Harnessing the Power of MLLMs for Transferable Text-to-Image Person ReID

Supplementary Material

This supplementary material comprises three sections. Section A provides a detailed explanation of how we obtain static and dynamic instructions. In our conversation with ChatGPT [4], the given raw instructions have their respective limitations. Based on experimental results, we make appropriate modifications. Section B showcases how to compute similarity between $\mathbf{F}_t$ and $\mathbf{F}_v$. We further calculate the noise level of T^{full} based on the similarity and illustrate the implementation of Noise-Aware Masking (NAM). We also provide visualization results of NAM, confirming its efficacy in identifying and masking noisy words. In Section C, we establish new validation sets for the downstream tasks to comply with the principle of transferable ReID. We conduct experiments on the NAM layer and p again and draw consistent conclusions with those in the main text.

A. Dialogues with ChatGPT

Static Instruction. We utilize MLLMs [1, 2] for image captioning. Designing effective instructions to fully leverage the image captioning capabilities of MLLMs to generate accurate and comprehensive descriptions is challenging. To address this, we turn to ChatGPT [4] for assistance. As shown in Fig. A, we send our requirements to ChatGPT through multiple rounds of conversation, and in response, ChatGPT provides six instructions. After receiving these instructions, we test them on the Qwen [1] and observe variations in the quality of the generated descriptions, as shown in Fig. B. Some instructions generate noisy descriptions, *e.g.*, *Instruction* 1, 2, and 5, while others lead to redundant information (*Instruction* 3, 4, and 6).

After analyzing the situation, we identify the reasons behind these issues. Errors in *Instruction* 1, 2, and 5 occur because the specified aspects of the description ("accessories", "from head to toe", "from the headwear to the footwear") are absent in the images, leading the MLLMs to imagine invisible contents. The redundancy issue arises from instructions that demand detailed descriptions ("distinctive", "details", "thorough").

Based on these insights, we make appropriate modifications: (1) The MLLMs strictly follows instructions. To ensure comprehensive descriptions that encompass all pedestrian features, we proactively incorporate attributes into the instructions, requiring the MLLMs to provide detailed descriptions. (2) However, the MLLMs may generate contents that are not present in the images. Motivated by [5], we add the following requirement to the instructions: *"Do not imagine any contents that are not in the image"*. (3) To avoid redundancy, we no longer explicitly demand "detailed" descriptions from the MLLMs.

Through these modifications and observations of experimental results, we ultimately establish the *static instruction*:

"Write a description about the overall appearance of the person in the image, including the attributes: clothing, shoes, hairstyle, gender and belongings. If any attribute is not visible, you can ignore it. Do not imagine any contents that are not in the image."

It is important to note that *static instruction* may not be the optimal choice, but it is effective. We also discover that *static instruction* leads to similar sentence structures for different images, which hinders the learning of transferable models. As a solution, we propose the Template-based Diversity Enhancement method (TDE) to improve description diversity.

Dynamic Instruction. Template-based Diversity Enhancement aims to enhance the diversity of captions by utilizing templates. It involves generating a large number of different templates. To reduce the manual effort, we delegate this task to ChatGPT. As depicted in Fig. D, we input multiple sentences generated from *static instruction* into ChatGPT and request it to extract sentence patterns. Subsequently, we encourage ChatGPT to generate additional templates on its own. During this process, the quality of templates generated by ChatGPT may not be high, so we provide appropriate requirements to assist ChatGPT in making modifications. Finally, we manually filter the generated templates, resulting in a total of 46 descriptive templates. Similarly, the combination of templates and *static instruction* is achieved like Fig. A and Fig. B. We finally create the *dynamic instuction*:

"Generate a description about the overall appearance of the person, including clothing, shoes, hairstyle, gender, and belongings, in a style similar to the template: '{template}'. If some requirements in the template are not visible, you can ignore them. Do not imagine any contents that are not in the image."

B. Details in Noise-Aware Masking

Pseudo-code for Noise-Aware Masking. NAM gives different masking probabilities according to the noise level of the token, the higher the noise level, the higher the masking probability. How to calculate the noise level of the tokens in T^{full} is shown in Algorithm A. After calculating the noise level for each tokens, we will save them for use in the next epoch. Following BERT [3], our masking operation consists of 80% probability of replacing the token with [MASK], 10% probability of replacing the token with some

Do you know the Instruction Model?

Yes, “Instruction Model” may refer to large language models that process instructions in a particular context. Such models are usually designed for processing instructions, commands, or tasks in natural languages.

Now I want to call these Instruction Models to describe the clothing appearance features of pedestrians in my input image. Please give some examples of instructions.

You can use some of the following examples when you want the Instruction Models to describe the clothing appearance characteristics of a pedestrian in an input image:
“Describe what colour blouse this person is wearing?”,
“Please tell me what colour his trousers are?”,
...
“Please describe his clothing style, is it casual or formal?”

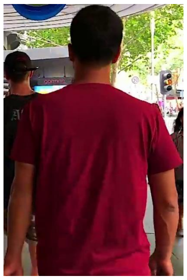
I’d like a full body description of my clothing and appearance.

When you want a full-body description of your clothing appearance, you can use the following command example:
“Describe the person's overall appearance, including the clothing, hairstyle, and any accessories.”,
“Provide a detailed description of the person’s outfit, from head to toe.”,
“Tell me about the person’s attire, including any distinctive features or accessories.”,
“Give me a comprehensive description of the person’s clothing, including colors, styles, and any notable details.”,
“Describe the individual’s entire ensemble, from the headwear to the footwear.”,
“Provide a thorough account of the person's appearance, including the garments, hair, and any adornments.”

Figure A. We create a context to consult the ChatGPT for instructions that can guide MLLMs to generate high-quality pedestrian descriptions.

Instructions:
(1) Describe the person’s overall appearance, including the clothing, hairstyle, and any accessories.
(2) Provide a detailed description of the person’s outfit, from head to toe.
(3) Tell me about the person’s attire, including any distinctive features or accessories.
(4) Give me a comprehensive description of the person’s clothing, including colors, styles, and any notable details.
(5) Describe the individual’s entire ensemble, from the headwear to the footwear.
(6) Provide a thorough account of the person’s appearance, including the garments, hair, and any adornments.

Image:



Descriptions:
(1) The person in the image is a man wearing a red shirt and a hat. He has short hair and is walking down the street. (Wrong)
(2) The person in the image is wearing a red shirt and shorts. (Wrong)
(3) The person is wearing a red shirt, which is a distinctive feature in the image. (Useless information)
(4) The person in the image is wearing a red shirt which is the most prominent piece of clothing in the scene. The rest of the person’s outfit is not fully visible, but it canbe inferred that they are likely dressed in other colors as well, such as black, white, or various shades of red. The person’s clothing style appears to be casual and comfortable, suitable for walking around a city street or engaging in everyday activities. (Useless information)
(5) The man is wearing a red shirt and black shoes. (Wrong)
(6) The person in the image is a man wearing a red shirt, and he is walking down the street. He has short hair, which is the most prominent feature of his appearance. There are no visible adornments or accessorieson the man, such as a hat or a tie. The focus of the image is primarily on his red shirt and his posture as he walks. (Useless information & repetition)

Figure B. We test the instructions provided by ChatGPT and analyze their shortcomings to make improvements.

Algorithm A: Pseudo-code for Noise-Level Estimation.

```
# fi_tokens: image tokens embeddings of shape (193,d)
# ft_tokens: full text tokens embeddings of shape (77,
    d)
# tokens_full: full text tokens array, length: 77
# p: average masking ratio (defalut:0.15)

fi_tokens = fi_tokens[1:,:] # patch tokens

# normalize the features
fi_tokens = F.normalize(fi_tokens, dim=1)
ft_tokens = F.normalize(ft_tokens, dim=1)

# calculate the similarity matrix S (Eq.1)
sim_matrix = torch.mm(ft_tokens, fi_tokens.t()) #
     shape (77,192)

# raw noise-level vector r (Eq.2)
sim_max = sim_matrix.max(-1)[0]
raw_r = 1 - sim_max.data.cpu().numpy()

# modify the expectation of r (Eq.3 & Eq.4)
if tokens_full[-1] == 0:
   valid_token_num = np.where(tokens_full == 0)[0][0]
else:
   valid_token_num = len(tokens_full)
# do not include [SOS], [EOS]
valid_noise = raw_r[1:valid_token_num-1]
mu = np.mean(raw_noise)
noise_ba = valid_noise - mu + p

r_ba = copy.deepcopy(raw_r)
r_ba[1:valid_token_num-1] = noise_ba

return r_ba
```

other token, and 10% probability of not doing any replacement. The specific NAM operation is shown in Algorithm B.

Visualization. NAM automatically identifies erroneous words in sentences and masks them with a higher probability. During the training process, we randomly sample some text-image pairs and decode the masked tokens back into sentences, comparing them with the original sentences to observe whether NAM can successfully locate the incorrect words. The results, as shown in Fig. C, demonstrate that NAM is able to correctly identify and mask the erroneous words in the majority of cases. It reflects the effective ability of NAM to combat the impact of noise.

C. More Experiments

Searching for optimal hyper-parameters using new validation splits. In Fig. 3 and 4, we conduct hyper-parameter selection experiments using the test splits of the downstream dataset, which contravenes the principle in transferable ReID that models are directly evaluated on downstream datasets with no access to the test splits. An alternative method is to use the validation splits for experimentation. However, most ReID datasets lack validation splits. Consequently, we decide to randomly sample 10% of IDs from the train splits to create three new validation sets. We conduct the hyper-parameter selection experiments on these newly established validation sets.

Algorithm B: Pseudo-code for Noise-Aware Masking.

```
# tokenizer: the tokenizer from CLIP
# tokens_full: full text tokens array, length: 77
# r_ba: probabillity array to apply NAM

# all token ids except [SOS], [MASK], [EOS]
token_range = list(range(1, len(tokenizer.encoder)-3))

# [MASK]
mask = tokenizer.encoder["<|mask|>"]

tokens_nam = copy.deepcopy(tokens_full)
for i, token in enumerate(tokens_nam):
   # skip [SOS], [EOS]
   if 0 < token < 49405:
      prob = random.random()

      # mask token with varied probability
      if prob < r_ba[i]:
         prob /= r_ba[i]

         # 80% randomly change token to mask token
         if prob < 0.8:
            tokens_nam[i] = mask

         # 10% randomly change token to random token
         elif prob < 0.9:
            tokens_nam[i] = random.choice(token_range)

         # -> rest 10% randomly keep current token

return torch.tensor(tokens_nam)
```

T^{full}: The man has brown hair and is dressed in a green shirt, brown pants, black belt, and black shoes. He is carrying a black bag.

T^{nam}: the man has brown hair units is dressed in a green [mask], brown pants [mask] black belt [mask] and [mask] [mask]. he is carrying [mask] black [mask].

T^{full}: The person in the image is a young adult with long hair, wearing a gray hoodie, dark pants, and a gray and white beanie.

T^{nam}: the person in the image is a [mask] adult with [mask] hair, wearing [mask] gray hoodie [mask] dark pants, run a gray and white beanie.

T^{full}: The person in the image is a woman with auburn hair, wearing a white jacket and a fanny pack. She is walking down the street and appears to be in a hurry.

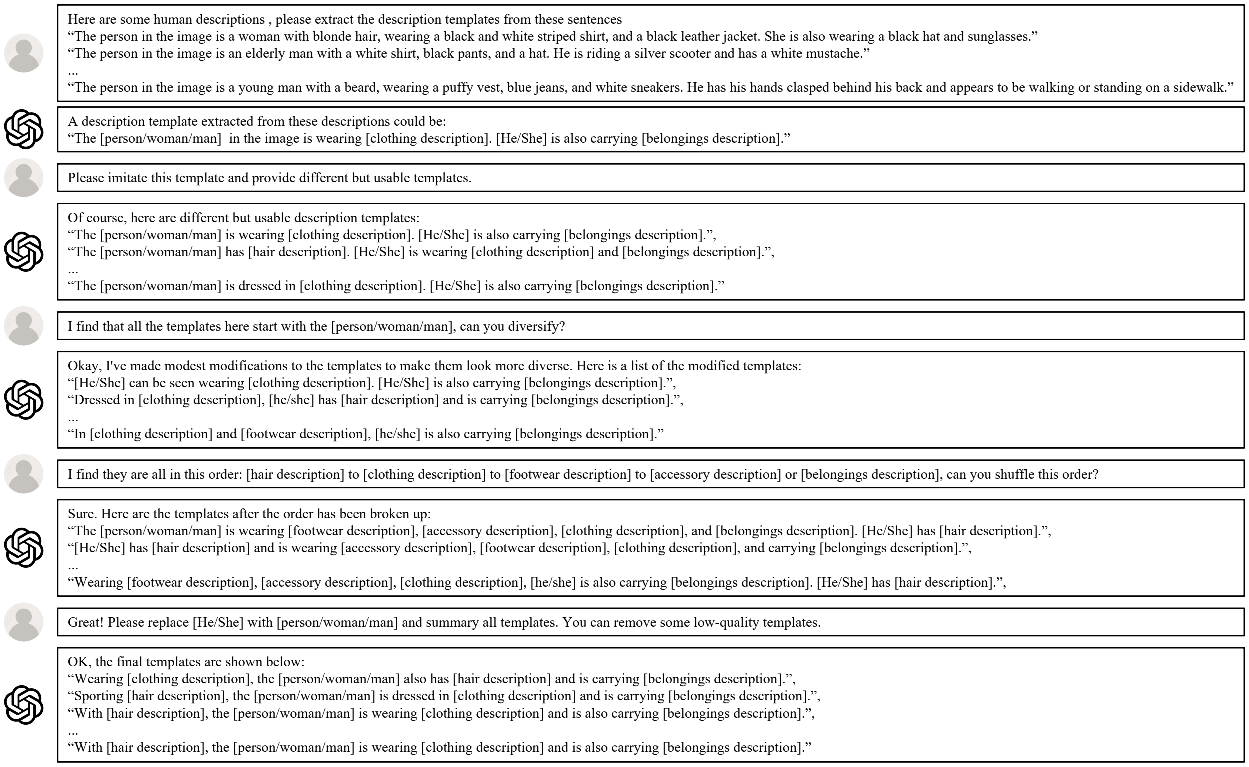
T^{nam}: the person in the [mask] is a woman with [mask] hair, wearing a white jacket and [mask] fanny pack. [mask] is walking down the street and appears to be in a hurry.

Figure C. In most cases, NAM is able to accurately identify erroneous tokens and mask them, effectively mitigating the impact of noise.

Table A. Searching for optimal hyper-parameters using new validation split. The best NAM layer and ρ are matching the main paper.

ρ=0.15	layer=8	layer=9	layer=10	layer=11	layer=12
average R1/mAP	46.13/34.88	46.15/35.01	46.60/35.41	46.06/34.66	45.32/34.14
layer=10	ρ=0.10	ρ=0.15	ρ=0.20	ρ=0.25	ρ=0.30
average R1/mAP	46.07/34.75	46.60/35.41	46.53/35.33	45.85/34.96	45.88/34.88

Specifically, we train ReID models using 0.1 million images from our LUPerson-MLLM dataset with a variety of hyper-parameters. We then directly evaluate and record the average performance on the three newly constructed validation sets. The final results are shown in Table A. This sug-



Here are some human descriptions , please extract the description templates from these sentences
“The person in the image is a woman with blonde hair, wearing a black and white striped shirt, and a black leather jacket. She is also wearing a black hat and sunglasses.”
“The person in the image is an elderly man with a white shirt, black pants, and a hat. He is riding a silver scooter and has a white mustache.”
...
“The person in the image is a young man with a beard, wearing a puffy vest, blue jeans, and white sneakers. He has his hands clasped behind his back and appears to be walking or standing on a sidewalk.”

A description template extracted from these descriptions could be:
“The [person/woman/man] in the image is wearing [clothing description]. [He/She] is also carrying [belongings description].”

Please imitate this template and provide different but usable templates.

Of course, here are different but usable description templates:
“The [person/woman/man] is wearing [clothing description]. [He/She] is also carrying [belongings description].”,
“The [person/woman/man] has [hair description]. [He/She] is wearing [clothing description] and [belongings description].”,
...
“The [person/woman/man] is dressed in [clothing description]. [He/She] is also carrying [belongings description].”

I find that all the templates here start with the [person/woman/man], can you diversify?

Okay, I've made modest modifications to the templates to make them look more diverse. Here is a list of the modified templates:
“[He/She] can be seen wearing [clothing description]. [He/She] is also carrying [belongings description].”,
“Dressed in [clothing description], [he/she] has [hair description] and is carrying [belongings description].”,
...
“In [clothing description] and [footwear description], [he/she] is also carrying [belongings description].”

I find they are all in this order: [hair description] to [clothing description] to [footwear description] to [accessory description] or [belongings description], can you shuffle this order?

Sure. Here are the templates after the order has been broken up:
“The [person/woman/man] is wearing [footwear description], [accessory description], [clothing description], and [belongings description]. [He/She] has [hair description].”,
“[He/She] has [hair description] and is wearing [accessory description], [footwear description], [clothing description], and carrying [belongings description].”,
...
“Wearing [footwear description], [accessory description], [clothing description], [he/she] is also carrying [belongings description]. [He/She] has [hair description].”,

Great! Please replace [He/She] with [person/woman/man] and summary all templates. You can remove some low-quality templates.

OK, the final templates are shown below:
“Wearing [clothing description], the [person/woman/man] also has [hair description] and is carrying [belongings description].”,
“Sporting [hair description], the [person/woman/man] is dressed in [clothing description] and is carrying [belongings description].”,
“With [hair description], the [person/woman/man] is wearing [clothing description] and is also carrying [belongings description].”,
...
“With [hair description], the [person/woman/man] is wearing [clothing description] and is also carrying [belongings description].”

Figure D. Through multiple rounds of dialogue and iterative optimization, we eventually obtain diverse and high-quality templates.

gests that the optimal NAM layer and ρ remain at 10 and 0.15, respectively, aligning with the findings presented in the main paper. Additionally, it is evident that our methods are robust to hyper-parameter changes.

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 1

[2] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 1

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[4] OpenAI. Chatgpt. `https://openai.com/blog/chatgpt/`, 2022. 1

[5] Deyao Zhu, Jun Chen, Kilichbek Haydarov, Xiaoqian Shen, Wenxuan Zhang, and Mohamed Elhoseiny. Chatgpt asks, blip-2 answers: Automatic questioning towards enriched visual descriptions. *arXiv preprint arXiv:2303.06594*, 2023. 1