K-Sort Arena: Efficient and Reliable Benchmarking for Generative Models via K-wise Human Preferences

Supplementary Material

A. Derivation of Bayesian Updating

In this section, we provide a more detailed derivation of the formulas in Section 3.1 to further clarify the theoretical underpinnings.

A.1. Derivation of Eq. 6 in Paper

$$
\begin{aligned}
P\left(\theta_1|D\right) &= \int_{-\infty}^{\infty} P\left(\theta_1, \theta_2|D\right) d\theta_2 \\
&\propto \int_{-\infty}^{\infty} \phi\left(\frac{\theta_1-\mu_1}{\sigma_1}\right) \phi\left(\frac{\theta_2-\mu_2}{\sigma_2}\right) \Phi\left(\frac{\theta_1-\theta_2}{\sqrt{\beta_1^2+\beta_2^2}}\right) d\theta_2 \\
&\propto \phi\left(\frac{\theta_1-\mu_1}{\sigma_1}\right) \int_{-\infty}^{\infty} \phi\left(\frac{\theta_2-\mu_2}{\sigma_2}\right) \Phi\left(\frac{\theta_1-\theta_2}{\sqrt{\beta_1^2+\beta_2^2}}\right) d\theta_2
\end{aligned} \tag{15}
$$

Now let's focus on the integral part. We first write $\Phi(x)$ as an integral of $\phi(x)$, as follows:

$$
\begin{aligned}
&\Phi\left(\frac{\theta_1-\theta_2}{\sqrt{\beta_1^2+\beta_2^2}}\right) d\theta_2 \\
&= \int_{-\infty}^{\theta_1} \frac{1}{\sqrt{2\pi(\beta_1^2+\beta_2^2)}} e^{-\frac{(y-\theta_2)^2}{2(\beta_1^2+\beta_2^2)}} dy d\theta_2
\end{aligned} \tag{16}
$$

For simplicity, let $\beta^2 = \beta_1^2 + \beta_2^2$, and the integral part is as follows:

$$
\begin{aligned}
&\int_{-\infty}^{\infty} \phi\left(\frac{\theta_2-\mu_2}{\sigma_2}\right) \int_{-\infty}^{\theta_1} \frac{1}{\sqrt{2\pi\beta^2}} e^{-\frac{(y-\theta_2)^2}{2\beta^2}} dy d\theta_2 \\
&= \int_{-\infty}^{\theta_1} \left(\int_{-\infty}^{\infty} \phi\left(\frac{\theta_2-\mu_2}{\sigma_2}\right) \frac{1}{\sqrt{2\pi\beta^2}} e^{-\frac{(y-\theta_2)^2}{2\beta^2}} d\theta_2 \right) dy \\
&= \int_{-\infty}^{\theta_1} \left(\phi\left(\frac{\theta_2-\mu_2}{\sigma_2}\right) * \phi\left(\frac{y-\theta_2}{\beta}\right) \right) dy \\
&= \int_{-\infty}^{\theta_1} \left(\phi\left(\frac{y-\mu_2}{\sqrt{\sigma^2+\beta^2}}\right) \right) dy \\
&= \Phi\left(\frac{\theta_1-\mu_2}{\sqrt{\beta_1^2+\beta_2^2+\sigma_2^2}}\right)
\end{aligned} \tag{17}
$$

Where "$*$" denotes the convolution of two Gaussian functions. Finally, Bringing the above result into Eq. 15, we have:

$$
\begin{aligned}
P\left(\theta_1|D\right) &= \int_{-\infty}^{\infty} P\left(\theta_1, \theta_2|D\right) d\theta_2 \\
&\propto \phi\left(\frac{\theta_1-\mu_1}{\sigma_1}\right) \Phi\left(\frac{\theta_1-\mu_2}{\sqrt{\beta_1^2+\beta_2^2+\sigma_2^2}}\right)
\end{aligned} \tag{18}
$$

A.2. Derivation of Eq. 7 in Paper

$$
\begin{aligned}
\hat{\mu}_1 = E\left[\theta_1|D\right] &= \frac{\int_{-\infty}^{\infty} \theta_1 P(\theta_1|D) d\theta_1}{\int_{-\infty}^{\infty} P(\theta_1|D) d\theta_1} \\
&= \frac{\int_{-\infty}^{\infty} \theta_1 \phi\left(\frac{\theta_1-\mu_1}{\sigma_1}\right) \Phi\left(\frac{\theta_1-\mu_2}{\sqrt{\beta_1^2+\beta_2^2+\sigma_2^2}}\right) d\theta_1}{\int_{-\infty}^{\infty} \phi\left(\frac{\theta_1-\mu_1}{\sigma_1}\right) \Phi\left(\frac{\theta_1-\mu_2}{\sqrt{\beta_1^2+\beta_2^2+\sigma_2^2}}\right) d\theta_1}
\end{aligned} \tag{19}
$$

We begin with the derivation of the numerator of Eq. 19. Again, we write $\Phi(x)$ as an integral of $\phi(x)$, as follows:

$$
\begin{aligned}
&\Phi\left(\frac{\theta_1-\mu_2}{\sqrt{\beta_1^2+\beta_2^2+\sigma_2^2}}\right) \\
&= \int_{-\infty}^{\theta_1} \frac{1}{\sqrt{2\pi(\beta_1^2+\beta_2^2+\sigma_2^2)}} e^{-\frac{(y-\mu_2)^2}{2(\beta_1^2+\beta_2^2+\sigma_2^2)}} dy
\end{aligned} \tag{20}
$$

The computation of the integrals is analogous to the procedure described in Eq. 17, which requires reordering the integrals and performing the necessary convolutions. Here, we omit the repetitive steps and directly show the final result as follows:

$$
\Phi\left(\frac{\mu_1-\mu_2}{\sqrt{\beta^2+\sigma^2}}\right) \left(\mu_1 + \frac{\sigma_1^2}{\sqrt{\beta^2+\sigma^2}} \frac{\phi\left(\frac{\mu_1-\mu_2}{\sqrt{\beta^2+\sigma^2}}\right)}{\Phi\left(\frac{\mu_1-\mu_2}{\sqrt{\beta^2+\sigma^2}}\right)} \right) \tag{21}
$$

where $\sigma^2 = \sigma_1^2 + \sigma_2^2$ and $\beta^2 = \beta_1^2 + \beta_2^2$. Similarly, the derivation result for the denominator of Eq. 19 is as follows:

$$
\Phi\left(\frac{\mu_1-\mu_2}{\sqrt{\beta^2+\sigma^2}}\right) \tag{22}
$$

Thus, bringing the numerator and denominator results into Eq. 19, we have the following:

$$
\begin{aligned}
\hat{\mu}_1 &= E\left[\theta_1|D\right] \\
&= \frac{\int_{-\infty}^{\infty} \theta_1 P(\theta_1|D) d\theta_1}{\int_{-\infty}^{\infty} P(\theta_1|D) d\theta_1} \\
&= \mu_1 + \frac{\sigma_1^2}{\sqrt{\sum\left(\beta_i^2+\sigma_i^2\right)}} \frac{\phi\left(\frac{\mu_1-\mu_2}{\sqrt{\sum\left(\beta_i^2+\sigma_i^2\right)}}\right)}{\Phi\left(\frac{\mu_1-\mu_2}{\sqrt{\sum\left(\beta_i^2+\sigma_i^2\right)}}\right)}
\end{aligned} \tag{23}
$$

B. Proof of theoretical advantages of UCB

The cumulative regret of the UCB policy grows logarithmically with the number of comparisons n, $R_n = \mathcal{O}(\log n)$, providing better long-term performance compared to the linear growth of cumulative regret, $R_n = \mathcal{O}(n)$, of the random selection policy.

Proof: For all K>1, if policy UCB is run on K machines having arbitrary reward distributions $P_1 \cdots P_k$ with support in [0,1], then its expected regret after n plays is bounded by:

$$R_n^{UCB} \leq \left[8 \sum_{i:\mu_i<\mu^*} \left(\frac{\ln n}{\Delta_i} \right) \right] + \left(1 + \frac{\pi^2}{3} \right) \left(\sum_{j=1}^{K} \Delta_j \right) \tag{24}$$

where $\mu_1 \cdots \mu_k$ are the expected values of $P_1 \cdots P_k$, μ^* is the maximum expected value, and $\Delta_i = \mu^* - \mu_i$ for suboptimal selections. Please refer to [1] for a detailed derivation of the above equation.

When adopting random selection, *i.e.*, choosing an arm uniformly at random at each play, the expected regret after n plays is:

$$R_n^{Rand} = n \cdot \left(\mu^* - \frac{1}{K} \sum_{i=1}^{K} \mu_i \right) \tag{25}$$

In the R_n^{UCB} bound in Eq. 24, the first component is a logarithmic term, and the second component is a constant term and independent of n, thus R_n^{UCB} has a logarithmic growth $\mathcal{O}(\log n)$. In Eq. 25, R_n^{Rand} has a linear growth $\mathcal{O}(n)$. This indicates that UCB can makes better selections over time, thus achieving a significantly lower cumulative regret compared to random selection.

In our K-Sort Arena system, the lower regret of the applied UCB policy indicates that it makes higher-reward player groupings. This yields more ranking benefits in a single comparison, thus allowing the system to converge more quickly with fewer comparisons.

C. List of Evaluated Models

The lists of text-to-image and text-to-video models covered by K-Sort Arena are shown in Table 2 and Table 3, respectively. The data is in no particular order. We will continue to add new models. In the future, besides distilled models [38, 46], we also plan to include the evaluation of models that are compressed through quantization [27–30, 34] and pruning [8, 15].

D. Analysis of Votes

After several months of internal testing, we have collected over 1,000 votes from experts in the field of visual generation. Note that in each vote, four models participate in a free-for-all comparison, which is equivalent to $\frac{K(K-1)}{2} = 6$ pairwise comparisons. This means our voting process can be approximately converted to over 6,000 pairwise comparisons. Figure 9 illustrates the number of comparisons in which each model is involved, with the data representing the number of pairwise comparisons after conversion. Thanks to the UCB algorithm and the pivot specification strategy, all models are fully and balanced evaluated.

Table 2. List of text-to-image models in K-Sort Arena (in no particular order). Here, we show the name and license of each model.

Task	Model	License	Organization
Text2Image	Dalle-3	Commercial	OpenAI
	Dalle-2	Commercial	OpenAI
	Midjourney-v6.0	Commercial	Midjourney
	Midjourney-v5.0	Commercial	Midjourney
	FLUX.1-pro	Open source	Black Forest Labs
	FLUX.1-dev	Open source	Black Forest Labs
	FLUX.1-schnell	Open source	Black Forest Labs
	SD-v3.0	Open source	Stability AI
	SD-v2.1	Open source	Stability AI
	SD-v1.5	Open source	Stability AI
	SD-turbo	Open source	Stability AI
	SDXL	Open source	Stability AI
	SDXL-turbo	Open source	Stability AI
	Stable-cascade	Open source	Stability AI
	SDXL-Lightning	Open source	ByteDance
	SDXL-Deepcache	Open source	NUS
	Kandinsky-v2.2	Open source	AI-Forever
	Kandinsky-v2.0	Open source	AI-Forever
	Proteus-v0.2	Open source	DataAutoGPT3
	Playground-v2.5	Open source	Playground AI
	Playground-v2.0	Open source	Playground AI
	Dreamshaper-xl	Open source	Lykon
	Openjourney-v4	Open source	Prompthero
	LCM-v1.5	Open source	Tsinghua
	Realvisxl-v3.0	Open source	Realistic Vision
	Realvisxl-v2.0	Open source	Realistic Vision
	Pixart-Sigma	Open source	PixArt-Alpha
	SSD-1b	Open source	Segmind
	Open-Dalle-v1.1	Open source	DataAutoGPT3
	Deepfloyd-IF	Open source	DeepFloyd

Table 3. List of text-to-video models in K-Sort Arena (in no particular order). Here, we show the name and license of each model.

Task	Model	License	Organization
Text2Video	Sora	Commercial	OpenAI
	Runway-Gen3	Commercial	Runway
	Runway-Gen2	Commercial	Runway
	Pika-v1.0	Commercial	Pika
	Pika-beta	Commercial	Pika
	OpenSora	Open source	HPC-AI
	VideoCrafter2	Open source	Tencent
	StableVideoDiffusion	Open source	Stability AI
	Zeroscope-v2-xl	Open source	Cerspense
	LaVie	Open source	Shanghai AI Lab
	Animate-Diff	Open source	CUHK etc.

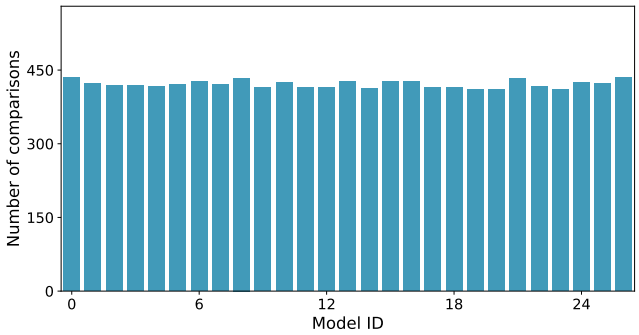


Figure 9. The number of comparisons in which each model is involved. Model IDs are aligned with the order in Table 2. The data is as of Aug 2024.

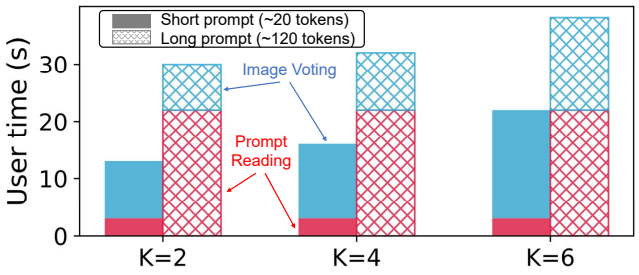


Figure 10. Analysis of user voting times with different K values (2, 4, 6) and prompt complexities.

E. Interface Layout

K-Sort Arena is served by Huggingface Space, and we carefully design the interface based on gradio to achieve a proper layout and user-friendly interaction. The interface layout is shown in Figure 11. First, we describe the initial interface before model running, which is divided into three main regions.

- Region ① describes the background of the project and the evaluation rules, and serves as a guide for users to vote.
- Region ② is the prompt input window, which allows users to enter their own prompts or click "Random Prompt" to randomly select from the data pool.
- Region ③ is some completed samples, including the prompt-image sample pairs, which allow users to quickly complete an experience without running the model.

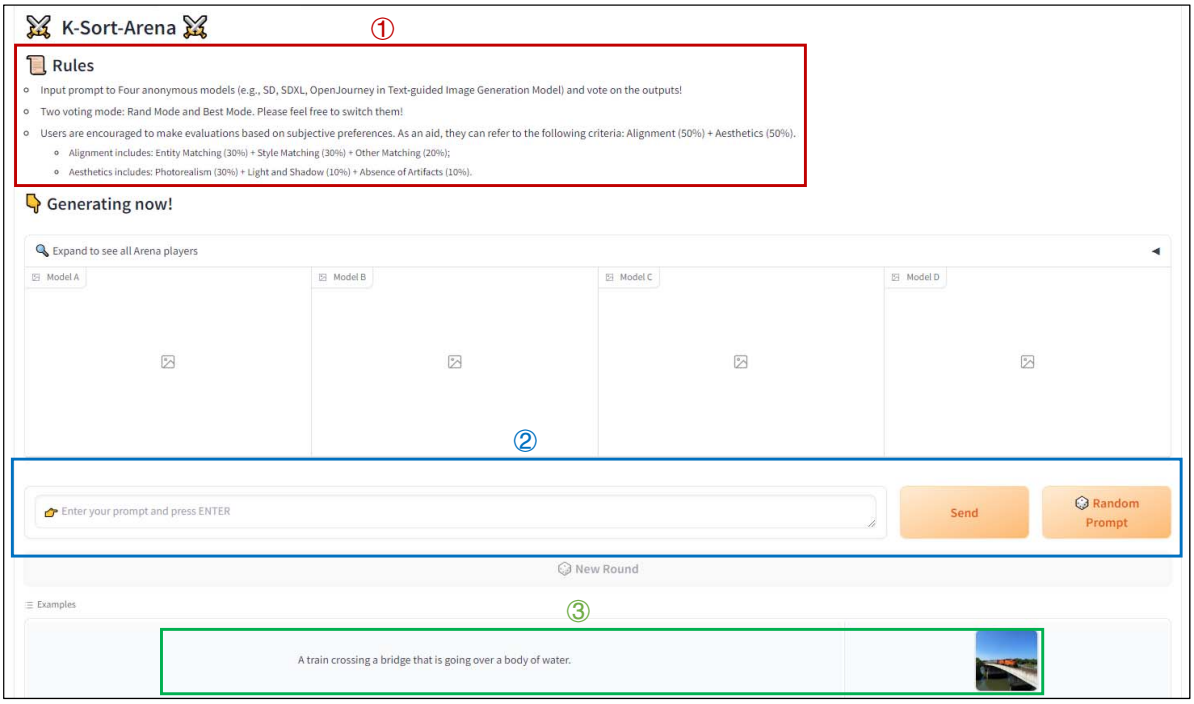
After finishing the model running, the interface automatically jumps to the voting interface. It supports two voting modes, and users can click "Mode" to switch between them.

- In Rank Mode, there are 4 buttons below each image to indicate its rank. Whenever a user clicks on it, the image is retouched with responsive borders and markup.
- In Best Mode, users can choose the best model or a tie.

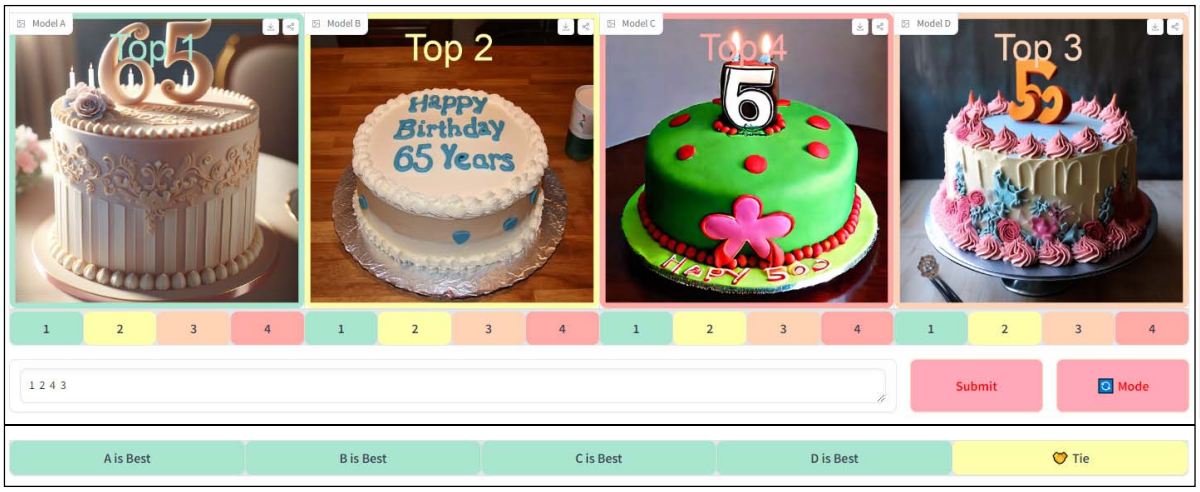
F. User Behavior Analysis

We conduct a comprehensive analysis of user effort in a visual voting task by collecting behavioral data from ten trained participants, and the results are shown is Figure 10. Our study examines effort expenditure across different values of K and varying levels of prompt complexity. Notably, we observe that the additional effort required for K=4 compared to K=2 remains within an acceptable range due to the perceptual intuitiveness of the task. This suggests that while increasing K introduces more choices, the cognitive load does not escalate significantly, allowing users to make selections with relative ease.

Furthermore, as prompt complexity increases, particularly with long prompts derived from the DiffusionDB dataset, users naturally spend more time reading and processing the information. This extended reading phase effectively diminishes the relative differences in effort when engaging in visual voting, as the majority of cognitive load is shifted towards comprehension rather than selection.



(a) Display of the initial interface.



(b) Display of the voting interface.

Figure 11. Interface of K-Sort Arena served by Huggingface Space.