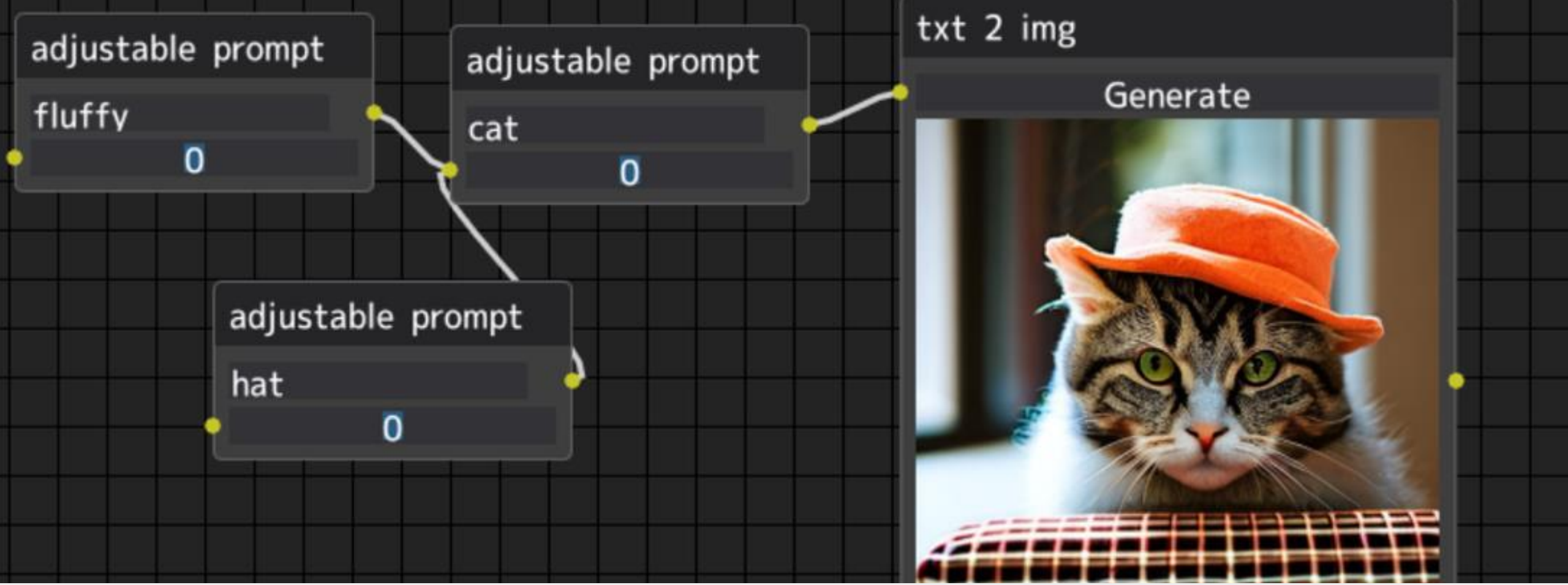


ノードとスライダで細部調整を追い込む画像生成システムの提案と評価

大友千宙　宮下芳明（明治大学）

背景

- **Text-to-Imageモデルで意図した通りの画像を生成することは困難**
  - プロンプトの**単語の順序**や**係り受け構造**ですら生成に影響
    - ↳ 容易な単語の入れ替えや係り受け構造の明示的な入力ができない

- **プロンプトをいじって後から画像を編集することは基本的にできない**
  - 画像生成向けの編集手法を利用すれば可能
    - ↳ 都度スクリプト言語によるプログラミングが必要

目的

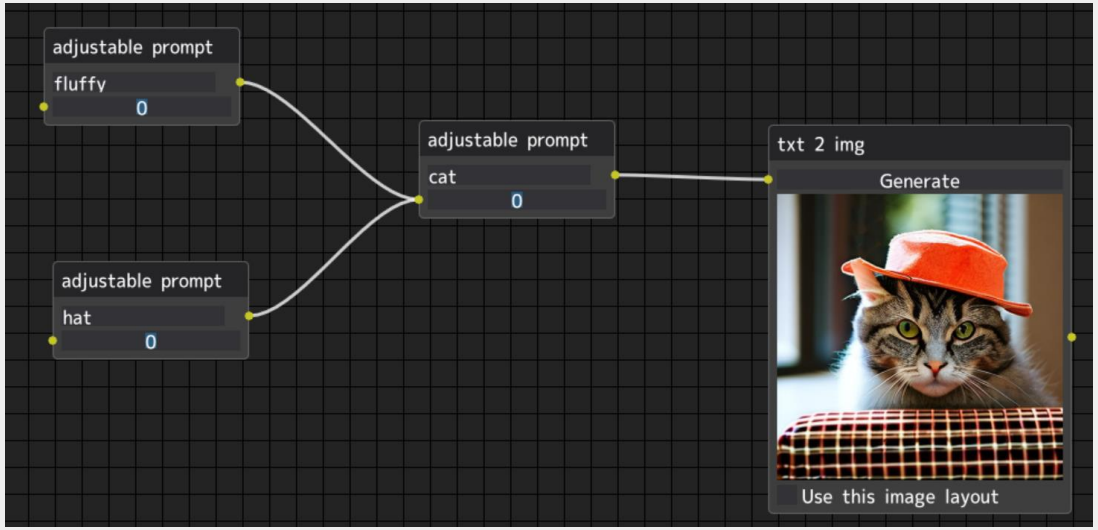
- Text-to-Imageモデルで意図した通りの画像を生成することは困難
  - プロンプトの**単語の順序**や**係り受け構造**ですら生成に影響
    - ↳ 容易な単語の入れ替えや係り受け構造の明示的な入力ができない

- プロンプトをいじって後から画像を編集することは基本的にできない
  - 画像生成向けの編集手法を利用すれば可能
    - ↳ 都度スクリプト言語によるプログラミングが必要

➡ **画像生成においてユーザ自身が求める表現へと**
追い込めるようにすること

提案システム

- ノードやスライダの操作で画像の生成と編集を行うノードベースシステム
  - ノードベースシステム
    - ↳ 単語の入れ替えが容易になり係り受けの明示的な入力が可能に
  - ノードやスライダによる操作
    - ↳ プログラミングなしで画像の編集が可能に



提案システムのスクリーンショット

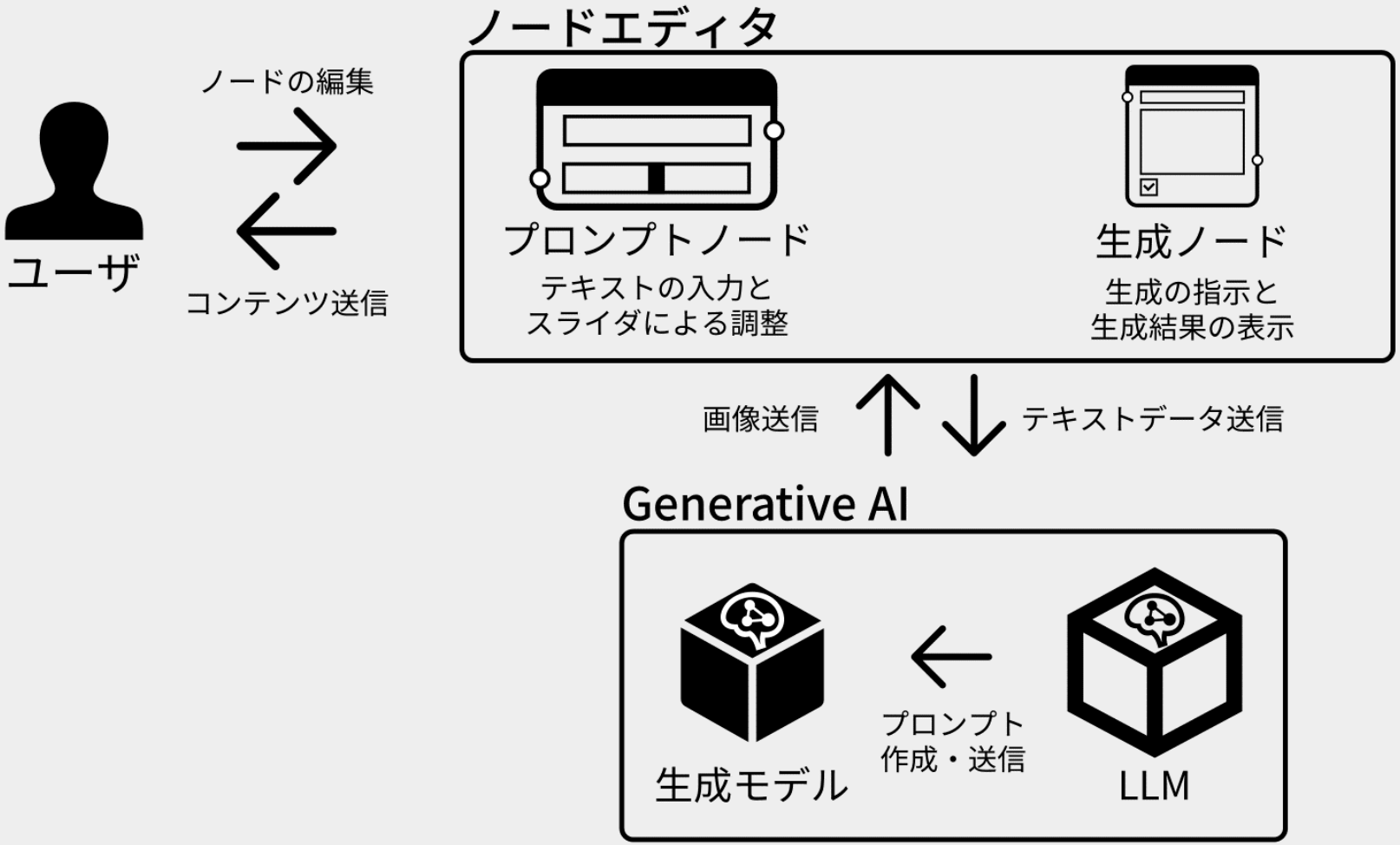
関連研究

- **Text-to-Image**
  - Diffusion Model [1]をベースにしたStable Diffusion [2]を利用
  - プロンプトの変更と画像の変化が対応しないという問題がある
- **画像生成向けの編集手法**
  - Prompt-to-prompt [3]やplug-and-play [4]
  - プロンプトの変更と画像の変化を対応させることが可能

プロンプトの変更を画像の変化と対応させている※[3]より引用

提案システム

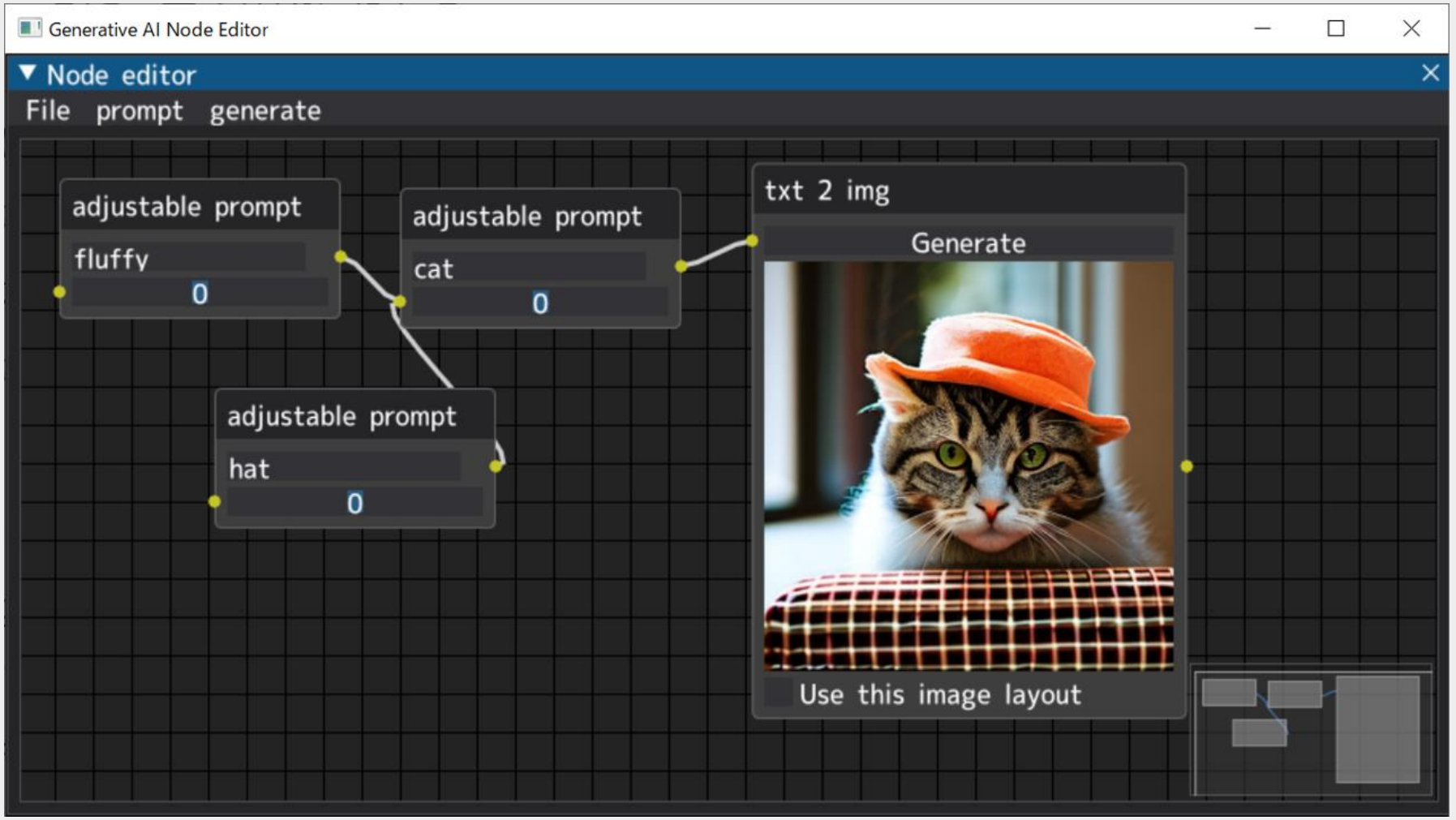
- ノードやスライダの操作で画像の生成と編集を行うノードベースシステム



提案システムの概要

提案システム

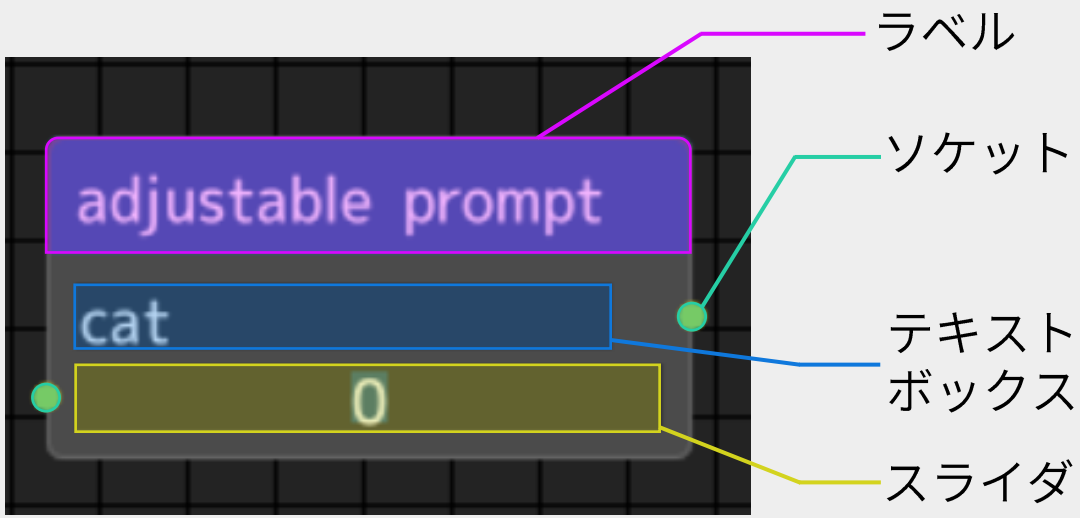
- ノードやスライダによって画像の生成と編集を行うノードベースシステム



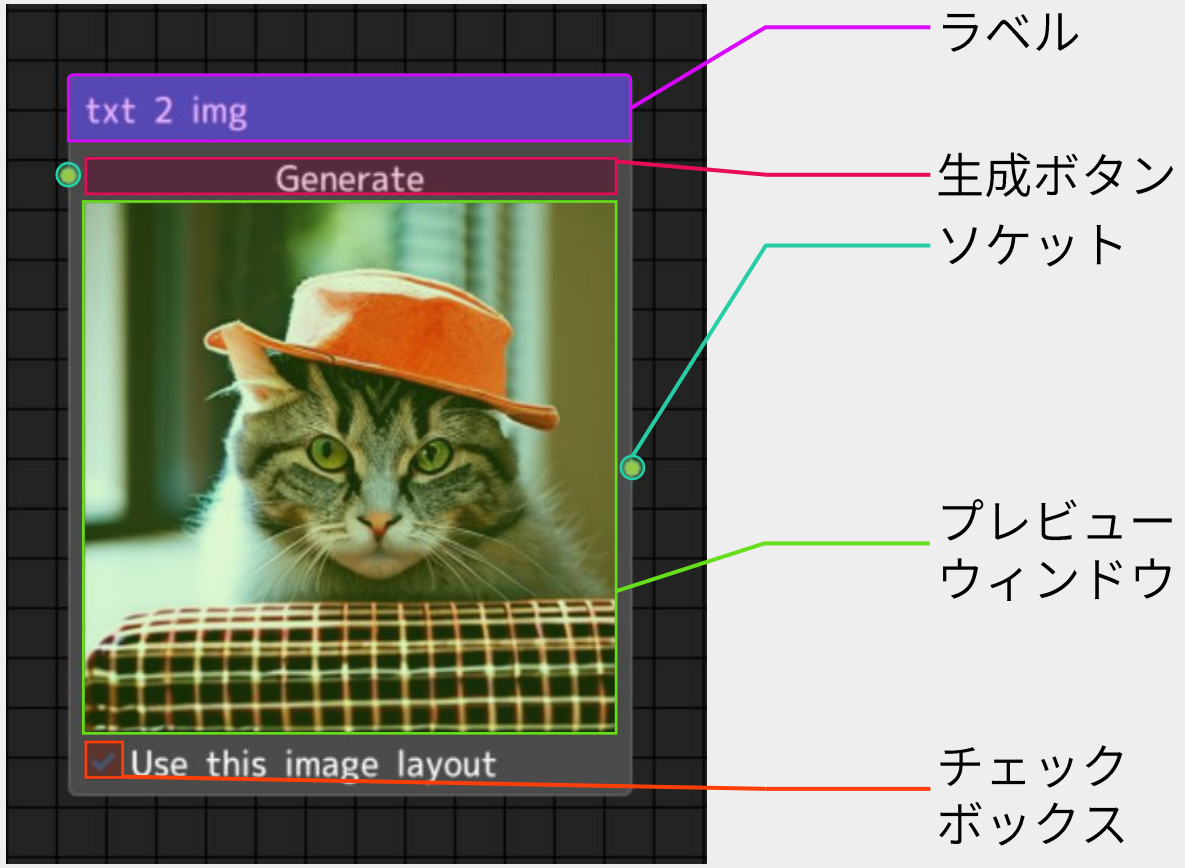
提案システムの画面

提案システム

- 2種類のノードで画像を生成し編集



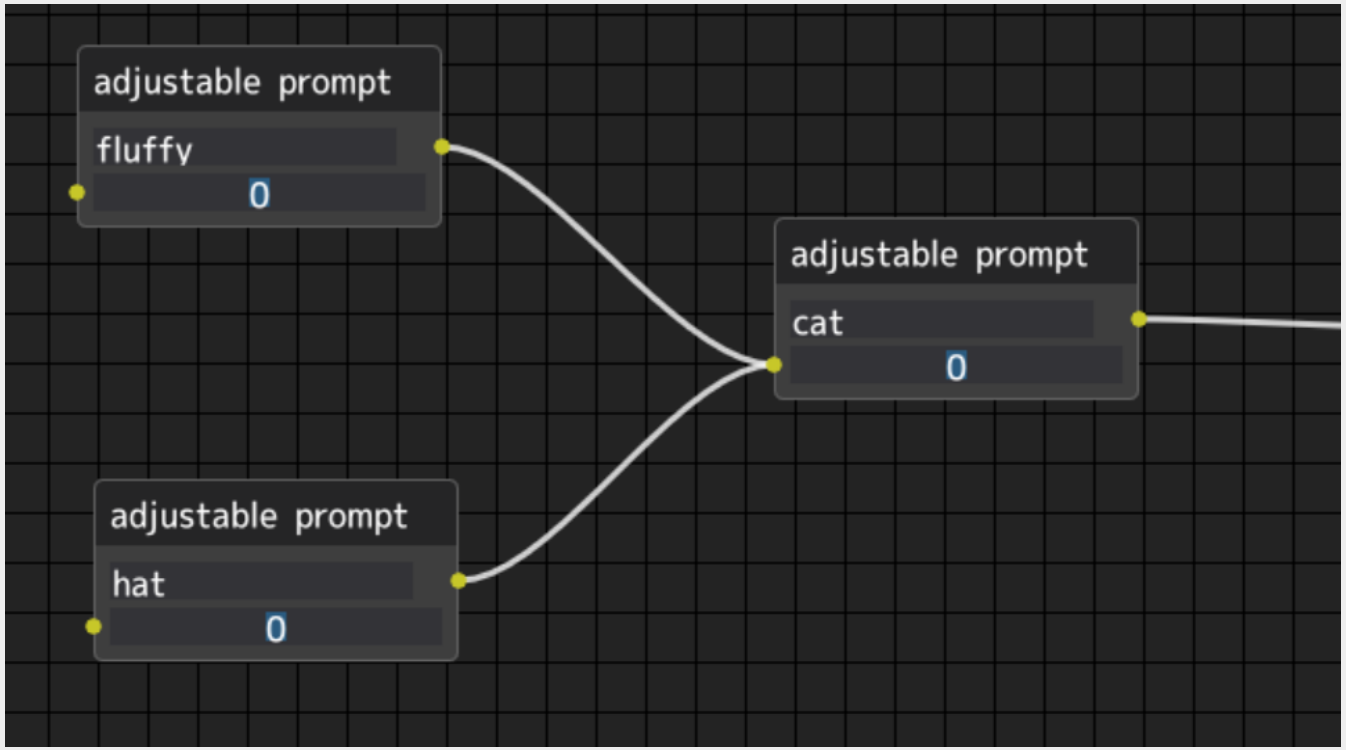
プロンプトノード



生成ノード

提案システム

- ノードやスライダによって画像の生成と編集を行うノードベースシステム



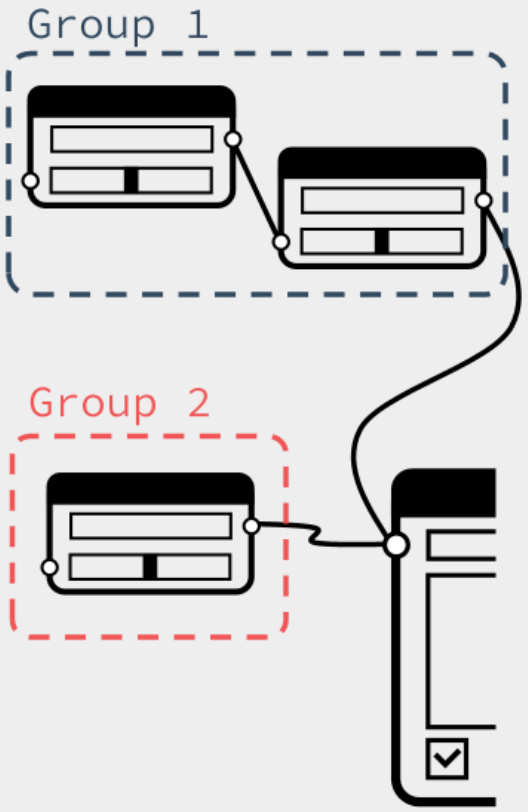
編集を行う前のノード

編集前の画像

提案システム

画像を生成する際の提案システムの動作

ノードのグルーピング → グループごとに情報のリストを作成 → GPT-4によるプロンプト生成 → Text-to-Imageによる画像生成



Text lists

```
[Group 1 Text list],
[Group 2 Text list],
⋮
```

Slider Value lists

```
[Group 1 Value list],
[Group 2 Value list],
⋮
```

Text lists

[], [] , ⋯

GPT-4

T

Prompt

Slider Value lists

[], [] , ⋯

T

Prompt

Text-to-Image モデル

生成結果

提案システム

- 画像の編集には3種類の操作が存在

ノードの追加

スタイルの変更

属性の指定

テキストの入れ替え

物体の入れ替え

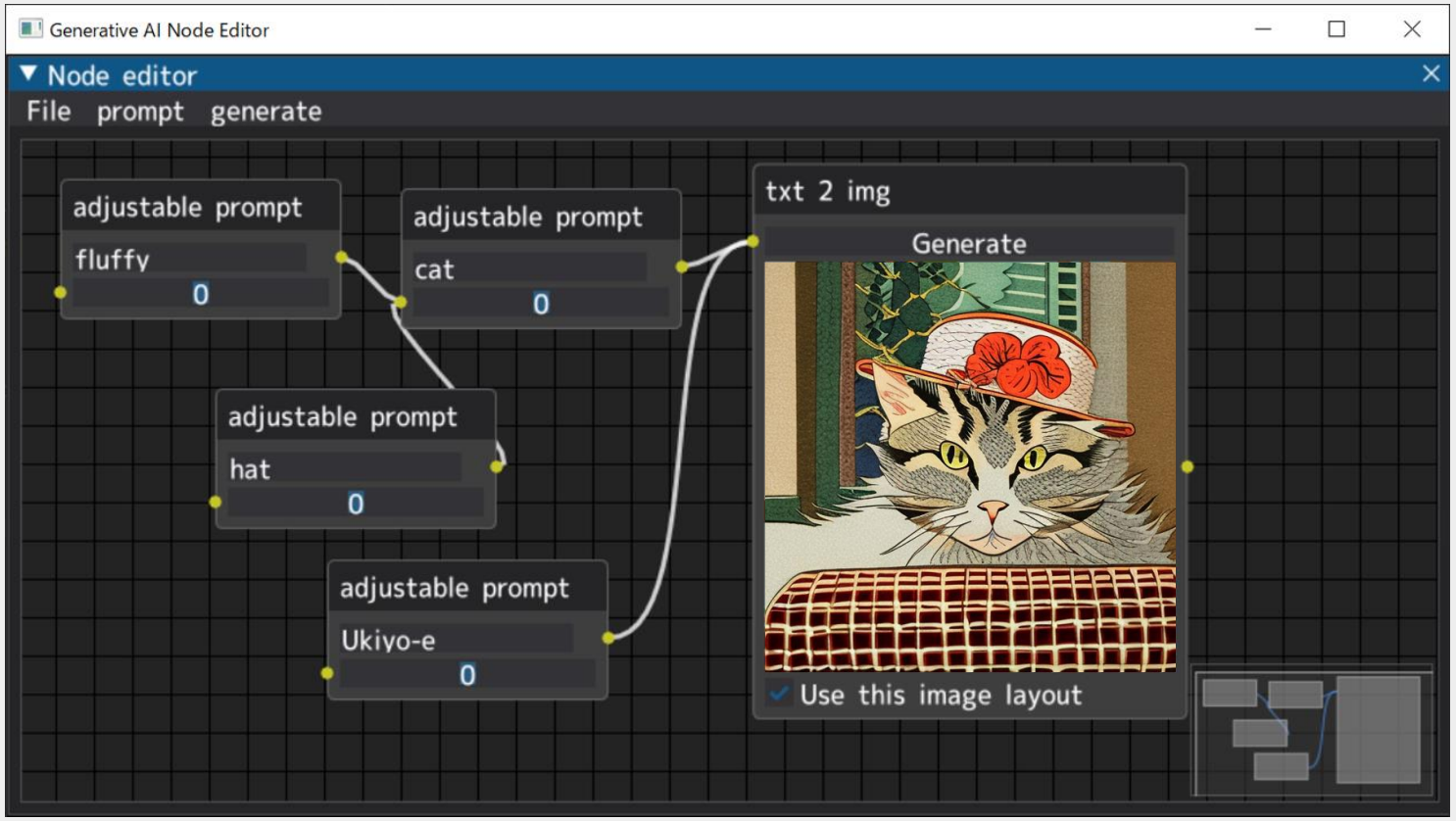
スライダの操作

効果の増幅

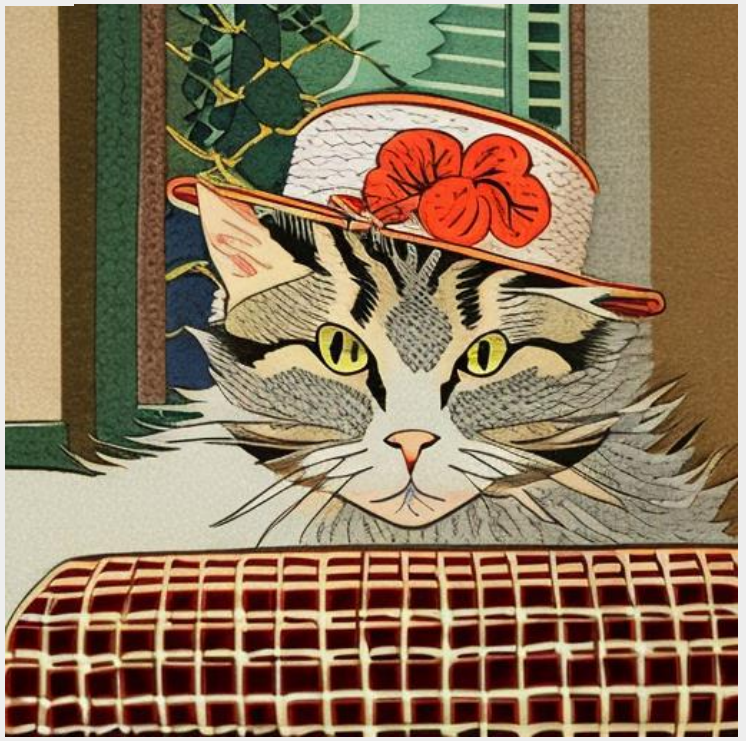
効果の減衰

提案システム

- ノードの追加
  - 全体のスタイルや物体の属性を指定



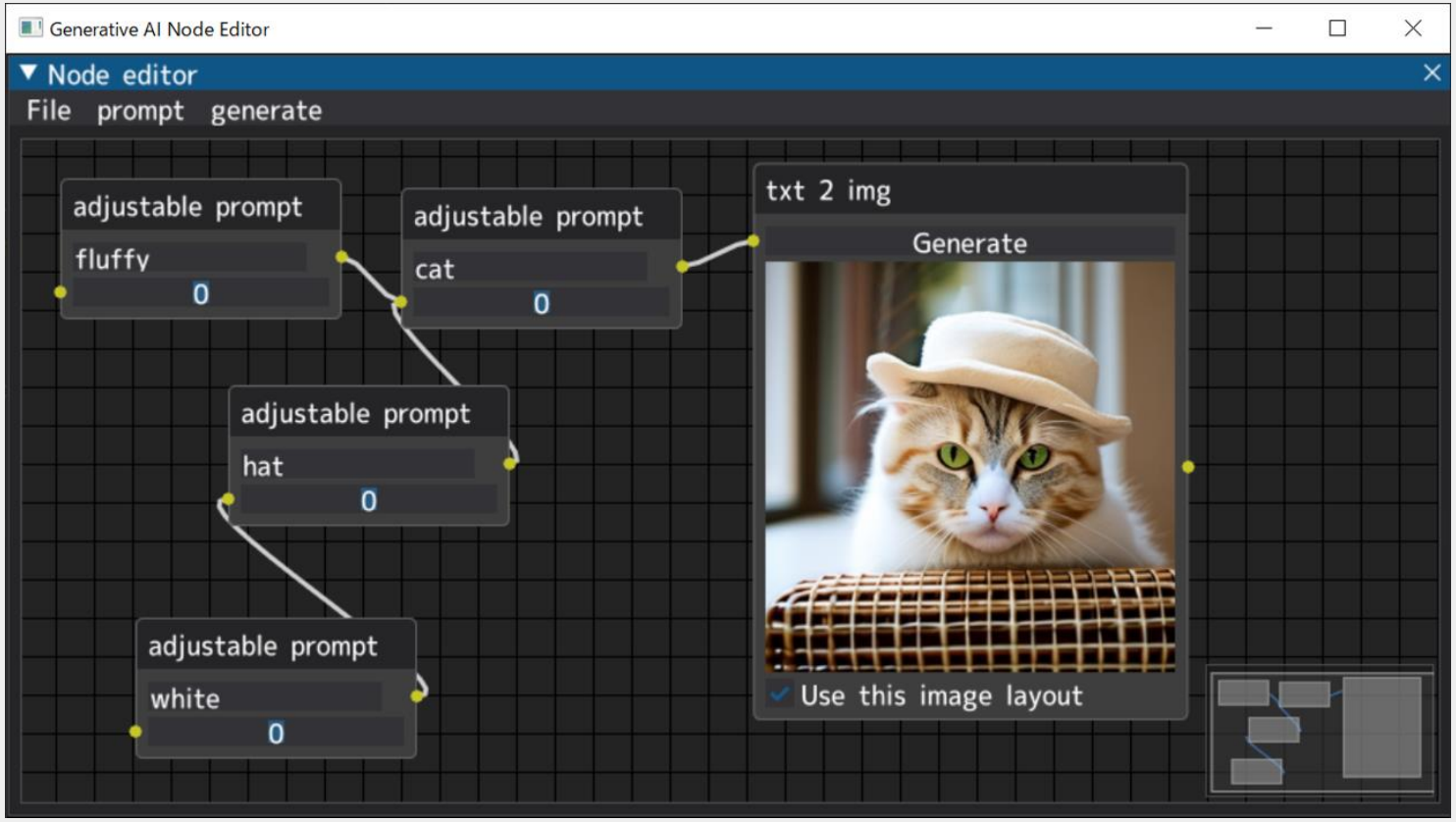
全体のスタイルを浮世絵風に変更※30倍速



編集後の画像

提案システム

- ノードの追加
  - 全体のスタイルや物体の属性を指定

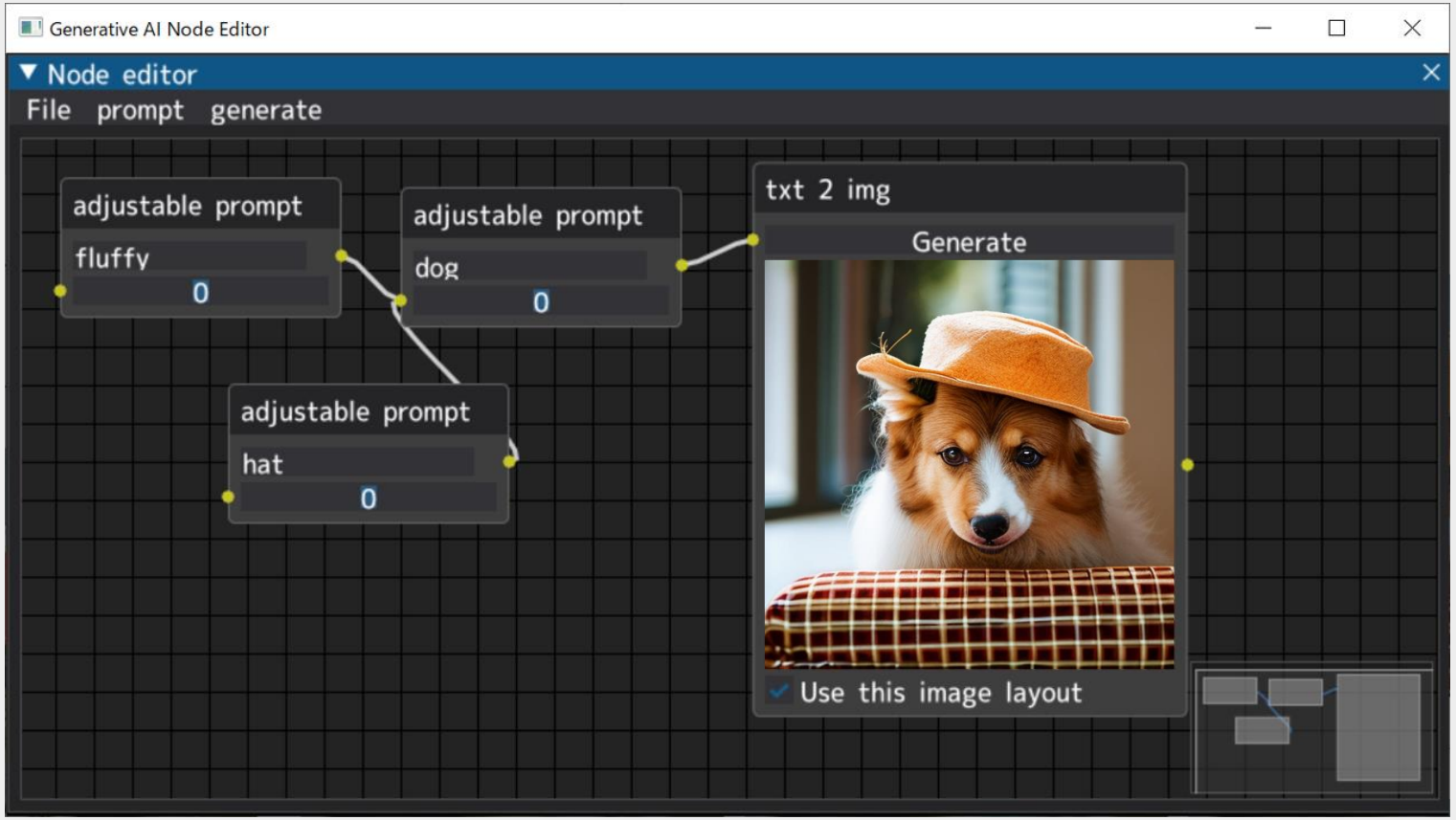


帽子と猫を白色に変更※30倍速

編集後の画像

提案システム

- テキストの入れ替え
  - 全体の構図は維持したまま物体を入れ替える

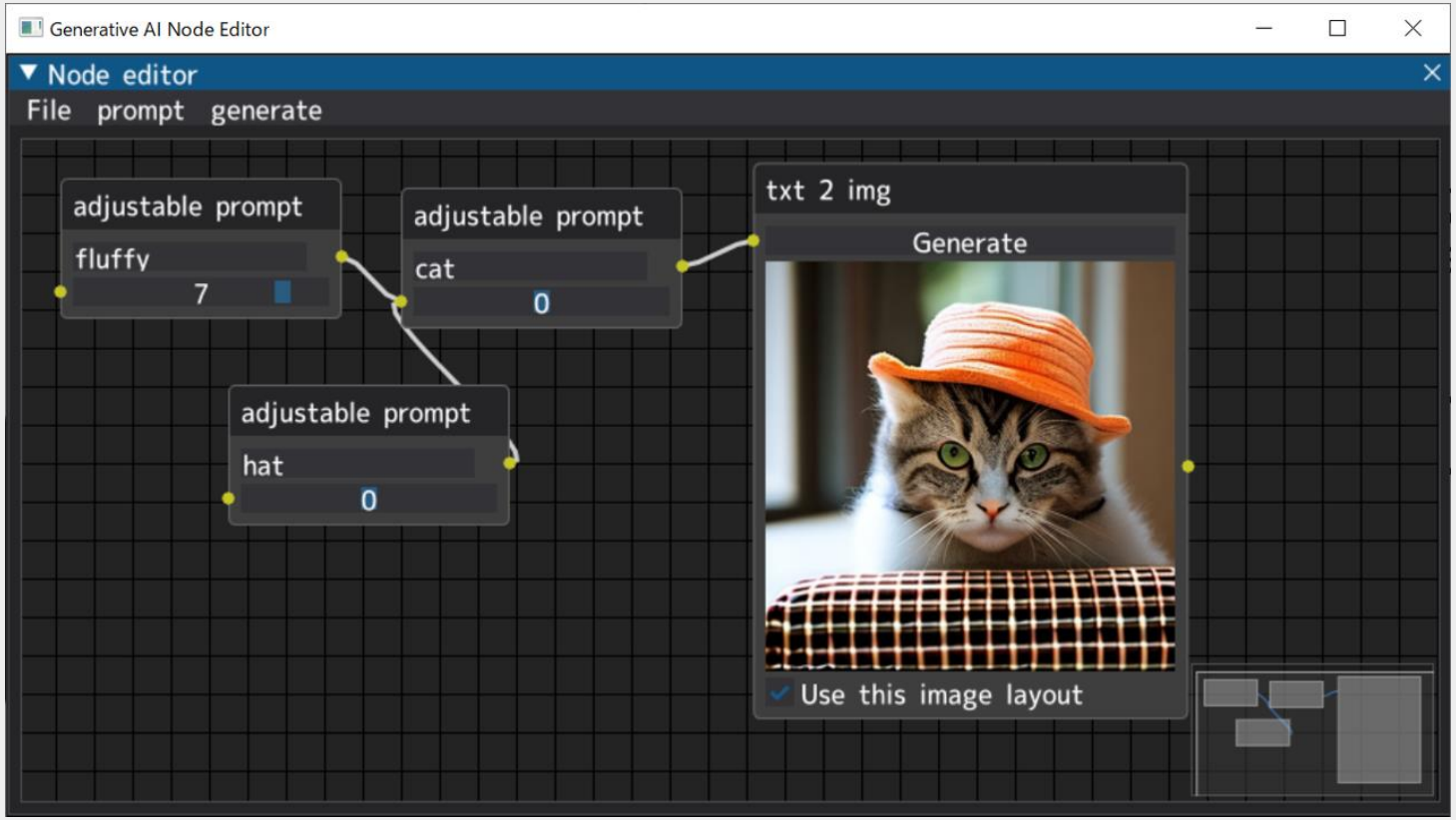


猫を犬に入れ替え※30倍速

編集後の画像

提案システム

- スライダの操作
  - 単語の効果を強めたり弱めたりできる

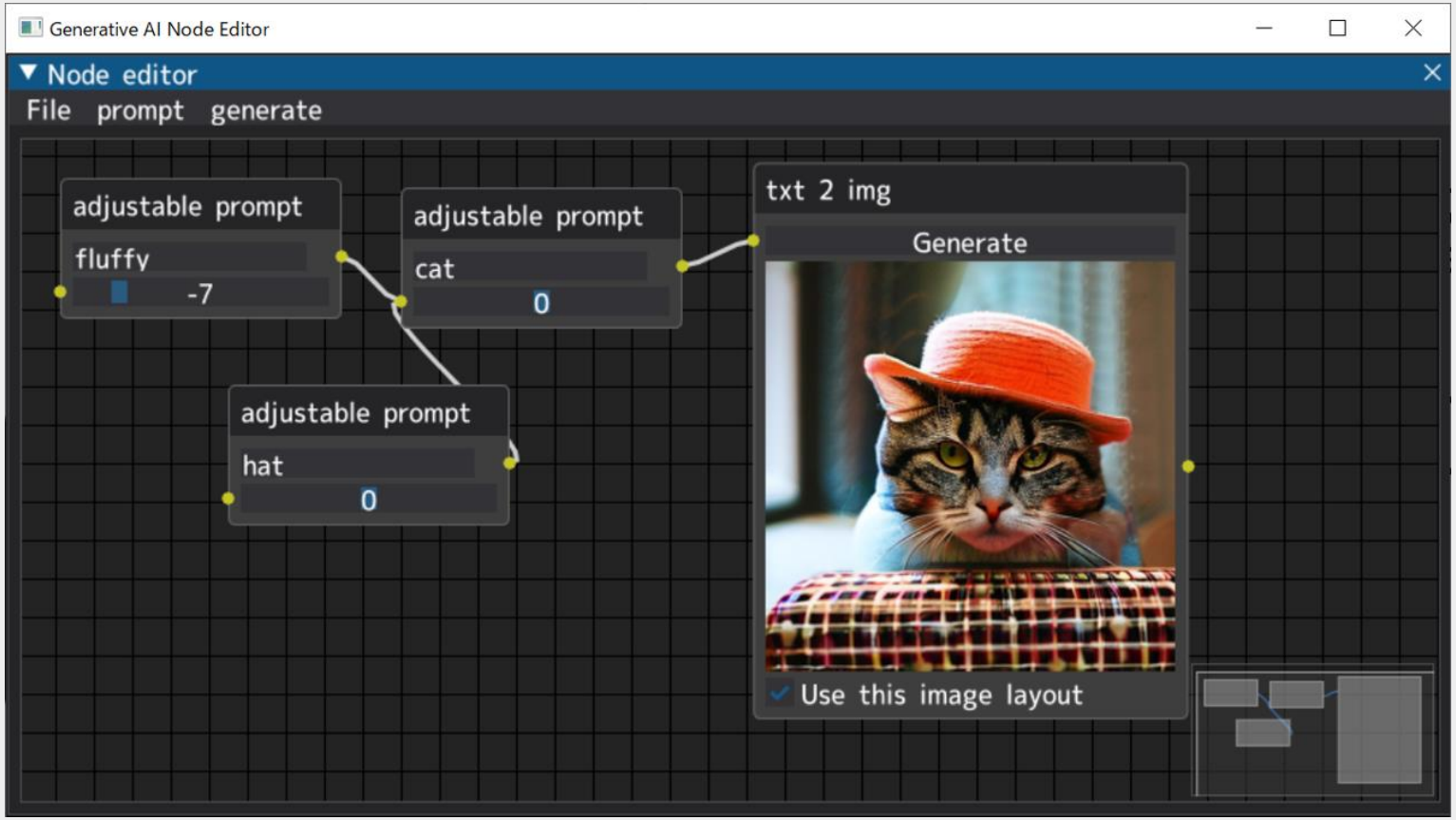


「fluffy」の効果を増幅 ※180倍速

編集後の画像

提案システム

- スライダの操作
  - 単語の効果を強めたり弱めたりできる

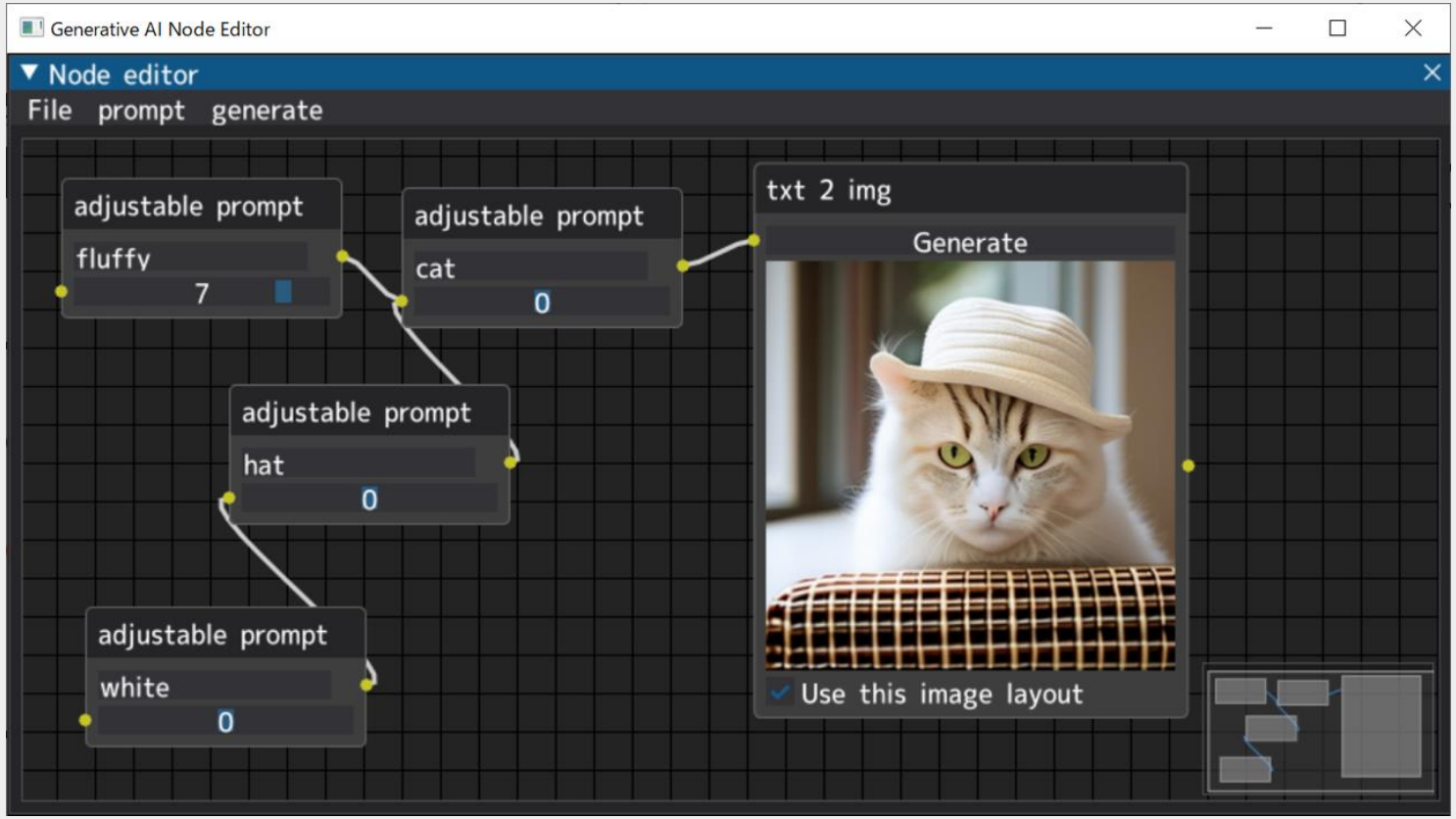


「fluffy」の効果を減衰 ※180倍速

編集後の画像

提案システム

- 複数操作の組み合わせ
  - 操作に応じて編集を重ねることが可能



毛量を増やしつつ猫と帽子を白色に変更※180倍速

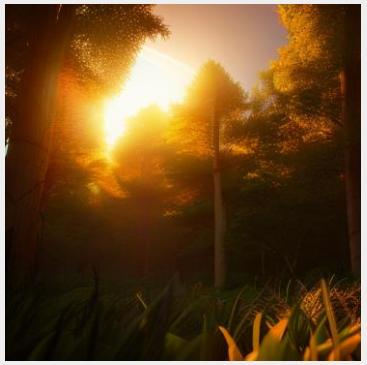
編集後の画像

作例

- 作例を3つ制作

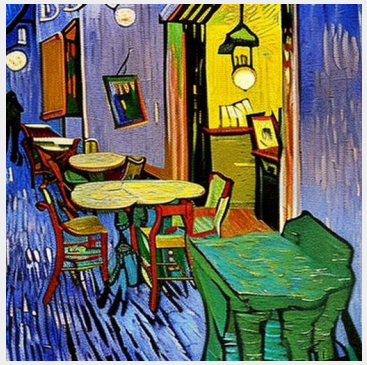
「夕暮れの風景画」

編集前

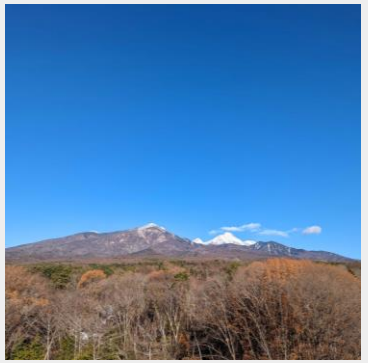


編集後

van Goghの
「夜のカフェテラス」



学習データに
ない画像の再現



作例

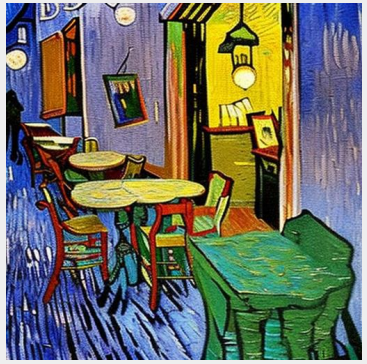
- 作例を3つ制作

「夕暮れの風景画」

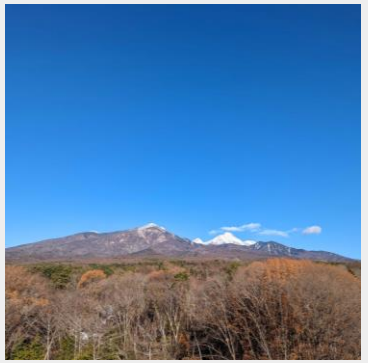
編集前

編集後

van Goghの
「夜のカフェテラス」

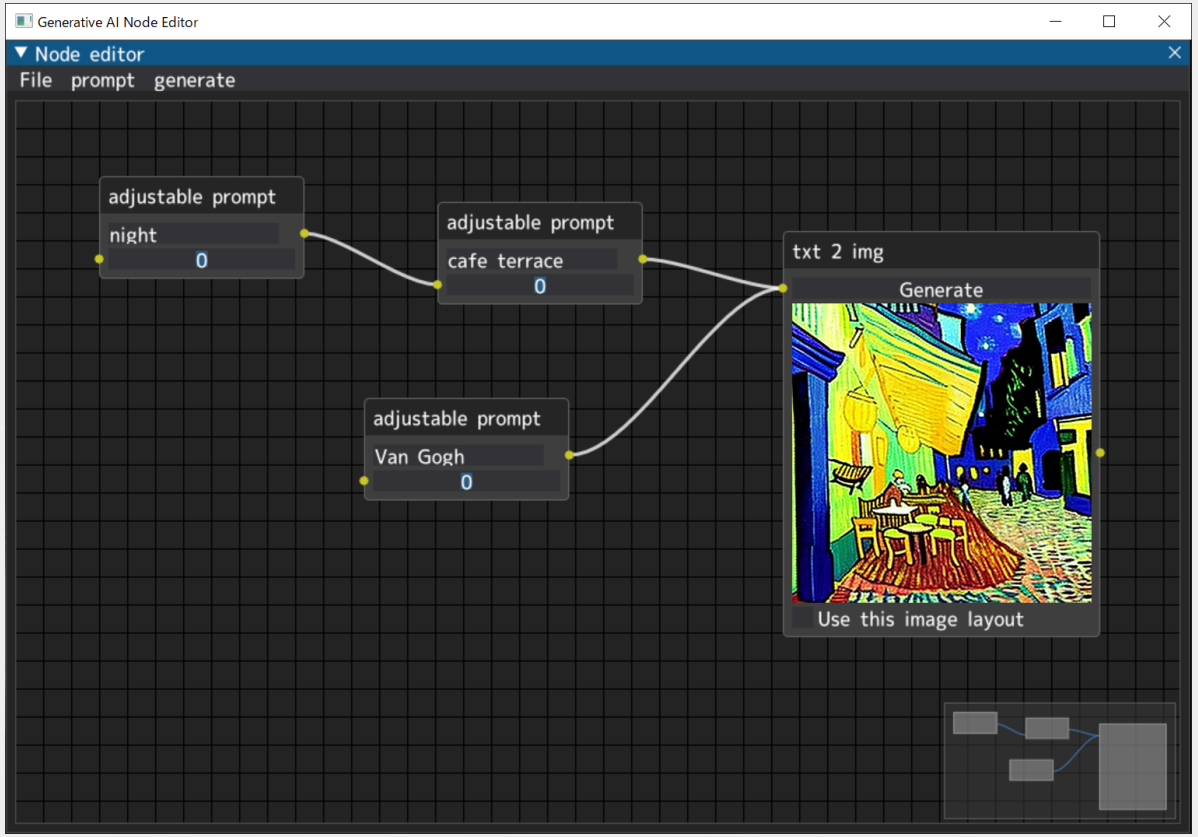


学習データに
ない画像の再現

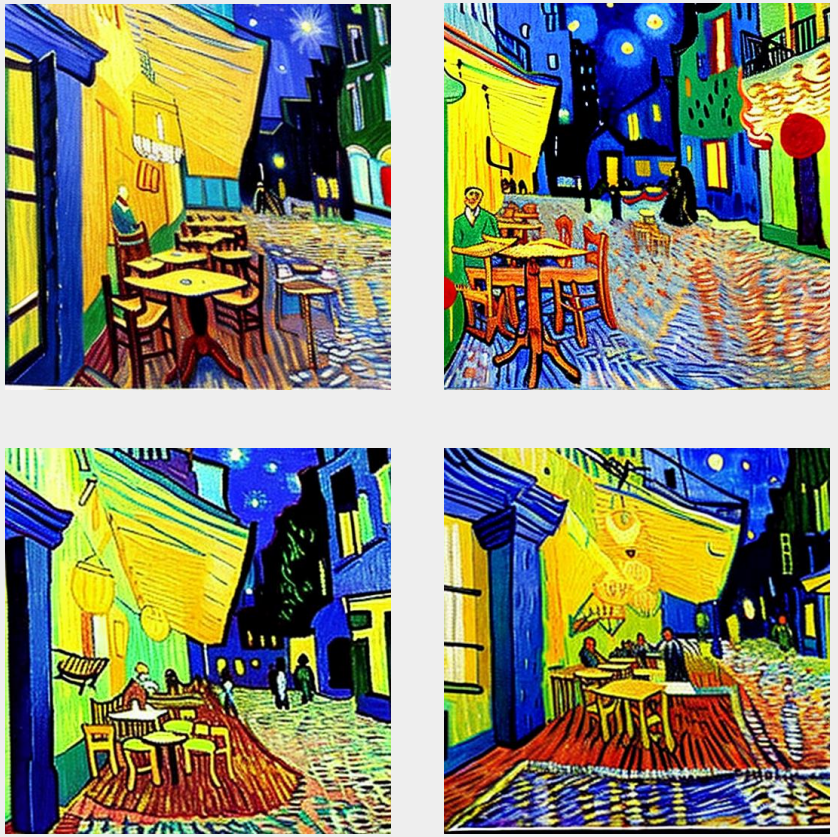


作例

- 学習データと異なる「van Goghの夜のカフェテラス」の生成
  - 似たような画像が生成される



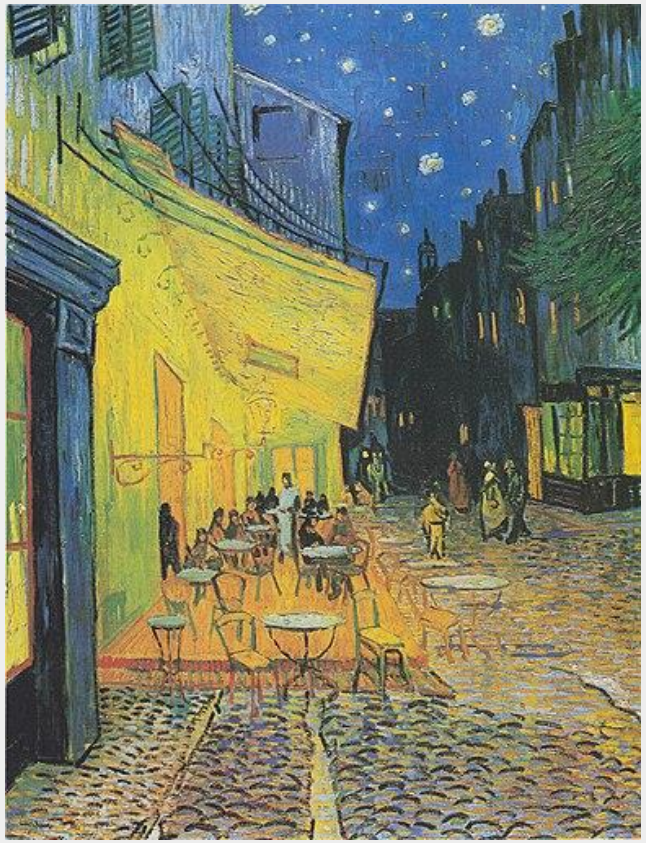
「van Goghの夜のカフェテラス」を生成した際の様子



生成された画像

作例

- 「van Goghの夜のカフェテラス」が学習データ[5]に存在
  - 学習データに寄った画像が生成されてしまう



van Goghの「Café Terrace at Night」
（1888年発表）

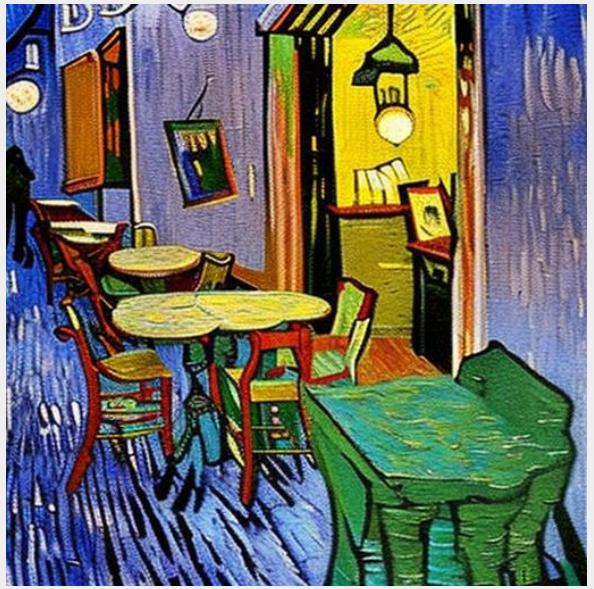
生成された画像

作例

- ただの「夜のカフェテラス」を「van Goghの夜のカフェテラス」にすればよい
  - ノードの追加によってスタイルを変化させることで実現

「café terrace at night」
で生成した画像

「night」ノードの
スライダを操作

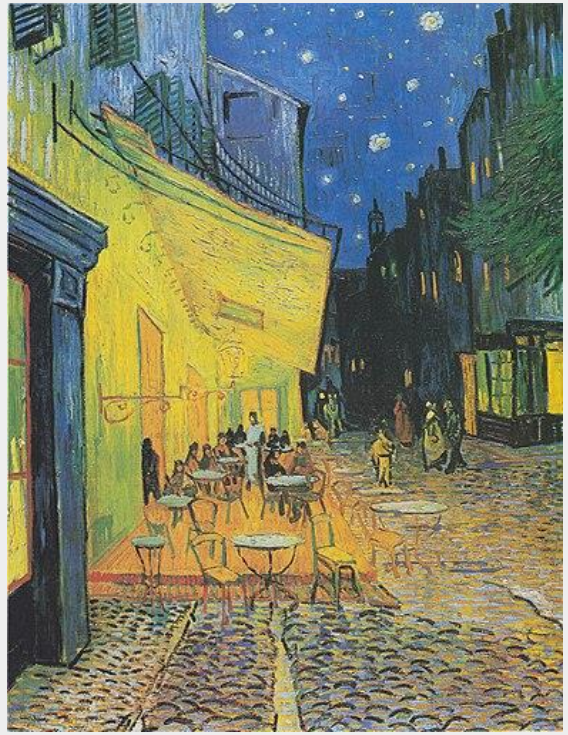


「van Gogh」ノードを追加

作例

- **「café terrace at night」で生成した画像**
  - 構図の時点でvan Goghの「café terrace at night」と異なる
  - 「night」とするには明るすぎるのではないか？

生成した画像



van Goghの「Café Terrace at Night」
（1888年発表）

作例

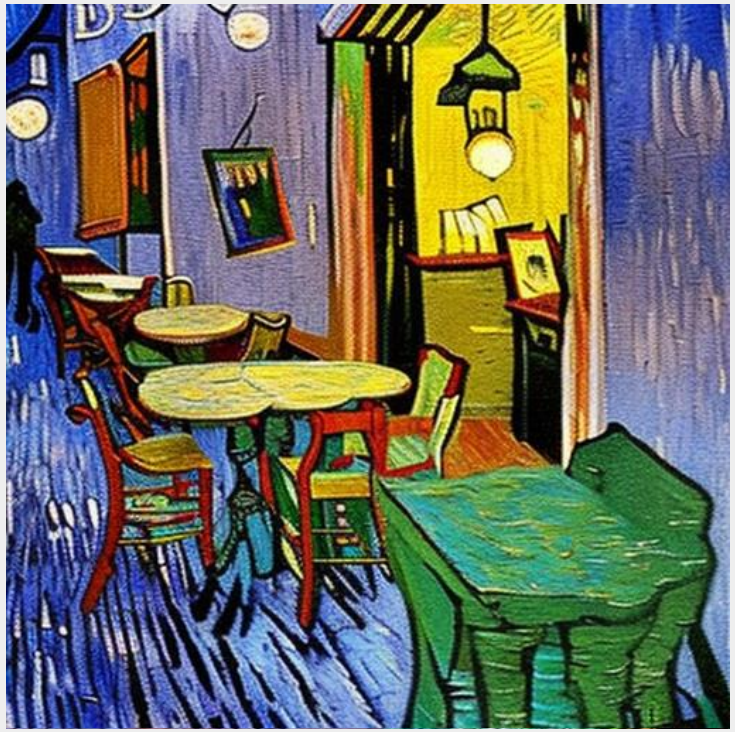
- **「night」ノードのスライダを操作**
  - 「night」という単語をより強く画像内に反映させた
  - 全体的な明度が低下し影が濃くなった

「night」を増幅した画像

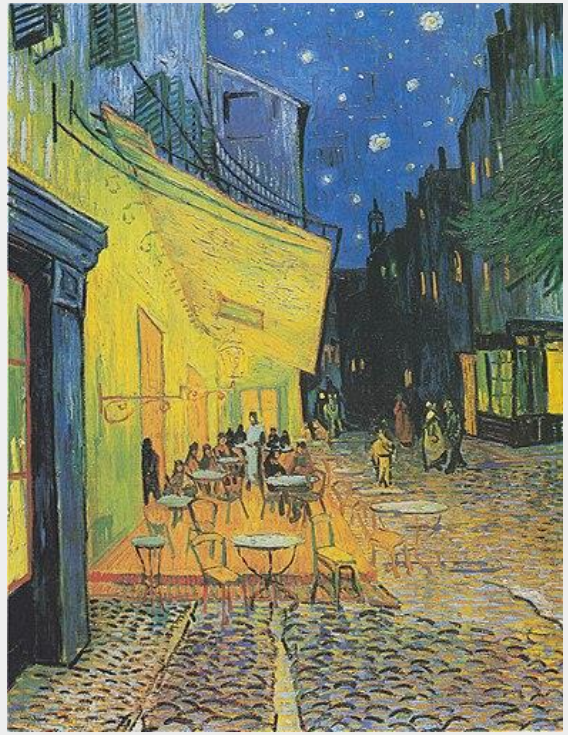
「night」増幅前の画像

作例

- 「van Gogh」ノードを追加
  - 全体のスタイルがvan Goghらしくなった
  - 1888年の「café terrace at night」とは異なる画像が生成できた



「van Gogh」を適用した画像



van Goghの「Café Terrace at Night」
（1888年発表）

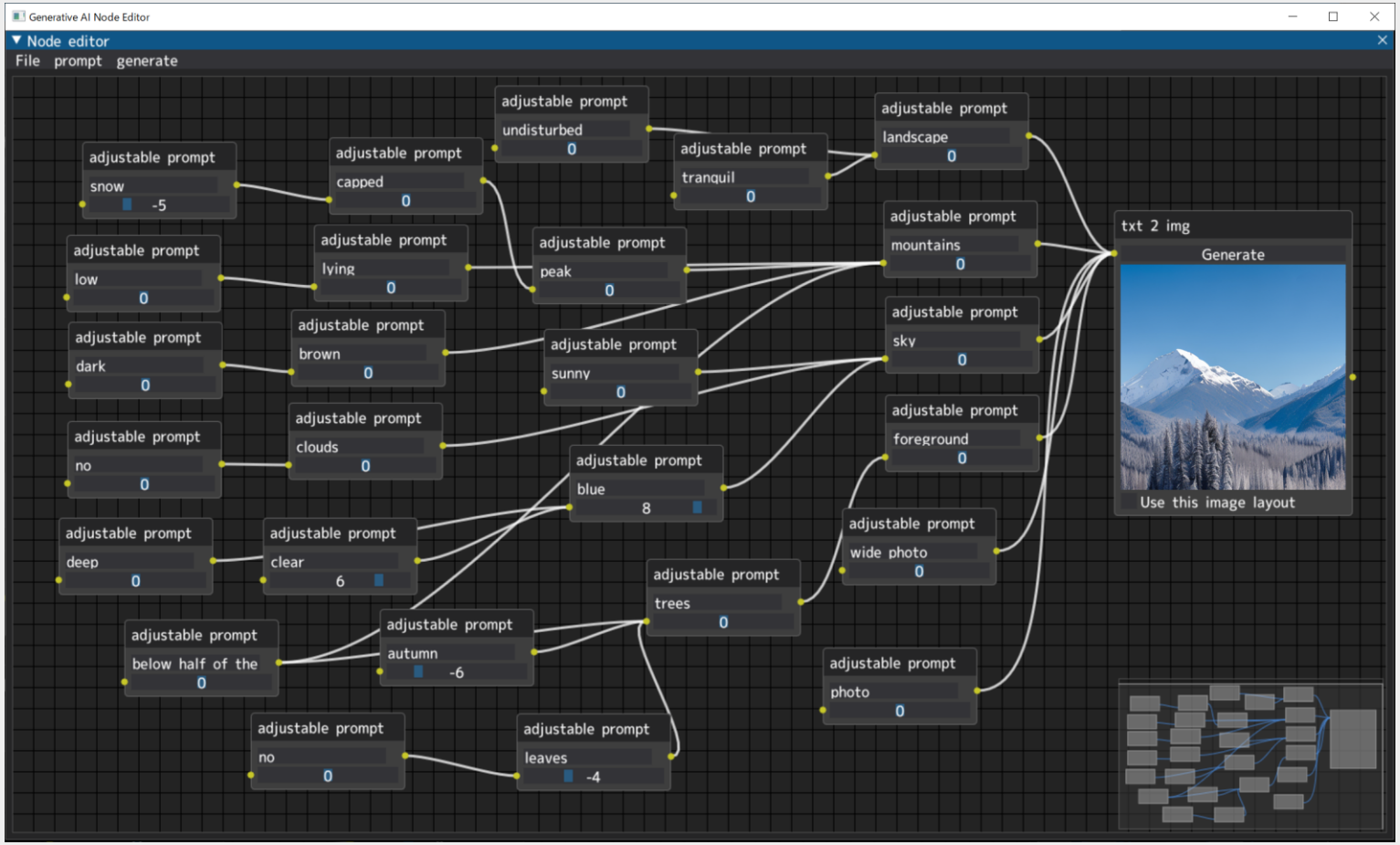
作例

- 学習データにない画像の再現

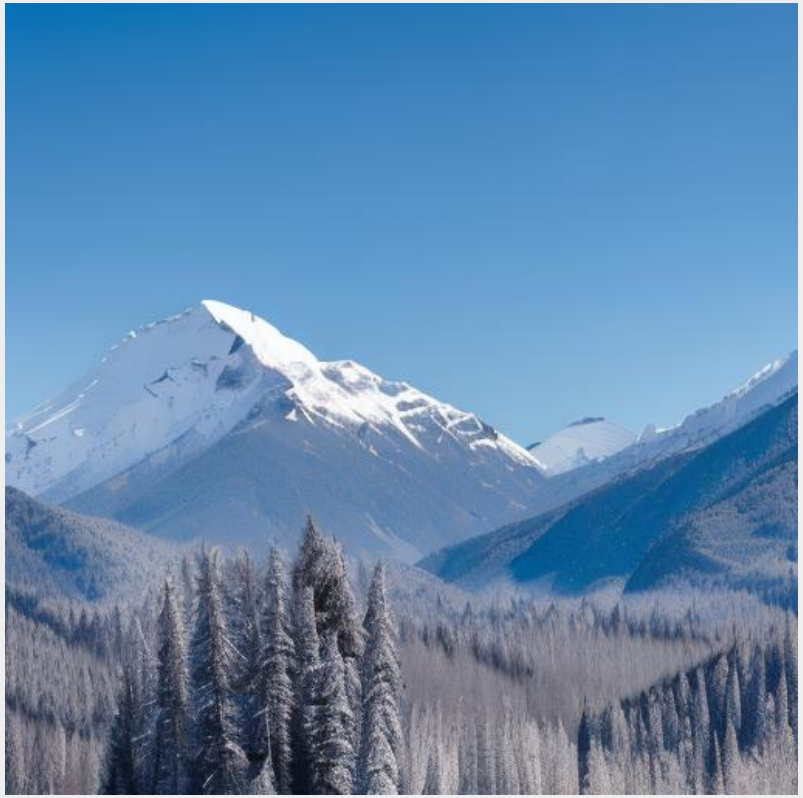
筆者が撮影しどこにもアップロードしていない写真

作例

- 提案システムで再現を試みた際の結果



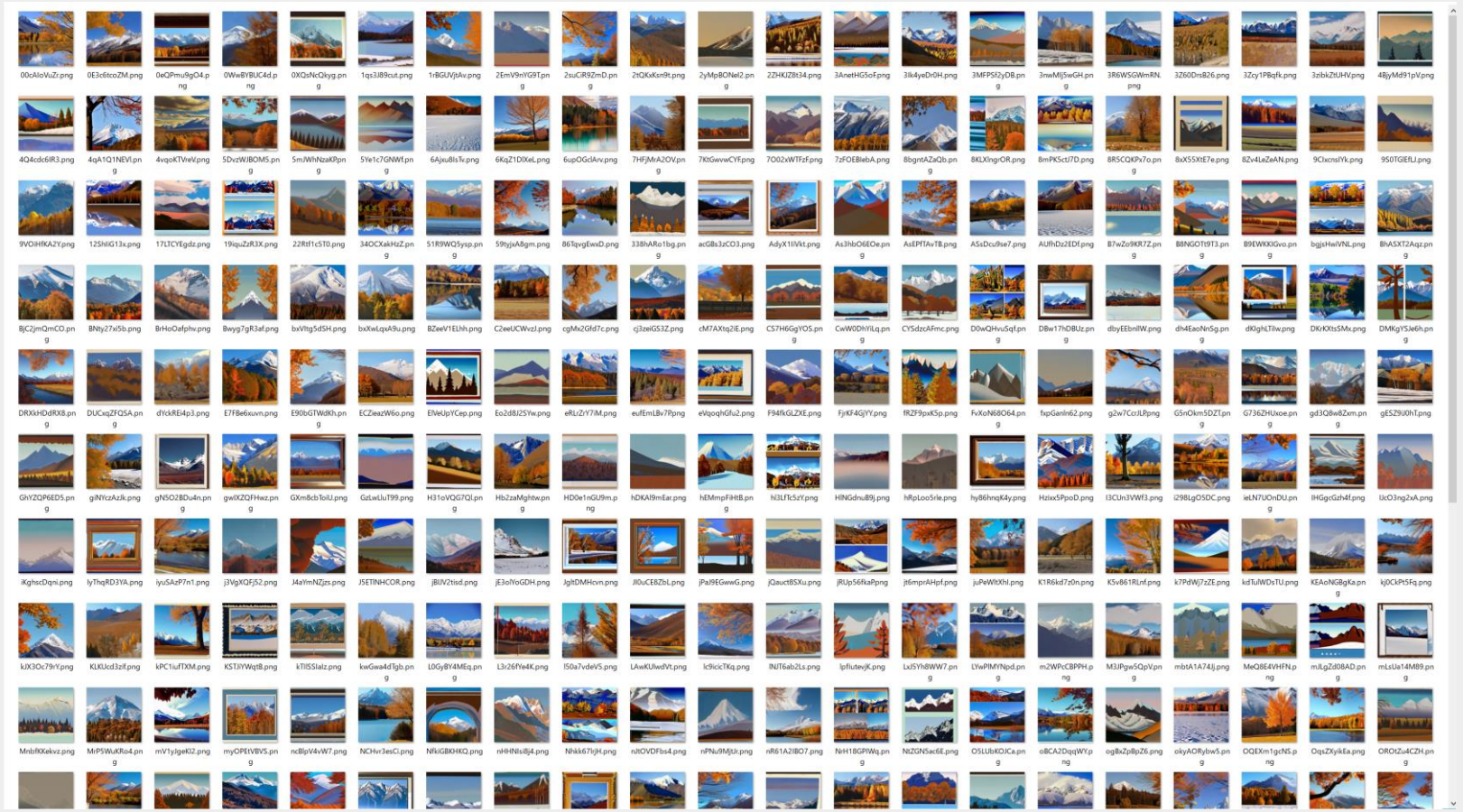
再現を試みた際の様子



編集後の画像

作例

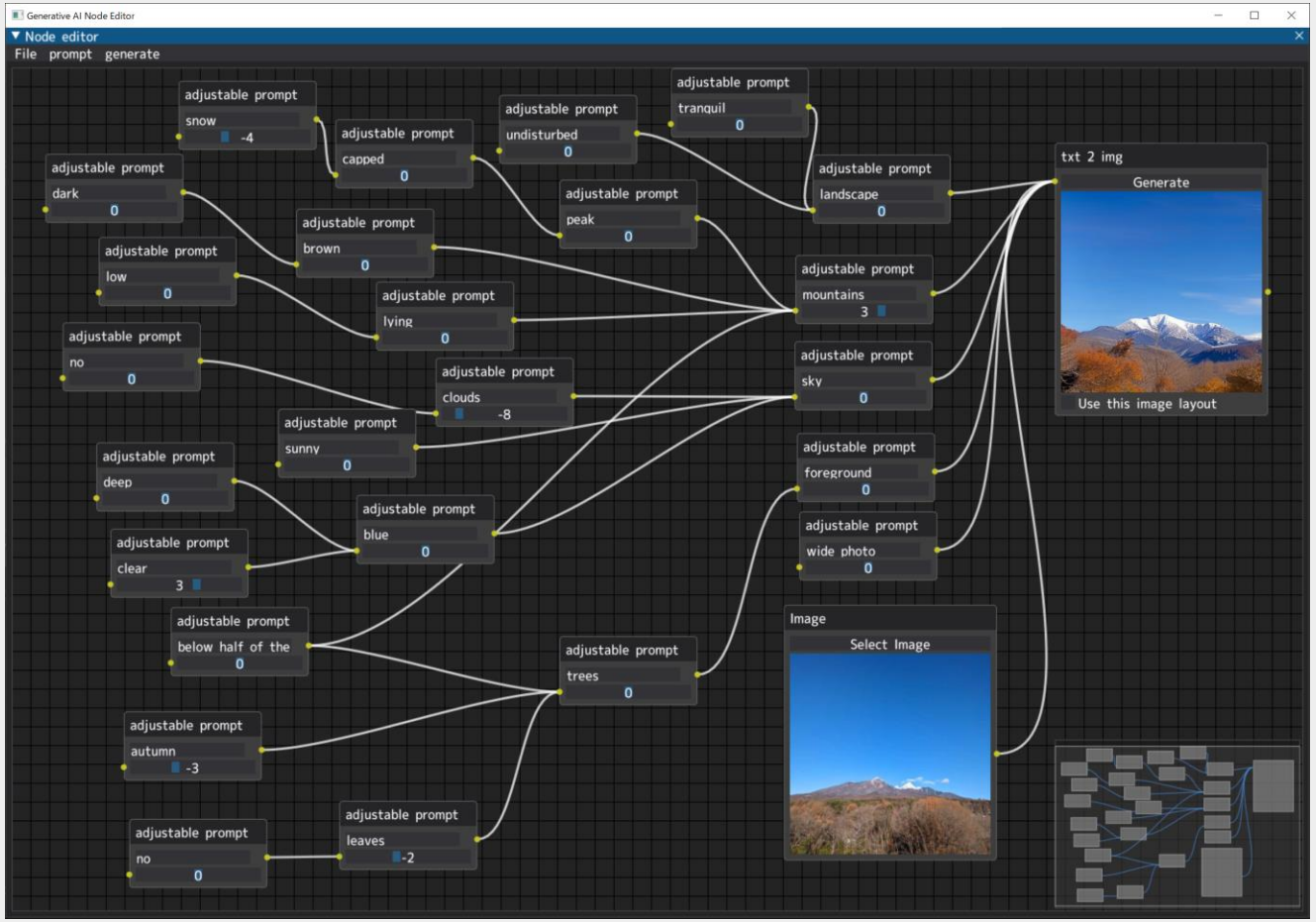
- 提案システムにおいて任意の構図を引き当てることは難しい
  - 任意の構図が来るまで何度も生成するしかない



提案システムで生成した「ハズレ」の画像

作例

- 新規ノードを追加しシステムを拡張



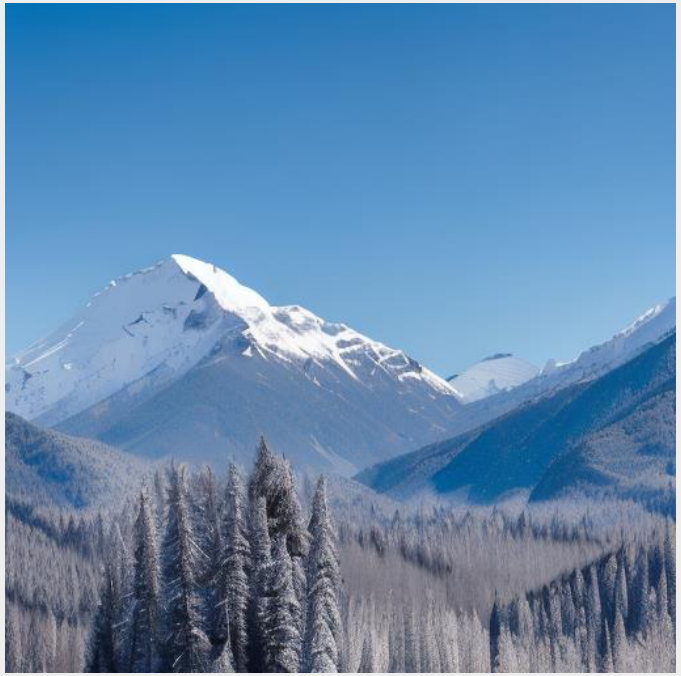
Imageノードを追加して再現を試みた際の様子

編集後の画像

作例

- システムの拡張によってユーザが求める表現へと追い込めるようになる

再現目標



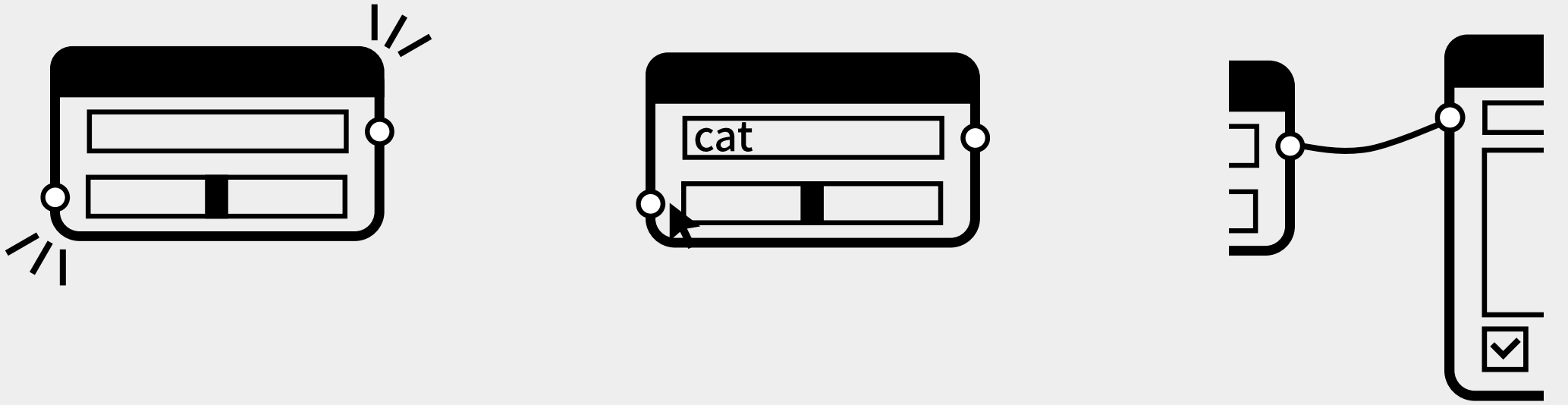
システム拡張前

システム拡張後

議論

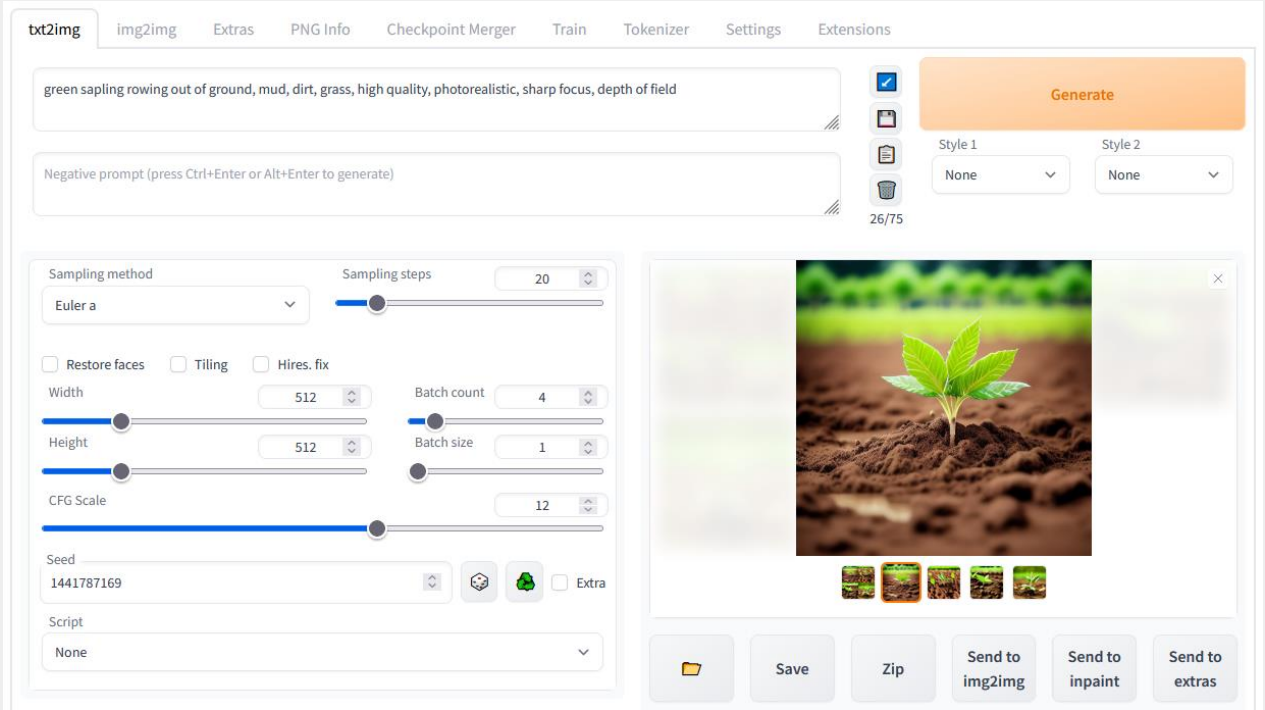
- ノードベースのシステムであることによる手間
  - 画像の生成・編集には最低でも以下の3つの操作が必要

①ノードの追加 ➡ ②テキストの入力 ➡ ③ノードの接続



議論

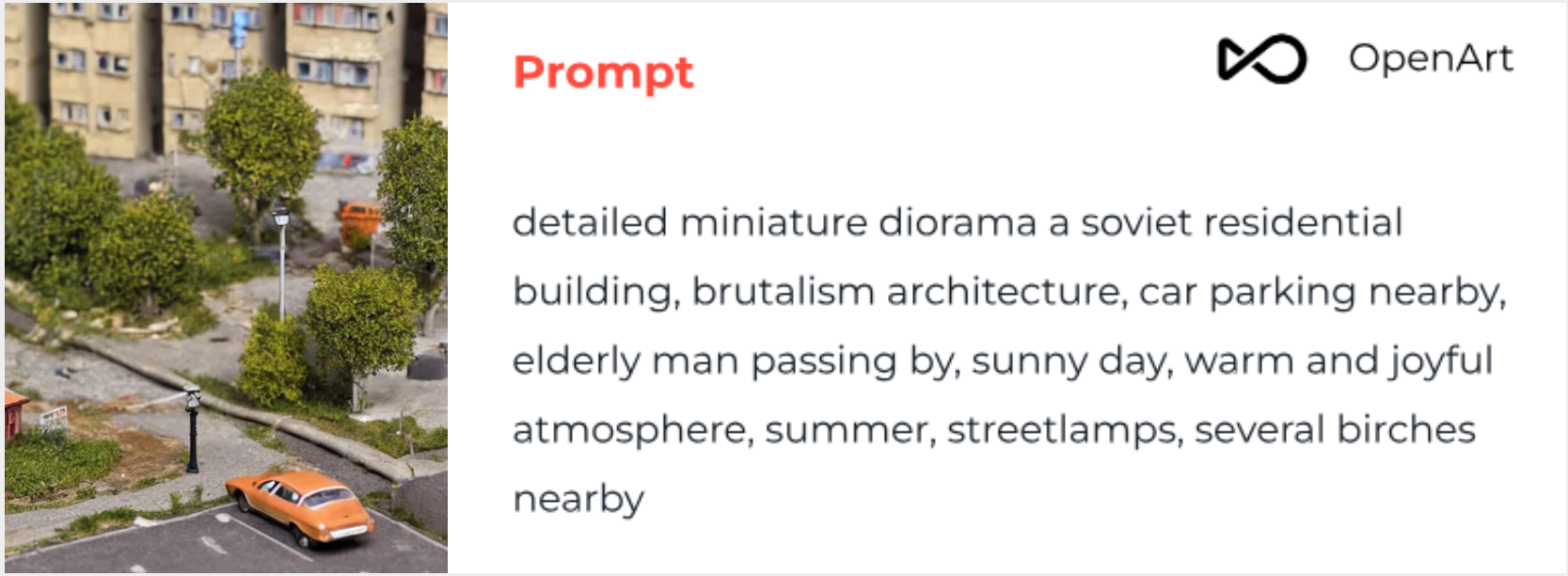
- 従来のシステムはテキストボックスへのテキスト入力だけでOK
  - 提案システムの方が手間が多い
  - 画像編集までを対象とすると提案システムの方が手間が減少する



一般的なテキストボックスベースのシステム ※[5]より引用

議論

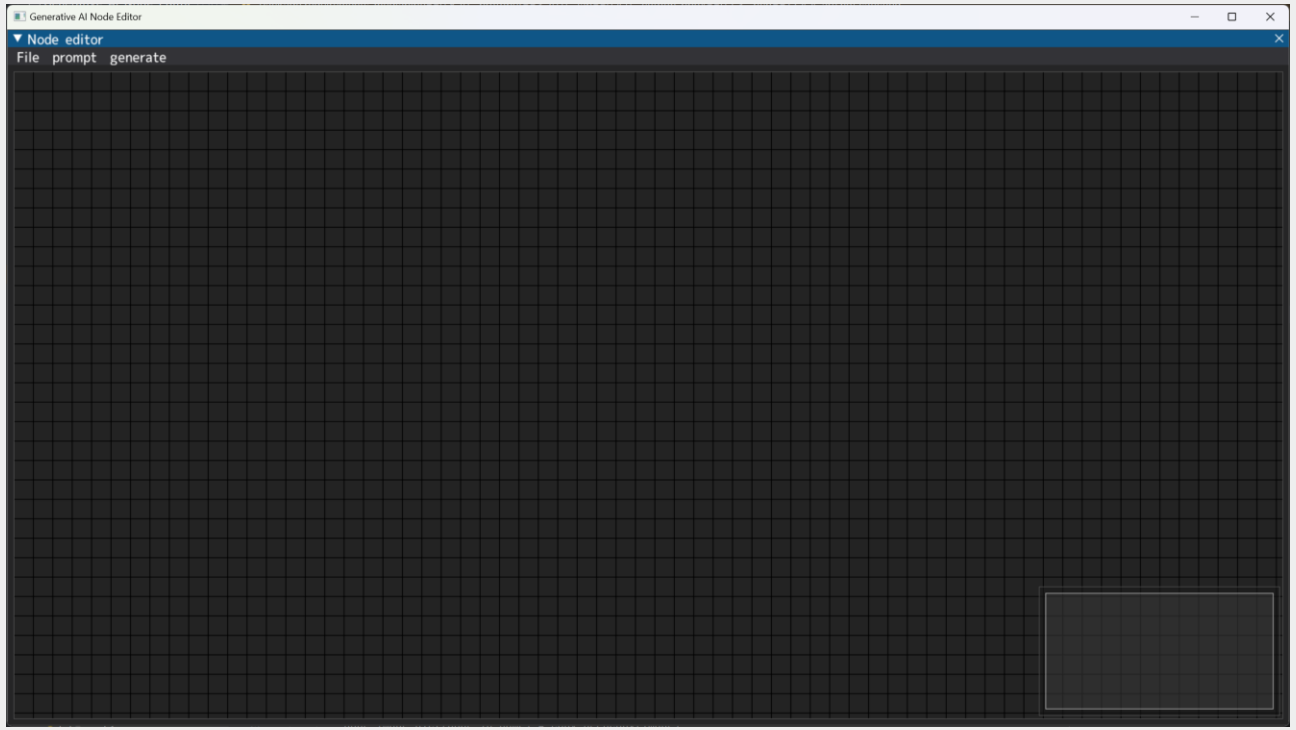
- プロンプトのノウハウをそのまま使えない
  - ノードの形に再構成する必要があるため
    - ↳ プロンプトのコピペができない



画像とプロンプトがセットでシェアされている※[6]より引用

議論

- 提案システムはハードルが高い
  - ↳ とりあえず生成できる状態になるまでが長い



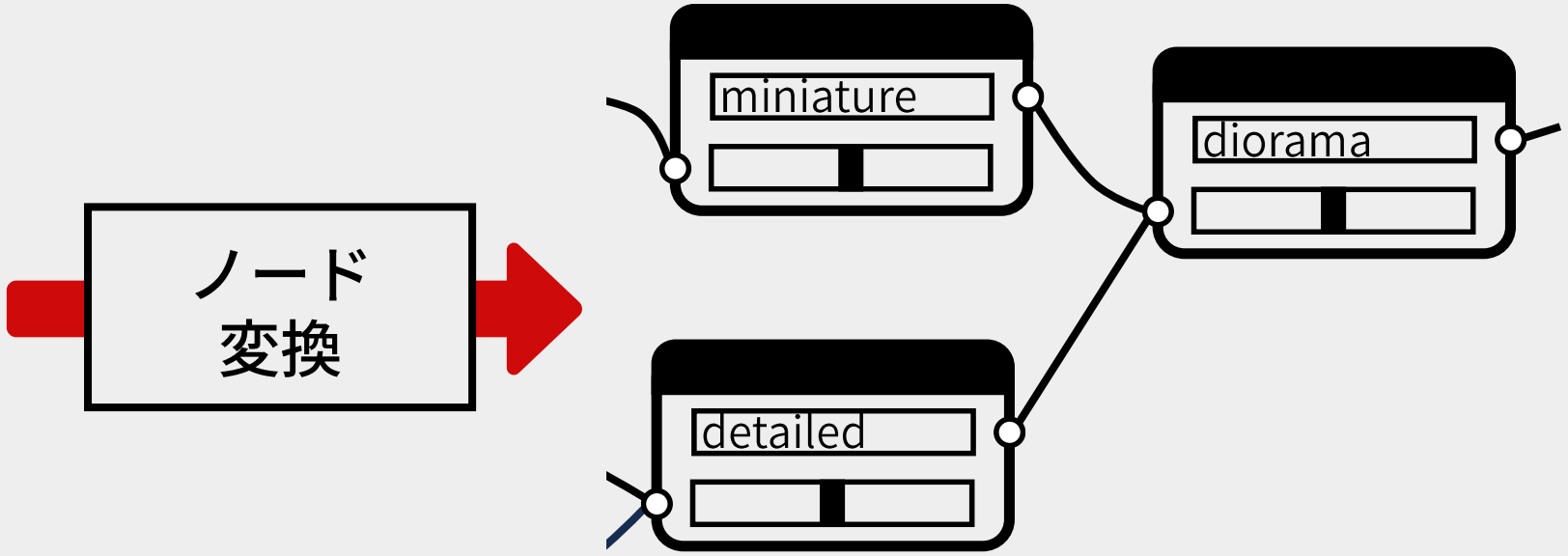
起動直後の提案システムの画面
テキストのコピペだけでは生成できない

議論

- 平文のプロンプトから自動でノードに変換する機能が必要
  - ↳ プロンプトのコピペだけで生成可能

プロンプトtoノードのイメージ

「detailed miniature diorama a
…
several birches nearby」



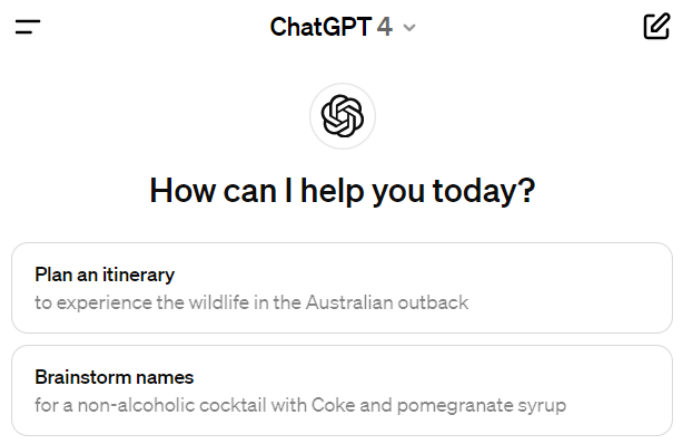
まとめと展望

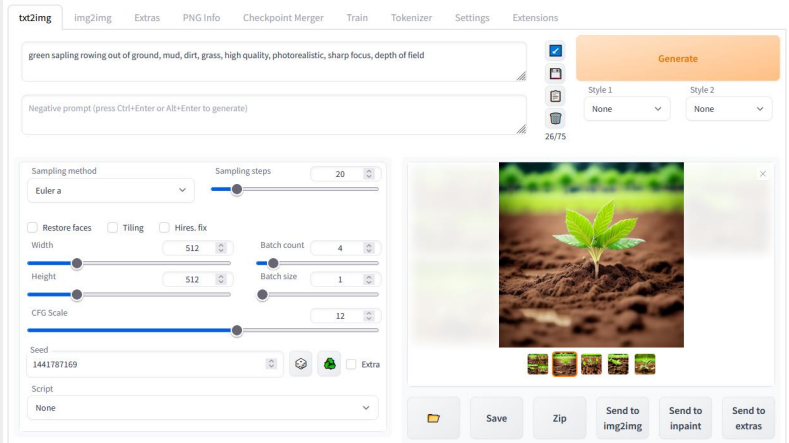
- **画像生成においてユーザが求める表現を追い込めるようにする**
  - テキストボックスとスライダからなるノードを操作して
    画像を生成し編集するシステムの提案

- **作例制作を通してユーザが求める表現へと追い込めるようになることを確認**
  - 学習データと異なるvan Goghの「夜のカフェテラス」
  - 学習データにない画像の再現

まとめと展望

- より多くのAIをノードとして自由に追加し配置できるシステムへの発展
  - 現状はそれぞれのAIを使うためにそれぞれのシステムが必要





- 統合した環境内で使い分け・コラボレーションできるようにしたい
  - ↳ 人間とAI，AI同士の掛け合いによる新たな表現への到達

参考文献　最終アクセス:2024/03/18

[1] Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., & Cohen-Or, D. 2023. Prompt-to-prompt image editing with cross attention control The Eleventh International Conference on Learning Representations.

[2] Tumanyan, N., Geyer, M., Bagon, S., & Dekel, T. 2023. 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 (pp. 1921-1930).

[3] Robin Rombach, et al. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10684–10695, June 2022.

[4] Jonathan Ho, et al. Denoising diffusion probabilistic models. Advances in neural information processing systems, Vol. 33, pp. 6840–6851, 2020.

[5] AUTOMATIC111. stable-diffusion-webui. https://github.com/AUTOMATIC1111/stable-diffusion-webui

[6] OpenArt. Stable Diffusion Prompt Book. https://openart.ai/promptbook

補足―背景

- 単語の順序…前半の単語は画像に反映されやすい

「cat, beach, dog」で生成

「dog, beach, cat」で生成

補足―背景

- 単語同士の係り受け関係…単語の関係が意図していないものになることがある

「white cat and hat」で生成
白くなる対象が生成するごとにばらつく

補足―背景

- 生成した画像の編集はプロンプトではできない
  - ↳ プロンプトの変更と画像の変化は対応しない

「bronze cat sculpture」
で生成

「silver cat sculpture」
で生成

補足―背景

- プロンプトの変化と画像内の変化を対応させる手法が研究されている
  - ↳ 使用には都度スクリプト言語でのプログラミングが必要

「bronze cat sculpture」
で生成

- Prompt-to-Prompt [1]
- Plug-and-Play [2]
  を利用

「silver cat sculpture」
で生成

補足―背景

- プロンプトの変化と画像内の変化を対応させる手法が研究されている
  - ↳ 使用には都度スクリプト言語でのプログラミングが必要

```python

if __name__ == '__main__':
    # コマンドライン引数からdict情報が入ったpickleのパスを取得
    parser = argparse.ArgumentParser()
    parser.add_argument( *name_or_flags: "--config_path", metavar="<path>")
    args = parser.parse_args()
    config_path = args.config_path

    with open(config_path, "r", encoding='utf-8') as f:
        config = yaml.safe_load(f)
    os.makedirs(config["output_path"], exist_ok=True)
    with open(os.path.join(config["output_path"], "config.yaml"), "w") as f:
        yaml.dump(config, f)

    seed_everything(config["seed"])
    print(config)
    pnp = PNP(config)
    pnp.run_pnp()
```

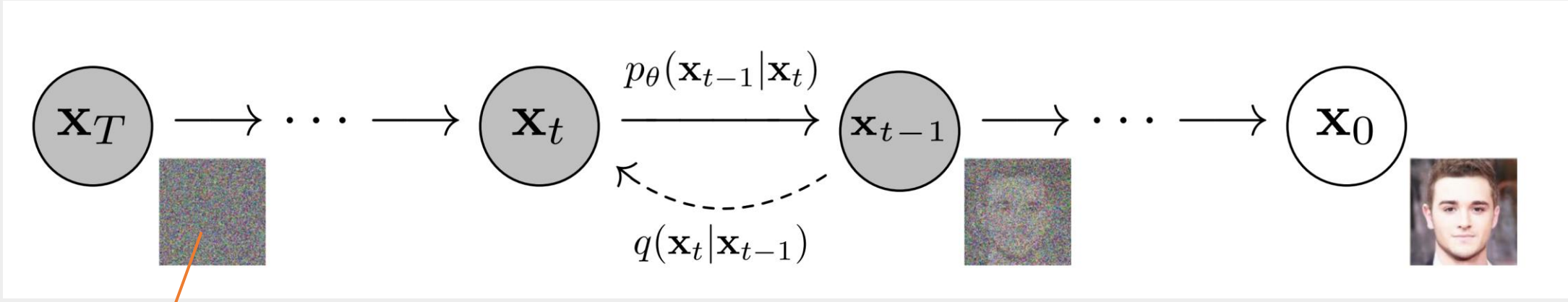
```python
    # wordとvalueのタプルを作成する
    word_list = []
    value_list = []
    for key in text_value_dict:
        print(text_value_dict[key][1])
        word_list.append(key)
        value_list.append(text_value_dict[key][1])
    word_tuple = tuple(word_list)
    value_tuple = tuple(value_list)

    # x-tのload
    old_x_t = torch.load(path)
    # promptの用意
    prompts = [args.base_prompt] * 2

    equalizer = get_equalizer(prompts[1], word_select: ("silver",), values: (5,))
    controller = AttentionReweight(prompts, NUM_DIFFUSION_STEPS, cross_replace_steps=.8,
                                   self_replace_steps=.4,
                                   equalizer=equalizer)
    _, xt = run_and_display(prompts, controller, img_name, adjust, latent=old_x_t, run_baseline=False)
```

実際に記述したプログラムの一部

補足―関連研究

Diffusion Modelによる画像生成の流れ

Reverse Process



Forward Process

seed値により決定

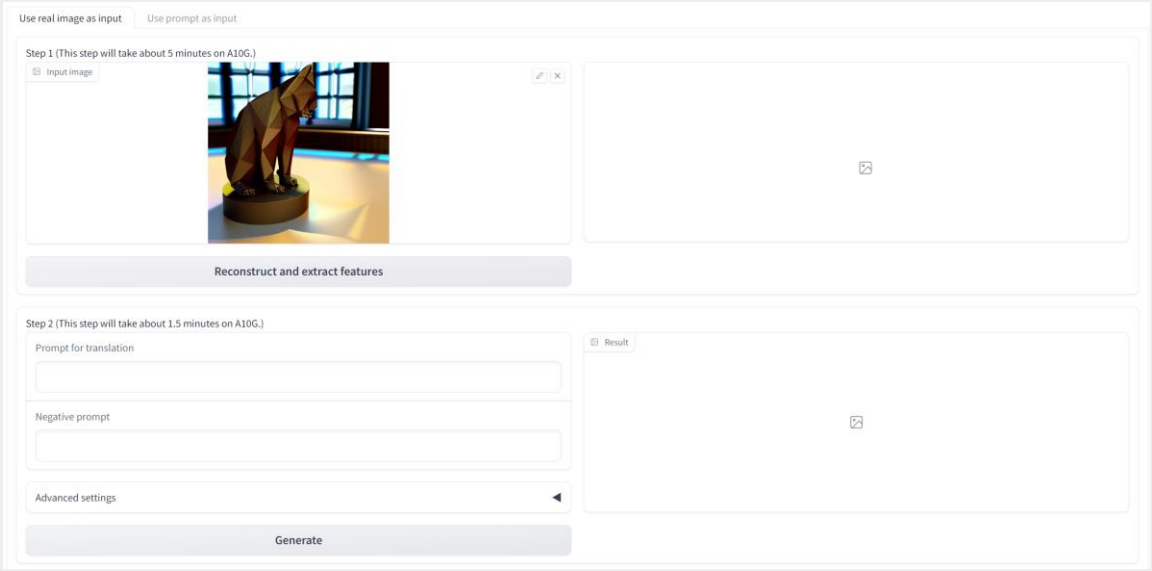
補足―関連研究

- プロンプトの編集のみで生成する画像を制御する手法
  - 都度スクリプト言語でのプログラミングを要する

```
prompts = ["A painting of a squirrel eating a burger",
           "A painting of a lion eating a burger"]

controller = AttentionReplace(prompts, NUM_DIFFUSION_STEPS, cross_replace_steps=.8, self_replace_steps=0.4)
_ = run_and_display(prompts, controller, latent=x_t, run_baseline=True)
```

```
prompts = ["a smiling bunny doll"] * 2

### pay 3 times more attention to the word "smiling"
equalizer = get_equalizer(prompts[1], ("smiling",), (5,))
controller = AttentionReweight(prompts, NUM_DIFFUSION_STEPS, cross_replace_steps=.8,
                               self_replace_steps=.4,
                               equalizer=equalizer)
_ = run_and_display(prompts, controller, latent=x_t, run_baseline=False)
```

Prompt-to-Promptを利用するための
プログラム例[7]



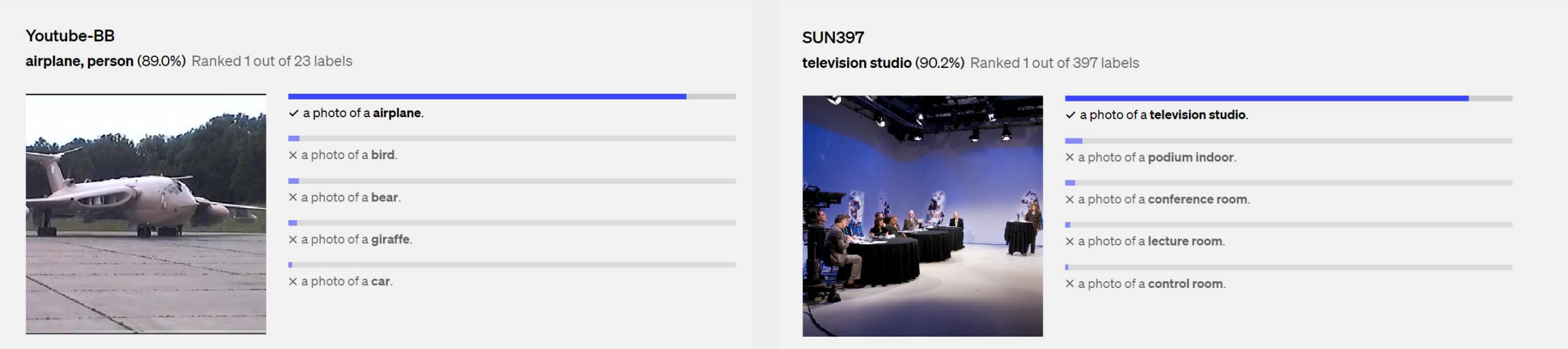
Plug-and-Playのデモページ[8]

[7] amirhertz. prompt-to-prompt_stable.ipynb. https://github.com/google/prompt-to-prompt/blob/main/prompt-to-prompt_stable.ipynb. (Accessed on 2024/03/18).

[8] hysts. hysts/PnP-diffusion-features. https://huggingface.co/spaces/hysts/PnP-diffusion-features . (Accessed on 2024/03/18).

補足―作例

- CLIPスコア
  - CLIP…言語と画像のマルチモーダルモデル
  - CLIPスコア…テキストと画像の類似度



画像の内容をもっともよく表すテキストに高いスコアを示している ※[9]より引用

[9] OPEN AI. CLIP: Connecting text and images. https://openai.com/research/clip (Accessed on 2024/03/18).

補足―作例

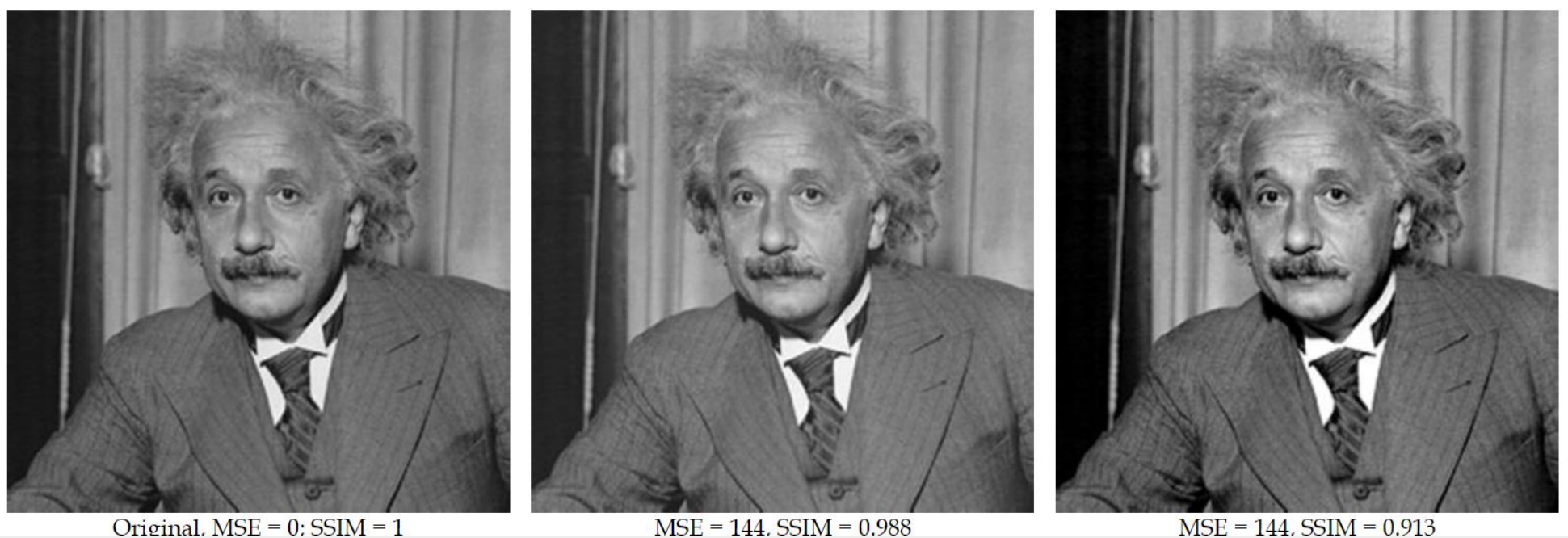
- CLIPスコアによる検証
  - いずれの画像もStable Diffusionから直接得にくい画像であることが分かった

Van Gogh's Café terrace at night	0.24	0.10	0.04
Café terrace at night	0.76	0.90	0.96

特に右の画像とのスコアが一番低い

補足―作例

- SSIM（Structual SIMilarity）
  - 画像の輝度・コントラスト・構造を考慮した画像の評価手法

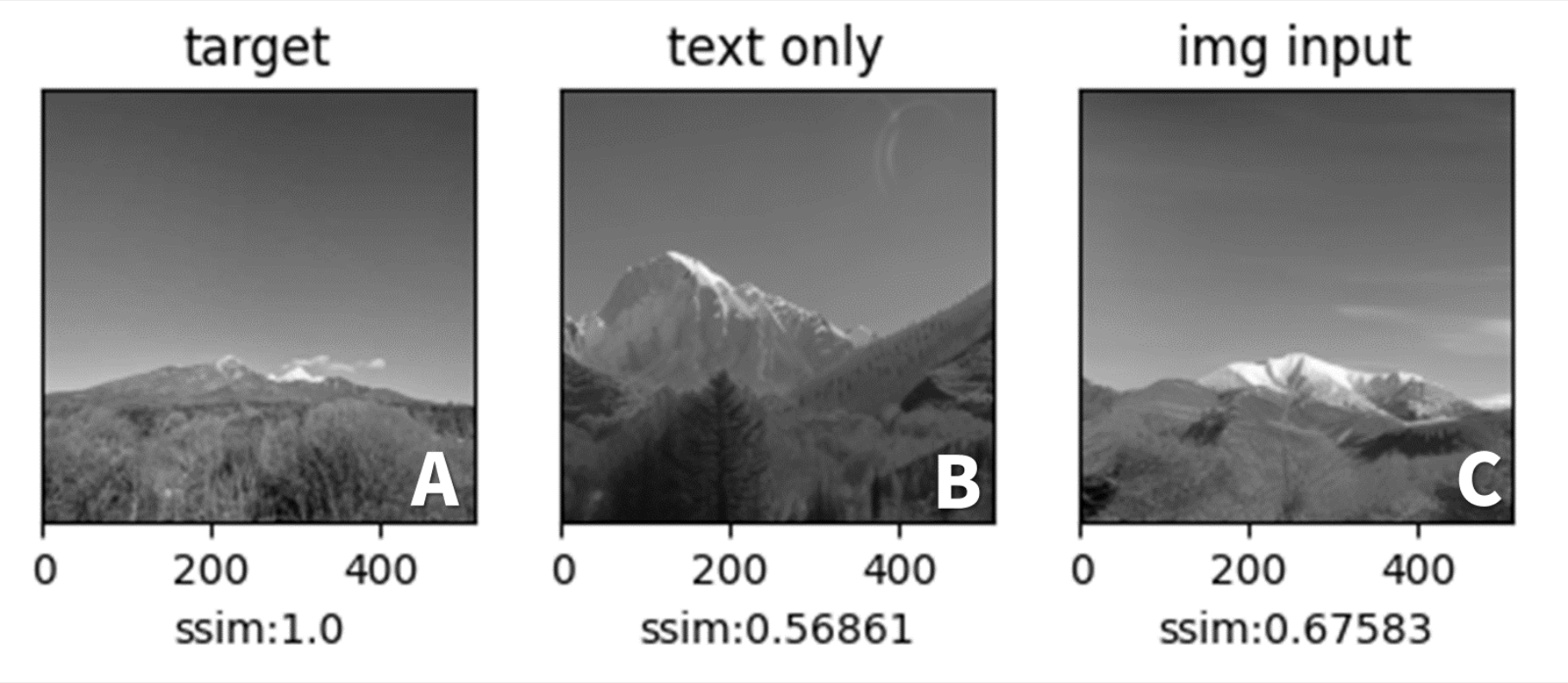


SSIMの算出例 ※[10]より引用

[10] Wang et al. The SSIM Index for Image Quality Assessment. https://ece.uwaterloo.ca/~z70wang/research/ssim/ (Accessed on 2024/03/18).

補足―作例

- SSIMの算出結果
  - システムを拡張して制作した画像の方が高い値を示した



図内Aは再現目標，BとCは再現した画像