

A Appendix

A.1 Broader impact statement

Our work offers a new path to improve text-to-image generation, but this improvement is not without possible social impacts. Text-to-image models can be used to create harmful or misleading content, and improving their output can increase this risk.

Moreover, our work relies on the abundance of community-created, fine-tuned specialized models and adapters. These are rarely developed with safety in mind, and do not typically undergo red team assessments. Hence, they may increase the risk of the user generating biased or unsafe content. However, this can be mitigated by carefully curating the generative models seen during training, or by black-listing specific models in the output flow strings.

Future work may be able to further refine the reward model used at training to also align it with safety, for example by reducing the score for content deemed unsafe by a detector.

A.2 ComfyUI Overview

ComfyUI is a popular (77,300 stars on GitHub at the time of this writing), open-source workflow engine designed for flexible and extensible automation of generative AI tasks. Its node-based interface allows users to visually construct and execute complex processing pipelines, with users across the community often implementing and sharing new nodes to accommodate the changing landscape of generative tools. The pipelines constructed in ComfyUI can be exported to a JSON format, which we then map to a more compact representation and use for both our training data and our LLM output representation. To run our generated workflows, we convert them back to the JSON format and run them through the ComfyUI API. A dedicated user could also load these workflows through the UI and further manually refine them.

A.3 Additional Qualitative results

Here, we give more qualitative examples of FlowRL generations.

Figures 6, 7 and ?? provide additional generations of CivitAI prompts using FlowRL. We give a detailed list of the relevant prompt (ordered by appearance order, from top-left to bottom right)

We provide additional qualitative comparisons between FlowRL and the baselines in Figure 8 for both CivitAI prompts and GenEval prompts.

In addition in figure 9 we give a qualitative comparison between FlowRL with and w/o the usage of the dual model guidance mechanism (CFG).

List of prompts for example generations

1. "Amazing detailed photography of a cute adorable samurai kitten holding Katana with 2 paws, Cherry Blossom Tree petals floating in air, high resolution, piercing eyes, lifelike fur, Anti-Aliasing, FXAA, De-Noise, Post-Production, SFX, insanely detailed & intricate, hypermaximalist, elegant, ornate, hyper realistic, super detailed, noir coloration, serene, 16k resolution, full body"
2. "masterpiece, best quality, high quality, intricate, absurdres, very aesthetic, no humans, landscape, outdoors, mountain tops, wind, windy, wind lines, clouds, above clouds, cliff, wind magic, aurora, ultra wide angle shot, cinematic style, highly detailed, extremely detailed, sharp detail, majestic, shallow depth of field, movie still, soft light, circular polarizer, colorful, wallpaper, professional illustration, anime"
3. "pixar style of turtle, as a pixar character, tinny cute, luminous, wearing hawaiian hat, at the sea shore, tropical beach, smile, high detailed, photorealistic, 8k"
4. "Medieval German castle, surrounded by mountains, high fantasy, epic, digital art."
5. "style of Edvard Munch, Piercing, sagacious eyes, mirage-like, the Sandswept dreamdweller, a trickster of dunes, clad in a wind-whispered turban, eternally smirking, sandswagging over a dune-freckled miragepath in an ancient zephyr-twisted cactidle wilderness of towering

- dustfrond phantasmagorias, paying no heed to the sun-scorched skyripples above, Arid, Sand-whirled, Mirage, Cacti, Mystical Desert, oasis illusions. Edvard Munch style"
6. "full body, Fat cats at Elrond's council from the movie Lord of the Rings, fluffy paws, background action-packed"
 7. "detailed, vector art, thick lines, oil painting, vibrant, colorful, candy pink, scarlet red, orange, smooth coloring, nature, landscape, stone pillars, long wild trees, moody streaks sky, natural lighting, river, reflections, best composition, background"
 8. a woman with red hair and a white shirt is shown in this painting style photo with a pink background, Charlie Bowater, stanley artgerm lau, a painting, fantasy art masterpiece, best quality, depth of field, backlighting, intricate details"
 9. "cinematic shot of stone giant walking in lush forest, dappled sunlight, high resolution"
 10. "Majestic jagged rocky mountains, red mesas, wind eroded colorful rock formations, twilight, starry night, petrified forest national park, arizona, astrophotography"
 11. "Cubist inspiration, A landscape represented with planes and flat colors. The landscape could show a field, forest or city, and flat planes and colors could be used to create a sense of depth and perspective, surrealism, aesthetic, bold gorgeous colours, high definition, super clear resolution, iridescent watercolor ink, acid influence, fantastic view, crisp quality, complex background, medium: old film grain, tetradic colors, golden hour, rust style, vantablack aura, golden ratio, rule of thirds, cinematic lighting Dark realism and magical. Complementary poisonous colors with deep zoom Memphis style abstract bokeh background with deep zoom"
 12. "FrostedStyle Highly detailed Dynamic shot of a transparent frosted ruby reindeer, glowing with rage from within extremely detailed"
 13. "vertical symmetry, vntblk, movie poster art, blood moon, red moon, darkest night, stone-henge, low angle:famous artwork by caspar david friedrich and stephan martinieri, perfectly round scifi portal, ominous dark surreal and unique landscape with towering obelisks piercing the sky, glowing ornate lovecraftian artifact, jagged rock formations, night sky, mysterious, ethereal, deserted, dark corners, burgundy, anthrazit grey, crimson, sunset orange, yellow, teal:16, ultra detailed"
 14. "by Peter Holme III and Roger Dean and Vitaly Golovatyuk and Mark Lovett, cinematic, shallow depth of field"
 15. "grainy, extremely detailed, intricate detail, dynamic lighting, photorealistic, filmg, natural lighting, low light, cat, slime, red glowing eyes, :P, fluffy, hairy, fluff, glowing stripes, raining, wet, dark theme, open mouth, lot of teeth, abyss, lurking in shadow"
 16. "The art of Origami, Paper folding, Swan on a lake, Amazing colours, Intricate details, Painstaking Attention to Details, UHD"
 17. "amateur analog photo, The creature monster brown fur Easter bunny character covered in yeast, evil, creepy, in dark forest, fine textures, high quality textures of materials, volumetric textures, natural textures"
 18. "In a wondrously gleaming futuristic realm composed entirely of ripe peaches, a towering palace made of glistening peach flesh and pitted stone stands as the focal point of the image. The palace's walls are adorned with intricate carvings of peach vines and blossoms, while peach juice flows like streams through the city streets. This vivid and surreal painting captures the ethereal beauty of a world where nature and architecture are seamlessly intertwined, every detail rendered with unparalleled precision and depth, making viewers feel as if they could reach out and touch the succulent fruit structures."
 19. "high-contrast palette, cinematic quality, fashion photography, chimp wearing a black suit with a black shirt with a black vest with a black necktie with black Rayban style sunglasses, natural skin texture, realistic skin texture, skin pores, skin oils"
 20. "faistyle, retro artstyle, painting medium, lake, mountain, forest"
 21. "close up Portrait photo of muscular bearded guy in a worn mech suit, light bokeh, intricate, steel metal rust, elegant, sharp focus, photo by greg rutkowski, soft lighting, vibrant colors, masterpiece, streets, detailed face"

22. "detailed ink, pen and ink, mail art, best quality, detailed epic ice transparent ethereal otherworldly ghost castle in the blue sky, clouds, smoke, fog, detailed landscape, ghost figures, lake, boat, green forest, detailed flying dragon at the sky, detailed scales, warm lights, glittering, Craola, Dan Mumford, Andy Kehoe, 2d, flat, art on a cracked paper, patchwork, stained glass, cute, adorable, fairytale, storybook detailed illustration, cinematic, ultra highly detailed, tiny details, beautiful details, mystical, luminism, vibrant colors, complex background"
23. "crystal scorpion"
24. "the image portrays a tranquil scene of a boat floating gently on the water, surrounded by an expansive landscape. the moon, full and glowing with a warm, reddish orange hue, casts a mystical ambiance over the entire scene. its reflection shimmers off the surface of the water, adding to the serene atmosphere. in the distance, mountains loom under the moon's soft glow, their peaks partially obscured by the low hanging clouds. they appear majestic yet gentle, as if watching over the peaceful night below. trees line the shore in the foreground, their silhouettes faintly visible against the darkening sky. this picturesque setting evokes a sense of calm and tranquility, inviting viewers to take a moment and appreciate the beauty of nature. it is a symphony of colors and shapes, each element working harmoniously together to create a visually captivating and emotionally soothing composition."

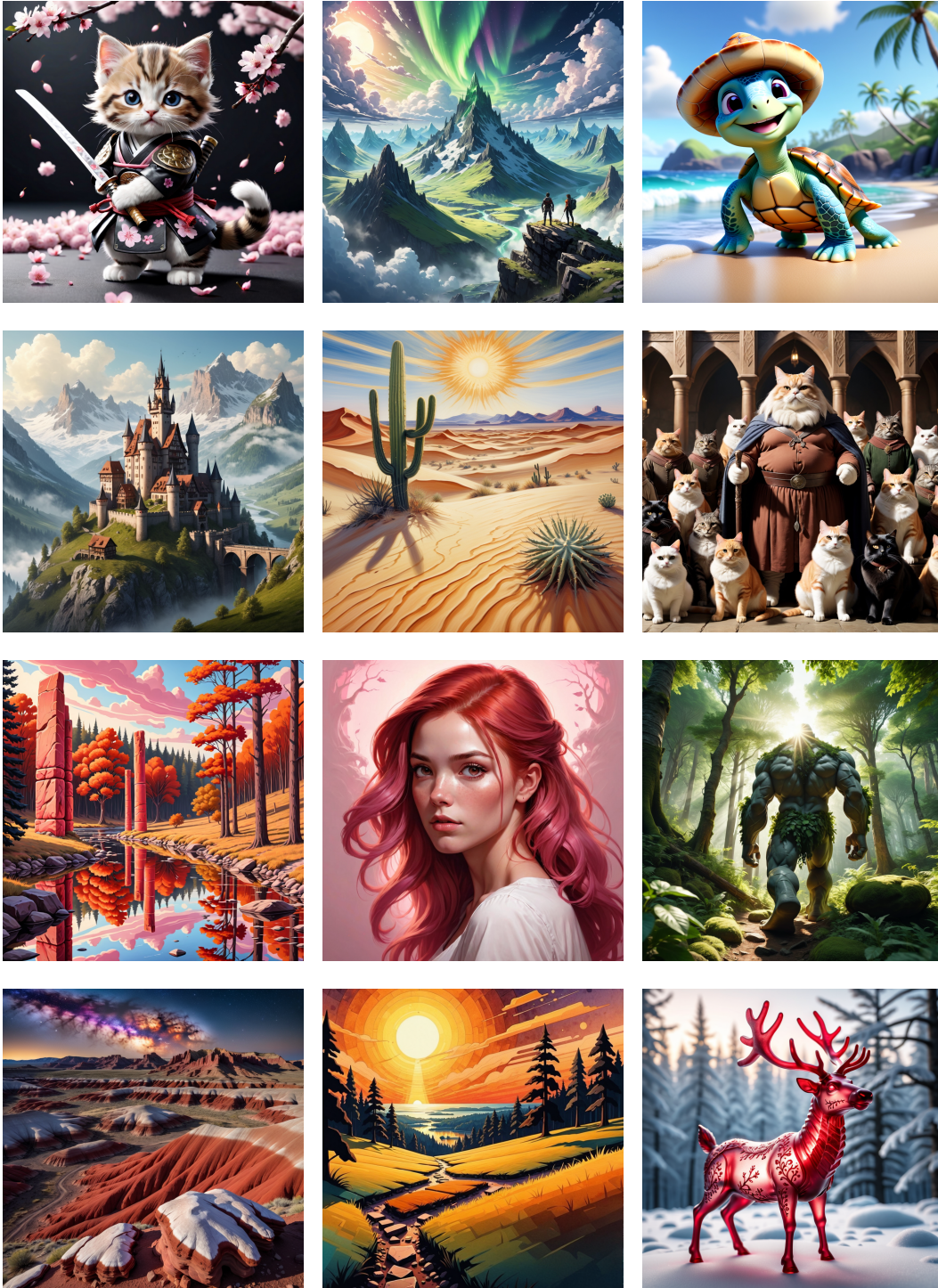


Figure 6: More qualitative generations using FlowRL

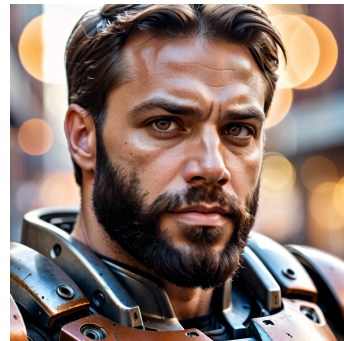


Figure 7: More qualitative generations using FlowRL

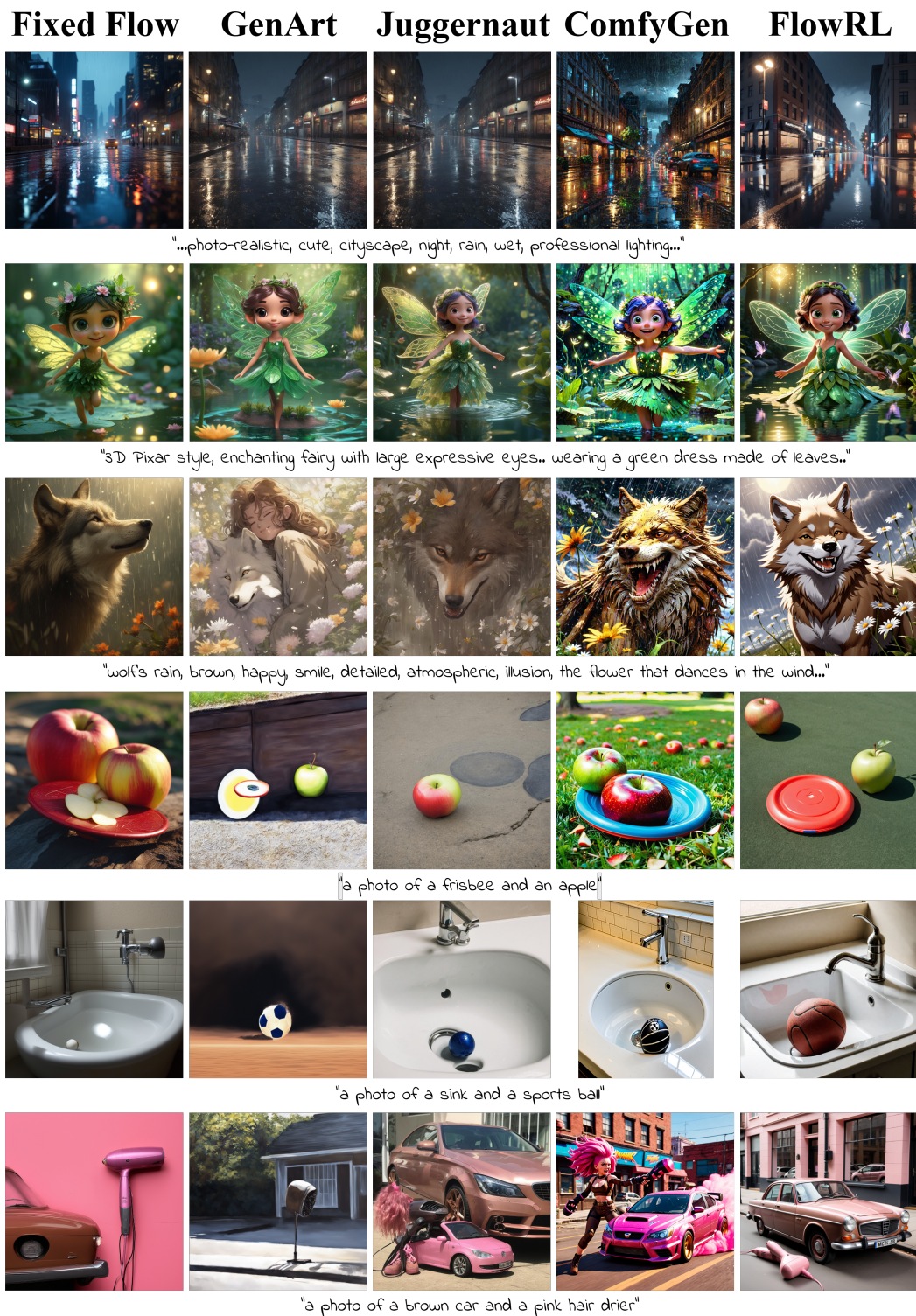


Figure 8: Additional qualitative comparisons on CivitAI prompts (top 3) and GenEval prompts (bottom 3)

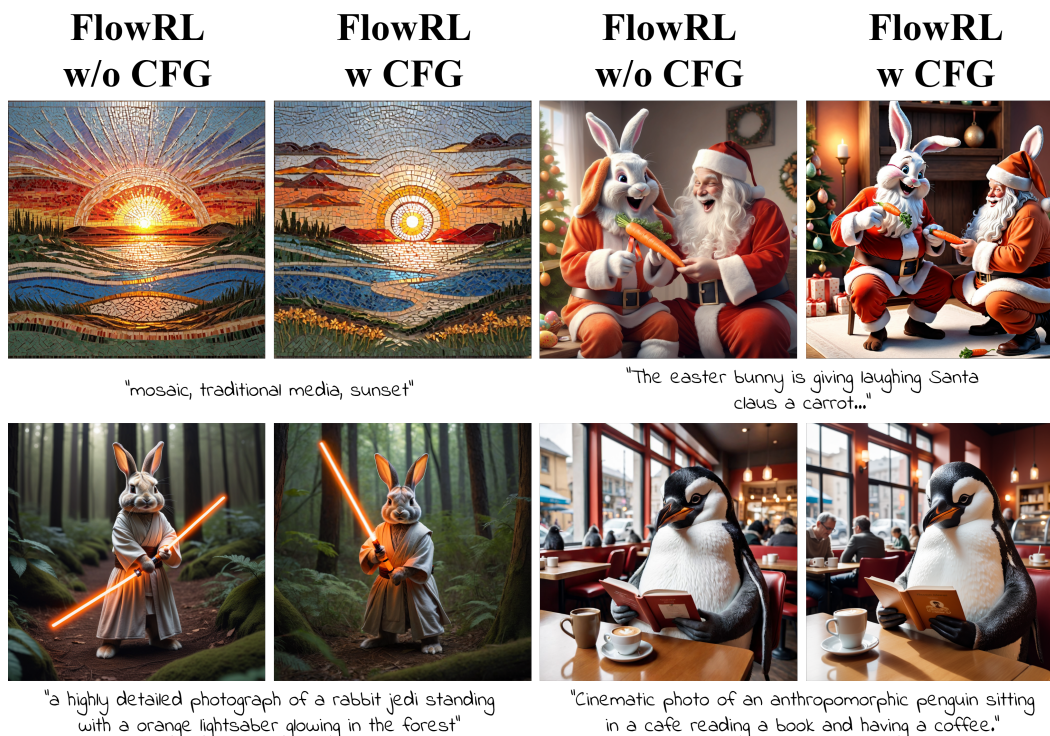


Figure 9: Qualitative example on influence of CFG on the output image

A.4 Tokenization Encoding Method

We developed a systematic procedure to transform JSON-based workflow representations into a compact, encoded format. This process utilizes schema learning to ensure both accuracy and efficiency in data transformation.

Methodology First, we infer a schema from a collection of workflow JSON files by iterating through each file and extracting the class types and field information for all utilized nodes. The resulting schema is stored for future encoding tasks. In the encoded representation, nodes are sorted and formatted to include their class types and input values, with explicit references to connected nodes. Each line in the encoded output corresponds to a node from the original JSON structure, providing a clear and organized mapping.

Incorporating Workflow-Specific Tokens To more effectively capture the structure of ComfyUI workflows, we enhanced the base tokenizer by introducing custom tokens that represent key workflow elements such as node types, connections, and parameters. This enriched tokenization scheme helps the model better understand relationships between workflow components. Below, we provide examples of some of the custom tokens added to the tokenizer:

- ng Everywhere3
- AspectSize
- Automatic CFG
- BNK_AddCLIPSDXLParams
- BNK_CLIPTextEncodeAdvanced
- BasicPipeToDetailerPipe
- Image Levels Adjustment
- Image Remove Background (rembg)

- CLIP Positive-Negative XL w/Text (WLSH)
- CLIP=
- CLIPLoader
- CLIPMergeSimple
- CLIPSetLastLayer
- CLIPTextEncode
- CLIPTextEncodeSDXL
- CLIPTextEncodeSDXLRefiner
- CLIP_NEGATIVE
- CONDITIONING=
- CR Apply LoRA Stack
- CR Apply Model Merge
- SDXL 1.0/animagineXLV31_v30.safetensors
- SDXL 1.0/crystalClearXL_ccxl.safetensors
- SDXL 1.0/dreamshaperXL_turboDpmppSDEKarras.safetensors
- SDXL 1.0/envyhyperdrivexl_v10.safetensors
- SDXL 1.0/faces_v1.safetensors
- SDXL 1.0/jibMixRealisticXL_v90BetterBodies.safetensors
- SDXL 1.0/juggernautXL_v9Rdphoto2Lightning.safetensors

A.4.1 Example of encoded flow

Following is a short example of the original ComfyUI flow in it's JSON form and it;s encoded text form:

JSON Form

```
{
  "2": {
    "inputs": {
      "ckpt_name": "SDXL 1.0/realvisxlV40_v40LightningBakedvae.safetensors"
    },
    "class_type": "CheckpointLoaderSimple",
    "_meta": {
      "title": "Load Checkpoint"
    }
  },
  "8": {
    "inputs": {
      "seed": 62282230408842,
      "steps": "50",
      "cfg": "7.5",
      "sampler_name": "ttm",
      "scheduler": "ddim_uniform",
      "denoise": 1,
      "noise_mode": "GPU(=A1111)",
      "batch_seed_mode": "incremental",
      "variation_seed": 0,
      "variation_strength": 0,
      "variation_method": "linear",
      "model": ["2", 0],
      "positive": ["9", 0],
      "negative": ["10", 0],
    }
  }
}
```

```

    "latent_image": ["15", 0]
  },
  "class_type": "KSampler //Inspire",
  "_meta": {
    "title": "KSampler (inspire)"
  }
},
"9": {
  "inputs": {
    "text": "{positive_prompt}",
    "token_normalization": "length+mean",
    "weight_interpretation": "comfy++",
    "clip": ["2", 1]
  },
  "class_type": "BNK_CLIPTextEncodeAdvanced",
  "_meta": {
    "title": "CLIP Text Encode (Advanced)"
  }
},
"10": {
  "inputs": {
    "text": "blurry, low quality, deformed, disfigured",
    "token_normalization": "length+mean",
    "weight_interpretation": "comfy++",
    "clip": ["2", 1]
  },
  "class_type": "BNK_CLIPTextEncodeAdvanced",
  "_meta": {
    "title": "CLIP Text Encode (Advanced)"
  }
},
"12": {
  "inputs": {
    "vae_name": "SDXL 1.0/sharpspectrum_vae1.safetensors"
  },
  "class_type": "VAELoader",
  "_meta": {
    "title": "Load VAE"
  }
},
"13": {
  "inputs": {
    "tile_size": 512,
    "samples": ["8", 0],
    "vae": ["12", 0]
  },
  "class_type": "VAEDecodeTiled",
  "_meta": {
    "title": "VAE Decode (Tiled)"
  }
},
"15": {
  "inputs": {
    "width": "1024",
    "height": "1024",
    "batch_size": 1
  },
  "class_type": "EmptyLatentImage",
  "_meta": {

```

```

    "title": "Empty Latent Image"
  }
},
"22": {
  "inputs": {
    "filename_prefix": "{save_prefix}",
    "images": ["13", 0]
  },
  "class_type": "SaveImage",
  "_meta": {
    "title": "Save Image"
  }
}
}
}

```

Encoded Form

```

N2: CheckpointLoaderSimple
[ckpt_name=SDXL 1.0/realvisxlV40_v40LightningBakedvae.safetensors]
N8: KSampler //Inspire
[steps=50, cfg=7.5, scheduler=ddim_uniform, model=N2, --
positive=N9, negative=N10, latent_image=N15]
N9: BNK_CLIPTextEncodeAdvanced
[clip=N2]
N10: BNK_CLIPTextEncodeAdvanced
[text=blurry, low quality, deformed, disfigured, clip=N2]
N12: VAELoader
[vae_name=SDXL 1.0/sharpspectrum_vae1.safetensors]
N13: VAEDecodeTiled
[samples=N8, vae=N12]
N15: EmptyLatentImage
N22: SaveImage
[images=N13]

```

A.5 User study

To evaluate our method against baselines, we conducted a user study using a structured survey. For the study, we randomly sampled 50 prompts and generated corresponding images with each baseline. From these, we filtered out results which contained unsafe content (e.g., nudity, violence), resulting in 7–11 comparison questions per baseline. These comparisons were aggregated into a survey where participants were shown a prompt and the outputs from FlowRL and one baseline, and asked to select their preferred image.

We collected approximately 200 responses per baseline. Figure 10, provides an example of a question from our survey.

A.6 Reward model generalization

We conducted an experiment where we re-trained the reward model using half of our original training set, and evaluated the generalization capabilities of our reward model on the remaining half. We further ensured that some graph structures appear only in the hold-out set, so we can evaluate performance on entirely unseen flow graphs.

The results show good overall performance on the hold-out set, with an R^2 score of 0.643 and a pearson correlation of 0.816, indicating a strong relationship between predicted and actual values. A more in-depth look shows that: (1) The model generalizes very well to scenarios which contain only parameter or prompt changes compared to what it saw during training (Pearson 0.928). (2) Performance remains good for flows with novel graph structures, but using only seen components

Below is a text prompt describing an image that we want to generate, and * two text-to-image model outputs corresponding to this prompt. Please select the result that you prefer, taking into account both the quality of the image and its adherence to the text prompt:

Text prompt: "refreshing, vibrant glowing coconut juice drink, dew drops, refreshing, in the style of a product hero shot in motion, dynamic magazine ad image, photorealism, sleep and mystical elements around the background"

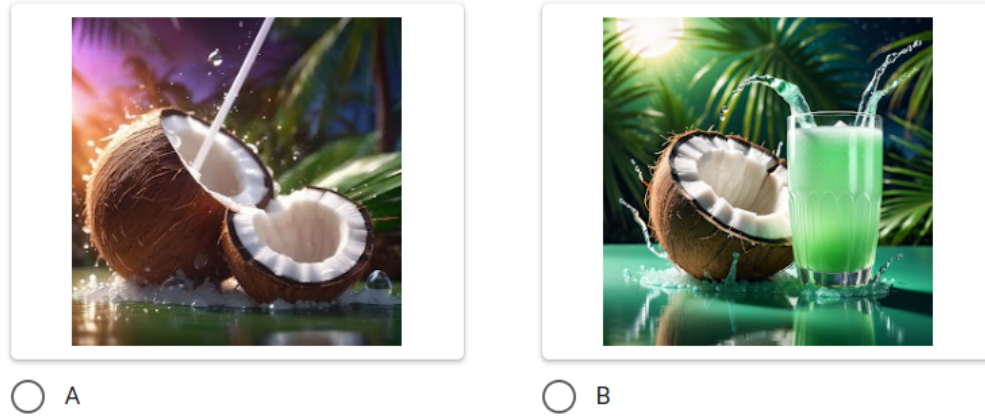


Figure 10: An example question from the user study.

(models / blocks) (Pearson 0.667). (3) Performance drops significantly for flows with entirely unseen components, which contain tokens that the reward model has never seen (Pearson 0.152).

After running this experiment, we also evaluated the GRPO stage using the new reward model. The model trained on the full dataset outperforms the model trained on the partial data (71% HPSv2 win-rate), showing the benefit of additional data.

A.7 Implementation details

A.7.1 SFT stage

We implement our model based on a pre-trained Meta Llama3.1- 8B [19]. We used the unsloth [10] library to fine-tune the model using LoRA [22]. The SFT stage was trained on a single NVIDIA H100 80GB HBM3 GPU for 10 hours.

LoRA Configuration: To enable parameter-efficient fine-tuning, we applied LoRA (Low-Rank Adaptation) to the model’s attention and feed-forward layers. The LoRA rank was set to $r = 16$, with an alpha value of $\alpha = 16$, and a dropout rate of 0.0. Target modules included "q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj", as well as "lm_head" and "embed_tokens" since we added new tokens to our vocabulary.

Prompt structure During the supervised fine-tuning (SFT) stage, the LLM is provided with both the prompt and one of the encoded flows:

```
">>> Prompt:
      {p_i}
>>> Flow:
```

{f_\$}"

In contrast, during the reinforcement learning (RL) fine-tuning stage, only the prompt is given to the LLM, and it is tasked with generating one or more candidate flows. This setup encourages the model to learn to produce the most appropriate flow for each prompt:

```
">>> Prompt:
      {p_i}
>>> Flow: "
```

A.7.2 Reward model training

For training the Reward BERT model, we utilized the "answerdotai/ModernBERT-base" [57] as the foundational architecture. Beyond its improved classification performance, we selected ModernBERT because it was trained on sequence lengths that match our expected prompt and encoded-flow format. We used the Adam optimizer with the default parameters and a learning rate of $8e-5$. The maximum sequence length was set to 4096 tokens, with a batch size of 128 over 10 epochs. Fine-tuning was done on a single NVIDIA A100-SXM4-80GB for approximately 4 hours.

Dataset: Each data-point consisted of the triplet (f_i, p_i, s_i) : flow, prompt and human-preference normalized score. and was inserted to the model in this format:

"[PROMPT] {p_i} [FLOW] {f_i}" .

The model was tasked with prediction the output score s_i for each pair, using an MSE loss.

A.7.3 GRPO Fine-Tuning Hyperparameters

Below, we detail the key hyperparameters and configurations used in the GRPO (Group Relative Policy Optimization) fine-tuning stage:

LoRA Configuration: To enable parameter-efficient fine-tuning, we applied LoRA (Low-Rank Adaptation) to the model’s attention and feed-forward layers. The LoRA rank was set to $r = 16$, with an alpha value of $\alpha = 16$, and a dropout rate of 0.0. Target modules included "q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj", following best practices for large language model adaptation. Note that this step does not optimize the "lm_head" or "embed_tokens" layers as this step aims to further tune the SFT model, which already knows the flow vocabulary.

Optimization Settings: We used the Adam optimizer with a learning rate of $5e-6$, $\beta_1 = 0.9$, $\beta_2 = 0.99$, and a weight decay of 0.1. Training was performed with a batch size of 16 per device.

GRPO-Specific Parameters: We used the group size of 4 (number of generations per prompt for group-based reward calculation). clipping coefficient of 0.2, max grad norm of 0.5 and KL-regularization coefficient of 0.2. We also used generation temperature of 0.9, and maximal output tokens of 500.

Training Procedure: Fine-tuning was conducted for 2 epochs over the CivitAI prompt train set. We trained on a single NVIDIA A100-SXM4-80GB node (8 GPUs) for approximately 10 hours. We used an ensemble of 7 BERT reward models and used their mean as the surrogate reward. we set the uncertainty threshold to 0.08 and set the "uncertain reward value" to 0. For the prefix-reward mechanism, we used 5 different Bert models.