Learning Prompt-Level Quality Variance for Cost-Effective Text-to-Image Generation

Dongkeun Lee and Wonjun Lee

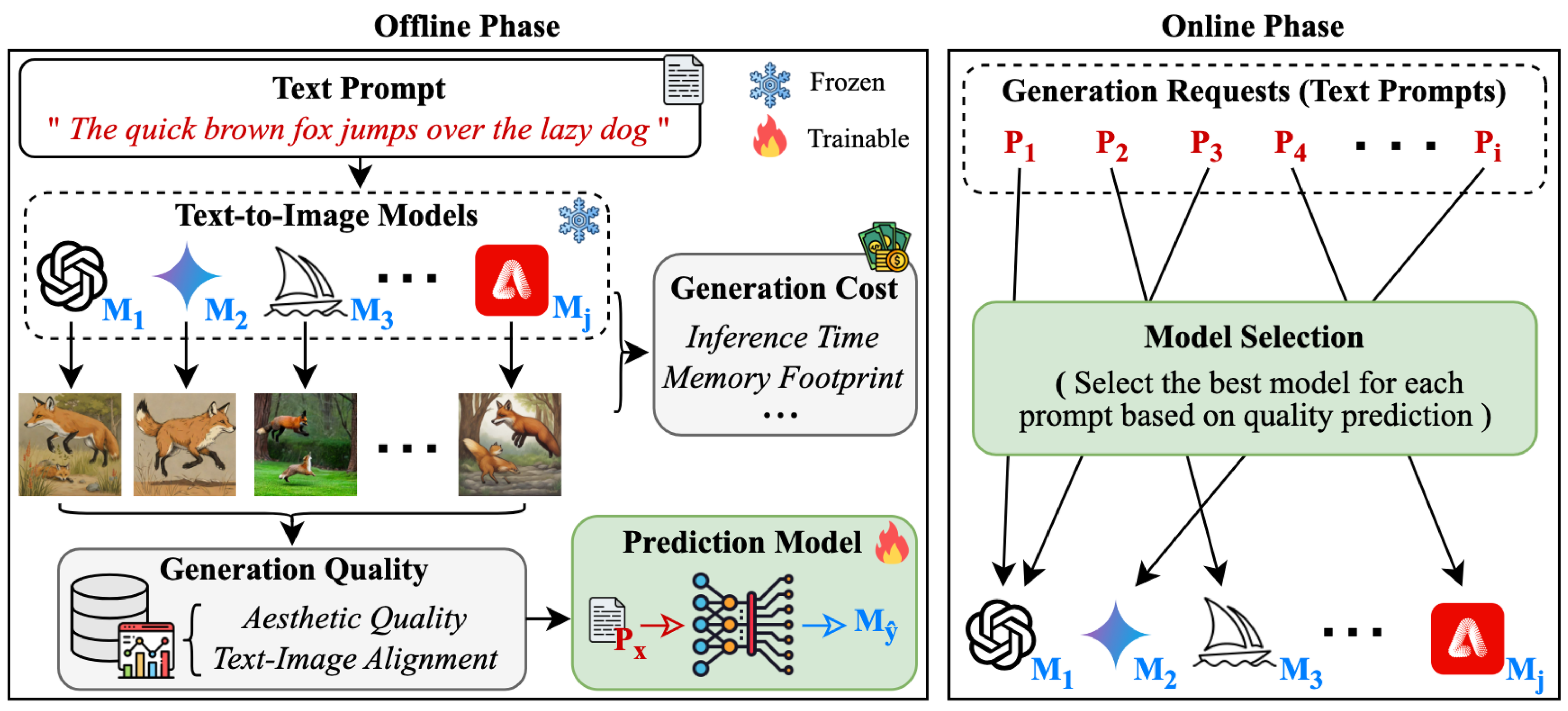
Korea University, Seoul, Republic of Korea

Key Contributions

1. First to utilize quality variance induced by difference in types of prompts to enhance **cost-effectiveness** in text-to-image generation
2. Empirical analysis on **inter-model** and **intra-model** quality variance according to the linguistic features of input prompts
3. A novel approach: Cost-Effective Model Selection
   - ✓ Select the best-performing model for each prompt based on its linguistic features
   - ✓ Reduce total generation cost by **29.25%** with comparable or even higher quality outcomes

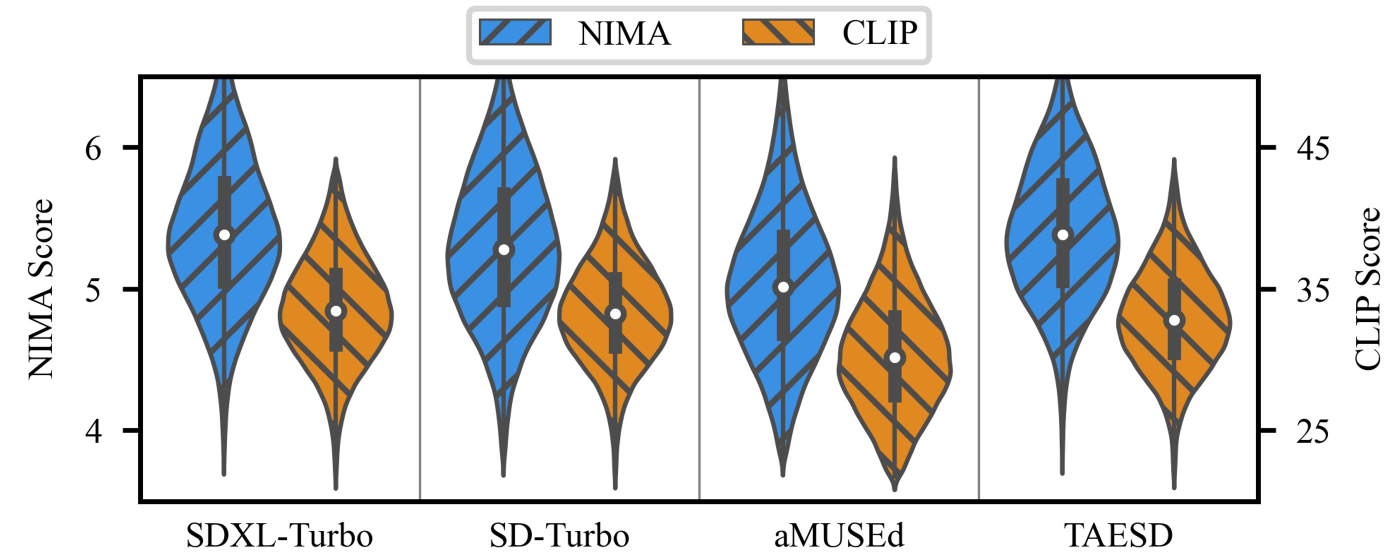
The Proposed Approach

- **Framework Overview**
  - ✓ Run performance tests and train a quality prediction model (**Offline Phase**)
    - – Evaluate generation quality in terms of both **aesthetic quality** and **text-image alignment**
    - – Jointly consider these metrics in selecting the best-performing model
  - ✓ Assign each generation request to the most suitable model (**Online Phase**)
    - – Maximize total generation quality at a lower cost → Increase cost effectiveness
  - ✓ Cost of generation request depends on the pricing model (e.g., API pricing)
    - – We set the cost of each model based on its inference speed and memory footprint



Motivation

- Text-to-image generation is a **multivariable** process
  1. Model properties and training data → *Inter-model* quality variance
  2. Linguistic features of input prompts → *Intra-model* quality variance
- No **single model** excels at handling all types of input prompts
  - ✓ Previous efforts → Enhance the model itself or reformulate prompts
  - ✓ Instead, select the best-performing model based on quality prediction



Problem Definition

- **Prompt-Level Quality Prediction**
  - ✓ Formulate the task as a classification problem
  - ✓ Predict which model will generate an image with the **highest quality** based on the linguistic features of **input prompts**

1. Set the best-performing model M_y for a benchmark prompt P_x^B as:

$$y = \arg\max_{m \in \{1,\dots,j\}} Q(M_m(P_x^B)) \tag{1}$$

2. Train the quality prediction model $F(\cdot)$ to minimize:

$$\sum_{P_x^B} l(F(P_x^B), M_y) \tag{2}$$

3. For generation requests $P^R = \{P_1, \dots, P_i\}$, assign each request P_x^R to $M_{\hat{y}}$:

$$M_{\hat{y}} = F(P_x^R) \tag{3}$$

Constructing Text-to-Image Performance Dataset

- **Experimental Setup**
  - ✓ An Intel i7-8700K CPU with GeForce RTX 2080 Ti GPU
  - ✓ All models generate images of size 512 × 512
  - ✓ CLIP score measured using OpenCLIP ViT-g/14

- **Evaluation Benchmarks**

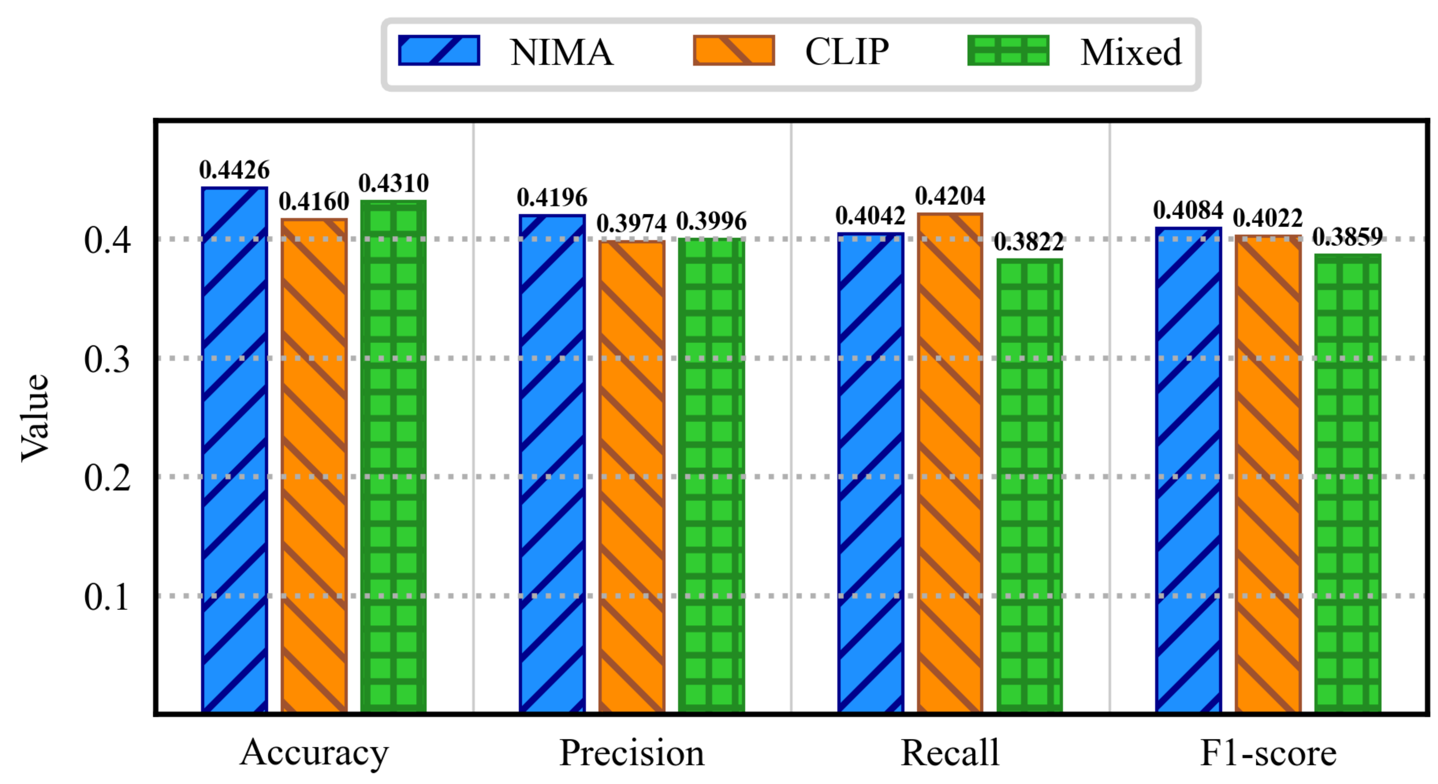
Benchmark	Number of Prompts	Number of Words / Prompt		
		Min.	Max.	Avg. $(\mu(\pm\sigma))$
MS-COCO	31,427	6	45	10.46 (±2.41)
LN-COCO	8,573	6	181	40.45 (±18.75)
DrawBench	200	1	51	11.68 (±9.62)
PartiPrompts	1,632	1	67	9.12 (±7.34)
DiffusionDB	8,168	1	217	24.31 (±16.10)

- **Performance Comparison between Text-to-Image Models**

Model (Sampling Steps)	NIMA Score ↑	CLIP Score ↑	Inf. Time $(\mu(\pm\sigma))$	Memory Footprint
SDXL-Turbo (4 steps)	5.405	33.59	0.616 s (±0.071)	9.51 GB
SD-Turbo (1 step)	5.292	33.34	0.176 s (±0.018)	4.64 GB
aMUSEd (12 steps)	5.024	30.09	0.489 s (±0.047)	3.75 GB
TAESD (25 steps)	5.397	32.90	1.588 s (±0.053)	3.48 GB

Evaluation Result #1: Prediction Performance

- **RQ #1: How well does our quality prediction model find the best-performing text-to-image model?**
  - ✓ Implementation
    - – CLIP text encoder (ViT-B/16) with a classification head on top
    - – Trained for 10 epochs using AdamW optimizer and a learning rate of 6.4×10^{-6}
  - ✓ Lower performance when using Mixed score (mixture of NIMA & CLIP score)
    - – Still, 51.53% of sub-optimal selections generate images with the second-highest quality
  - ✓ Non-linear relationship between NIMA score and CLIP score
    - – Pearson correlation coefficient of 0.1883



Evaluation Result #2: Cost Effectiveness

- **RQ #2: How effective is our approach in reducing cost while preserving generation quality?**
  - ✓ Pricing model (cost per generation request)

$$\text{Inference Time (s)} \times \lceil \text{Memory Footprint (GB)} \rceil \times 0.0000166667 \tag{4}$$

  - ✓ Average quality and total cost of each model selection strategy

Strategy	NIMA Score		CLIP Score		Mixed Score		
	NIMA ↑	Cost ↓	CLIP ↑	Cost ↓	NIMA ↑	CLIP ↑	Cost ↓
Oracle	5.625	0.3876	35.16	0.3461	5.562	34.47	0.3864
SDXL-Turbo	5.405	0.5133	33.66	0.5133	5.405	33.66	0.5133
SD-Turbo	5.303	0.0733	33.40	0.0733	5.303	33.40	0.0733
aMUSEd	5.034	0.1630	30.13	0.1630	5.034	30.13	0.1630
TAESD	5.401	0.5293	32.92	0.5293	5.401	32.92	0.5293
CEMS †	5.462	0.3833	33.75	0.3476	5.434	33.60	0.3586

Contacts

Network and Security Research Lab. (**NetLab**)
- Homepage: https://netlab.korea.ac.kr

Email:
- Prof. Wonjun Lee (wlee@korea.ac.kr)
- Dongkeun Lee (dklee98@korea.ac.kr)

NetLab   Dongkeun Lee