CONSTANTLY IMPROVING IMAGE MODELS NEED CONSTANTLY IMPROVING BENCHMARKS

Anonymous authorsPaper under double-blind review

ABSTRACT

Recent advances in image generation, often driven by proprietary systems like GPT-40 Image Gen, regularly introduce new capabilities that reshape how users interact with these models. Existing benchmarks often lag behind and fail to capture these emerging use cases, leaving a gap between community perceptions of progress and formal evaluation. To address this, we present ECHO, a framework for constructing benchmarks directly from real-world evidence of model use: social media posts that showcase novel prompts and qualitative user judgments. Applying this framework to GPT-40 Image Gen, we construct a dataset of over 35,000 prompts curated from such posts. Our analysis shows that ECHO (1) discovers creative and complex tasks absent from existing benchmarks, such as re-rendering product labels across languages or generating receipts with specified totals, (2) more clearly distinguishes state-of-the-art models from alternatives, and (3) surfaces community feedback that we use to inform the design of metrics for model quality (e.g., measuring observed shifts in color, identity, and structure).

1 Introduction

When new generative image models are released, users often find new and unanticipated capabilities not captured by existing benchmarks. These capabilities are discussed on social media, where users document their interactions with new models and qualitatively discuss their performance. The release of GPT-40 Image Gen (OpenAI, 2025a) exemplified this behavior with the introduction of "Ghiblification," the style-transfer task of turning a natural image into a cartoon version emulating a particular animated studio. This new "task" was not only shared widely on social media, but used as a personal measure of model quality by many members the online community. As of today, explicit benchmarks have now been developed for this task (Jiang et al., 2025), but the benchmarks that we traditionally use to evaluate models do not have the capability to evolve with community feedback, and instead, must react to changes in a delayed cycle.

Indeed, despite significant changes in what constitutes a "good" image generation model, current popular crowdsourced text-to-image benchmarks (Wang et al., 2022; Kirstain et al., 2023) are often still tailored towards older models such as Stable Diffusion (Rombach et al., 2022), with extensive art-centric keyword lists that are not representative of now-feasible use cases. Popular image editing benchmarks (Zhang et al., 2023a;b; Liu et al., 2025) contain overly simple instructions. These instructions were challenging at their inception but do not actually require complex language understanding or reasoning. Furthermore, these tasks can already be solved by many models, new and old. This slow adaptation rate is reflected in model benchmark scores. As we see in Figure 1b, human ratings indicate that 4o Image Gen is substantially better than the current best open-source unified model (Deng et al., 2025), yet even when benchmarking on a recent image editing benchmark (Liu et al., 2025), the gap appears less significant.

With the rapid releases of new image generation models, each revealing a range of new capabilities to be tested, it has become clear that we need more responsive mechanisms for adapting benchmarks to emergent user observations. In this work we present ECHO: Extracting Community Hatched Observations, a re-usable framework that converts community discussion on social media into a structured benchmark. Our proposed method bypasses the traditional "observation to benchmark" cycle, and provides us with a framework for automatically converting real-world ideas and capabilities surfaced by users on social media directly to metrics that we can use to measure and improve SOTA models. The ECHO framework operates by searching social media for mentions of a target model and automatically filtering for coherent image generation prompts specified via text and/or images, while extracting community insights and feedback on particular prompt ca-

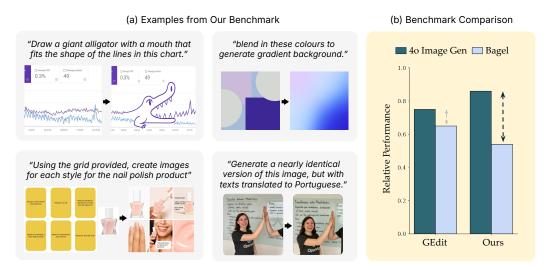


Figure 1: ECHO distills collective discussion about a new generative model into a structured benchmark. As a case study, we apply ECHO to GPT-40 Image Gen (OpenAI, 2025a) on Twitter/X. Left: ECHO automatically surfaces highly diverse and novel tasks not covered in prior benchmarks. Right: Consequently, our image-to-image split shows a 3.2x larger relative performance gap compared to a prior image editing benchmark, GEdit (Liu et al., 2025).

pabilities. It is designed to address a number of common challenges associated with social media, including the tradeoff between post volume and relevance, the splitting of context across posts, and noisy formatting.

Using ECHO, we are able to surface and formalize a number of qualitative observations related to the most recent image generation methods. By running our framework on the 4o Image Gen release, we introduce a new dataset containing more than 35,000 user-sourced prompts which: (1) surfaces creative and complex tasks absent from existing benchmarks, (2) is more diverse and more closely resembles natural user language (contains 2.3x more unique first bigrams and are 1.2x lower in LLM perplexity), (3) better separates state-of-the-art models from prior models, and (4) *automatically* surfaces several new quantifiable indicators for image generation quality, including identity preservation and color shift, which we show can be operationalized into secondary evaluation metrics that could inform future model losses and development.

2 BACKGROUND & RELATED WORK

Model benchmarks often mirror the capabilities of the models themselves, and are designed by model developers in order to evaluate and understand how these models perform on tasks of interest. For example, traditional text-to-image benchmarks (Huang et al., 2023; Ghosh et al., 2023; Lee et al., 2023) and image-to-image benchmarks (Brooks et al., 2023; Wang et al., 2023; Sheynin et al., 2024; Hui et al., 2024; Zhang et al., 2023a) are not collected in-the-wild. These benchmarks contain short and overly simple instructions such as "A cat in front of a chair" or "Add fireworks in the sky" that fail to reflect real user intent, but provide strong diagnostic signal for understanding simple generative understanding.

On the other hand, community-driven benchmarks are often designed to collect real user prompts, and more closely mirror what a downstream user might desire from a model. For example, previous methods (Wang et al., 2022; Xu et al., 2023; Kirstain et al., 2023) have collected real user prompts of Stable Diffusion models from an explicit interface (Rombach et al., 2022). These benchmarks require a significant amount of human intervention to decide which user prompts to model, and how to model them. In addition, the model interface itself can lead to two further limitations: (i) prompt intent is bounded by the capabilities of the model itself, and (ii) prompt style is tailored towards the model rather than natural user language. For example, it has already been demonstrated that users will adjust their prompting behavior to account for limitations of the CLIP (Radford et al., 2021) text encoder, which behaves more like a bag-of-words representation, where they use extensive sets of "phrases rather than complete sentences" (ComfyUI Wiki, 2025), with prompts like "colorful stars, galaxies, space, artstation." Unlike interface-collected datasets, our framework draws on prompts crafted for human audiences on social media, where the goal is to showcase creativity rather than to optimize around model quirks. While prompts inevitably reflect the capabilities

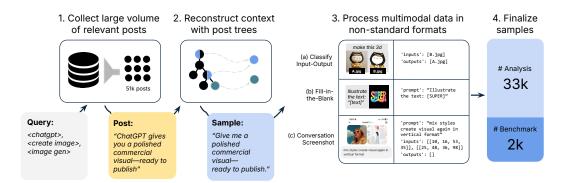


Figure 2: **ECHO Framework.** ECHO is motivated by several challenges inherent to social media. (1) We start with broad queries followed by relevance filtering, since basic querying presents a volume-relevance tradeoff. (2) We then extract prompts from these posts, making sure to utilize the full post tree, as context can be spread across posts. (3) We then apply multimodal processing, since useful data also exists in non-standard formats. (4) Finally, we reserve the highest quality data for benchmarking, while the rest is used for analysis.

of the current best models, our framework is re-runnable and can adapt as models and user behaviors evolve, reducing the risk of per-model biases.

GEdit (Liu et al., 2025) proposed scraping the internet for real image editing prompts. However, these prompts are limited by the imagination of the authors, leading to a restricted set of 11 specific single image editing tasks, such as changing the color or changing the background. Most closely related to our work, IntelligentBench (Deng et al., 2025) and KontextBench (Batifol et al., 2025) were designed to highlight the capabilities of new models released by the same authors. However, details about their data source and creation method are largely unknown, and neither benchmark is publicly available.

Outside of image generation, Chatbot Arena (Chiang et al., 2024) uses an online platform to collect use cases in the wild, incentivizing users to provide data by providing a free platform for interacting with the model. While such a process does collect real user prompts, unlike this approach, we investigate social media, which represents a notably different prompt distribution: since users are seeking reciprocal engagement, they are more incentivized to produce novel and creative examples, rather than tasks that are already well-within model capabilities.

3 CROWDSOURCING A BENCHMARK

Our primary goal is to distill collective discussion about a new generative model into a structured dataset. Such discussion often involves users sharing interesting prompts and outputs, novel task ideas, or commentary on model behavior. We aim to capture all of these cases, in a standardized format:

```
<input text, input image(s)*, output image, community feedback*>
```

where * denotes optional fields; the full set of data we collect is given in Table D.1. However, this objective poses several challenges:

- Collection: A large volume of relevant data is desired, which requires identifying the right platform and gathering the data.
- Processing: A non-trivial amount of processing is required, e.g., the input prompt and images may be embedded in a single screenshot or the prompt may not be written explicitly.
- Filtering: Data quality varies widely, e.g., a user may provide more general commentary or exactly document their input prompt.

We propose a framework, ECHO, that addresses these challenges, illustrated in Figure 2. Our framework first collects relevant posts (Section 3.1), converts posts into self-contained samples (Section 3.2), and finally expands coverage via multimodal processing (Section 3.3).

3.1 IDENTIFYING RELEVANT POSTS

There is an inherent tradeoff between the volume of posts and their relevance. When querying with broader keywords, the average post relevance goes down, and with narrower ones, the available post pool is quickly exhausted. To address this, we implement a two-stage pipeline where we first query for a large volume of posts then use an LLM to filter irrelevant ones.

Designing Keywords. First, our goal is to maximize the post pool. However, we found two issues: (1) LLM-based filtering is expensive, so the pool cannot be too large, and (2) there is a temporal shift in which keywords lead to relevant posts (e.g., in the initial two weeks of the 4o Image Gen release, generic terms like "openai" often retrieve relevant posts, but later on relevancy decreases). Therefore, we use two sets of keywords to query posts within vs. outside the first two weeks of the 4o Image Gen release (see Figure D.1).

Classifying Relevance. We then use an LLM to classify the post text on a 5-point relevance scale (see Figure D.2). We initially collected 68k posts in total, of which 47% passed our relevance filter as 'very likely relevant' or "certainly relevant." Nearly half of collected posts pass this filter, amounting to 32k posts, indicating that our query design is fairly efficient and has a high yield rate.

3.2 RECONSTRUCTING CONTEXT ACROSS POSTS

Posts can be context dependent. For example, a user may write "prompt below" in the first post then include the actual prompt text in a reply. We want self-contained samples, characterized as: a unique prompt some user tried, community feedback towards that prompt and its resulting outputs, and a label for its quality. To achieve this, our framework attempts to collect as much of the reply tree as possible, then use this full context when processing posts into samples.

Constructing Reply Trees. For each post obtained via keyword query, we extract the full reply tree, or URLs pointing to the parent post or child replies. We then recursively expand the dataset by querying these discovered posts and traversing their respective reply chains, introducing 19k new posts from the replies. This procedure enables our framework to discover relevant posts that may not otherwise appear with keyword-based queries. After reply collection, each post contains ancestor chain $\mathcal{P}_{\uparrow} = \langle P_0, ..., P_n \rangle$ and direct replies $\mathcal{C}_{\downarrow} = \{C_0, ..., C_m\}$. We then search for the unique reply trees across all collected posts. We iterate through each post, referred to as the "main post" P_{main} . For every $P_i \in \mathcal{P}_{\uparrow}$, we attach P_{i+1} as its sole child, producing the path $P_0 \rightarrow ... \rightarrow P_n \rightarrow P_{\text{main}}$. Each $C_j \in \mathcal{C}_{\downarrow}$ becomes a child of the main post, giving edges $P_{\text{main}} \rightarrow C_j$. Since the same posts can appear in multiple trees, we remove duplicates via URL and recursively union their children.

Extracting Self-contained Samples. We then use an LLM to convert trees into samples, as illustrated in Figure D.3. This processing step first identifies the spans of text corresponding to the prompt and discards any unrelated remarks, and performs minor fixups such as combining disjoint spans. Next, this step collects any commentary either from the original author or other user replies (e.g., "amazing result," "didn't work," etc.) as a list of community feedback. Finally, the sample is assigned one of three quality labels: "Benchmark" (high quality prompts that are coherent and show clear user intent), "Analysis" (moderate quality partial prompts or commentary that could not be associated with another sample), or "Trash" (off-topic and malformed content that should be discarded). While we want only the highest quality data for benchmarking, we are also interested in retaining any other relevant data for large-scale analysis.

3.3 MULTIMODAL PROCESSING OF POSTS WITH IMAGES

While Section 3.2 can extract text prompts and community feedback, other metadata requires multimodal processing. This step marks the input and output images associated with each sample, updates prompts with fill-in-the-blank, and produces new samples by parsing screenshots, addressing three common cases:

Classifying Input vs. Output Images. There does not exist a standardized format for marking input and output images. For example, the output image could be the first or the last in a series of images, or there may be irrelevant images that are neither inputs nor outputs. Nevertheless, users expect viewers to infer this distinction, and thus we use a VLM to make this same inference (see Figure D.4).

Completing Fill-in-the-Blank. A common user behavior is "fill-in-the-blank" prompts, where users post a template intended for commenters to infill in the replies. Keeping these templates as-is presents

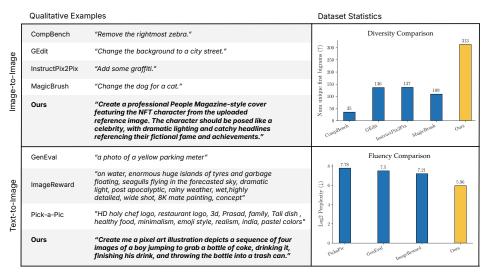


Figure 3: **Dataset Comparison.** The prompts in ECHO are significantly different from prior benchmarks. Top: the image-to-image split is more diverse and complex, with more unique first bigrams. Bottom: the text-to-image split is more fluent, as measured by perplexity under Pythia 12B (Biderman et al., 2023).

a problem, because they are not fully specified and effectively omit the completions that commenters find most interesting. Instead, we use a VLM to reverse-engineer these completions conditioned on the template and the images provided by commenters (see Figure D.4).

Extracting Conversation Screenshots. Another behavior is sharing screenshots of interactions with 4o Image Gen, which may contain prompt text, reference images, and image outputs all within the same frame. This is an especially high-quality source of data, since the inputs and outputs are exactly documented without paraphrasing or summary. Extracting the raw data requires a multi-task computer vision system that can localize images to bounding boxes, classify the sub-images as inputs vs. outputs, and detect what is prompt text vs. unrelated conversation. While one could chain together specialized models for each subtask, we instead opt for a more generalizable solution using a VLM. We first detect these cases with the general multimodal processing prompt, which is then routed to the parsing prompt depicted in Figure D.5. The VLM can not only parse the 4o Image Gen interface but also other non-standard layouts, for example side-by-side collages of input and output images. For the VLM we opt to use Qwen-2.5-VL (Bai et al., 2025), which is specifically trained for bounding box detection.

4 ECHO: A Social-Media Post-Release Benchmark

We initially run our framework to explore the 4o Image Gen release on Twitter/X, which yields 35k total samples. After the LLM quality filter in Section 3.2, we find that 20% of samples are marked as high-quality (usable for benchmarking) and 66% are marked as moderate-quality (usable for analysis). For our final benchmark, we limit each split to up to a thousand samples, to keep the downstream costs of benchmarking (generating outputs and rating them) manageable. The image-to-image split contains 777 prompt-image pairs, consisting of the high-quality samples with input images that passed manual review (see more details in Section 6). The text-to-image split contains 1,000 prompts, acquired via random sampling of high-quality samples without input images.

ECHO Surfaces Diverse and Novel Tasks. While most benchmarks are limited to templated image editing tasks, such as changing the background, changing the color, adding or replacing an object, as shown in Figure 3 (top left), ECHO incorporates several tasks not captured by existing tasks, such as novel view synthesis, image editing that requires cognitive reasoning, virtual try-on, template-based product generation, multi-image subject-driven generation, colorization, image translation, and code-based style transfer (see Figure B.3-B.6). We also can see this diversity effect in the language distribution itself. In Figure 3 (top right), we show the unique first-bigrams of each dataset's editing instructions. ECHO also exhibits a substantially larger variety of first bigrams, indicating more diverse instruction types and image operations.



Figure 4: **Common Failures Observed by Users.** A word cloud of failure cases, derived from community feedback, showing practical capabilities that users care about in real use cases and curiosity-driven tests that reveal deeper model limitations. Common failures include identity shift, color drift, text rendering errors, and style mismatches; more exploratory failures include originality and reasoning about volume.

In addition to diversity, ECHO also remains more natural in the language domain. As shown in Figure 3 (bottom right), our instructions exhibit consistently lower perplexity, indicating that they align more closely with natural language, and suggesting that users now prefer to interact with generative models using fluent, coherent instructions (compared to previous keyword-centric methods).

ECHO Surfaces How Users Interact With Models. To capture failure modes that users explicitly care about, we first use an LLM to label each piece of community feedback as denoting a success or a failure. Then, for each failure case, the LLM generates a short keyword summary describing the underlying issue (e.g., a failure to render "a transparent helmet" correctly will get the keyword "transparency"). We visualize these keywords as a word cloud and highlight representative cases, as shown in Figure 4.

Figure 4 shows us that users are generally most sensitive to failure types such as identity shift, color drift, text rendering errors, style mismatches, and aspect ratio inaccuracy. These failure modes reflect practical use cases where users expect reliability and usefulness, and thus indicate areas where improving models would directly enhance satisfaction. Beyond these common issues, ECHO also surfaces corner-case failures that users found interesting. These often come from probing interesting model behaviors, such as reasoning failures in scientific contexts, misunderstandings of concepts such as originality, and difficulty with counting. Such cases reveal deeper limitations of current models and highlight opportunities for future research.

The community feedback from ECHO also reveals practical strategies that users employ to work around model limitations. As shown in Figure 5a, users discuss ways to construct valid mazes or mitigate identity mismatches. In this way, ECHO records crowdsourced prompting solutions to certain issues, which also reflects what users care about, and can help to motivate future model development.

Exploratory Behaviors. Interestingly, ECHO also surfaces cases where users explore the model's behavior itself, rather than pursuing a concrete task. As shown in Figure 5b, some examples include prompting 4o Image Gen to generate a self-portrait (where it refers to itself as DALL-E) or its favorite color (creating speculation about "invisible colors" beyond human vision). These examples illustrate how users collectively probe and reflect on how models behave under novel edge cases, and reveal interesting behaviors that are not captured in standard benchmarks, yet might be of interest to model developers.

5 ECHO DIFFERENTIATES MODELS

Given our newly curated in-the-wild benchmark, we can now use it to differentiate models. We evaluate three types of models:

• Unified Models. To capture the open-source community's progression, we include early models like Anole (Chern et al., 2024; Team, 2024) and recent models like Bagel (Deng et al., 2025).

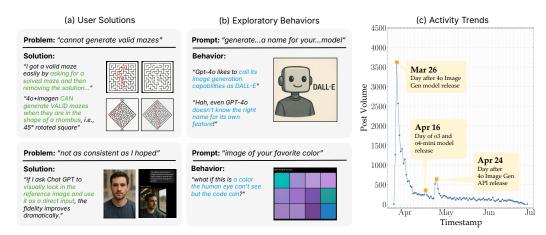


Figure 5: **How Users Interact with Models.** We depict qualitative examples of (a) user solutions and (b) exploratory behaviors, discovered via community feedback. We also visualize (c) activity trends using the timestamps of collected posts.

We also evaluate proprietary models like 40 Image Gen (OpenAI, 2025a), as well as Gemini 2.0 Flash (Comanici et al., 2025) and the more recent 2.5 Flash (Nano Banana) (Gemini, 2025).

- LLM+Diffusion. A good baseline for unified models is its most naive implementation: an LLM chained to a diffusion model, where the LLM rewrites the input prompt before diffusion image generation. We follow the best-performing method from Zhou et al. (2025), a pipeline with GPT-40 (OpenAI, 2024) as the LLM and DALL-E 3 (Betker et al., 2023) as the diffusion model.
- Image Editing Models. Another natural baseline is a specialized image editing model without a
 sophisticated text encoder. To represent this category, we use Flux Kontext (Batifol et al., 2025),
 which demonstrates state-of-the-art image editing performance.

Our overall evaluation metric for the benchmark is head-to-head "win rate", a relative rather than absolute metric. Given that our benchmark is composed of in-the-wild prompts that are intrinsically open-ended, it is very challenging to define a notion of "accuracy." The win rate is calculated across all $\binom{n}{2}$ pairwise model comparisons, where each model earns 1 for a win, 0 for a loss, and 0.5 for a tie. Therefore, the final win rate of a model can be interpreted as its average win rate compared with all other models.

Automatic Evaluation. Due to the cost of collecting human evaluations, we primarily leverage automated evaluation through VLM-as-a-judge. We follow the "single answer grading" setup from MT-Bench (Zheng et al., 2023). In this setup, a score is directly assigned to each output, then converted into "pseudo pairwise" comparisons: for any pair of models, the one with the higher score is treated as the winner. This setup is more scalable as the number of models being evaluated increases, and simplifies the benchmarking process. Furthermore, MT-Bench validates that both true pairwise and pseudo pairwise grading show high agreement with human judgements. To mitigate any biases VLM-as-a-judge might have towards models from the same developer, we ensemble the judgements of three evaluators. We use GPT-40 (OpenAI, 2024), Gemini 2.0 (Team et al., 2023), and Qwen2.5-VL-32B-Instruct (Bai et al., 2025), then take the majority vote to determine the winner of each model pair. Following MT-Bench, we instruct the model to produce a chain-of-thought and consider factors like prompt following, fidelity to any reference images, and realism and aesthetics, before producing a score (see Figure E.1).

Human Correlation. As a secondary validation of the automatic evaluator beyond those in Zheng et al. (2023), we compare our automated evaluations against against gold label human annotations. Specifically, we present five expert raters with outputs of all 8 models for 200 samples, and ask the annotators to rank the outputs from best to worst for both the text-to-image and image-to-image splits. We found that our VLM-as-a-judge measure correlates weakly, but significantly, with human ratings (GPT: $\tau_b = 0.117_{p=0.0036}$, Gemini: $\tau_b = 0.083_{p=0.0199}$, Qwen: $\tau_b = 0.045_{p=0.1327}$, Human-Human concordance: $W = 0.49_{p<0.0001}$). While the correlation is positive, and significant for Gemini and GPT, this result suggests that further research into judge models may be necessary for stronger results overall. See Appendix C for more details.

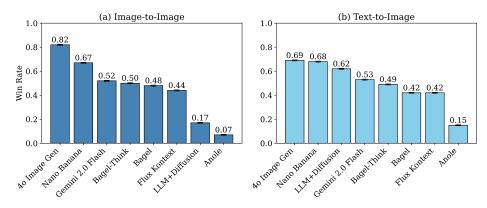
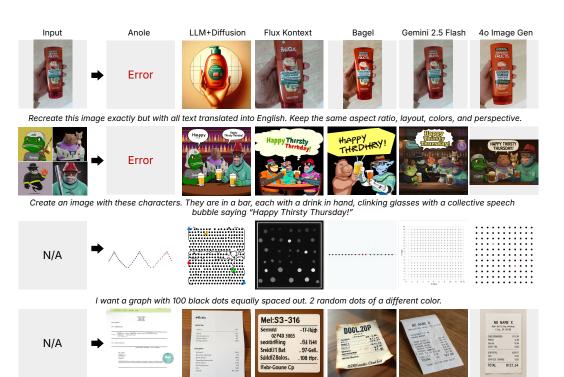


Figure 6: **Overall Evaluation.** We compare a range of unified models, as well as an image editing (Flux Kontext) and shallow fusion (LLM+Diffusion) baseline. We report the win rate, or percentage of pairwise comparisons won. The win rate is calculated automatically with an ensemble of three VLMs-as-a-judge.



Generate a photorealistic image that appears to be a photograph of a detailed paper receipt. The receipt should display a total of \$127.54, with all numerical values (item prices, taxes, and any additional charges) accurately adding up. The restaurant name: No Name X and the address: Non-Existing Avenue, City, CA 94130

Figure 7: **Qualitative Model Comparison**. Challenging tasks from our benchmark, ranging from translation to multi-concept combination to mathematical reasoning, elicit diverse model responses. We mark samples where the model fails to generate an output as "Error."

5.1 RESULTS

We present the win rate comparison on the image-to-image and text-to-image splits in Figure 6. Qualitative results of representative models are shown in Figure 7. On the image-to-image split (Figure 6a), model performance separates into five distinct tiers. First, 4o Image Gen significantly outperforms the other models, followed by Gemini's Nano Banana. Next, Gemini 2.0 Flash, Bagel-Think, Bagel, and Flux Kontext exhibit similar performance. Finally, LLM+Diffusion then Anole perform much worse. We observe similar trends on the text-to-image split (Figure 6b), although the gaps are less pronounced and LLM+Diffusion makes a large jump forward in its ranking.

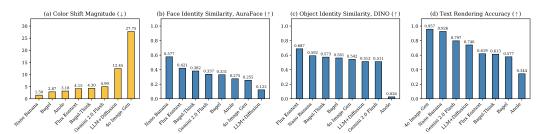


Figure 8: **Specialized Metrics from Community Feedback.** Based on qualitative community observations, we validate that 40 Image Gen exhibits large shifts in color (a) and face identity (b), moderate shifts in structure distance (c), but superior text rendering accuracy (d).

5.2 CLOSING THE LOOP WITH COMMUNITY FEEDBACK METRICS

In addition to the automated evaluations in Section 5, we also wanted to see how community feedback extracted using ECHO could help to differentiate model performance in fine-grained ways. Based on the failure categories extracted by ECHO, and illustrated in Figure 4, we designed several specialized automated metrics: color shift magnitude, face identity similarity, structure distance, and text rendering accuracy. For each metric, described below, we used an LLM to classify samples where each metric is applicable (Figure E.3), and computed the metric over these samples, with the results presented in Figure 8.

Color Shift Magnitude. We quantify the "yellow tint" frequently reported in community feedback with a color shift metric, computed as the average difference between the color histogram of the input versus output images. As shown in Figure 8a, 4o Image Gen indeed exhibits the largest color shift. Interestingly, the only other method from the same developer, LLM+Diffusion (implemented with DALLE-3), also exhibits an abnormally large color shift. Users theorize that the yellow tint could be a "watermarking method, potentially trying to do something kinda fancy with low level pixel encoding."

Face Identity Similarity. Community feedback critiques face identity shifts, which we quantify with a face embedding metric. Specifically, we use AuraFace (Deng et al., 2019; fal, 2025) to detect faces and extract their embeddings, then select the input-output face pair with the highest cosine similarity. **Figure 8b** confirms user observations that 4o Image Gen struggles with face preservation, which could be attributed to a lossy visual encoder or insufficient identity-oriented training data.

Structure Distance. Users are perceptive towards drift in visual structure, such as object positioning or human pose, which we measure using a DINO-based (Caron et al., 2021) structure metric. Following the setup of Tumanyan et al. (2023b), we compute the Frobenius norm of the Gram matrices derived from DINO key features (Tumanyan et al., 2023a) for input-output image pairs. As expected, methods not specifically trained on image-to-image data (LLM+Diffusion and Anole) perform the worst in structure preservation, as shown in Figure 8c. Outside of this category, 4o Image Gen also exhibits non-negligible drift, matching observations that it tends to re-approximate images rather than faithfully copy image structure.

Text Rendering Accuracy. Users are also sensitive towards rendered text, which we measure via VLM-as-a-judge. Unlike OCR-based string matching, VLMs can produce a more holistic score that takes into account factors like legibility in addition to spelling, punctuation, and grammar (see Figure E.2). Figure 8d shows that 4o Image Gen achieves near-perfect text rendering accuracy, consistent with its popularity as a tool for generating infographics and other text-heavy media.

Together, these results show how community feedback can be systematically translated into targeted quantitative metrics that expose fine-grained tradeoffs across models. Beyond confirming user observations, this approach produces concrete, interpretable signals that can guide model development.

6 Conclusion

In this work, we introduced ECHO, the first framework for evaluating image generation in alignment with emerging, real-world use cases of modern image models. Applied to social media posts about GPT-4o Image Gen, ECHO surfaces novel use cases not captured by prior benchmarks, differentiates proprietary from open-source models, and motivates targeted metrics grounded in common failure cases such as text rendering. As both models and user needs evolve, so too must the benchmarks that guide their development.

ETHICS STATEMENT

In this work, we primarily study discussion of 4o Image Gen on Twitter/X, a public social media platform where users voluntarily share content, for academic research purposes. Our collection process implicitly benefits from existing moderation systems: Twitter/X removes posts that violate its content policies (X Help Center, 2025), and ChatGPT refuses to generate images that violate its usage guidelines (OpenAI, 2025b). For this reason, the collected posts are relatively benign, as illustrated by qualitative examples from our dataset (see Appendix B). During our manual review, we also take additional steps to remove potentially harmful material. We did not observe any images that were overtly violent, hateful, or sexual in nature, although we removed a small number of risque images out of an abundance of caution. To minimize privacy risk, we manually exclude images that plausibly depict anyone under eighteen.

REFERENCES

- Pierre Andrieu, Sarah Cohen-Boulakia, Miguel Couceiro, Alain Denise, and Adeline Pierrot. A unifying rank aggregation framework to suitably and efficiently aggregate any kind of rankings. *International Journal of Approximate Reasoning*, 162:109035, 2023. ISSN 0888-613X. doi: https://doi.org/10.1016/j.ijar.2023.109035. URL https://www.sciencedirect.com/science/article/pii/S0888613X23001664.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv e-prints*, pp. arXiv–2506, 2025.
- James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions (2023). *URL https://cdn. openai. com/papers/dall-e-3. pdf*, 6, 2023.
- Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.
- Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.
- Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.
- Ethan Chern, Jiadi Su, Yan Ma, and Pengfei Liu. Anole: An open, autoregressive, native large multimodal models for interleaved image-text generation. *arXiv preprint arXiv:2407.06135*, 2024.
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*, 2024.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv* preprint arXiv:2507.06261, 2025.
- ComfyUI Wiki. Basic syntax tips for comfyui prompt writing, 2025. URL https://comfyui-wiki.com/en/tutorial/basic/stable-diffusion-prompt-basic.

543

544

546 547

548 549

550

551

552

553

554

556

559

560

561

563

565

566

567 568

569

570

571

572

573

574 575

576

577

578 579

580

581 582

583

584

585

586

587

588 589

590

591

- 540 Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. arXiv preprint 542 arXiv:2505.14683, 2025.
 - Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4690-4699, 2019.
 - fal. Auraface, 2025. URL https://huggingface.co/fal/AuraFace-v1.
 - Gemini. Nano banana, 2025. URL https://gemini.google/overview/ image-generation.
 - Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. Advances in Neural Information Processing Systems, 36: 52132-52152, 2023.
 - Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. Advances in Neural Information Processing Systems, 36:78723–78747, 2023.
 - Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. Hq-edit: A high-quality dataset for instruction-based image editing. arXiv preprint arXiv:2404.09990, 2024.
 - Yuxin Jiang, Liming Jiang, Shuai Yang, Jia-Wei Liu, Ivor Tsang, and Mike Zheng Shou. Balanced image stylization with style matching score. arXiv preprint arXiv:2503.07601, 2025.
 - Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. Advances in neural information processing systems, 36:36652–36663, 2023.
 - Tony Lee, Michihiro Yasunaga, Chenlin Meng, Yifan Mai, Joon Sung Park, Agrim Gupta, Yunzhi Zhang, Deepak Narayanan, Hannah Teufel, Marco Bellagente, et al. Holistic evaluation of text-to-image models. Advances in Neural Information Processing Systems, 36:69981–70011, 2023.
 - Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. arXiv preprint arXiv:2504.17761, 2025.
 - OpenAI. Gpt-4o system card, 2024. URL https://openai.com/index/ gpt-4o-system-card.
 - OpenAI. Addendum to gpt-4o system card: Native image generation, 2025a. Accessed: 2025-08-24.
 - OpenAI. Creating images and videos in line with our policies, 2025b. URL https://openai.com/ policies/creating-images-and-videos-in-line-with-our-policies/.
 - 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, pp. 8748–8763. PmLR, 2021.
 - Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10684-10695, 2022.
 - Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8871–8879, 2024.
 - Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. arXiv preprint arXiv:2405.09818, 2024.

- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
- Narek Tumanyan, Omer Bar-Tal, Shir Amir, Shai Bagon, and Tali Dekel. Disentangling structure and appearance in vit feature space. *ACM Trans. Graph.*, nov 2023a. ISSN 0730-0301. doi: 10.1145/3630096. URL https://doi.org/10.1145/3630096.
- Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1921–1930, June 2023b.
- Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J Fleet, Radu Soricut, et al. Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18359–18369, 2023.
- Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv* preprint arXiv:2210.14896, 2022.
- X Help Center. The x rules, 2025. URL https://help.x.com/en/rules-and-policies/x-rules.
- Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. Advances in Neural Information Processing Systems, 36:15903–15935, 2023.
- Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In *Advances in Neural Information Processing Systems*, 2023a.
- Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, Caiming Xiong, and Ran Xu. Hive: Harnessing human feedback for instructional visual editing. *arXiv preprint arXiv:2303.09618*, 2023b.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=uccHPGDlao.
- Pengfei Zhou, Xiaopeng Peng, Jiajun Song, Chuanhao Li, Zhaopan Xu, Yue Yang, Ziyao Guo, Hao Zhang, Yuqi Lin, Yefei He, Lirui Zhao, Shuo Liu, Tianhua Li, Yuxuan Xie, Xiaojun Chang, Yu Qiao, Wenqi Shao, and Kaipeng Zhang. Opening: A comprehensive benchmark for judging open-ended interleaved image-text generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 56–66, June 2025.

APPENDIX

The Appendix is organized as follows:

- Appendix A discusses the limitations of our framework.
- Appendix B gives some additional qualitative examples from ECHO.
- Appendix D provides the prompts for, and some additional information on, the data collection pipeline.
- Appendix E provides the prompts used for automatic evaluation with VLM-as-a-judge.
- Appendix C discusses the results and process used for our human validation of VLM-as-a-judge.
- Appendix F discusses the use of LLMs in the preparation of this manuscript.

A LIMITATIONS

While ECHO is diverse and markedly distinct from prior benchmarks, it may not be representative of *all* possible user queries. First, there is a bias towards certain topics; for example there is an unusually large number of requests for "*Ghibli style*" due to social media trends. Second, users are more likely to post examples where 40 Image Gen succeeds rather than fails, which affects the distribution of tested capabilities. However, these quirks are inherent to crowdsourced datasets; DiffusionDB (Wang et al., 2022) is similarly biased towards "*artstation style*" and keyword lists favorable towards Stable Diffusion. As such, these benchmarks should be viewed as comparisons to the current best model in the community consciousness, rather than arbiters of the "universally best" model for any user query. For this reason, we present not only a benchmark but also a reproducible framework, which can be re-run as soon as a new model with new capabilities is released, or as soon as community interests change.

Figure B.1: **Per-Model Average Color Histogram.** For each model, we compute the average color histogram of its outputs on the image-to-image split (top), then overlay it on top of a real image as a visual aid (bottom). Evidently, 40 Image Gen exhibits a substantial yellow tint.



Figure B.2: **Qualitative Comparison of Identity and Spatial Shift.** Given the prompt "Billy the Kid cleaned up and colorized from the famous photo of him," each model retains the input identity to varying degrees. For the prompt "giving it a fresh twist with a more detailed, realistic touch" each model retains the input image's spatial layout by a different amount.

B ADDITIONAL EXAMPLES

Examples Illustrating Specialized Metrics. In Figure B.1 and Figure B.2 we display examples illustrating the range of drift in color, identity, and spatial structure across different methods.

Qualitative Examples from ECHO. In Figure B.3, Figure B.4, Figure B.5, Figure B.6 we highlight further qualitative examples surfaced through the ECHO framework. These examples demonstrate the breadth of tasks that naturally arise from community use of current image generation models, going beyond traditional templated image editing tasks and crowdsourced prompts centered around Stable Diffusion.

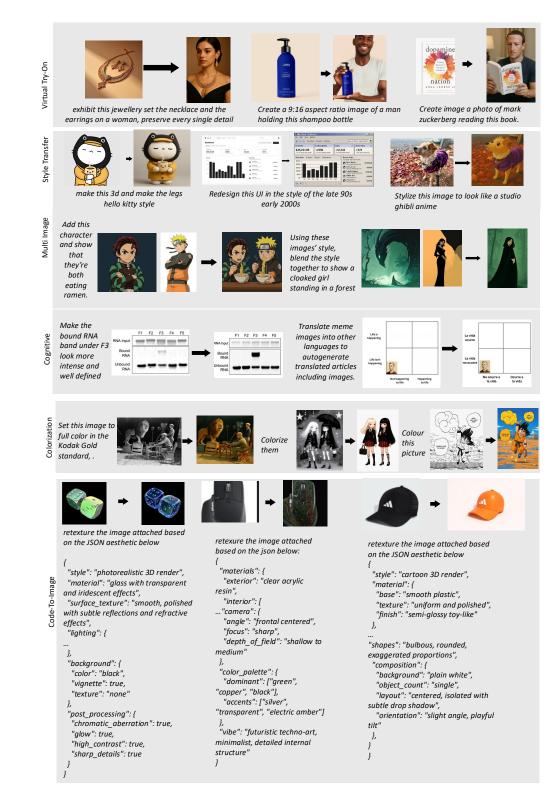


Figure B.3: Image-to-image examples from ECHO.

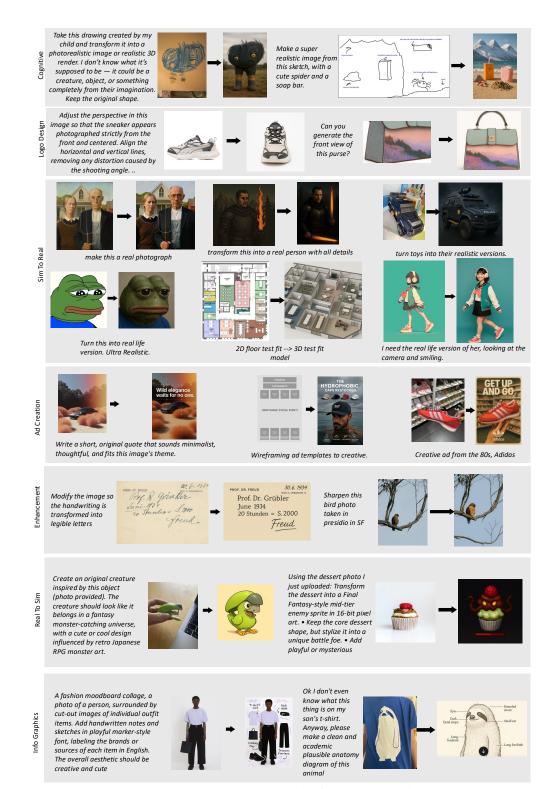


Figure B.4: Image-to-image examples from ECHO, continued.





870

871

872

873

874 875

876

877

878

879

880

882 883 884

885

887

888

889

890

891

892

893

894

895

897

899

900

901

902

903

904

905

906

907

908

909

910

911

912 913

914

915

916

917

"Make an image of a calculator app for the calculation



Create a digital artwork featuring an arrangement of black keyboard keycaps forming the shape of a dragon. Use keycaps with white letters and some additional symbols (like @, #, *, %, +) to complete the design... The shape should be easily recognizable and arranged either in portrait or landscape mode depending on the dragon's natural orientation.



a real photograph of a whiteboard solving the integral of 4x and showing all steps to get to answer



writing the prompt I'm writing now to ChatGPT 4o and receiving the screenshot I asked for



Create a solved "triple maze" with three entrances through the top (labeled Red Start, Blue Start, Green Start) with colored, dotted lines for each path and three exits (also labeled) through three corresponding openings at the bottom.



Chatgpt drawing a Chatgpt drawina a Chatapt .



Generate me a photorealistic iPhone picture of a \$277.02 wrinkled receipt on a wooden table with reasonable numbers. Make the math add up. The address should be Y



"How a bat sees the world"

"instruction": "Generate an image of a high quality product render of a CHANEL designed for brand presentation. It should prominently display the brand name as a central logo and use a color palette that represents the brand. The image should be clean, realistic, and suitable for e-commerce or promotional

```
"image_style": {
"type": "product",
"material": {
    "primary_surface": "cotton or canvas",
"finish": "matte",
     "color_profile": {
    "base_color": "match brand identity",
       "secondary_tones": ["complementary to
brand"]
      .
"panel lines":{
       "material": "stitched fabric",
"visual_treatment": "detailed stitching"
   "lighting": {
     "tvne": "studio".
     "key_light": {
    "position": "top-front",
        'effect": "highlight form and texture"
    "reflections": {
  "character": "subtle glossy bounce"
```

.. "shadows": "soft, layered, directional with slight floor gradient" "color": "contrasting gradient (light grey to dark, with a soft spot light behind "style": "minimal with a faint branded pattern or diagonal texture "composition": {.. "style": "modern, clean, transparent with subtle alow'

"opacity": "60%"

tone": "bold and dynamic",

}, "visual style":{

energetic"

"inspiration": "premium sportswear ads", "aesthetic": "high-contrast, sharp,



Design a 3D miniature store shaped like a giant Starbucks® iced coffee cup, with a glowing neon "Sip & Chill" sign above the entrance. Incorporate tiny human figures enjoying the store, soft clay textures, and a pastel mint and cream color scheme. The scene should exude a whimsical and playful atmosphere, viewed isometrically with high detail. capturing the essence of the Starbucks®



A surreal and minimalist brand logo design, where the brand's emblem is transformed into a fully mechanical object—its exact shape preserved and rendered with anatomical precision using hyper-polished, chrome-like metallic components.

The logo should maintain full fidelity to the original design: ensure that all lines, curves, and proportions are rendered accurately and vividly, without being cropped, altered, or distorted in any way.

Beneath the logo, elegant serif typography presents the brand name ("[GPT Breeze]") and a refined, poetic slogan ("[ChatGPT AI shortcuts in your current tab]"), both centered and minimal. The entire composition feels ethereal, luxurious, and visionary—ideal for a future-forward, design-conscious brand.



Generate a square-format campaign image by reimagining a specific product from [Dyson] as if it were originally invented and manufactured in [Germany]. Go beyond surface-level decoration—redesign the core shape, structure, and materials of the product using that country's traditional techniques. materials, and aesthetic principles.



3D render of [BAG TYPE] by [BRAND NAME], glowing brand name above Neon lighting in [color] and [color], cinematic shadows, ultra-detailed textures, reflective surface, dark minimal backdrop, HD product photography style, sharp, eye catching, made for high-end marketing.



Create a high-resolution illustration of the word "METAL" in the style of sharp-edged heavy metal logos. Use jagged, aggressive letterforms with pointed extensions and torn, asymmetrical outlines. Apply a metallic chrome texture with icy blue gradients and bright white highlights to simulate a reflective surface. Add thick black shadows behind each letter to enhance depth and legibility. The overall style should look danaerous and cold, like frozen steel shards. Only the stylized text should appear, with no additional elements or borders. Center the word on a solid black background. Square aspect ratio.



A high-resolution, studio-lit macro photograph of a pastry shaped like a tech company logo, with a partial bite taken out, placed on a neutral matte surface with visible crumbs and soft shadows, highlighting texture and detail.

Figure B.5: Text-to-image examples from ECHO.



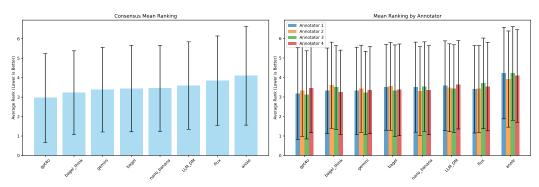


Figure C.1: The consensus ranking, and mean ranking by annotator for each of the models. As we can see, because of the limited size of the annotation sets, the standard deviations of the bars is quite high, meaning that we can draw very few conclusions about model performance overall from the human data.

Table C.1: Significant model differences from human evaluation. We can see that even from our relatively limited human evaluation, anole and LLM_DM under-perform most models, primarily due to the image-editing split, where both perform quite poorly.

| Model A | Model B | Z-Statistic | P-Value (raw) | P-Value (Bonf.) | Signficance |
|---------|-------------|-------------|---------------|-----------------|-------------|
| anole | gpt4o | 7.429 | 0.000000 | 0.000000 | *** |
| anole | nano_banana | 6.218 | 0.000000 | 0.000000 | *** |
| anole | flux | 6.175 | 0.000000 | 0.000000 | *** |
| anole | bagel_think | 6.127 | 0.000000 | 0.000000 | *** |
| anole | gemini | 5.755 | 0.000000 | 0.000000 | *** |
| anole | bagel | 4.377 | 0.000012 | 0.000337 | *** |
| LLM DM | anole | 4.011 | 0.000061 | 0.001694 | ** |
| LLM DM | gpt4o | 3.418 | 0.000631 | 0.017675 | * |
| bagel | gpt4o | 3.052 | 0.002274 | 0.063676 | - |
| LLM DM | nano banana | 2.207 | 0.027314 | 0.764788 | _ |
| LLM DM | flux | 2.164 | 0.030461 | 0.852901 | - |
| LLM DM | bagel think | 2.117 | 0.034287 | 0.960036 | - |

C HUMAN RANKING & CORRELATION WITH LLMS

To evaluate the performance of our LLM as a judge models, we performed a limited human evaluation using five expert raters in our group. Each rater fully ranked each of the 8 models over 200 samples (100 from the text-only split, and 100 from the interleaved split), flagging any samples that were impossible to rank fairly. Figure C.1 shows the aggregate of the rankings for each model.

While the number of annotations is somewhat low for determining model performance, we wanted to understand if the samples that we collected (200) could show significant results in terms of model ordering. To do so, we first ran a Friedman Test on the rankings, and found that with p < 0.001 there was a significant difference between the means of the rankings. To determine which pairs are actually significant, we further performed a Dunn's test for significant pairwise differences, and found that after Bonferroni correction, only 8/28 model pairs were significant, shown in Table C.1.

To compute annotator-LLM agreement, we first constructed a consensus ranking for the human raters using the Kemeny-Young method (Andrieu et al., 2023). The split rankings were then merged, giving a total of 200 samples. The LLM as a judge methods produce a single floating point score for each sample. In order to compare the methods, we construct a ranking for each LLM judge from these scores, breaking ties randomly. We then computed Kendall's τ_b with each of the LLM judges, giving us the presented results in Section 5, GPT: $\tau_b = 0.117_{p=0.0036}$, Gemini: $\tau_b = 0.083_{p=0.0199}$, Qwen: $\tau_b = 0.045_{p=0.1327}$. We notice here that while GPT and Gemini both have weak, but significant correlations, Qwen does not correlate significantly with human judgment across the raters, and is thus, unlikely to serve as a strong judge for human performance. Interestingly, however, the LLMs correlate with each other. We computed the pearson-r correlation between the scores of pairs of annotators: GPT \leftrightarrow Gemini: $r=0.575_{p=0}$, Gemini \leftrightarrow Qwen: $r=0.627_{p=0}$, GPT \leftrightarrow Qwen: $r=0.480_{p=0}$.

Another interesting finding is that the Kendall's tau-b for each of our raters differed dramatically. Figure C.2 shows the correlation between each of the human raters, and each of the LLM judges independently

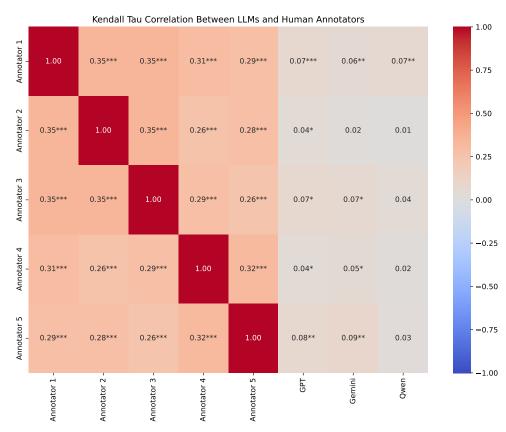


Figure C.2: Kendall's τ_b for pairs of individual human raters and LLM judges (without consensus ranking). We can see that human annotators have fairly high inter-rater correlation, while LLM judges have slight positive correlations, with most correlations signficant among them. In the figure, $*** \rightarrow p < 0.001$, $** \rightarrow p < 0.05$.

(no consensus ranking). We can see that while three of our raters (graduate students on the project) have high inter-annotator correlation, two other raters (undergraduates on the project) have notably different preferences, some of which correlate better with models than others.

Table D.1: Full sample metadata before and after processing with ECHO.

| Raw Retrieved Fields | | | | |
|-----------------------|--|--|--|--|
| text | Post text content. | | | |
| timestamp | Posting time of the tweet. | | | |
| replies_above | Context tweets obtained by scrolling upward in the thread. | | | |
| keyword | Search keyword used to retrieve the post. | | | |
| url | Direct URL of the post. | | | |
| author | Username of the post author. | | | |
| image_urls | List of image URLs with associated ALT text. | | | |
| replies_below | Replies obtained by scrolling downward in the thread. | | | |
| engagement | Engagement statistics of the post, including likes/views/reposts/bookmarks. | | | |
| Post-Processed Fields | | | | |
| prompt | User instruction or query text extracted from the original post. | | | |
| prompt_modified | Boolean flag indicating whether the prompt was manually edited during cleaning. | | | |
| post_urls | List of original tweet URLs from which the content was collected. | | | |
| quality | Label describing the intended use of the sample (e.g., "Benchmark"). | | | |
| community_feedback | List of replies or comments from other users, each stored with its post URL and feedback text. | | | |
| id | Unique identifier for the sample, corresponding to the primary post ID. | | | |
| images | Local filenames of all downloaded images associated with the sample. | | | |
| images_to_posts | Mapping from each image file to the URL of the tweet it originated from. | | | |

D DATA COLLECTION PIPELINE

Dataset Fields. In Table D.1 we display the full set of metadata associated with each sample after running the entire ECHO framework.

Data Collection and Processing. We discuss the design of keywords for querying posts in Figure D.1. We also display the prompts used for each step of data processing in the ECHO framework, including relevance filtering (Figure D.2), extracting trees into samples (Figure D.3), multimodal processing (Figure D.4), and parsing screenshots of conversations (Figure D.5).

1154

1155

1156

1157

1158

1185

1186

```
1135
             Keywords for Querying Posts
1136
1137
             Broader Terms
1138
               • "40"
1139
               • "gpt"
1140
               • "gpt-4o"
1141
               • "openai"
1142
               • "chatgpt"
1143
             Narrower Terms
1144
               • "create image"
1145
               · "gpt image"
1146
               • "4o image"
1147
               • "prompt share"
1148
               • "gpt prompt" and "4o prompt"
1149
               • "create image gpt"
1150
               • "image gen"
1151
               • "生成"
1152
               • "画像"
1153
```

Figure D.1: Keywords used to query posts, described in Section 3.1. For the initial two weeks following the 4o Image Gen release we favor volume: we query more generic terms, over daily intervals. In later weeks we favor precision: we query more targeted terms often used for sharing image generation results, over weekly intervals. To increase coverage in foreign languages, we also use calligraphic keywords applicable to Chinese and Japanese, while the alphanumeric keywords are sufficient to also cover Romance languages like Spanish and French.

```
Instructions
1163
1164
1165
                You are a helpful assistant that evaluates the relevance of Twitter posts to OpenAI's GPT-40 image-
                generation feature. The goal is to assign a relevance score to each tweet.
1166
               Scoring Scale
1167
                  1 - Definitely trash
                Contains "40" or "gpt" only by coincidence and has no relation to image generation (e.g., political
1168
               Contains "40" or gpt c..., -2 commentary, education topics).

- 2 - Very likely irrelevant

Mentions "40" or "gpt" but clearly not about generating or editing images (e.g., "40" as slang, or
1169
1170
1171
                Could plausibly refer to GPT-40 image generation but lacks clear indicators (e.g., "This is insane..." or
                 mild excitement without explicit "image" context).
1172
                 4 - Very likely relevant
1173
               Contains clear prompt-like language or references to creating or sharing images (e.g., "turn myself into
                              "prompt share!",
                                                 "new prompt").
1174
                - 5 - Certainly relevant
               Explicitly about using GPT-4o for image generation or editing, often including sample prompts or direct
1175
               praise (e.g., "GPT-40 image gen is amazing!", "tried this with GPT-40 image gen, prompt: ...").
1176
               Prompt for X data cleaning
1177
               - Determine which level best matches the content.
- Output exactly one JSON object with a "score" field set to an integer 1-5.
1178
1179
                - If you choose 3, you may optionally add a "note" field (one sentence) explaining the uncertainty.
1180
                Input Example
1181
                Output Example (Score Only)
1182
1183
1184
```

Figure D.2: Prompt for relevance filtering after raw data collection with GPT-4o (OpenAI, 2024), described in Section 3.1.

1236

1237

```
1190
1191
1192
                 Instructions
1193
1194
                  You are an extractor of multimodal prompts for image generation.
1195
                 You will be given a JSON that represents a Twitter post and its reply tree. Each post in the tree may
1196
                  contain an image generation prompt; your job is to extract them into unique samples.
1197
                 For every input, try to extract at least one sample rather than returning an empty list. We want to
                 extract as many samples as possible, and use a quality score for filtering. Please output a JSON list of samples in the format '''json [...]'''.
1198
1199
                  ## Post to Prompt
                 Each sample should include the following keys:
1200
                  {"prompt": <str>, "prompt_modified": <bool>, "post_urls": <list of strs>}
1201
1202
                   Identify each post that discusses a unique image generation task. Set "prompt" as the post text that
                 describes this task. Be very broad in the definition of "prompt"; any instruction, description, comment,
1203
                 or question that hints at an image generation task is fine.
                  - Make a new sample for every new prompt, even if it is a slight modification of another sample's prompt. - Try to extract the prompt from the post text exactly, without modification. You may modify the prompt
1204
1205
                 when the modification is obvious, for example, piecing together text from multiple posts or filling in placeholder text. Set the flag "prompt_modified" to True or False accordingly.
1206
                   Omissions of text should not be considered as modifications; you should omit statements that are
1207
                 obviously not part of the prompt.
                   Many main posts say something like "Prompt Below" or "Prompt in Next Comment"; this means that the tree
1208
                   is likely to have a really good sample, and the prompt needs to be found in the replies.
1209
                 To determine "post urls":
                  - For each "prompt", set "post_urls" to the urls of posts in the tree that likely contain images that are
1210
                   related inputs or outputs for that prompt, which you can determine from the post text.
1211
                   Order "post_urls" by importance; the first url should contain the main task information.

    Many replies use a similar prompt as the root post and attach an output image. These should be grouped in the "post_urls" of the main post. Try to infer if this is happening from the reply text.
    If the reply's text indicates a new task, it should be a new sample. If the reply's text indicates it

1212
1213
                 is irrelevant to image generation, it should be omitted. If the reply contains no text and an image, it should be included in "post_urls" so that it can be further processed later.
1214
                  - Each url/post should appear at most once; images should not be duplicated across samples.
1215
                  ## Quality Score
1216
                 Each sample should also classify the prompt quality: {"quality": <str>}
1217
1218
                 - Classify the guality as one of the following categories: ["Benchmark", "Analysis", "Trash"].
1219
                  - "Benchmark" are the highest quality prompts, which instruct a single coherent image generation task,
                 that can be used for benchmarking. Be fairly strict about the quality.

- "Analysis" are moderate quality prompts that are not in "prompt" format, which are often comments or
1220
                 questions relevant to image generators but do not query a specific task, and are still usable for
1221
1222
                    "ar{	ext{r}}rash" are low quality prompts that have no clear task or are clearly irrelevant. Our focus is on
                 OpenAI's gpt-image-1 or 40 image generation; if the post clearly uses another model or platform like DALL-
E, Stable Diffusion, some video generator, etc. it should be classified as "Trash".

- Make sure to collect as many "Analysis" samples as possible, while maintaining relevancy. For these
1223
1224
                 samples, set "prompt" to be the relevant text or commentary about image generation.
1225
                  ## Community Feedback
1226
                 Each sample should contain a list of community feedback:
                  {"community feedback": [{"post url": <str>, "feedback": <str>}, ...]}
1227
                 To extract "community_feedback":
- For each post in the tree, determine whether it discusses the sample's success / quality (e.g., "really
1228
1229
                   cool", "does not work", etc.).
                  - If a post obviously does not have feedback, do not include it.
1230
                 - The feedback may come from the main post's author or from other authors in the replies.
                    Include the full feedback text without modification such that there is sufficient context, but also
1231
                 omit obviously irrelevant text.
                  - Each url/post should appear at most once in the "community_feedback"; feedback should not be duplicated
1232
                   across samples.
1233
                  json_post_tree: <tree>
1234
                  extracted:
1235
```

Figure D.3: Prompt for tree-to-sample extraction with GPT-40 (OpenAI, 2024), described in Section 3.2.

```
1243
1244
1245
                Instructions
1246
1247
1248
                You are an extractor of multimodal prompts for image generation.
1249
                Your job is to process input-output image pairs from raw user prompts for image generation collected from
1250
                You will be given a prompt, a dictionary mapping image ids to images, and a dictionary mapping image ids
1251
                Please output a JSON list of samples in the format '''json [...]'''.
1252
1253
                Each sample should include the following keys, which categorize images as inputs or outputs:
{"inputs": <list of ids>, "outputs": <list of ids>, "post_urls": <list of strs>}
1254
1255
                To classify "inputs" and "outputs":
                - Inputs, combined with the prompt, should produce a fully specified and coherent image generation task.
1256
                - Outputs should be plausible results given the inputs and prompt.
1257
                 - You may encounter tasks like text-to-image generation (no inputs), image editing (one input), or multi-
                image conditioned generation (multiple inputs).

- Set "post_urls" the list of urls associated with the inputs and outputs. Order "post_urls" by
1258
                importance; the first url should contain the main task information.
1259
1260
                General rules:
                - Each category is mutually exclusive. Each image should not be assigned to multiple categories.
1261
                - Some images are low quality and irrelevant to any task. They should not be assigned to any category.
                - Some samples are low quality where it is not possible to extract any coherent task. Simply return an
1262
                empty dictionary { }.
                  If there are no relevant images, assign an empty list [] to the category.
1263
                - If an image is duplicated, use the smaller index as the id and ignore the others.
1264
                - Each id should appear at most once. Each post_url should appear at most once.
1265
                ## Fill in the Blank
                The input prompt may be a "fill in the blank" case with placeholders. Infer these placeholders and update
1266
                the following keys:
{"prompt": <str>, "prompt_fill_blank": <bool>}
1267
1268
                To update the prompt if it is "fill in the blank":
- If the prompt is not "fill in the blank", which should happy the majority of the time, you should by
1269
                default copy the input prompt exactly and set "prompt_fill_blank" to False
                - Otherwise update the prompt and update the flag "prompt_fill_blank" to True.
1270
                - Often fill in the blank prompts include brackets of the form "[keyword]"
1271
                - Often you can infer the right keyword to replace the placeholder using the output images.
                - Often you will generate multiple infilled prompts, because there are often multiple output images that
1272
                represent different instantiations with different sets of keywords.
               - Only fill in the blank only when it makes sense to do so, and when you are fairly confident about what the keyword should be. Otherwise, if highly uncertain, don't "fill in the blank".
1273
                  You should make a new sample for each new instantiation of the "fill in the blank". If there are
1274
                multiple outputs that infill with different keywords, you should create multiple samples.
1275
                ## Screenshots of Conversations
1276
                For special images that show a screenshot of a conversation with the image generator, mark their image id:
1277
                {"conversation": <id>}
1278
                To extract a "conversation":
1279
                - For each conversation, you should create a new sample that represents the task expressed in the
                conversation.
1280
                  If there exist multiple images showing screenshots of the same conversation, select the main one
                showing the most task information and omit the others.

- Combined related samples and their fields like "inputs", "outputs", "post_urls", "prompt" to minimize
1281
1282
                - A conversation is defined as a screenshot that shows a conversation (which may involve a prompt and image(s)) in OpenAI's ChatGPT window.
1283
                - If the image shows any other platform, it is not a conversation.
1284
                - If the image generation task is not clear (e.g., the screenshot seems to be using ChatGPT's LLM rather than image generation capabilities, the screenshot is extremely low quality, the images are extremely
1285
                small), it is also not a conversation.
1286
                - If the sample does not contain a conversation, set "conversation" to the empty string "".
1287
                prompt: prompt>
1288
                images: <images>
                images_to_posts: <images_to_posts>
1289
                extracted.
1290
```

Figure D.4: Prompt for multimodal processing with GPT-40 (OpenAI, 2024), described in Section 3.3.

```
1309
1310
1311
                Instructions
1312
1313
                You are an extractor of multimodal prompts for image generation.
1314
                Your job is to extract the text prompt and bounding boxes of individual images from screenshots of
1315
                conversations with an image generator.
                 You will also be provided relevant text that may be helpful for determining the input and output images
1316
1317
                Please output only a valid JSON dictionary according to this schema:
1318
                  "json {"prompt": <str>, "inputs": <list of bounding boxes>, "outputs": <list of bounding boxes>}
1319
                - If there is text, run OCR and extract the raw text input by the user exactly.
1320
                 - The extracted text should produce a fully specified and coherent image generation task; ignore other
1321
                - If there is no relevant text, output the empty string "".
1322
                To extract "inputs" and "outputs":

- Extract a list of bounding boxes for every individual image.

- Each bounding box should be formatted as [x1, y1, x2, y2]; (x1, y1) is the top-left and (x2, y2) is the
1323
1324
1325
                 - Also sort bounding boxes as "inputs" vs. "outputs" of the image generator.
                - The extracted images should produce a fully specified and coherent image generation task; ignore other
1326
                irrelevant images.
                - Each image should only appear once. Ignore exact duplicates.
- If there are no "inputs" output an empty list [].
- If there are no "outputs" output an empty list [].
1327
1328
1329
                relevant_text: <relevant_text>
1330
                images: <images>
                extracted:
1331
1332
```

Figure D.5: Prompt for parsing screenshots of conversations with Qwen-2.5-VL (Bai et al., 2025), described in Section 3.3.

E AUTOMATIC EVALUATION METRICS

1350

1351 1352

1353

1354

1355 1356

1357

1358

1380

1381

1384

1385 1386

1387

1388

1389

1390

1391

1392

1393

1394

1395

1396

1398 1399

1400

1401 1402 1403 **Overall Metrics.** In Figure E.1 we depict the prompt used for VLM-as-a-judge in our overall benchmark evaluation. We follow the "single answer grading setup" of MT-Bench (Zheng et al., 2023), and convert absolute scores into pseudo pairwise comparisons, which can be used to compute the win rate.

Specialized Metrics. We display our prompt for rating the accuracy of rendered text in Figure E.2, and classifying the applicability of each sample to each specialized metric in Figure E.3.

```
1359
              Instructions
1360
              Please act as an impartial judge and evaluate the quality of the image output produced by an image
              generation model in response to an input instruction (expressed via text and/or image(s)).
1362
1363
              Begin your evaluation by forming your own expectation of what a good output should look like for the
              given prompt. Describe this briefly before judging the output.
1364
              Then compare the model's output with your expectation. Point out errors, inaccuracies, or failures to
1365
               follow the instruction, and identify missing details that would make the output better satisfy the
1366
              instruction.
1367
              Make sure to consider the following factors equally:
               - **Prompt Following**: Does the output interpret the text correctly and execute the requested task
1368
              accurately?
1369
               - **Reference Fidelity**: Does the output preserve key details from the input images when relevant?
               - **Realism and Aesthetics**: Is the output photorealistic (e.g., accurate anatomy, no artifacts, etc.)
1370
              and visually appealing (e.g., balanced colors, well-framed composition, etc.) when relevant?
1371
              After providing your explanation, please rate the output on a scale of 1 to 10 by strictly following this
               format: "[[rating]]", for example: "Rating: [[5]]".
1372
1373
              <|The Start of Input Instruction|>
              input_prompt: <input_prompt>
input_images: <input_images>
1374
              <|The End of Input Instruction|>
1375
              <|The Start of Model Output|>
1376
              output_image: <output_image>
1377
              <|The End of Model Output|>
1378
```

Figure E.1: Prompt for automatic evaluation with GPT-40 (OpenAI, 2024), Gemini 2.0 (Team et al., 2023), and Qwen2.5-VL-32B-Instruct (Bai et al., 2025), described in Section 5. Our prompt closely follows the format from MT-Bench (Zheng et al., 2023), but adapted for rating image generation outputs.

```
Instructions
Check if all text in the image is accurate and readable.
For exact copy requests: spelling, punctuation, grammar match exactly, with no missing or extra
characters, and text is not cropped.
For created text: content is coherent, relevant, and fits the available space and design.
Begin your evaluation by reading through the image and OCR the text
Point out spelling errors, punctuation errors, grammar errors, and missing characters of the text.
Point out if the text is cropped.
Then, look at the image again and check if the text is coherent, relevant, and fits the available space
Please rate the output on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example:
 "Rating: [[5]]'
<|The Start of Input Instruction|>
input prompt: <input prompt
<|The End of Input Instruction|
<|The Start of Model Output|>
output_image: <output_image>
<|The End of Model Output|>
```

Figure E.2: Prompt for judging text rendering accuracy with GPT-40 (OpenAI, 2024), described in Section 5.2.

```
1405
               Instructions
1406
1407
                 For each metric in the provided list, decide if it is applicable for the given image generation
1408
                  The instruction can be defined with text and/or images. Some instructions may contain no input images.
1409
                  If is a metric is marked as applicable, it will be used as an axis to score and rank outputs for the
               given input
1410
                  ## Metric List
1411
                  <metric_list>
1412
                  ## Output Format
1413
                 Respond only with a JSON dictionary containing all the metric names as keys, and the value 0 (is not
               applicable) or 1 (is applicable).
1414
                 Also include a short global rationale for your overall decision-making process.
1415
                  ```json
1416
 "<metric1>": <integer 0 or 1>,
 "<metric2>": <integer 0 or 1>,
1417
 "rationale": "<a short rationale, 20 words or less>"
"prompt": "<the input prompt repeated again>"
1418
1419
1420
 ## Your Turn
1421
 task: <task>
1422
 input prompt: <input prompt>
 input_images: <input_images>
1423
1424
 The <metric_list> is replaced with the defined metrics name, its description, and its applicability:
1425
 Name: "Face Identity Preservation"
1426
 Description:

Check if the person's identity matches the reference or intended person, keeping facial structure and
1427
 distinctive features the same.
 Examples to Penalize: Changes in hairstyle, beard length, scars, facial expression, accessories, etc.
1428
 that do not match the reference.
 Applicability:
1429
 This metric is often applicable, especially for tasks involving subject-driven generation.
1430
 However, it is not applicable when:
 - The prompt does not explicitly or implicitly request face identity preservation.
- No person's face is visible (because there are no people, or faces are occluded).
1431
 - The task is stylization, where the creative freedom allows for many valid outputs and correctness is
1432
 too subjective.
1433
1434
 Name: "No Color Shift"
 Description:
1435
 Check if the overall color tone, brightness, and contrast match the reference or intended look.
 Examples to Penalize: Added yellow tint, overexposure, or darkening compared to the reference.
1436
 Applicability:
 This metric is often applicable, especially for tasks like local editing.
1437
 However, it is not applicable when:
1438
 - The task is colorization or image-to-image translation, where color change is inherent to the task.
1439
 Name: "Spatial Position Preservation"
1440
 Description:
 Check if the structure and spatial layout of the reference are copied correctly, including positions,
1441
 relative layout, and scale of key objects.
1442
 Examples to Penalize: A dog is slightly moved from its original position during stylization; a table
 that was centered is shifted.
1443
 Applicability:
 .
This metric is only applicable for tasks that involving image-to-image translation, stylization, or
1444
 local editing that requires strict structure preservation. However, it is not applicable when:
1445
 - The prompt does not expect the resulting image to be strictly preserving spatial structure with the
1446
 reference image
 - The prompt can allow some structure changes (eg, sketch-to-image, 2D-to-3D stylization)
1447
1448
 Name: "Text Rendering Accuracy"
 Description:
1449
 Check if rendered text contains mistakes that hinder readability.
1450
 Examples to Penalize: Characters are garbled; there are missing or extra characters; there is incorrect
 spelling or punctuation; there is incorrect grammar.
1451
 Applicability:
 This metric is often applicable, but only when the prompt explicitly requests rendered text.
1452
1453
```

Figure E.3: Prompt for classifying the set of applicable samples for each specialized metric with GPT-40 (OpenAI, 2024), described in Section 5.2. The number of applicable samples is as follows: Face Identity Preservation (244), No Color Shift (271), Spatial Position Preservation (180), Text Rendering (240).

1455

1456

## F LLM DISCLOSURE

Some portions of this work were generated with the assistance of large language models (LLMs). Their primary role was to support editing, rephrasing, and formatting of existing text to improve clarity and readability. While human authors created and reviewed the core content, LLMs were used as a tool to streamline refinement and presentation. All factual information, analysis, and conclusions remain the responsibility of the authors, and every effort has been made to ensure accuracy and integrity.