DLSR-IQA: Image Quality Assessment Model Based on Deep Learning and StairReward Matric

Shanghai Jiao Tong University, *Qi Siyuan, 521030910012*

***Abstract*—This document proposes a Image Quality Assess(IQA) model: DLSR-IQA to assess both perception and text-image alignment of AI Generated Content(AIGC). For both of these, I use deep learning and StairReward [1]methods respectively. And these two models perform excellent on test datasets by calculating Spearman rank-order correlation coefficient(SRCC) and Pearson linear correlation coefficient(PLCC).**

***Index Terms*—IQA, AIGC, perception, text-image alignment, deep learning, StairReward, SRCC, PLCC.**

I. Introduction

NOWADAYS more and more text-to-image AI models are emerging, and evaluating the quality of the images they generate has become a critical issue. Subjective evaluation methods are accurate but time-consuming and costly, making the design of **IQA models** crucial.

Based on existing datasets and subjective evaluation scores for perception and text-image alignment within the dataset, we can separately employ **deep learning** and **StairReward** method to obtain perception assessment and text-image alignment evaluation models.

II. Data preprocessing

A. Extract Image Features

ResNet152V2 [2] is an improved version of the ResNet (Residual Network) family, which belongs to deep **convolutional neural network (CNN)**. This model is designed to solve the "gradient vanishing" and "gradient explosion" problems that are common in deep networks, making it can significantly improve the performance of image processing tasks. Therefore, this network is appropriate for our IQA model to extract image features.

In addition, I performed **image data augmentation** [3], that is, applying a series of random transformations (such as rotation, translation, flip, etc.) to the original image to generate a new, slightly different image to improve the generalization ability of the model.

B. Extract Text Features

To make use of prompt, we need to extract its text features. We can use the Tokenizer to build a vocabulary based on the words that appear in the text and map each word to a unique integer index. Next, the Tokenizer converts each text (a series of words) into a series of integers, which are indexes of the corresponding words in the vocabulary. This transformation transforms the text into a digital format that the model can understand.

C. Normalize MOS Scores

Originally I trained my model only on dataset AGIQA-3K, and I found that if using STD scores, then the model perform badly. One possible reason is that STD data may lose some useful information because they scale all the features to a similar range. Therefore, the MOS score may contain features that make more sense to the model and I trained using MOS scores.

Therefore, when I want to expand data in dataset AIGCIQA2023, I need to standardize its scores into the same scale with those in AGIQA-3K. Since the MOS scores are subjective and the two groups of scores differ only in their full score, it is sufficient to use linear transformations:

$$\text{scaled\_score} = \frac{(b-a)(\text{AIGCIQA2023\_score} - c)}{d-c} + a \quad (1)$$

where $[a, b]$ and $[c, d]$ are MOS scores range of AGIQA-3K and AIGCIQA2023 respectively.

III. IQA Model on perception

A. Brief Introduction of Model

Since perception is only related to image itself, not prompt, then in this model we don't need to consider prompts information.

To begin with, I train a deep learning model on all data from both AGIQA-3K and AIQCIQA2023. Here gives its design detail:

(1) **Inputs**: image features vector;
(2) **Outputs**: MOS quality score;
(3) **Network Structure**: 2 hidden layers with 128 and 64 neurons respectively which uses **relu** as activation function;
(4) **Optimizer**: Adam optimizer with learning rate $\alpha = 0.001$;
(5) **Loss Function**: Mean Squared Error(MSE):

$$L = \frac{1}{n}\sum_i (y_i - \hat{y_i})^2 \quad (2)$$

However, the model behaves just so so. Then I do some improvement and parameters fine-tuning.

This paper was produced by Qi Siyuan, 521030910012, Shanghai Jiao Tong University, majoring in CS(IEEE)
Finished in Novermber, 2023.

B. Improvement and Fine-tuning

About improvement,I tried as followed:

(1) **Dropout**: prevent overfitting;
(2) **Logistic Mapping Function**: inspired by the paper of AGIQA-3K, I add that five-parameter funtion after the second hidden layer in the neural network and train its parameters. The function is given as followed:

$$\hat{y} = \alpha_1(0.5 - \frac{1}{1+e^{\alpha_2(y-\alpha_3)}}) + \alpha_4 y + \alpha_5 \quad (3)$$

(3) **Different Activation**: tried **sigmoid**, **tanh** and so on;
(4) **Different Optimizer**: tried **SGD** and **RMSprop**;
(5) **Different Loss Function**: tried **MAE** and **Huber Loss**.

Next, I tried the following fine-tuning:

(1) Number of neurons in each hidder layer;
(2) Dropout percentage;
(3) Learning rate of optimizer.

By comparing the value of SRCC and PLCC, I finally determined the model architecture and parameters as follows:

TABLE I: Perception Model Architecture and Parameters

Neuron Number	Dropout Percentage	Learning Rate
(128,64)	20%	0.0015
Optimizer	Activation and Loss	Mapping Layer Parameters
Adam	sigmoid, MSE	[1.48 -0.34 0.81 -0.18 0.31]

And next table shows performance before and after improving:

TABLE II: Performance Before and After Improving

\	Before Improving		After Improving	
k_fold	SRCC	PLCC	SRCC	PLCC
$k=1$	0.6892	0.7040	0.7332	0.7644
$k=2$	0.6771	0.7075	0.7231	0.7636
$k=3$	0.6733	0.6988	0.7227	0.7624
AVERAGE	0.6799	0.7034	0.7263	0.7635
VAR	6.89e-5	1.92e-5	3.54e-5	1.01e-6

And the improvement percentage is shown as followed:

TABLE III: Improvement Percentage

	SRCC	PLCC
AVERAGE	6.83%	8.53%
VAR	48.65%	94.71%

C. Result Analysis

We can see that after improving, average SRCC and PLCC both **increase**. Also, In different fold corss-validation, the data variance is very small, implied the results of the model performance are very similar. It shows that the model has a strong **adaptability to different data distributions**.

D. Further Exploration

Imspired by the AGIQA-3K paper, I want to test my model on different subsets of original dataset. Howerer, it is not appropriate to use the model described above directly for testing, since I train on the whole dataset. Therefore, the subset we split must include data that are used for training, causing **overfitting**.

Then, I thought two methods. One is I still train one model once again, but in this case I need to split train and test sets carefully. That is, make sure in train and test sets, **different characters make up the same proportion of the total set**. For example, if in the train set data with Baroque style accounts for 10%, then in the test set that also needs to account for 10%. For only one character it is easy to achieve, but to make all characters satisfy that requirement is hard since one image has many characters and **they are bound together**.

Another method is to **train different models on the specific subset**. Then each model should have a good predictive power for the dataset with this particular feature. For each direction, I process data of two datasets as followed:

(1) **Quality of AI Model**: In AGIQA-3K, they divide T2I AGI models into three groups: bad model (**AttnGAN** [4], **GLIDE** [5]), medium model (**DALLE2** [6], **Stable Diffusion** [7]), and good model (**Midjourney** [8], **Stable Diffusion XL**). Howerer, in AIGCIQA2023, the T2I models are not the same, so I only make use of models that are mentioned in AGIQA-3K(**GLIDE, DALLE2 and Stable Diffusion**) for training;
(2) **Length of Prompt**: In AGIQA-3K, prompts length is divided into 0, 1, 2, 3 according to **number of punctuation marks**. However, AIGCIQA2023 doesn't consider this point. So I only use AGIQA-3K dataset for training;
(3) **Style in Prompt**: In AGIQA-3K, they divide 4 styles: **Abstract & Scifi**, **Anime & Realistic**, **Baroque** and **No** style. But in AIGCIQA2023, most prompts don't include style, except 051: a portrait of a man wearing sunglasses and a business suit, painting in **pop art style**. I put this data into **Baroque** style and others into **No** style.

The experiment results are as followed(From top to bottom, k=1, k=2, k=3, Average and Var):

TABLE IV: Performance on Different Subsets for Perception

Performance for Different T2I Groups

All		bad models		medium models		good models	
SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
0.7332	0.7644	0.4386	0.4871	0.4753	0.5650	0.7113	0.7751
0.7231	0.7636	0.5162	0.5229	0.4705	0.5975	0.6652	0.6941
0.7227	0.7624	0.5044	0.5723	0.4250	0.5497	0.6464	0.7142
0.7263	0.7635	0.4864	0.5274	0.4569	0.5707	0.6743	0.7278
3.54e-5	1.01e-6	1.75e-3	1.83e-3	7.71e-4	5.96e-4	1.12e-3	1.78e-3

Performance for Different Prompt Length Groups

prompt0		prompt1		prompt2		prompt3	
SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
0.6428	0.7142	0.6644	0.7207	0.7363	0.7859	0.6452	0.7016
0.6782	0.7248	0.7001	0.7741	0.6519	0.7483	0.7119	0.7797
0.6812	0.7365	0.6721	0.7442	0.6612	0.7386	0.5546	0.6961
0.6674	0.7252	0.6789	0.7463	0.6831	0.7576	0.6372	0.7258
4.56e-4	1.24e-4	3.53e-4	7.16e-4	2.14e-3	6.24e-4	6.23e-3	2.19e-3

Performance for Different Style Groups

Abstract & Sci-fi		Anime & Realistic		Baroque		No Style	
SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
0.7053	0.7574	0.6429	0.7209	0.6067	0.6713	0.6941	0.7215
0.7140	0.7645	0.6760	0.7571	0.4321	0.6231	0.7153	0.7379
0.7064	0.7580	0.7231	0.8088	0.6880	0.7753	0.7251	0.7444
0.7086	0.7600	0.6807	0.7623	0.5756	0.6899	0.7115	0.7346
2.24e-5	1.55e-5	1.62e-3	1.95e-3	1.71e-2	6.05e-3	2.51e-4	1.39e-4

Similar to the AGIQA-3K paper, we can find that on the specific subset, the performance of my IQA model doesn't get improvement in most cases, even behaves worse. Only in a little case the performance gets some improvement(such as Anime & Realistic Style, in one fold the PLCC reaches to **0.8088**). There may be 2 main reasons:

(1) The model doesn't get fine-tuning. I use model on all data and don't tune parameters, therefore it may behave not so well;

(2) The number of data in each subset is not enough. Deep learning relies on large amounts of data. For each subset, training data has been reduced, causing the model behaves worse.

IV. IQA MODEL ON TEXT-IMAGE ALIGNMENT

A. Overall Model on Total Dataset

For text-image alignment, I followed the approach of perception model. The difference is that we need to additionally enter the prompt feature of the image. Besides, I also enter T2I model feature as an attempt. This should be careful since we need to align model name(for example, in AGIQA-3K, there exists "glide" and "sd1.5", but in AIGCIQA2023, they are called "Glide" and "stable-diffusion" respectively).

In addition, model parameters should be readjusted. Next table gives the architecture and parameters:

TABLE V: Alignment Model Architecture and Parameters

Neuron Number	Dropout Percentage	Learning Rate
(128,64)	20%	0.0005
Optimizer	Activation and Loss	Mapping Layer Parameters
Adam	relu/sigmoid, MSE	[1.33 0.25 -0.59 0.16 0.25]

And next table gives the performance of my model:

TABLE VI: Performance of Alignment Assessment

k_fold	SRCC	PLCC
$k=1$	0.6860	0.7023
$k=2$	0.6859	0.7129
$k=3$	0.7023	0.7249
AVERAGE	0.6914	0.7134
VAR	8.91e-5	1.28e-4

It can be found that the performance of this model is not so good as the perception model, but its performance is stable. That is because this task is harder(prompt feature also needed as inputs).

B. Performance on Different Subsets

Similar to perception model, I also did some parallel experiments on different subsets. I use the same method about perception model to split subsets.

The experiment results are shown in Table.VII(From top to bottom, k=1, k=2, k=3, Average and Var).

In most cases, performance on specific subsets is also not so good as model on total dataset. The reason is similar, one is number of training data and the other is not getting well tuned.

TABLE VII: Performance on Different Subsets for Alignment

Performance for Different T2I Groups

All		bad models		medium models		good models	
SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
0.686	0.7023	0.5479	0.5045	0.4781	0.5555	0.4173	0.4061
0.6859	0.7129	0.5234	0.5396	0.5473	0.5656	0.4104	0.3818
0.7023	0.7249	0.5357	0.561	0.4818	0.5246	0.4352	0.3674
0.6914	0.7134	0.5357	0.5350	0.5024	0.5486	0.4210	0.3851
8.91e-5	1.28e-4	1.50e-4	8.14e-4	1.52e-3	4.56e-4	1.64e-4	3.83e-4

Performance for Different Prompt Length Groups

prompt0		prompt1		prompt2		prompt3	
SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
0.6428	0.7142	0.6644	0.7207	0.7363	0.7859	0.6452	0.7016
0.6782	0.7248	0.7001	0.7741	0.6519	0.7483	0.7119	0.7797
0.6812	0.7365	0.6721	0.7442	0.6612	0.7386	0.5546	0.6961
0.6674	0.7252	0.6789	0.7463	0.6831	0.7576	0.6372	0.7258
4.56e-4	1.24e-4	3.53e-4	7.16e-4	2.14e-3	6.24e-4	6.23e-3	2.19e-3

Performance for Different Style Groups

Abstract & Sci-fi		Anime & Realistic		Baroque		No Style	
SRCC	PLCC	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
0.7053	0.7574	0.6429	0.7209	0.6067	0.6713	0.6941	0.7215
0.7140	0.7645	0.6760	0.7571	0.4321	0.6231	0.7153	0.7379
0.7064	0.7580	0.7231	0.8088	0.6880	0.7753	0.7251	0.7444
0.7086	0.7600	0.6807	0.7623	0.5756	0.6899	0.7115	0.7346
2.24e-5	1.55e-5	1.62e-3	1.95e-3	1.71e-2	6.05e-3	2.51e-4	1.39e-4

C. Further exploration

Inspired by the **StairReward** metric proposed in AGIQA-3K, I also tried this metric. There are 3 key equations:

$$(p_1, p_2 \cdots p_K) = \text{Split}(p_0) \tag{4}$$

where p_0 represents the original prompt, K represents the number of morphemes and Split$(\cdot)$ is a prompt segmentation function based on prepositions and punctuation;

$$I_K = \text{Box}_{L=\frac{1}{2}+\frac{k-1}{2(K-1)}}(I_0) \tag{5}$$

where $k \in [1, K]$ is the index of morphemes and L is the box length cutting original image I_0;

$$F = A(p_0, I_0) + \sum_{k=1}^{K} \frac{A(p_k, I_k)}{2^k} / (1 - \frac{1}{2^K}) \tag{6}$$

where $A(\cdot)$ is the output of neural network and F is the overall score.

To simplify, I fill the empty content for attribute **adj1, adj2, style** using **none**. Therefore, for each data, $K = 3$, and we need to cut each original image by 50% and 75% for training.

About the neural network, I set 4 outputs in parallel(responding $A_0 \rightarrow A_3$), then compute the total score F, finally enter F into the mapping layer. The good news is that the performance of the model does improve after doing so.

TABLE VIII: Applying StairReward for Improving

\	Before Improving		After Improving	
k_fold	SRCC	PLCC	SRCC	PLCC
$k=1$	0.6715	0.7363	0.7045	0.8015
$k=2$	0.6250	0.7433	0.6331	0.7697
$k=3$	0.6954	0.7628	0.6840	0.7863
AVERAGE	0.6640	0.7475	0.6739	0.7858
VAR	1.28e-3	1.89e-4	1.35e-3	2.53e-4

V. DISCUSSION

A. Improvement of Fine-Tuning

About the perception, I draw a bar chart before and after tune the model architecture and parameters as followed:

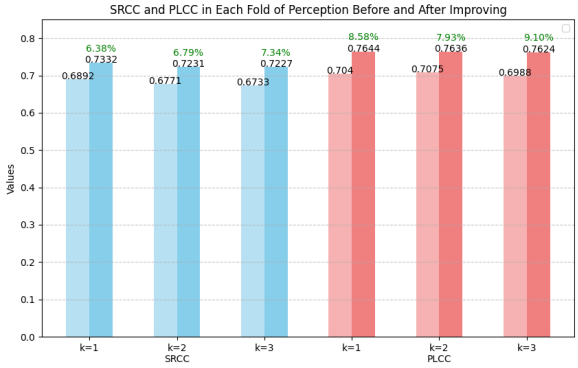


Fig. 1: Improvement of Fine-Tuning

We can see that model architecture such as **Dropout**, selection for **loss,activation** function and model parameters especiallty **learning rate** are really important. If they're well tuned, model performance does improve.

B. Improvement of StairReward

StairReward is a magic metric, and it is essentially a data augmentation method. By taking a screenshot of a part of the image and combining it with a specific part of the prompt that is related to it, we can get more useful information. Using these extra information as inputs, model performance can get much improvement. Next figure shows the improvement of StairReward:

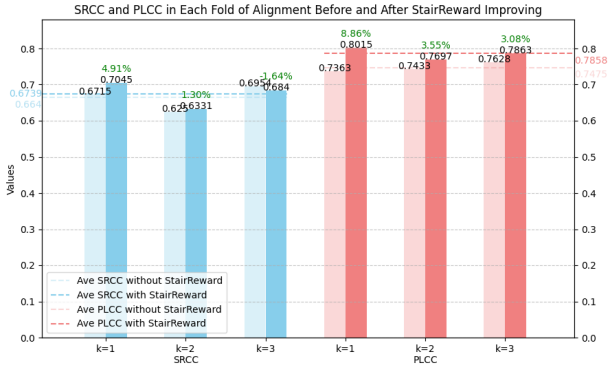


Fig. 2: Improvement of StairReward

We can see that both SRCC and PLCC gets improvement, especially PLCC. If prompt of AIQCIQA-2023 datasets can be aligned to AGIQA-3K, performance may get improved more.

C. Performance on Different Subsets

According to experiments result, I draw relavent comparision charts as shown in Fig. 3, and we can get some information from them:

(1) **Different T2I Models**: About, classification of model quality proposed in the AGIQA-3K paper, models on each specific subset all perform worse than the total model. Overall, what we want to do is to assess any image generated by one T2I model, then there is no need to distinguish quality of T2I model. Therefore, **we should train on total dataset to get a better model.**

(2) **Different Prompt Length**: For both perception and alignment model, the trend is the same: when prompt length increases from $0 \rightarrow 2$, the performace gets better, but gets worse if length reaches to 3. That is reasonable since **increasing the prompt length appropriately** can provide more information to T2I model, getting a better image, and they are easier to assess. But if the length gets too long, T2I model can't handle that complex task, and the image they generate is also harder to assess.

(3) **Different Style**: For both perception and alignment model, they underperform on the subset of Baroque style. Baroque means complex structure and strange form, causing that style is difficult to assess. Relatively, other styles are easier to assess than no styles.

VI. CONCLUSION

In the report, I proposed **DLSR-IQA**, a lightweight, high-performing IQA model. Combining Deep Learning and Stair-Reward metric, the performance on test dataset is excellent(both SRCC and PLCC nearly reach to 0.8). And in each fold cross-vilidation, the model performed stable, meaning the model is universal. It is hoped that the proposal of this model will motivate CV researchers to design models with better performance.

REFERENCES

[1] C. Li, Z. Zhang, H. Wu, W. Sun, X. Min, X. Liu, G. Zhai, and W. Lin, "Agiqa-3k: An open database for ai-generated image quality assessment," 2023.

[2] A. C. Sani Zulkarnaen, I. Gusti Ngurah Rejski Ariantara Putra, N. F. Reviana, R. Hidayah, N. Ibrahim, N. K. Caecar Pratiwi, and Y. N. Fuadah, "Application of convolutional neural network method with mobilenet v1 and resnet-152 v2 architecture in batik motif classification," in Advances on Broad-Band and Wireless Computing, Communication and Applications, L. Barolli, Ed. Cham: Springer Nature Switzerland, 2024, pp. 57–68.

[3] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image data augmentation for deep learning: A survey," 2023.

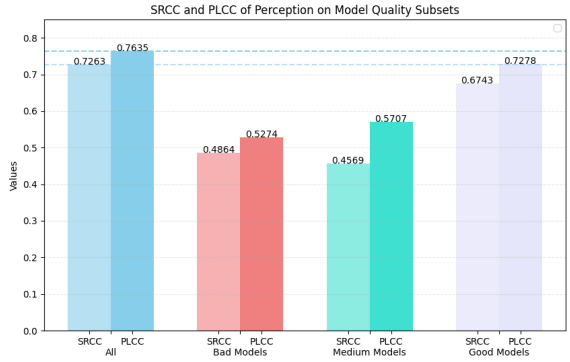
[4] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text to image generation with attentional generative adversarial networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[5] R. A. Friesner, J. L. Banks, R. B. Murphy, T. A. Halgren, J. J. Klicic, D. T. Mainz, M. P. Repasky, E. H. Knoll, M. Shelley, J. K. Perry, D. E. Shaw, P. Francis, and P. S. Shenkin, "Glide: a new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy," Journal of Medicinal Chemistry, vol. 47, no. 7, pp. 1739–1749, 2004, pMID: 15027865. [Online]. Available: https://doi.org/10.1021/jm0306430

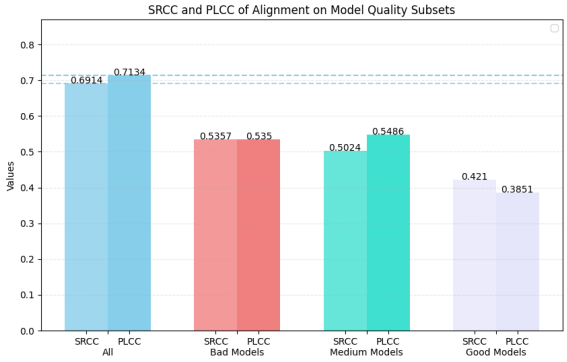
[6] Rassin R, Ravfogel S, Goldberg Y. DALLE-2 is seeing double: Flaws in word-to-concept mapping in Text2Image models[J]. arXiv preprint arXiv:2210.10606, 2022.

[7] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bj¨orn Ommer, "High-resolution image synthesis with latent diffusion models," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.

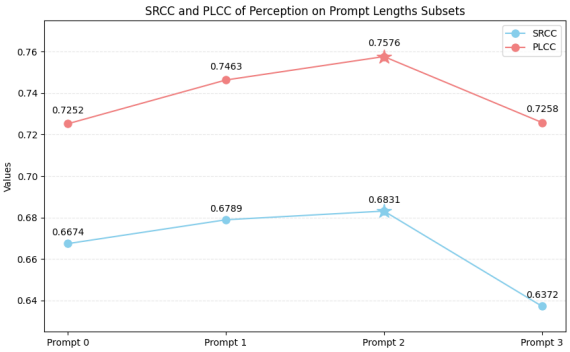
[8] David Holz, "Midjourney," https://www.midjourney.com/, 2023.



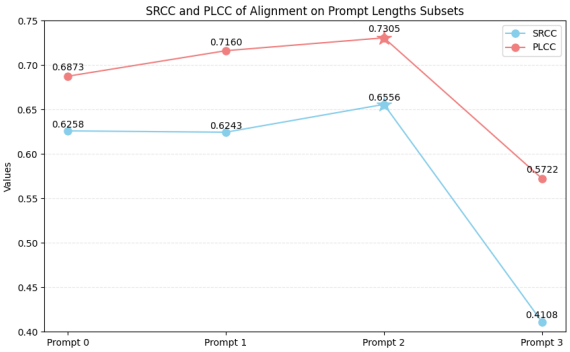
(a) Perception on T2I Subsets



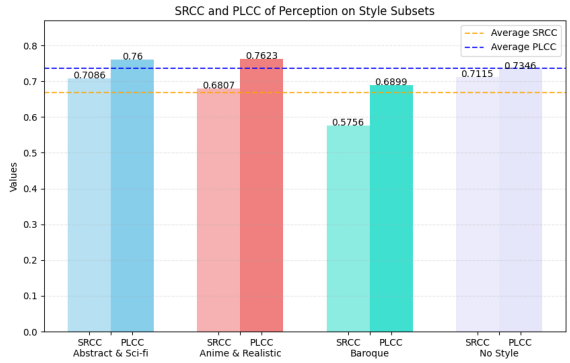
(b) Alignment on T2I Subsets



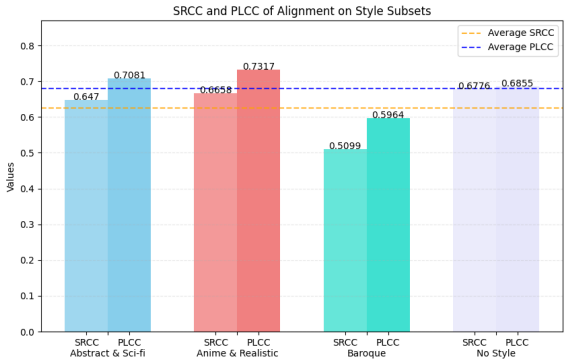
(c) Perception on Prompt Length Subsets



(d) Alignment on Prompt Length Subsets



(e) Perception on Style Subsets



(f) Alignment on Style Subsets

Fig. 3: Perception and Alignment Model on Different Subsets