Premier: Personalized Preference Modulation with Learnable User Embedding in Text-to-Image Generation

Supplementary Material

The following materials are provided in this supplementary file:

- Sec. A: Comparison of Premier with other methods in the PIP dataset evaluation.
- Sec. B: Generalization performance on real user preference images.
- Sec. C: Comparison with LoRA in terms of effectiveness and efficiency.
- Sec. D: Additional experimental analyses, including hyperparameter sensitivity analysis and training-set user scale analysis.
- Sec. E: More qualitative results.

A. Experimental Comparison on PIP-dataset

Evaluation Setup. Our method and several baseline approaches were evaluated on 20 randomly selected users from the PIP dataset, adhering to the experimental protocol established in the main paper. For each user, we generated preference images using Z-Image and obtained the user preference embedding via the linear combination approach.

Evaluation Results. In the PIP dataset, a single user's preferences often exhibit considerable diversity. Our approach leverages user preference embeddings to enable token-level preference modulation, thereby adding user preferences more precisely. From Tab. B, our method achieves the best performance in terms of ViPer Score, ViPer Rate, and LPIPS. From Fig. A, the images generated by our method are closer to the user's preferences.

B. Generalization Performance

Evaluation Setup. To evaluate the effectiveness of our method on real user data beyond the dataset, we select 4 real-world user preference cases and obtain their user preference embeddings for testing. 4 real-world user preferences are represented by the following: Honor of Kings, Arknights, Genshin Impact and Chiikawa. For each preference, we select 8 preferred images as the user's historical data and directly train the user embedding to obtain the new user's preference embedding, enhancing generalization to out-of-distribution preferences.

Evaluation Results. As shown in Fig. B, even on real-world preference data outside the training dataset, our method successfully captures key aspects of user preferences. The characters generated by our method closely resemble the target characters in the user's reference images. Specifically, for the *Honor of Kings* preference, our method produces images that better align with the user's favored 3D-anime aesthetic. For the *Arknights* preference, the outputs more faithfully reflect the stylistic conventions of 2D anime characters. These results demonstrate the generalization capability of our approach to real-world user preferences.

	Data	Flux [1]	LoRA [2]	Ours
ViPer [5] Score↑	0.8890	0.3953	**0.7096**	0.6889
ViPer Rate↑	-	-	**0.906**	0.876
CLIP [4] T2I↑	0.3027	0.3089	0.3011	**0.3183**
LPIPS [6]↓	-	-	0.6037	**0.5986**

Table A. **Quantitative comparison with LoRA training in terms of preference alignment in generated images.**

C. Comparison with LoRA

Evaluation Setup. In the comparison with LoRA, we train a LoRA adapter for each user on their 8 preference images, using the Prodigy [3] optimizer with a learning rate of 1. The LoRA rank is set to 1, and all linear layers in the attention and feed-forward modules are targeted for adaptation, with training conducted for 5,000 steps. We evaluate both methods using the same metrics as described in the main paper. The results are reported in Tab. A and Fig. C.

Evaluation Results. As shown in Tab. A, our method achieves performance comparable to LoRA in terms of preference alignment, while preserving the base model's text-to-image alignment capability more effectively. Fig. C further shows that LoRA may unintentionally harm the base model's ability. In the first row of Fig. C, the LoRA-generated image shows only two airplanes, failing to depict the "three airplanes" specified in the prompt. And in the second row, it violates the instruction "passing through a green traffic light." In contrast, our method remains more faithful to the input text. The results indicate that our method has minimal impact on the generative capability of the base model.

Efficiency Analysis. Moreover, our method is significantly more efficient: each user embedding occupies only 61 KB of storage, compared to 10.7 MB for LoRA. Our preference adapters occupy 1.8 GB of storage in total. When the number of users exceeds 170, our method requires less storage than LoRA. Training takes 30 minutes per user, versus 1.2 hours for LoRA. During inference, our approach introduces only a 1-second overhead over the base model at any resolution, demonstrating high inference efficiency.

	Data	Ours	Bagel	Qwen-Image-Edit	ViPer	InstantStyle	DrUM
ViPer Score↑	0.8919	**0.7204**	0.7149	0.6696	0.5779	0.6232	0.6815
ViPer Rate↑	-	**0.6796**	0.6591	0.5399	0.4613	0.476	0.6134
CLIP T2I↑	0.2844	0.2929	0.2653	0.2877	0.2823	**0.2942**	0.293
LPIPS↓	-	**0.6002**	0.6297	0.6068	0.6306	0.643	0.6208

Table B. **Quantitative comparisons on the PIP-dataset.**

	Ours	Ours 100 users	Ours $\lambda_{\text{distinct}} = 0.1$, $\lambda_{\text{share}} = 0.01$	Ours $\lambda_{\text{distinct}} = 0.01$, $\lambda_{\text{share}} = 0.1$	Ours $\lambda_{\text{distinct}} = 0.1$, $\lambda_{\text{share}} = 1$	Ours $\lambda_{\text{distinct}} = 1$, $\lambda_{\text{share}} = 0.1$
ViPer Score↑	**0.7005**	0.5013	0.5635	0.6632	0.6703	0.6935
ViPer Rate↑	0.8788	0.7115	0.7884	0.875	0.85	**0.8807**
CLIP T2I↑	0.3147	0.3152	**0.3153**	0.3151	0.3139	0.3147
LPIPS↓	0.5946	0.6261	0.6131	0.6014	0.6087	**0.5923**

Table C. **Results for scaling and sensitivity analysis.**

D. Other Analysis Experiments

Training-set User Scale Analysis. To investigate the relationship between the size of the user training set and model performance, we trained the preference adapter on 100 randomly selected users who do not appear in the test set. User embeddings are trained using linear combination approach, with all other settings identical to the evaluation setup described in the main paper. As shown in the Tab. C, model performance significantly degrades as the number of users in the training set decreases. This also indicates that our method can accommodate a wider diversity of user preferences when trained with a larger user population.

Hyperparameter Sensitivity Analysis. The sensitivity analysis on hyperparameters follows the same setup as in the main paper on 20 users of test set, with adjustments only λ_{shared} and $\lambda_{\text{distinct}}$. As shown in Tab. C, the model is more sensitive to λ_{shared} because Δ_{shared} affects preference modulation across all DiT blocks, making performance highly dependent on its associated loss term. When λ_{shared} is too large, the model prioritizes separating users over aligning with preferences, degrading performance. When λ_{shared} is too small, the dispersion loss becomes ineffective, also causing a significant performance drop. The variation of $\lambda_{\text{distinct}}$ has a similar trend to that of λ_{shared} but with a smaller impact, as Δ_{distinct} already exhibits strong inherent diversity.

E. More Qualitative Results

Additional qualitative comparisons with other methods are provided in Figs. D to G.

References

[1] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. 1

[2] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. 1

[3] Konstantin Mishchenko and Aaron Defazio. Prodigy: An expeditiously adaptive parameter-free learner. In *Forty-first International Conference on Machine Learning*, 2024. 1

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 1

[5] Sogand Salehi, Mahdi Shafiei, Teresa Yeo, Roman Bachmann, and Amir Zamir. Viper: Visual personalization of generative models via individual preference learning. In *European Conference on Computer Vision*, pages 391–406. Springer, 2024. 1

[6] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 1

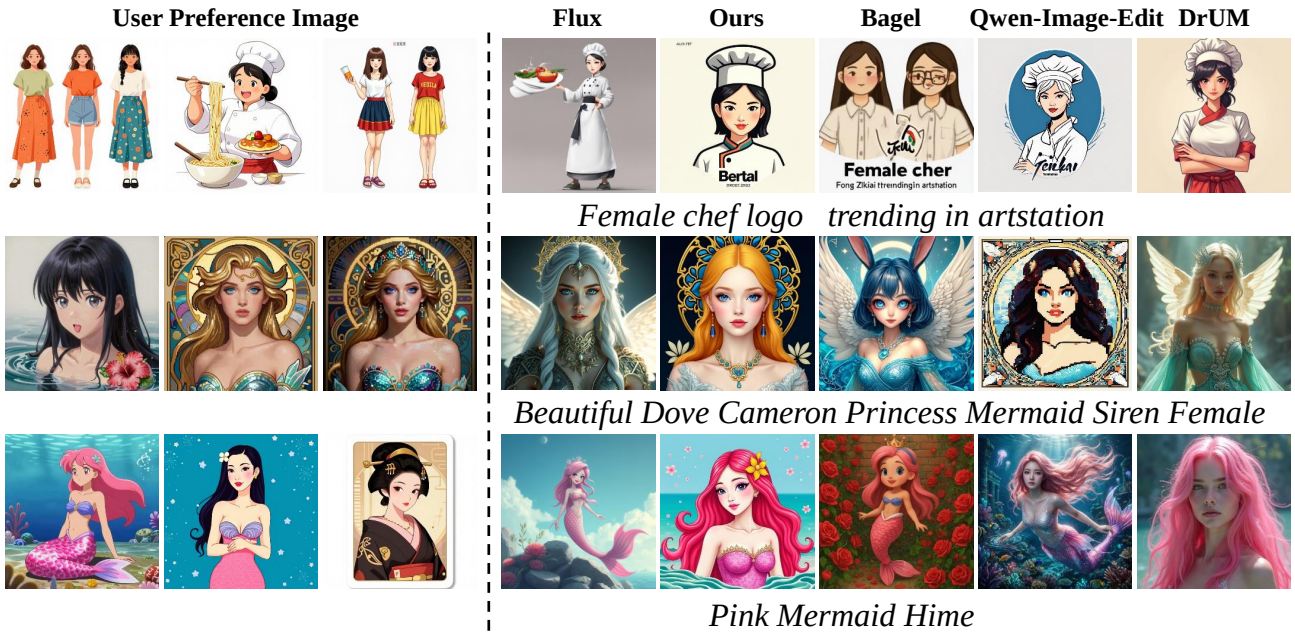

Figure A. **Qualitative comparison on PIP-dataset.**

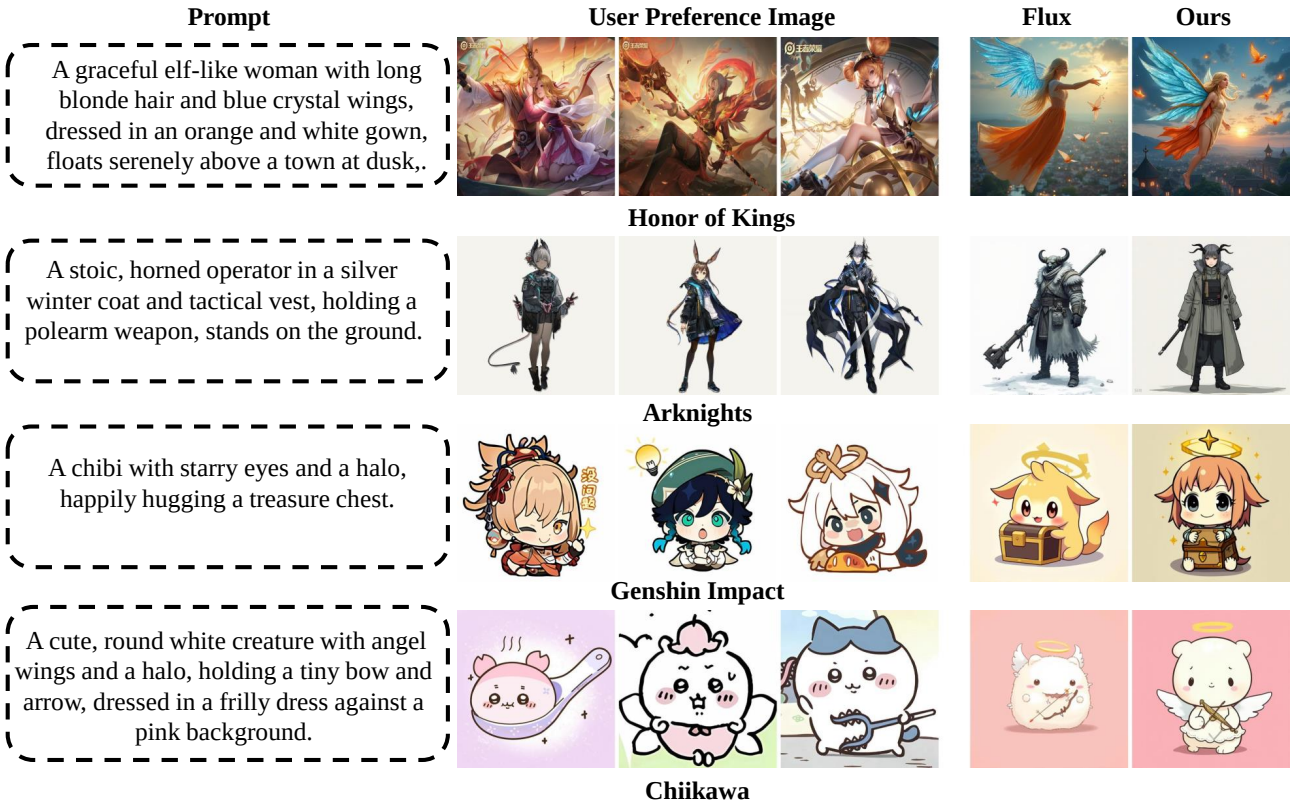

Figure B. **Qualitative results on real user preference data.**

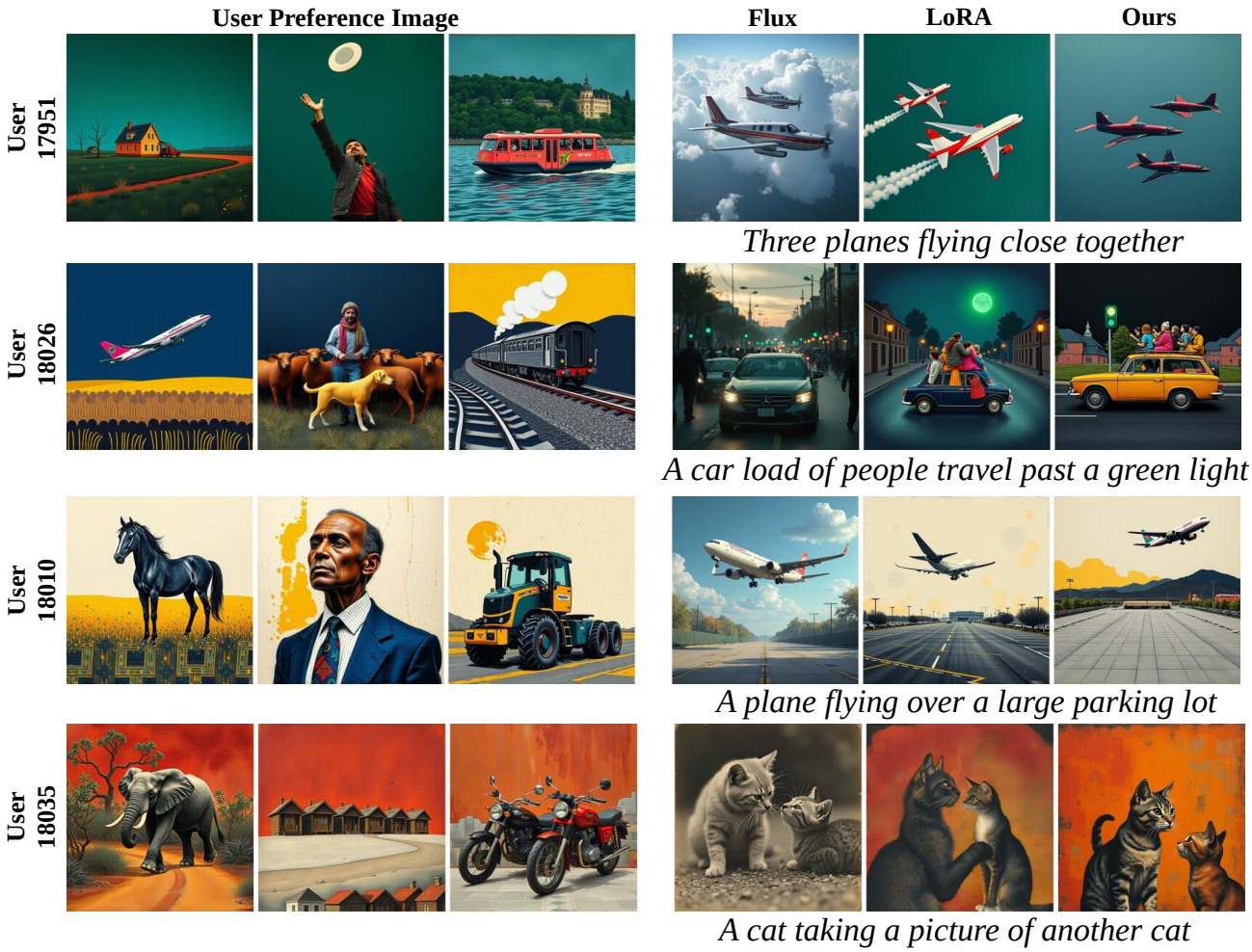

Figure C. **Qualitative comparison with LoRA-generated results.**

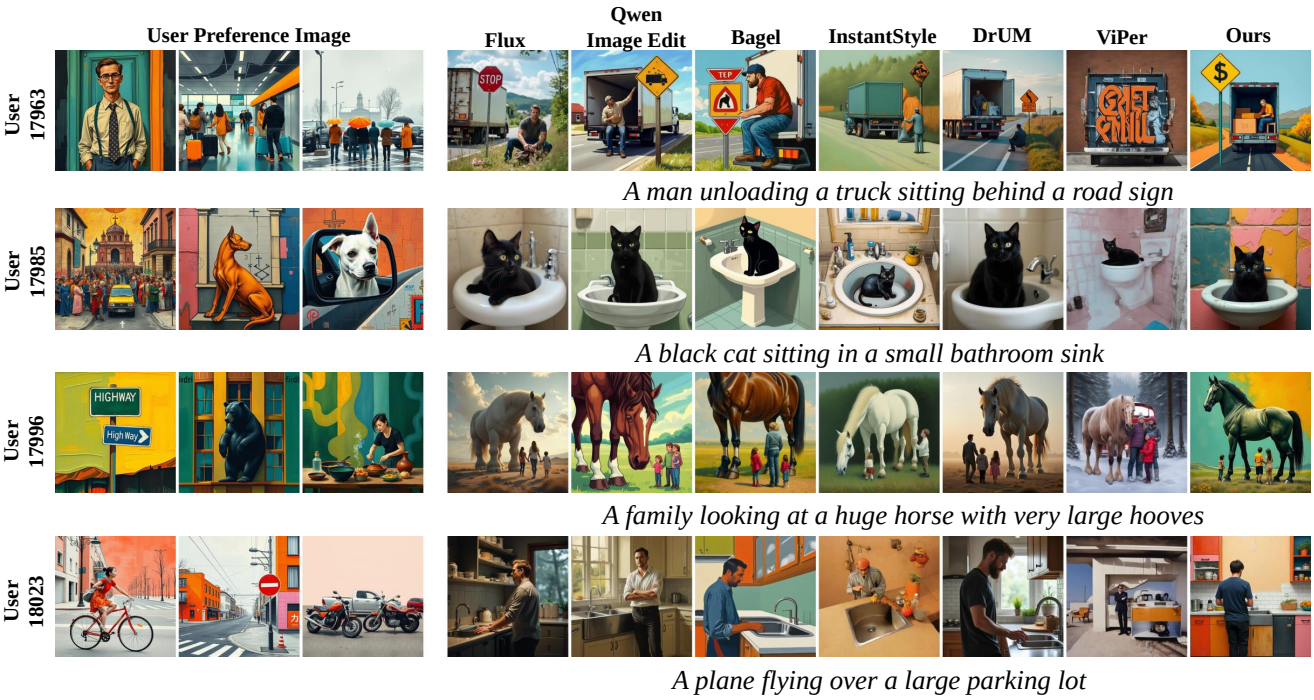

Figure D. **Qualitative comparison with other methods.**

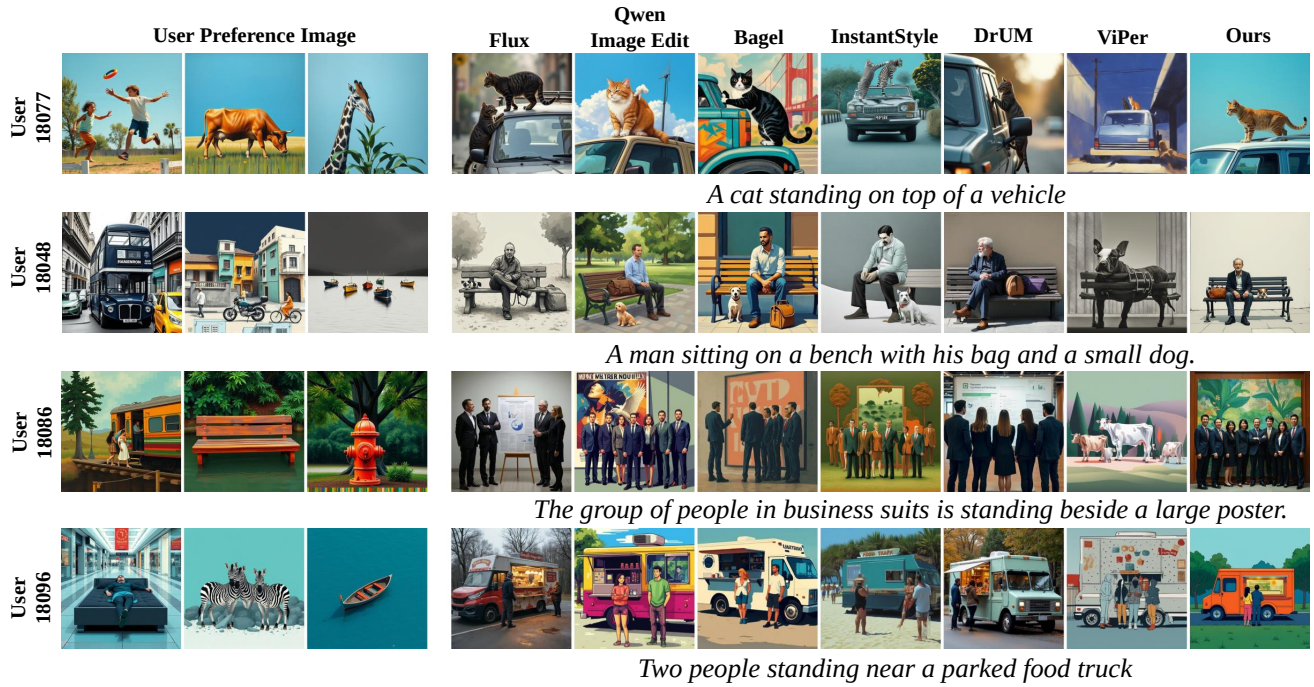

Figure E. **Qualitative comparison with other methods.**

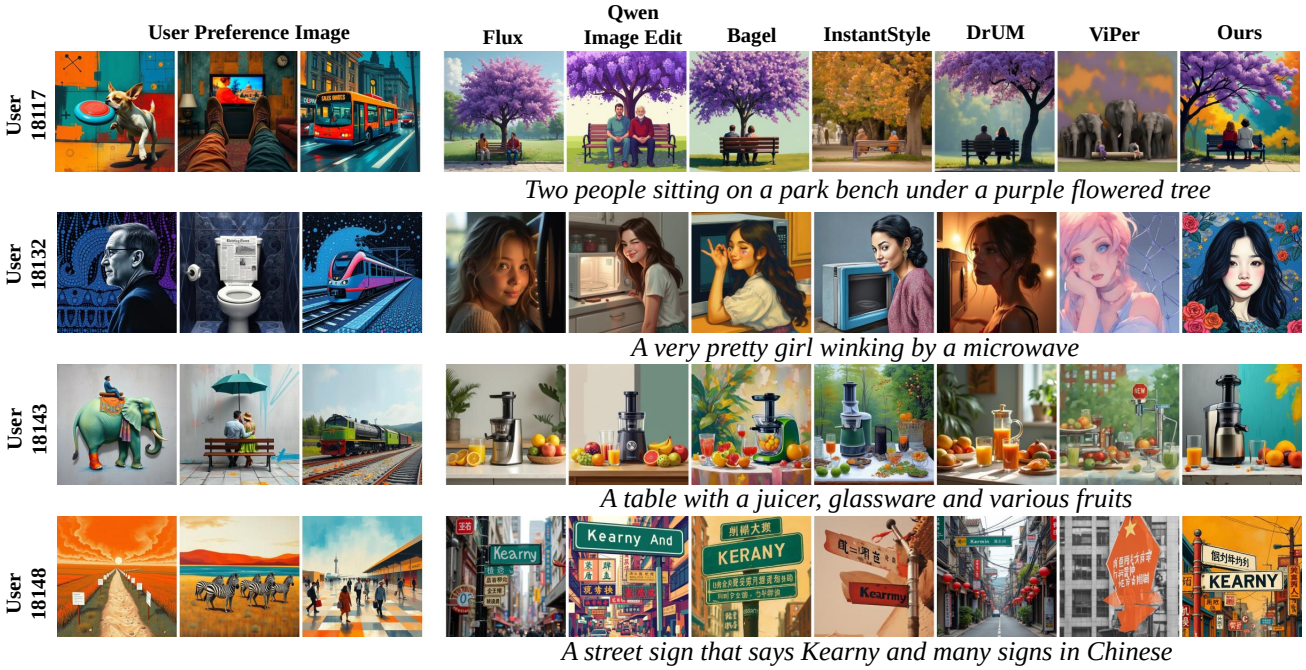

Figure F. **Qualitative comparison with other methods.**

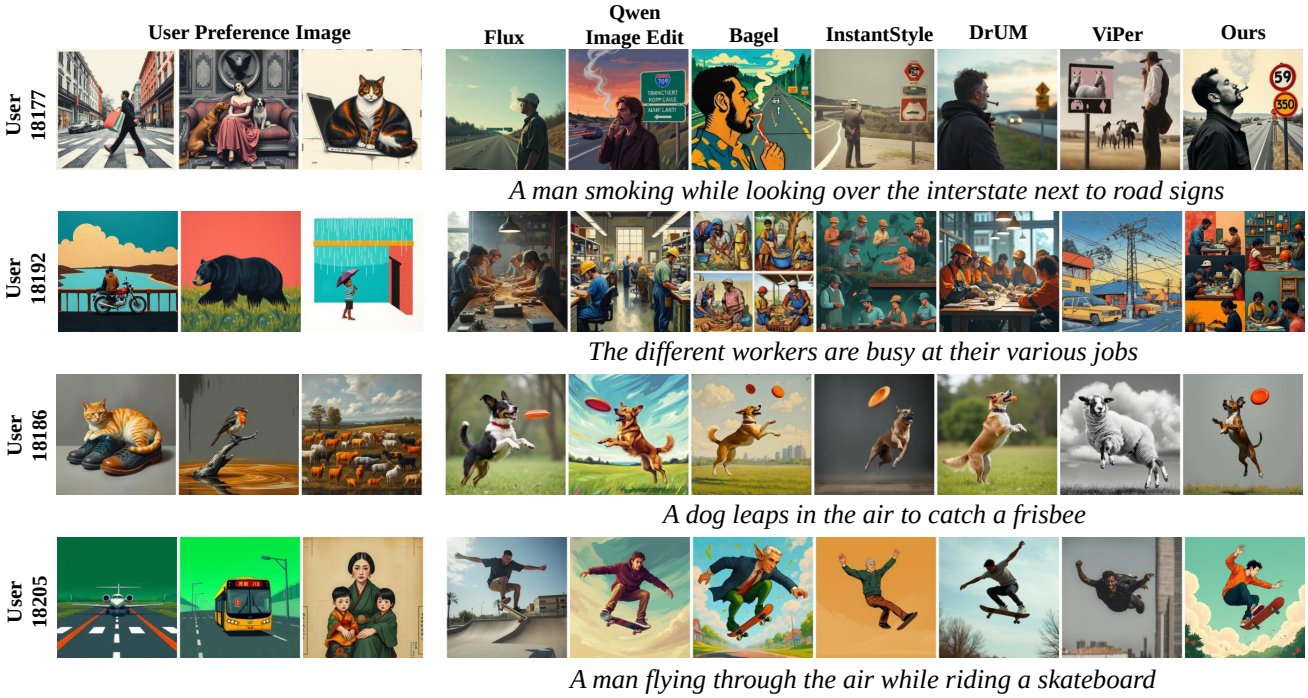

Figure G. **Qualitative comparison with other methods.**