sample domain
  - \+ Relatively competitive performance
  - \- Inherit limitation of using one-hot domain labels to train
  - \- Only learns clusters that are dominant in dataset

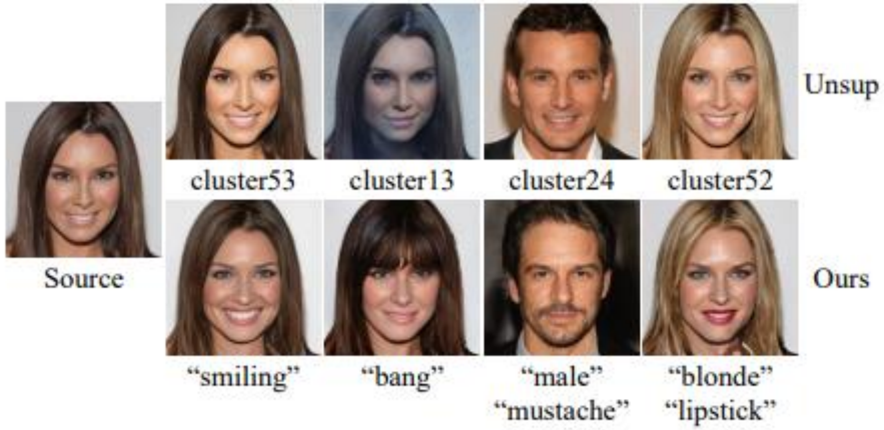
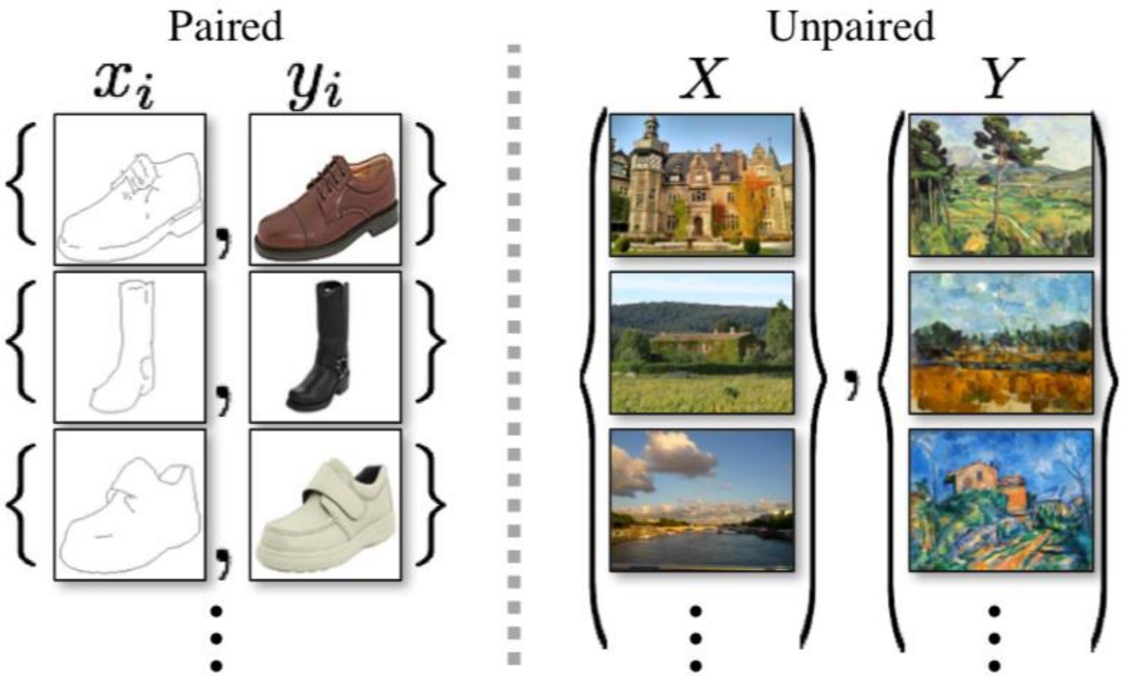


Figure 2. **Examples of semantic encoding.** The existing unsupervised method [3] allows users to translate one of several clusters. However, the learned clusters lack semantic meaning and sometimes some attributes do not appear in any clusters. Unlike this, our framework can select and train the domains that have explicit semantic meaning, which is more applicable.

2. Related Work

- Image-to-Image Translation Trend

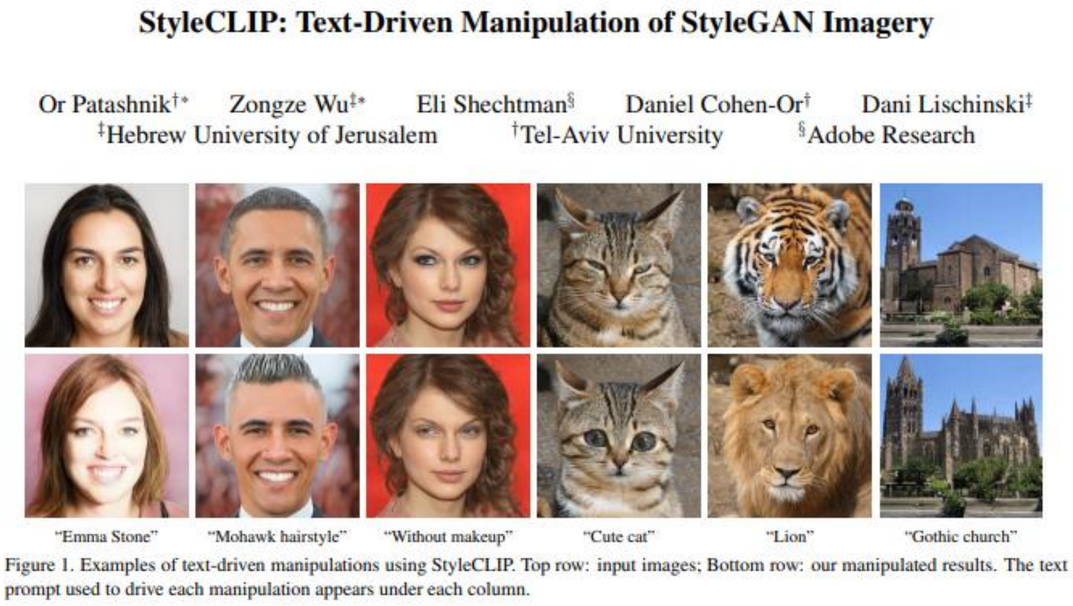


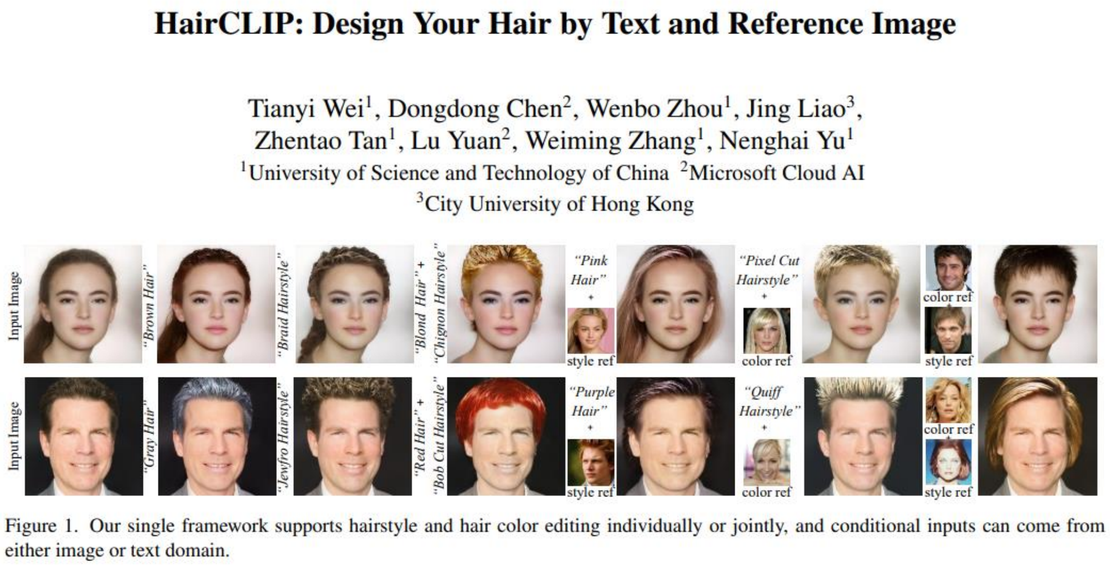
(ex. CycleGAN)

- Requires per-sample domain annotation
- Does not consider multi-hot domain labels

2. Related Work

- Vision-Language Model in Image Manipulation
- Learning semantic information of the image
- CLIP & its variation with GAN
    - –image-text pair





2. Related Work

- CLIP

사람이 제공

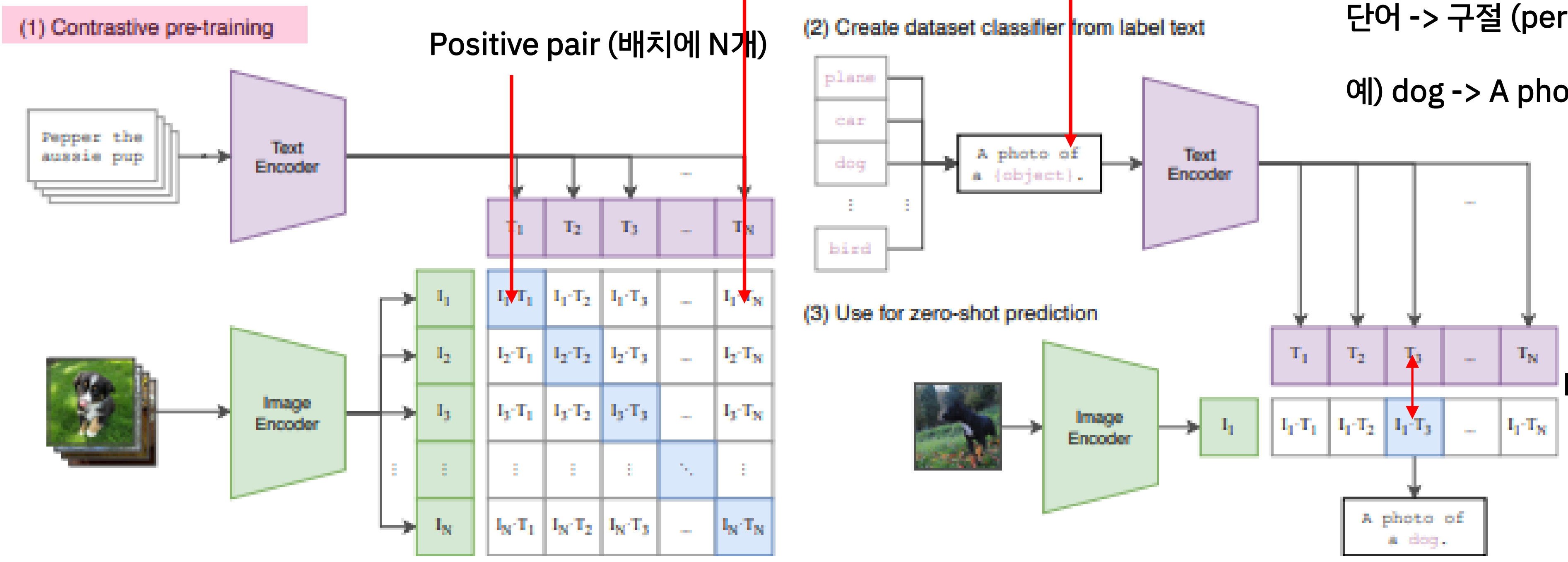
Negative pair (한 줄에 N-1개, *N줄 = N^2-N개)

Positive pair (배치에 N개)

단어 -> 구절 (performance가 좋아짐)

예) dog -> A photo of a dog

(1 Batch)



Highest cosine similarity

Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

2. Related Work

- Prompt Learning

- Success of pre-trained large-scale language models (GPT, BARD etc..)
- Inspired works of optimizing input prompt
- ex) CoCoOp -> learned continuous prompt could surpass manually-designed discrete prompt based method

Domain Adaptation via Prompt Learning

Chunjiang Ge[1] Rui Huang[1] Mixue Xie[2] Zihang Lai[3]
Shiji Song[1]

[1]Department of Automation, BN
[3]Carnegie Mellon Unive

Unsupervised Prompt Learning for Vision-Language Models

Tony Huang[1*], Jack Chu[1*], Fangyun Wei[2*†]
[1]Peking University [2]Microsoft Research Asia
tonyhuang_pku@outlook.com chuxicloud@icloud.com fawe@microsoft.com

3. Methodology

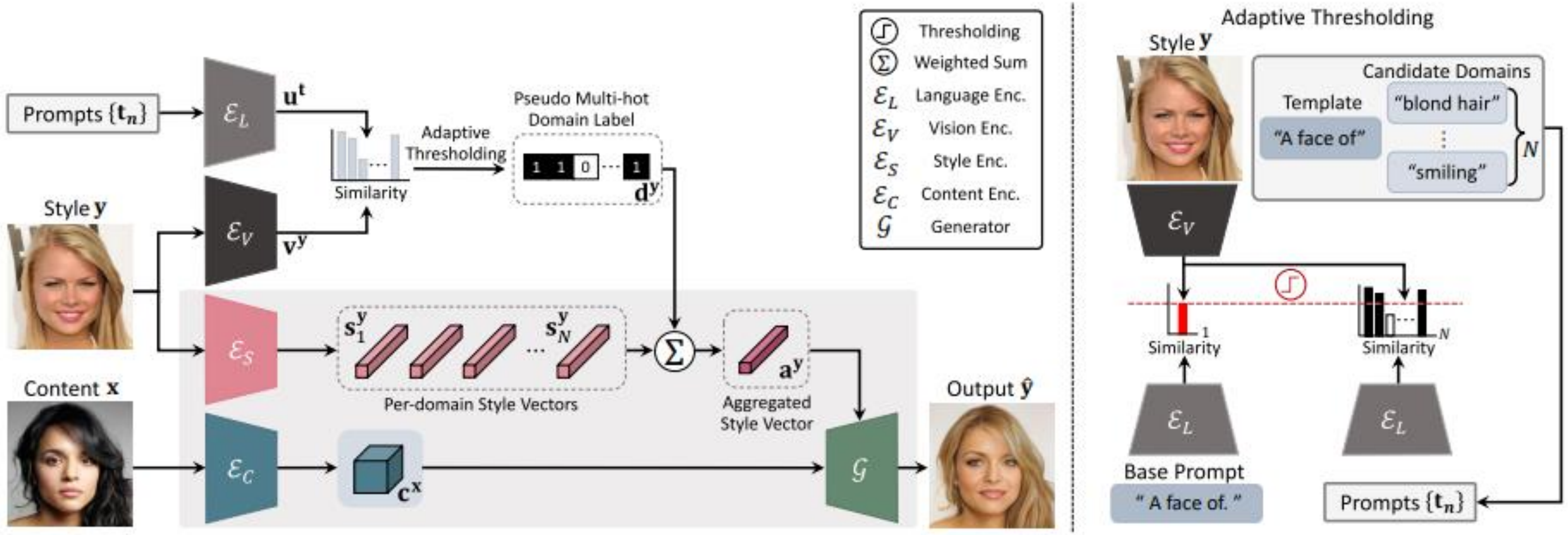


Figure 3. **Network configuration.** Our model consists of content and style encoders ($\mathcal{E}_C$, $\mathcal{E}_S$), vision-language encoders ($\mathcal{E}_V$, $\mathcal{E}_L$), and generator ($\mathcal{G}$). We extract content and style vectors from content $\mathbf{x}$ and style $\mathbf{y}$, respectively. By leveraging vision-language features and the proposed adaptive thresholding technique, we measure the pseudo domain label $\mathbf{d}^{\mathbf{y}}$ of $\mathbf{y}$. We generate $\hat{\mathbf{y}}$ with content vector $\mathbf{c}^{\mathbf{x}}$ and aggregated style vector $\mathbf{a}^{\mathbf{y}}$ through the generator.

3. Methodology

1. get Multi-hot domain label

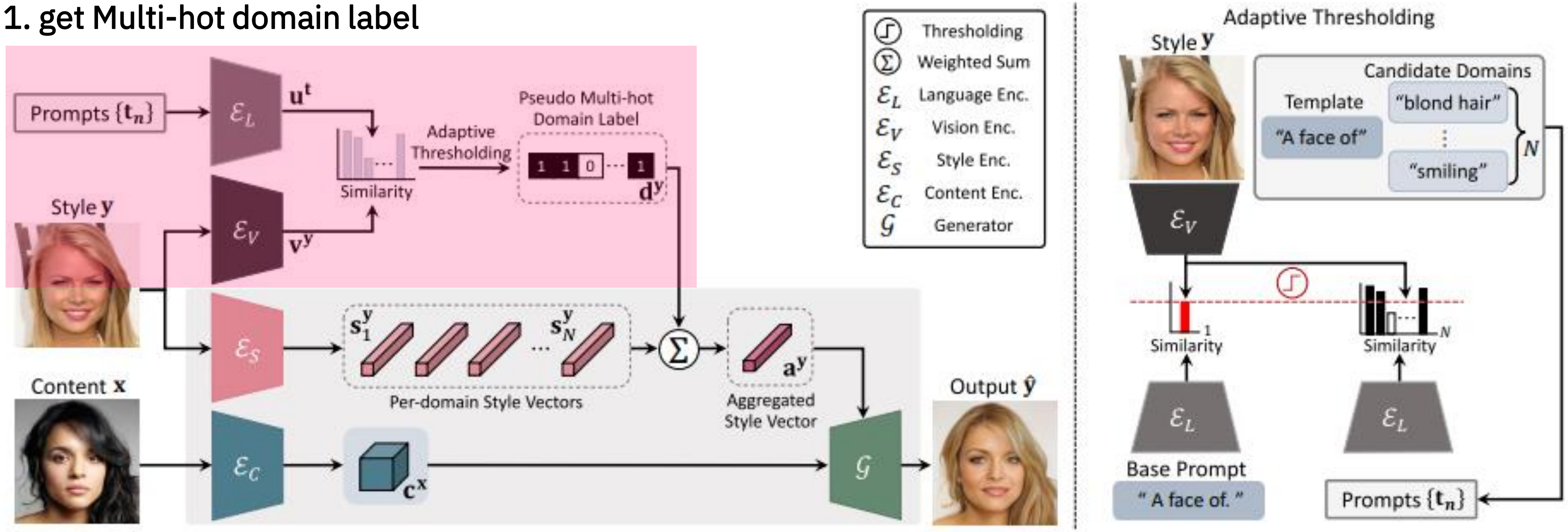


Figure 3. **Network configuration.** Our model consists of content and style encoders ($\mathcal{E}_C$, $\mathcal{E}_S$), vision-language encoders ($\mathcal{E}_V$, $\mathcal{E}_L$), and generator ($\mathcal{G}$). We extract content and style vectors from content $\mathbf{x}$ and style $\mathbf{y}$, respectively. By leveraging vision-language features and the proposed adaptive thresholding technique, we measure the pseudo domain label $\mathbf{d}^\mathbf{y}$ of $\mathbf{y}$. We generate $\hat{\mathbf{y}}$ with content vector $\mathbf{c}^\mathbf{x}$ and aggregated style vector $\mathbf{a}^\mathbf{y}$ through the generator.

3. Methodology

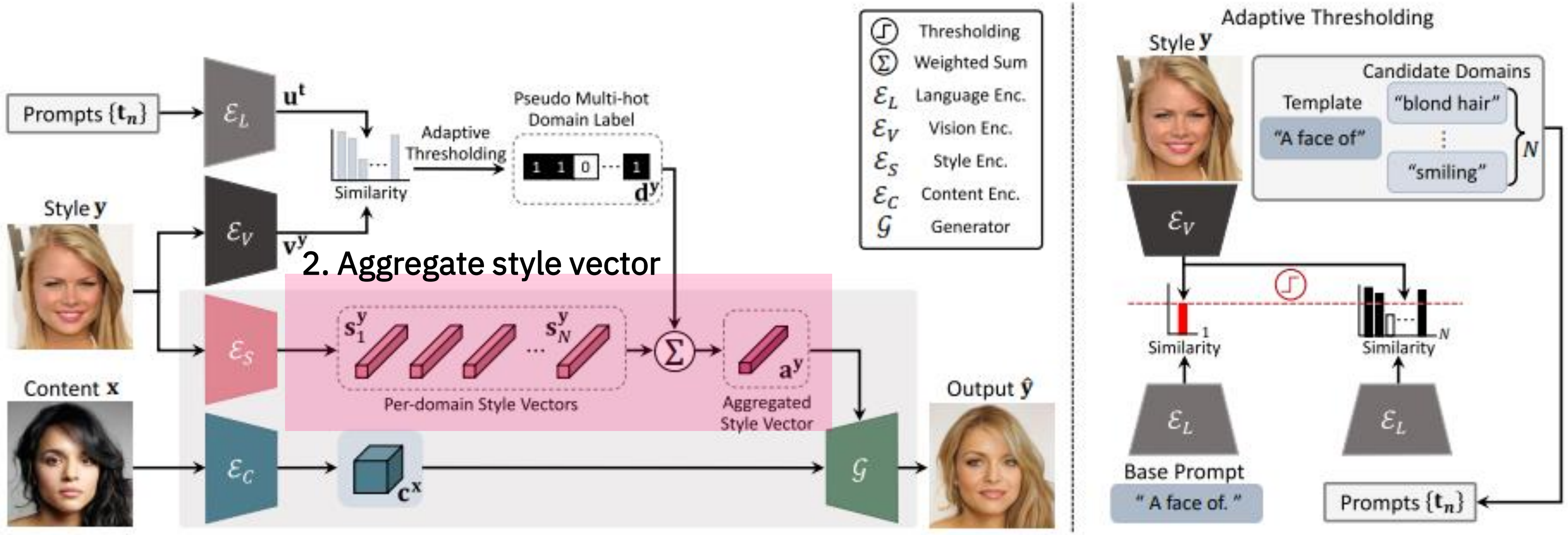
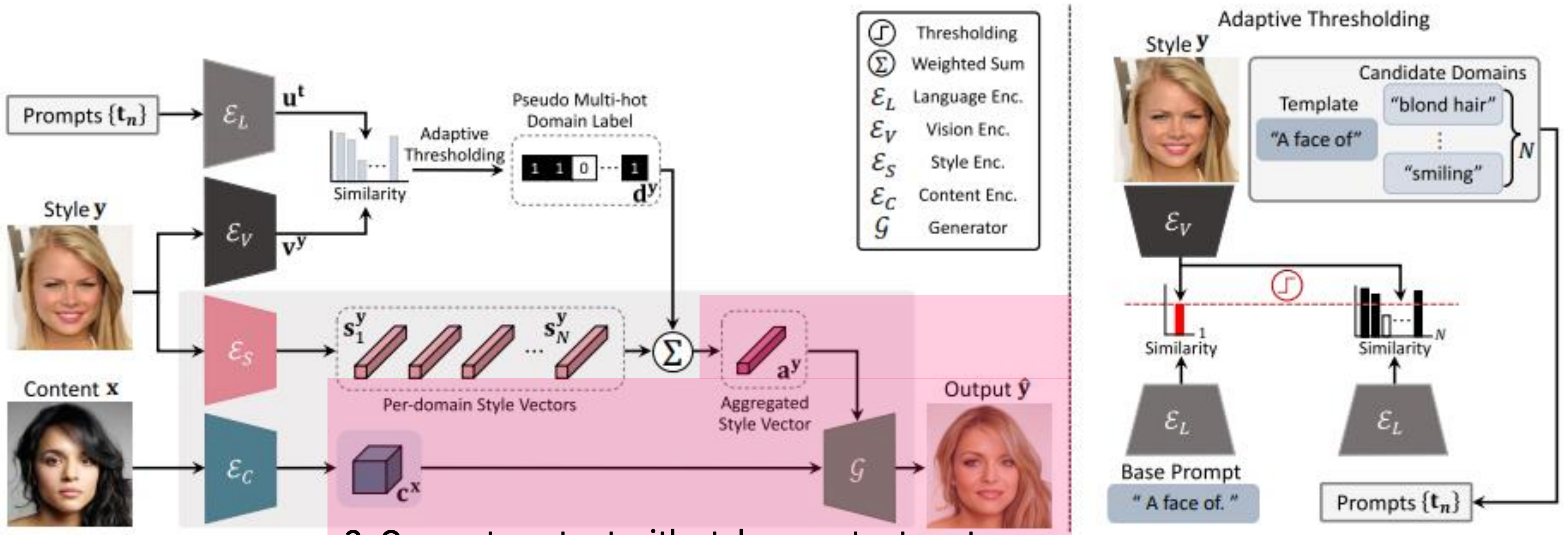


Figure 3. **Network configuration.** Our model consists of content and style encoders ($\mathcal{E}_C$, $\mathcal{E}_S$), vision-language encoders ($\mathcal{E}_V$, $\mathcal{E}_L$), and generator ($\mathcal{G}$). We extract content and style vectors from content $\mathbf{x}$ and style $\mathbf{y}$, respectively. By leveraging vision-language features and the proposed adaptive thresholding technique, we measure the pseudo domain label $\mathbf{d}^{\mathbf{y}}$ of $\mathbf{y}$. We generate $\hat{\mathbf{y}}$ with content vector $\mathbf{c}^{\mathbf{x}}$ and aggregated style vector $\mathbf{a}^{\mathbf{y}}$ through the generator.

3. Methodology

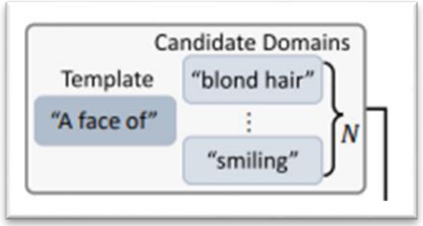


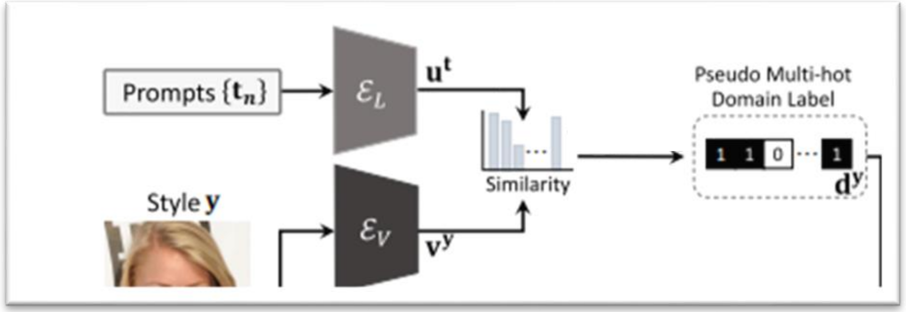
3. Generate output with style + content vector

Figure 3. **Network configuration.** Our model consists of content and style encoders ($\mathcal{E}_C$, $\mathcal{E}_S$), vision-language encoders ($\mathcal{E}_V$, $\mathcal{E}_L$), and generator ($\mathcal{G}$). We extract content and style vectors from content $\mathbf{x}$ and style $\mathbf{y}$, respectively. By leveraging vision-language features and the proposed adaptive thresholding technique, we measure the pseudo domain label $\mathbf{d}^\mathbf{y}$ of $\mathbf{y}$. We generate $\hat{\mathbf{y}}$ with content vector $\mathbf{c}^\mathbf{x}$ and aggregated style vector $\mathbf{a}^\mathbf{y}$ through the generator.

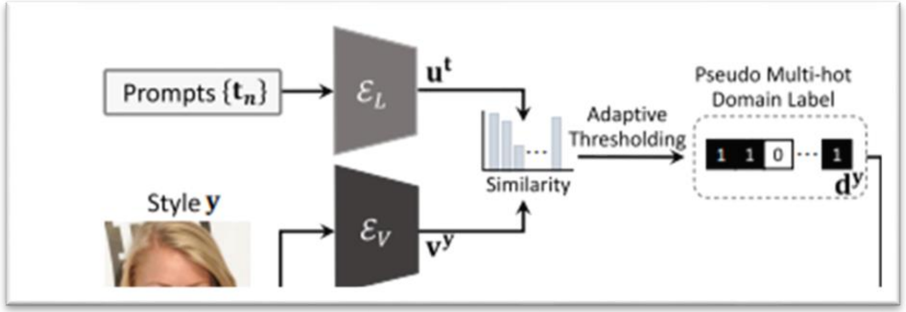
3. Methodology

- Per-sample annotation / One-hot encoding
- -> Dataset-level domain descriptions / multi-hot label setting
- 1. get pseudo multi-hot domain label





- IF only used CLIP(calculating similarity between image and text feature) to get domain label,
- -> Can cause inaccurate labeling / noisy translation



- Adaptive thresholding + prompt learning technique

3. Methodology

1. get pseudo multi-hot domain label

- 1) extract vision and language features using pre-trained vision-language models like CLIP

$$\mathbf{t}_n = [p_1, p_2, \ldots, p_L, p_n^{\mathrm{domain}}], \quad \Rightarrow \quad [\mathbf{u}_n^{\mathbf{t}}]_{n=1}^{N} = \mathcal{E}_L(\mathbf{T}) \in \mathbb{R}^{N \times k}$$

$$\Rightarrow \quad \mathbf{v}^{\mathbf{y}} = \mathcal{E}_V(\mathbf{y}) \in \mathbb{R}^{1 \times k}$$

- 2) measure a similarity using features

$$\mathbf{f}^{\mathbf{y}} = [f_n^{\mathbf{y},\mathbf{t}}]_{n=1}^{N} \in \mathbb{R}^{N \times 1} \qquad f_n^{\mathbf{y},\mathbf{t}} = \bar{\mathbf{v}}^{\mathbf{y}} \cdot \bar{\mathbf{u}}_n^{\mathbf{t}},$$

t = template + words = “a face with” + “lipstick”

p = template only = “a face with”

$$\mathbf{u}^{\mathbf{p}} = \mathcal{E}_L(\mathbf{p}) \in \mathbb{R}^{1 \times k}.$$

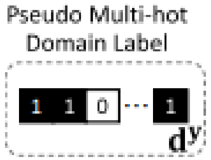
- 3) obtain multi-hot pseudo domain label

  - top-K (can only select ‘K’ numbers of attributes and ignores others)

  - simple thresholding (can limit the performance)

  - adaptive thresholding

$$d_n^{\mathbf{y}} = \begin{cases} 1, & \text{if, } f_n^{\mathbf{y},\mathbf{t}} > \bar{\mathbf{v}}^{\mathbf{y}} \cdot \bar{\mathbf{u}}^{\mathbf{p}}, \\ 0, & \text{otherwise.} \end{cases} \quad \Rightarrow$$



3. Methodology

2. get optimized prompt

t = template + words = “a face with” + “lipstick”

Template is given by human for the dataset

-> Template might not be optimal to describe all the images in the dataset

Domain regularization Loss

1) set prompt “tn" as learnable except “Pn domain”

$$\mathbf{t}_n = [p_1, p_2, ..., p_L, p_n^{\text{domain}}],$$

2) Make domain label pair, $\mathbf{d}^{\mathbf{y}}$ and $\mathbf{d}^{\mathbf{y}}_{\text{inv}}(n)$, dy-inv has same labels with dy except n-th label opposite. (so that y and y’ has opposite style for n-th domain)

3) generate $\hat{\mathbf{y}}' = \mathcal{G}(\mathbf{c}^{\hat{\mathbf{y}}}, \mathbf{a}^{\hat{\mathbf{y}}}_{\text{inv}})$ where ay-inv is obtained by dy-inv.

4) Calculate $\mathbf{f}^{\mathbf{y}} = [f_n^{\mathbf{y},\mathbf{t}}]_{n=1}^N \in \mathbb{R}^{N\times 1}$ for each

5) Minimize the loss with dy fy $\mathcal{L}_{\text{dl}} = \mathcal{H}(d_n^{\mathbf{y}}, f_n^{\hat{\mathbf{y}}}) + \mathcal{H}(d^{\mathbf{y}}_{\text{inv},n}(n), f^{\hat{\mathbf{y}}}_{\text{inv},n})$, = which learnable prompts primarily evoke similarity?

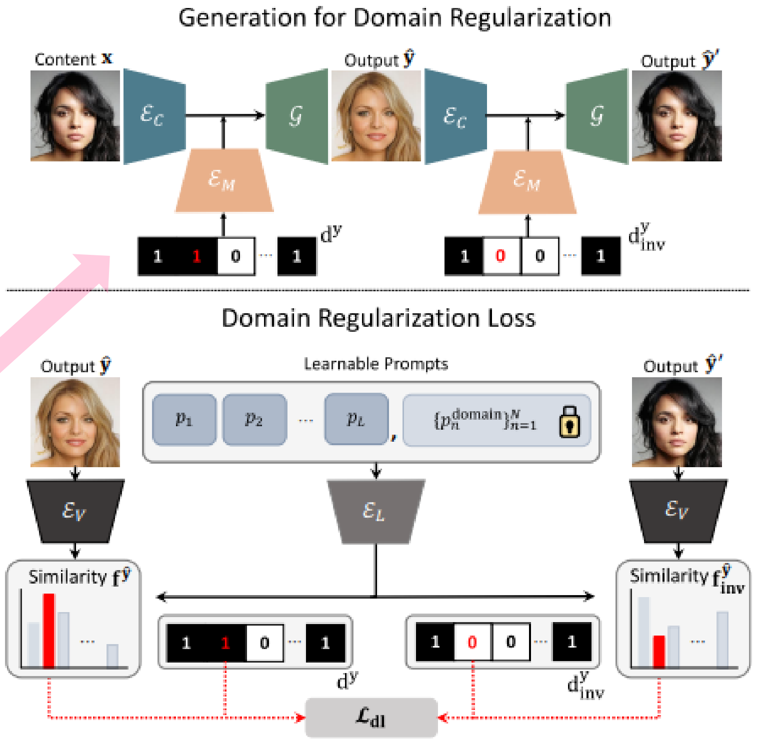


Figure 4. **Illustration of domain regularization loss.** We define a domain regularization loss $\mathcal{L}_{\text{dl}}$ utilizing two outputs $\hat{\mathbf{y}}$ and $\hat{\mathbf{y}}'$ that have the opposite styles at n-th domain, and minimize the loss function not to only learn the optimal prompt but also better learn the translation process.

3. Methodology

```
animal_imagenet_templates = [
    'a animalface photo with {}.',
    'a animalface photo of the {}.',
    'the animalface photo of the {}.',
    'a good animalface photo of the {}.',
    "high quality animal
    "a animalface image
    "the animalface ima
    "high quality anima
    "a high quality ani
]

animal_base_imagenet_te
    ['a animal photo wi
    ['a animal photo of
    ['the animal photo
    ['a good animal photo of the.'],
    ["high quality animal photo of."],
    ["a animal image of."],
    ["the animal image of."],
    ["high quality animal image of."],
    ["a high quality animal image of."],
]
```

```
food_imagenet_templates = [
    'a food photo with {}.',
    'a food photo of the {}.',
    'the food photo of the {}.',
    'a good food photo of the {}.',
    "high quality food photo of {}.",
    "a food image of {}.",
    "the food image of {}.",
    "high quality food image of {}.",
    "a high quality food image of {}.",
```

```
if 'animal' in args.dataset:
    init_prompt = 'a photo of the {}.'
    base_template = ["a photo of the animal face."]
    all_prompt = ['beagle', 'dandie dinmont terrier', 'golden retriever', 'malinois', 'appenzeller sennenhund', 'white fox',

    if args.num_domains == 4:
        prompt = ['beagle', 'golden retriever','tabby cat', 'bengal tiger']
    elif args.num_domains == 7:
        prompt = ['beagle', 'dandie dinmont terrier', 'golden retriever', 'white fox', 'tabby cat', 'snow leopard', 'bengal
    elif args.num_domains == 10:
        prompt =  ['beagle', 'dandie dinmont terrier', 'golden retriever', 'malinois',\
                  'appenzeller sennenhund', 'white fox', 'tabby cat', 'snow leopard', 'lion', 'bengal tiger']
    elif args.num_domains == 13:
        prompt =  ['beagle', 'dandie dinmont terrier', 'golden retriever', 'malinois',\
                  'appenzeller sennenhund', 'white fox', 'tabby cat', 'snow leopard', 'lion', 'bengal tiger',\
                  'french bulldog', 'mink', 'maned wolf']
    elif args.num_domains == 16:
        prompt =  ['beagle', 'dandie dinmont terrier', 'golden retriever', 'malinois',\
                  'appenzeller sennenhund', 'white fox', 'tabby cat', 'snow leopard', 'lion', 'bengal tiger',\
                  'french bulldog', 'mink', 'maned wolf', 'monkey', 'toy poodle', 'angora rabbit']
```

3. Methodology

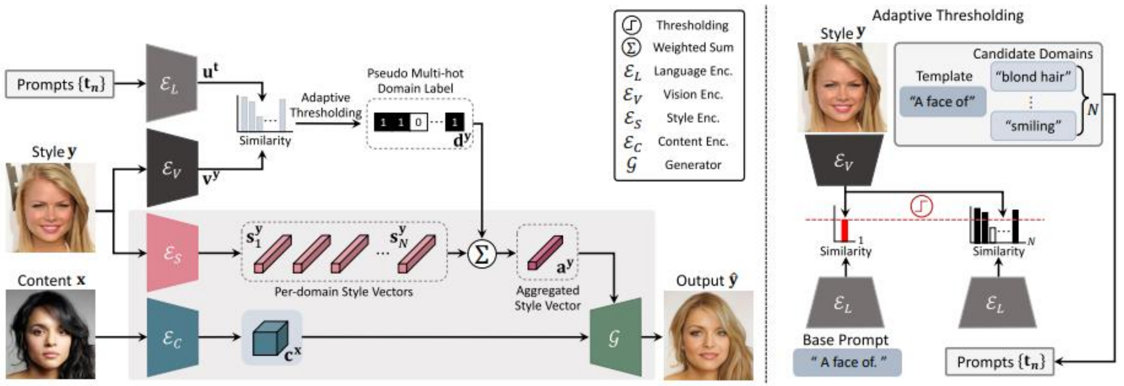


Figure 3. **Network configuration.** Our model consists of content and style encoders ($\mathcal{E}_C$, $\mathcal{E}_S$), vision-language encoders ($\mathcal{E}_V$, $\mathcal{E}_L$), and generator ($\mathcal{G}$). We extract content and style vectors from content $\mathbf{x}$ and style $\mathbf{y}$, respectively. By leveraging vision-language features and the proposed adaptive thresholding technique, we measure the pseudo domain label $\mathbf{d}^{\mathbf{y}}$ of $\mathbf{y}$. We generate $\hat{\mathbf{y}}$ with content vector $\mathbf{c}^{\mathbf{x}}$ and aggregated style vector $\mathbf{a}^{\mathbf{y}}$ through the generator.

using output y, Trained with

1) Adversarial loss – adopted multi-domain discriminators (same as StarGAN2)

(output of multi-domain discriminators weighted by multi-hot domain label dy)

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{\mathbf{x},\mathbf{y}} \sum_{n=1}^{N} [\log \mathcal{D}_n(\mathbf{y}) d_n^{\mathbf{y}} + \log(1 - \mathcal{D}_n(\mathcal{G}(\mathbf{x}, \mathbf{a}^{\mathbf{y}})) d_n^{\mathbf{y}})], \quad (5)$$

<< multi-hot domain label weighted

$$\mathcal{L}_{adv} = \mathbb{E}_x[\log D_{src}(x)] + \mathbb{E}_{x,c}[\log(1 - D_{src}(G(x,c)))], \quad (1)$$

<< StarGAN (one-hot encoding)

2) cycle-consistency loss

Generating input x with output y backward, Generator learns to preserve original

characteristics of x

$$\mathcal{L}_{\text{cyc}} = \mathbb{E}_{\mathbf{x},\mathbf{y}}[\|\mathbf{x} - \mathcal{G}((\hat{\mathbf{y}}), \mathbf{a}^{\mathbf{x}})\|_1], \quad (7)$$

3. Methodology

3) Style-reconstruction loss – l-1 loss between the style vector from

translated image and style image.

$$\mathcal{L}_{\text{sty}} = \mathbb{E}_{\mathbf{x},\mathbf{y}}[\|\mathbf{s}^{\mathbf{y}} - \mathcal{E}_S(\hat{\mathbf{y}})\|_1]. \quad (8)$$

4) Style-diversification loss –

(z1 & z2 random latent vectors from Gaussian distribution are used)

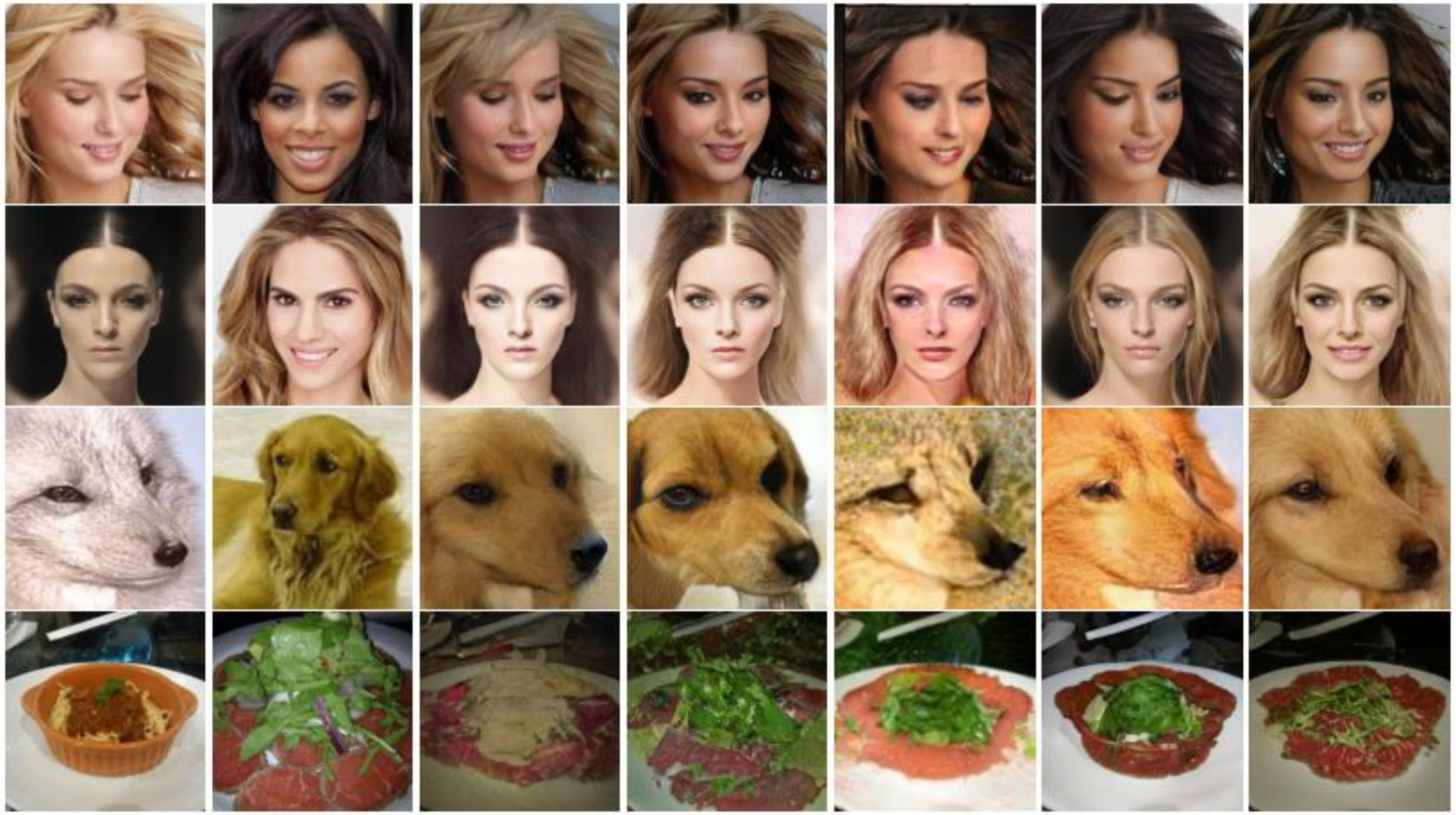
$$\mathcal{L}_{\text{ds}} = \mathbb{E}_{\mathbf{x},\mathbf{y}}[\|\mathcal{G}(\mathbf{x}, \mathcal{E}_M(\mathbf{z}_1)) - \mathcal{G}(\mathbf{x}, \mathcal{E}_M(\mathbf{z}_2))\|_1], \quad (9)$$

Overall Objective. Full loss functions are as follows:

$$\begin{aligned} \mathcal{L}_{\text{total}} =& \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \lambda_{\text{dl}}\mathcal{L}_{\text{dl}} + \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}} \\ &+ \lambda_{\text{sty}}\mathcal{L}_{\text{sty}} - \lambda_{\text{ds}}\mathcal{L}_{\text{ds}}, \end{aligned} \quad (10)$$

where λ_{adv}, λ_{dl}, λ_{cyc}, λ_{sty}, and λ_{ds} are hyper-parameters.

4. Results



(a) Content (b) Style (c) StarGANv2 [10] (d) Smoothing [40] (e) TUNIT [3] (f) Kim *et al.* [29] (g) Ours

???

Figure 5. **Qualitative comparison.**

4. Results

Method	CelebA-HQ [41]		AnimalFaces-10 [35]		Food-10 [7]	
	mFID ↓	D&C ↑	mFID ↓	D&C ↑	mFID ↓	D&C ↑
StarGAN2 [10] (sup.)	32.16	1.22 / **0.44**	**33.67**	**1.54** / **0.91**	65.03	1.09 / 0.76
Smoothing [40] (sup.)	35.93	**1.25** / 0.43	38.93	0.97 / 0.75	61.13	0.96 / 0.68
TUNIT [3] (unsup.)	61.29	0.24 / 0.13	47.70	1.04 / 0.81	52.20	1.08 / **0.88**
Kim *et al.* [29] (unsup.)	41.33	0.60 / 0.24	36.83	1.06 / 0.82	49.34	1.06 / 0.80
LANIT	**27.96**	0.91 / 0.34	34.11	1.46 / 0.89	**48.08**	**1.24** / 0.86

Table 1. **Quantitative comparison on CelebA-HQ [41], Animal Faces-10 [35] and Food-10 [7]** The configurations of StarGAN2 and Smoothing use ground-truth domain labels while TUNIT and Kim *et al.* use pseudo-labels generated from each image. Our LANIT uses only textual domain descriptions.

4. Results

N	Method	AnimalFaces-10 [35] mFID ↓	AnimalFaces-10 [35] D&C ↑	CelebA-HQ [41] mFID ↓	CelebA-HQ [41] D&C ↑
4	TUNIT	77.7	0.88 / 0.74	61.5	0.24 / 0.12
	LANIT	71.6	1.35 / 0.46	49.3	0.33 / 0.14
7	TUNIT	62.7	1.02 / 0.73	54.7	0.33 / 0.16
	LANIT	49.9	1.47 / 0.66	43.2	0.44 / 0.19
10	TUNIT	47.7	1.04 / 0.81	61.3	0.24 / 0.13
	LANIT	**34.1**	1.46 / **0.89**	**27.9**	**0.91** / **0.34**
13	TUNIT	56.8	0.99 /0.72	98.9	0.08 / 0.03
	LANIT	30.1	1.43 / 0.85	34.8	0.58 / 0.21
16	TUNIT	54.1	1.09 / 0.78	127.7	0.04 / 0.02
	LANIT	35.8	**1.49** / 0.82	**27.9**	0.76 / 0.23

Table 2. **Quantitative comparison of LANIT with TUNIT [3] by varying the number of domains.**

4. Results

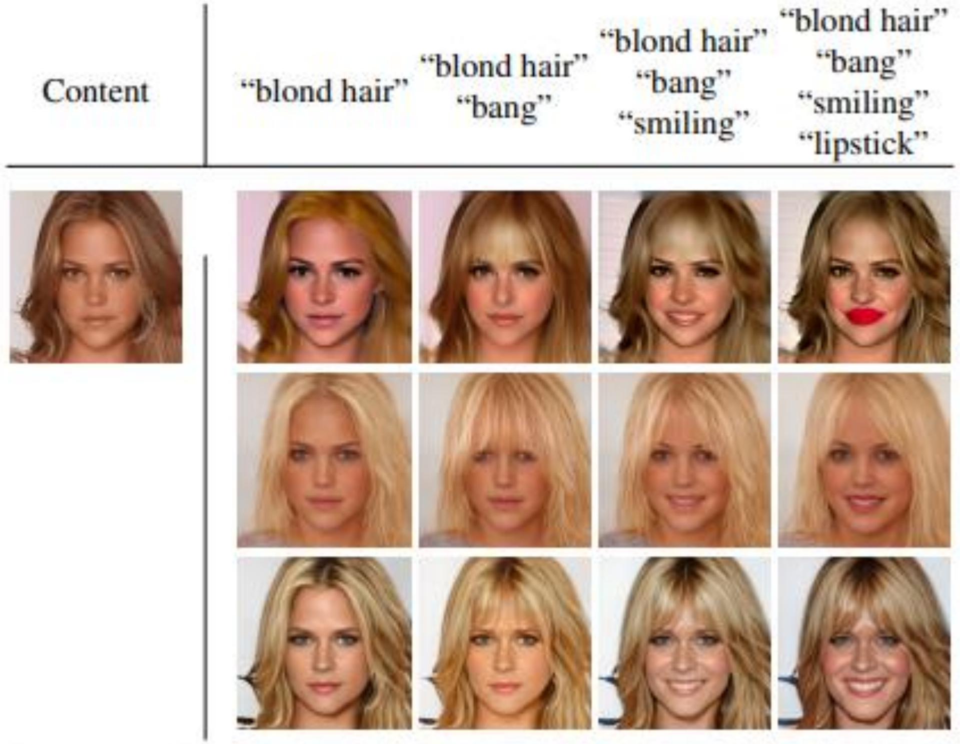


Figure 7. **Additional qualitative results by (from top to bottom) DiffusionCLIP [28], StyleCLIP [47], and our LANIT.**