画像生成モデルにおける直喩喩体の生成挙動分析

王 略丞 [*1]　尾崎 慎太郎 [*2]　上垣外 英剛 [2]　林 克彦 [1]　Kwon Jingun[3]　奥村 学 [4]　渡辺 太郎 [2]
[1] 東京大学　[2] 奈良先端科学技術大学院大学 (NAIST)　(*: 共同第一著者)
[3]Chungnam National University　[4] 東京科学大学
{wanglc,katsuhiko-hayashi}@g.ecc.u-tokyo.ac.jp
{ozaki.shintaro.ou6, kamigaito.h, taro.watanabe}@naist.ac.jp

概要

画像生成モデルは，テキスト記述に基づいて高品質な画像を生成し，複雑な言語表現への追従性も向上している．特に，メタファーのような非字義的な表現が与えられた場合，喩体を字義的に具象化するか抽象的に処理するかは，画像生成モデルの言語理解能力を評価する上で重要な課題である．本研究では，直喩に限定したメタファー文を対象とし，その生成および評価のための枠組みを提案する．具体的には，複数の大規模言語モデルを用いて直喩文を生成し，多モデルによる正誤判定を通じて，YOLO による物体検出が可能な名詞を喩体とする直喩文データセットを構築する．このデータセットを用いて，複数の画像生成モデルに対して画像生成および物体検出を行う．さらに，Diffusion Lens を用いて生成過程の内部挙動を分析し，喩体が生成過程のどの段階で出現するかを調査した．分析の結果，多くの文において，喩体は生成過程の途中で一時的に検出されるが，最終生成画像には必ずしも反映されないことが確認された．さらに，その出現タイミングにはテキストエンコーダ間で系統的な差異が観察された．

1　はじめに

画像生成モデルの急速な進展により，自然言語記述から高品質かつ多様な画像を生成できるようになっている．これらのモデルは字義的な表現に対して高い追従性を示す一方で，非字義的な表現，とりわけメタファーを入力とした場合に，喩体がどのように扱われるのかは，画像生成モデルの持つ言語能力を評価する上で本質的な問いである．

メタファーは，人間の言語理解において重要な表現形式であり，異なる概念の対応関係に基づいて意味が構成される (例: 心に重い石を抱えている)．このような表現が画像生成モデルに与えられた場合，喩体が字義的な物体として具象化されるのか，あるいは場面全体の関係性や属性として処理されるのかは，モデルの言語理解能力を考える上で重要である．先の「心に重い石を抱えている」という表現を例にとると，「重い石」は生成されず憂鬱な表現を画像として生成することが望ましい．

これまで，メタファー理解に関する研究は主に言語モデルを対象として行われてきたが [1]，画像生成モデルを対象とし，メタファー表現が生成過程に与える影響を体系的に分析した研究は限られている．さらに，T5 [2] と CLIP [3] のような異なるテキストエンコーダを用いた画像生成モデルにおいて [4, 5]，喩体の具象化が生成過程のどの段階で生じるかについては，十分に検討されていない．

本研究では，直喩形式に限定したメタファー文を対象とする．直喩は喩体が明示的に表現されるため，画像生成モデルによる喩体の処理を制御的に分析しやすい利点がある．そこで，図 1 に示すように，YOLO [6] による物体検出が可能な名詞を喩体とする直喩文データセットを構築し，画像生成結果に対する物体検出と Diffusion Lens [7] による生成過程の内部挙動分析を組み合わせた枠組みを提案する．

実験の結果，喩体の具象化傾向にはモデルやテキストエンコーダ間で差が見られ，最終画像に現れない場合であっても，生成過程の途中では一時的に具象化される例が多く観察された．このことから，喩体の扱いが生成過程の時間的推移の中で動的に変化することが明らかとなった．

2　関連研究

メタファーと非字義的意味処理　非字義的意味の処理は，自然言語処理において長年にわたって重要な課題とされてきた [1, 8]．非字義表現は，表層的な語の意味を超えた解釈を必要とするため，モデルの意味理解能力を検討する対象として広く用いられ

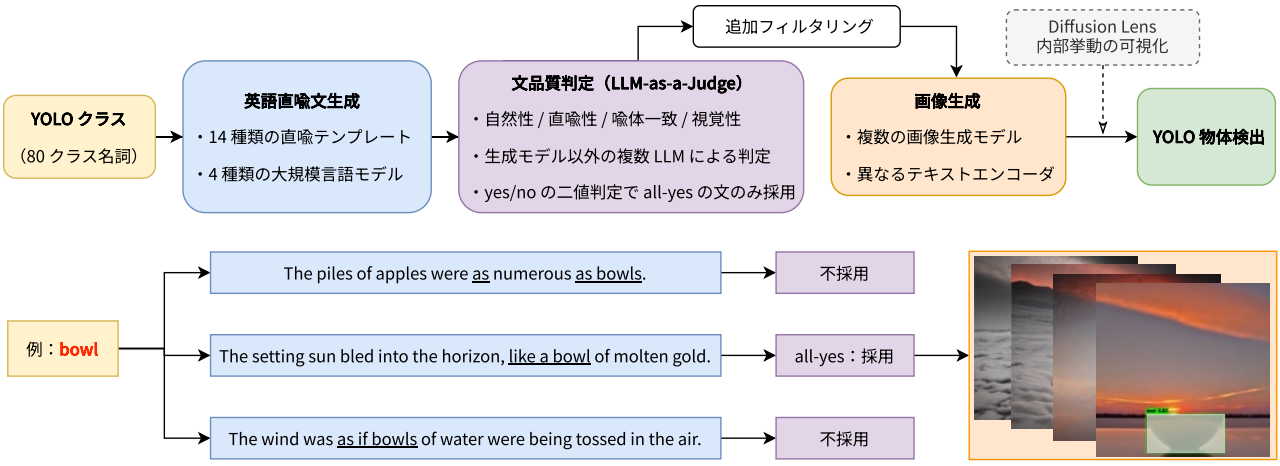


図 1　全体パイプラインの概要．YOLO のクラス名詞を起点とし，直喩文の生成，画像生成，Diffusion Lens の適用，ならびに YOLO による物体検出までの処理の流れを示す．下段には，1 つの YOLO クラスに対する処理例を記載している．

ている [9]．その中でも，メタファーは異なる概念領域間の対応関係に基づいて意味が構成される表現であり [10, 11]，非字義的意味処理を議論する上で代表的な現象とされている [12]．

近年，大規模言語モデル（LLM）の発展に伴い，メタファーの識別，判断，生成といったタスクにおいて高い性能が幅広く報告されている [13, 14, 15]．一方で，多くの研究はテキストのみを対象としており，視覚的生成を伴うモデルにおけるメタファー処理については十分に検討されていない [16]．特に，メタファー文を処理する場合に，喩体が視覚的にどの段階で活性化されるのかは，体系的な分析がほとんど行われていない [17]．

画像生成モデル　Stable Diffusion [5] に代表される画像生成モデルは，入力された任意のプロンプトを U-Net [18] や Diffusion Transformer [19] などを通じて画像を生成する．入力されたプロンプトを適切に表現するために，様々なモデルが文埋め込みのためのエンコーダとして用いられる．例えば，Dreamlike [20] は CLIP [3] のテキストエンコーダを，PixArt [21] は T5 [2] のエンコーダ側を用いている．さらに Stable Diffusion [22] や FLUX [4] などはその両方を用いることで表現力を豊かにしている．LLM の登場以降，デコーダモデルをエンコーダとして用いた Qwen-Image [23] なども登場している．

テキストエンコーダによる違いによって，生成結果に大きな影響を与えることが知られている [24]．テキストエンコーダがどのように中間層で表現をしているか解釈するために，Toker ら [7] は層ごとの表現から画像を生成する手法である Diffusion Lens を提案し，モデルがどのようにして言語能力を画像に反映させるか確認することを可能にした．

3　データセット構築

本研究では (1) テンプレート設計，(2) LLM による直喩文の生成，(3) フィルタリングの手順によってデータセットを構築する．

(1) 喩体語彙とテンプレート設計．　画像生成結果に対して客観的な評価を行うため，物体検出モデル YOLO [6] によって検出可能な 80 クラスの名詞 [25] を喩体語彙として用いた．これにより，生成画像において喩体が字義的な物体として具象化されているかを，物体検出に基づいて評価可能とした．また，直喩表現における表層的なばらつきを抑えるため，喩体を明示的に含む 14 種類のテンプレートを設計した．詳細は付録 B に示す．

(2) 直喩文の生成．　直喩文の生成には，指示学習された複数の LLM を用いた．すべてのモデルに対して同一のテンプレートおよび生成設定を統一している．生成時のプロンプトでは，直喩表現の使用，喩体の明示的指定，および具体的かつ視覚的な場面描写を制約として与えることで，生成文の構造的な一貫性と分析上の制御性を確保している [26]．生成および判定に用いた言語モデルの詳細ならびに実行設定については，付録 A に示す．

(3) 多モデル判定とフィルタリング．　生成された直喩文に対しては，後続の評価を安定させるため，生成時の引用符や強調記号などの表層的表記を除去し，文の内容を保持したままフィルタリングを行った．その上で，生成文の品質を担保するため，

LLM-as-a-judge [27, 28] に倣った生成に用いたモデル自身を除いた言語モデルによる判定を実施した．

各判定モデルは，生成文に対して (i) 文として自然かつ文法的に正しいか，(ii) 直喩表現を含んでいるか，(iii) 指定された名詞が喩体として用いられているか，(iv) 視覚的に具体化可能な場面を記述しているか，の 4 観点について *yes/no* 判定を行った．すべての観点についてすべての判定モデルから *yes* が得られた文のみを採用し，それ以外は除外した．さらに，直喩として本体が不明確な表現や，喩体が複数回出現する文については，画像生成および物体検出との整合性を考慮し，追加的に除外した．この過程により残った 1,139 件を用いて実験を行った．

4 画像生成とその評価

画像生成モデル． 本研究ではテキストエンコーダの構造の違いを考慮して T5 にエンコーダを持つ PixArt [21]，CLIP をエンコーダにもつ Dreamlike [29]，その両方を持つ FLUX [4] と Stable Diffusion 3.5 (SD3.5 と表記) [22]，LLM を用いた Qwen-Image [23] の 5 種類の画像生成モデルを用いて同一の直喩文に対する画像生成を行う．詳細な設定は付録 A に記載する．

Diffusion Lens による生成過程分析． 喩体の生成挙動を生成過程の時間的側面から把握するため，2 節で説明した Diffusion Lens [7] を用いて，テキストエンコーダの各層に対応する生成結果を可視化する．本研究では，テキストエンコーダの層ごとの表現を単一系列として明示的に取得可能な設定を対象とする．下記の式で表される:

$$\mathbf{Img}^{(l)} = \mathrm{Diff}\Big(\mathrm{LayerNorm}\Big(\mathbf{h}^{(l)}\Big)\Big), \quad l = 1, \ldots, L \quad (1)$$

ここで，$\mathbf{h}^{(l)}$ はテキストエンコーダの l 層目の隠れ状態，LayerNorm$(\cdot)$ は最終層正規化，Diff$(\cdot)$ はその表現を条件として画像を生成する拡散モデル，$\mathbf{Img}^{(l)}$ は層 l の表現から生成された画像を表す．

物体検出による評価． 生成された画像における喩体の字義的具象化を定量的に評価するため，本研究では物体検出モデルとして YOLO11 [6] を用い，直喩文中で指定された喩体名詞に対応するクラスが検出されたかどうかを主要な評価指標とする．

生成画像において喩体に対応する物体が検出された場合を，喩体が字義的に具象化された状態として扱い，検出されなかった場合を，視覚的には具象化されなかった状態として解釈する．同様の物体検出を，Diffusion Lens により各層の表現を条件として生成された拡散過程の各生成ステップに対応する中間画像にも適用し，喩体が生成過程のどの段階で初めて検出されるかを記録することで，生成過程の時間軸上における喩体の具象化挙動を分析する．

表 1 テンプレート別の YOLO における喩体の検出率 (%)．N はデータの数を表す．

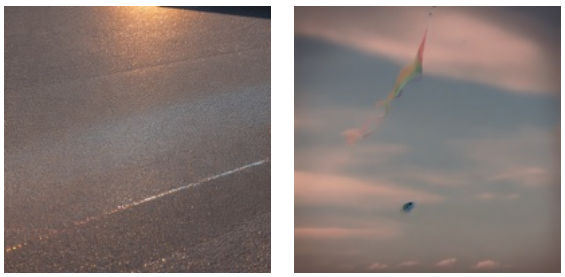
テンプレート	N	Dreamlike	FLUX	PixArt	Qwen-Image	SD3.5
as_as	52	**26.92**	40.38	55.77	51.92	55.77
as_as_plural	38	**28.95**	57.89	63.16	44.74	65.79
as_if	68	47.06	**44.12**	66.18	52.94	55.88
as_if_plural	50	**50.00**	66.00	64.00	72.00	76.00
as_though	75	**46.67**	56.00	69.33	61.33	60.00
as_though_plural	35	65.71	65.71	**57.14**	68.57	74.29
exactly_like	70	**22.86**	38.57	55.71	50.00	45.71
exactly_like_plural	70	**28.57**	31.43	52.86	72.86	41.43
just_like	97	**19.59**	26.80	42.27	46.39	39.18
just_like_plural	111	**25.23**	41.44	42.34	54.95	50.45
like	93	**21.51**	**21.51**	37.63	41.94	32.26
like_plural	108	**19.44**	32.41	39.81	61.11	47.22
look_like	133	**27.07**	35.34	42.11	52.63	38.35
look_like_plural	139	**23.74**	35.97	52.52	64.75	47.48
Total	1,139	**29.24**	38.98	50.31	56.45	48.64

表 2 クラス別ににおける物体検出率．80 クラスのうち，データ事例数 N の多い上位 10 件を記載している．

クラス	N	Dreamlike	FLUX	PixArt	Qwen-Image	SD3.5
oven	30	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
bowl	29	27.59	20.69	68.97	20.69	**13.79**
knife	29	27.59	**17.24**	24.14	27.59	**17.24**
kite	26	**69.23**	**57.69**	73.08	80.77	73.08
potted plant	24	**62.50**	83.33	79.17	70.83	79.17
vase	24	66.67	54.17	**50.00**	91.67	83.33
wine glass	24	**37.50**	70.83	70.83	87.50	70.83
teddy bear	23	**26.09**	30.43	86.96	56.52	78.26
cup	22	**22.73**	50.00	54.55	36.36	40.91
umbrella	22	**27.27**	54.55	54.55	59.09	81.82

5 結果と分析

最終層での画像における喩体の検出． 表 1 より，YOLO を用いた喩体のトータルの検出率にはモデル間で顕著な差が見られた．Dreamlike および FLUX では検出率が 30–40% 程度に抑えられている一方，PixArt，Qwen-Image，および SD3.5 では，50%前後の画像において喩体が検出されている．表 1 に，直喩文の生成テンプレートごとの検出率の違いを示す．この結果から，Dreamlike は直喩文の形式によらず，検出率を相対的に低く抑えられており，直喩文の理解度の高さを示す結果となった．一方で，as_if や



(a) oven (b) kite

図 2 Dreamlike による生成例. (a) では喩体は具象化されず，(b) では物体として生成されている．

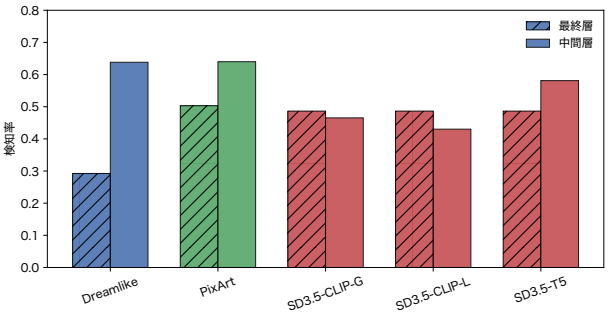


図 3 最終生成画像と生成過程における喩体検出率の比較．各モデルについて，最終画像における検出率（Final）と，Diffusion Lens により得られた生成過程全体で一度でも検出された割合（Process）を示す．SD 3.5 については，異なるテキストエンコーダ設定（CLIP-G/ CLIP-L/ T5）ごとに Diffusion Lens を適用した．

as_though のようなテンプレートでは，多くのモデルにおいて他のテンプレートよりも相対的に高い検出率が観察された．これらのテンプレートは，仮定的な状況や知覚的な描写を導入する表現であり，喩体が場面内の具体的な対象として解釈されやすい可能性がある．

表 2 には，全 80 クラスのうち，データ事例数の多い上位 10 クラスの喩体検出率を示した．この結果から，クラスごとの検出率がモデルによらず大きく異なることが分かる．例えば，oven は全てのモデルで喩体が検出されなかったが，kite では多くの喩体が検出される結果となった．図 2 に，oven と kite の直喩文を Dreamlike へ入力した際に生成された画像例を示す．これらの事例から，背景などと親和性高く登場しやすいクラスの物体は明確な形状を持って具現化される可能性が示唆される．

生成過程における喩体の出現． 喩体検出率の図 3 より，多くのモデルにおいて，生成過程における検出率（Process）は最終画像における検出率（Final）を上回っている．特に Dreamlike では，最終画像における検出率が 3 割程度にとどまる一方で，生成過程全体では約 6 割以上の文において喩体が一度は検出されている．この結果は，多くの文において，喩体が生成過程の途中で一時的に具象化される一方，最終的な生成結果には必ずしも反映されないことを示している．

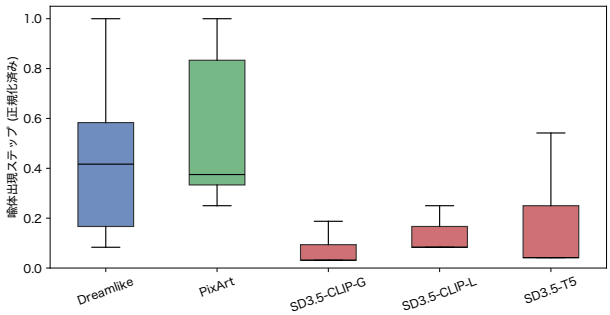


図 4 生成過程における喩体の初期出現ステップの分布．Diffusion Lens による中間生成画像に対して物体検出を適用し，喩体が初めて検出された生成ステップを，各モデルの総ステップ数で正規化して示す．

次に，生成過程における喩体の初期出現ステップの分布図 4 より，Dreamlike および PixArt では，喩体の初期出現が生成過程の中盤から後半にかけて広く分布しているのに対し，SD3.5 の一部設定では，より初期の生成ステップにおいて喩体が検出される傾向が見られる．

喩体の具象化が最終出力の段階で一度に決定されるのではなく，生成過程の時間的推移の中で動的に生起し，その後の生成によって抑制または消失する場合があり，メタファー処理が時間的側面を持つ動的な過程であることを示唆している．

6 おわりに

本研究では，直喩形式に限定したメタファー文を対象として，画像生成モデルが喩体をどのように処理するかを分析した．喩体として物体検出可能な名詞を用いたデータセットを構築し，最終生成画像に対する物体検出と，Diffusion Lens による生成過程分析を組み合わせることで，メタファー処理の時間的側面に着目した評価枠組みを提案した．

実験の結果，喩体は最終画像に必ずしも反映されない一方で，生成過程の途中では一時的に具象化される例が多く観察された．また，この挙動にはテキストエンコーダの設計に応じた系統的な差異が見られ，メタファー処理が生成過程の中で動的に制御されている可能性が示唆された．今後は，人手評価を通じて，人間のメタファー理解との対応関係を調査する予定である．

参考文献

[1] Ekaterina Shutova. Models of metaphor in NLP. In **Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics**, pp. 688–697, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[2] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. **Journal of machine learning research**, Vol. 21, No. 140, pp. 1–67, 2020.

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, et al. Learning transferable visual models from natural language supervision. In **International conference on machine learning**, pp. 8748–8763. PmLR, 2021.

[4] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

[5] 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.

[6] Glenn Jocher, Jing Qiu, and Jingyu Peng. Ultralytics yolo11. https://github.com/ultralytics/ultralytics, 2024.

[7] Michael Toker, Hadas Orgad, Mor Ventura, Dana Arad, and Yonatan Belinkov. Diffusion lens: Interpreting text encoders in text-to-image pipelines. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 9713–9728, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[8] Emily M. Bender and Alexander Koller. Climbing towards NLU: On meaning, form, and understanding in the age of data. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 5185–5198, Online, July 2020. Association for Computational Linguistics.

[9] Julia Birke and Anoop Sarkar. A clustering approach for nearly unsupervised recognition of nonliteral language. In **11th Conference of the European Chapter of the Association for Computational Linguistics**, pp. 329–336, Trento, Italy, April 2006. Association for Computational Linguistics.

[10] George Lakoff and Mark Johnson. **Metaphors we live by**. University of Chicago Press, Chicago, 1980.

[11] Dedre Gentner. Structure-mapping: A theoretical framework for analogy. **Cognitive Science**, Vol. 7, No. 2, pp. 155–170, 1983.

[12] Carlo Strapparava. Metaphor: A computational perspective by tony veale, ekaterina Shutova and beata beigman klebanov. **Computational Linguistics**, Vol. 44, No. 1, pp. 191–192, April 2018.

[13] Paul V. DiStefano, John D. Patterson, and Roger E. Beaty. Automatic scoring of metaphor creativity with large language models. **Creativity Research Journal**, Vol. 37, No. 4, pp. 555–569, 2024.

[14] Nicholas Ichien, Dušan Stamenković, and Keith J. Holyoak. Large language model displays emergent ability to interpret novel literary metaphors, 2024.

[15] Xiaoyu Tong, Rochelle Choenni, Martha Lewis, and Ekaterina Shutova. Metaphor understanding challenge dataset for LLMs. In **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 3517–3536, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[16] Manishit Kundu, Sumit Shekhar, and Pushpak Bhattacharyya. Looking beyond the pixels: Evaluating visual metaphor understanding in VLMs. In **Findings of the Association for Computational Linguistics: EMNLP 2025**, pp. 23137–23158, Suzhou, China, November 2025. Association for Computational Linguistics.

[17] Tuhin Chakrabarty, Arkadiy Saakyan, Olivia Winn, et al. I spy a metaphor: Large language models and diffusion models co-create visual metaphors. In **Findings of the Association for Computational Linguistics: ACL 2023**, pp. 7370–7388, Toronto, Canada, July 2023. Association for Computational Linguistics.

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In **International Conference on Medical image computing and computer-assisted intervention**, pp. 234–241. Springer, 2015.

[19] William Peebles and Saining Xie. Scalable diffusion models with transformers. In **Proceedings of the IEEE/CVF international conference on computer vision**, pp. 4195–4205, 2023.

[20] DeepFloyd. If-i-xl-v1.0, 2023. Available at https://huggingface.co/DeepFloyd/IF-I-XL-v1.0.

[21] Junsong Chen, Jincheng YU, Chongjian GE, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In **The Twelfth International Conference on Learning Representations**, 2024.

[22] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In **Forty-first international conference on machine learning**, 2024.

[23] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. **arXiv preprint arXiv:2508.02324**, 2025.

[24] Shintaro Ozaki, Kazuki Hayashi, Yusuke Sakai, Jingun Kwon, Hidetaka Kamigaito, Katsuhiko Hayashi, Manabu Okumura, and Taro Watanabe. Texttiger: Text-based intelligent generation with entity prompt refinement for text-to-image generation. **arXiv preprint arXiv:2504.18269**, 2025.

[25] rcland12. A list of all 80 YOLO classes and its index in JSON format. GitHub Gist, 2023. Version 2, created July 14, 2023.

[26] Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. Reframing instructional prompts to GPTk's language. In **Findings of the Association for Computational Linguistics: ACL 2022**, pp. 589–612, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[27] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. **Advances in neural information processing systems**, Vol. 36, pp. 46595–46623, 2023.

[28] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. A survey on llm-as-a-judge, 2025.

[29] Dreamlike Art. Dreamlike photoreal 2.0, 2023. Available at https://huggingface.co/dreamlike-art/dreamlike-photoreal-2.0.

[30] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, et al. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 38–45, Online, October 2020. Association for Computational Linguistics.

[31] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In **NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications**, 2021.

[32] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, et al. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

表 3　実際に生成された画像とプロンプト．下線は 3 節で説明した作成方法によって生成されたメタファーを示す．これらに含まれる物体 (すなわち「人々」や「ゾウ」) は生成されないことが望ましい．

プロンプト	Dreamlike	PixArt	FLUX	SD3.5	Qwen-Image
The swaying trees in the wind looked like people dancing.					
The towering redwoods stood as tall as elephants.					

付録

A　詳細なモデル設定

大規模言語モデル　表 4 に，文生成および判定に使用した LLM の詳細を示す．直喩文の生成は，全 4 モデルに対して同一のテンプレートおよび生成設定のもとで実行した．生成された文の判定は，生成に用いたモデル自身を除く残り 3 モデルによって行った．生成および判定はいずれも決定的な設定（seed = 42，temperature = 0）で実行している．実行には Transformers ライブラリを用いた [30].

表 4　文生成および判定に使用した大規模言語モデル．

モデル名	パラメータ	HuggingFace での名前
Llama-3.1	8B	meta-llama/Llama-3.1-8B-Instruct
Mistral	7B	mistralai/Mistral-7B-Instruct-v0.3
Gemma-2	9B	google/gemma-2-9b-it
Phi-4-mini	4B	microsoft/Phi-4-mini-instruct

画像生成モデル　表 5 に使用したモデルの詳細な名前を記載する．生成の際には解像度を 768 × 768 px に，seed を 42，guidance_scale [31] を 4.5，num_inference_steps を 30 に，その他の設定は標準に従って実験を行った．実験には Diffusers ライブラリを用いた [32].

表 5　画像生成モデルの詳細な名前とそのエンコーダ．

モデル名	テキストエンコーダ	HuggingFace での名前
Dreamlike	CLIP	dreamlike-art/dreamlike-photoreal-2.0
PixArt	T5	PixArt-alpha/PixArt-XL-2-1024-MS
FLUX	T5，CLIP	black-forest-labs/FLUX.1-dev
SD3.5	T5，CLIP	stabilityai/stable-diffusion-3.5-large
Qwen-Image	Qwen (LLM)	Qwen/Qwen-Image

B　直喩テンプレート

本研究では，以下の 14 種類の直喩テンプレートを使用した．各テンプレートの *singular* には不定冠詞付きの単数名詞（例: *a book*），*plural* には複数形名詞（例: *books*）を挿入する．一部の名詞（例: *a pair of scissors*）については，語彙的慣習に従い定型表現を単数形として扱った．

- like *(singular / plural)*
- just like *(singular / plural)*
- looks like *(singular)* / look like *(plural)*
- exactly like *(singular / plural)*
- as 〜 as *(singular / plural)*
- as if *(singular / plural)* 〜
- as though *(singular / plural)* 〜

C　実際に生成された画像

表 3 に 3 節によって作成されたプロンプトおよび下線で示された直喩，またそれを用いて生成された画像を示す．表 3 上段では「looked like people dancing (人が踊っているかのように) のような直喩が含まれており，木がそのような情景で立っている画像生成するのが望ましい．一方で，PixArt や SD3.5，Qwen-Image では実際に人々が踊っている画像を生成しており，これはメタファーを適切に理解していない．

同じように，表 3 下段でも「as tall as elephants (像と同じくらいの高さ)」が含まれているが，Dreamlike は木から像の顔が，FLUX では画像中央に像が含まれるような画像を生成しており，既存のモデルはメタファーの理解に乏しいことが明らかである．