因果追跡に基づく VLM のモジュール重要度推定と LoRA 適用先選択の検証

佐藤 勇元 [1]　Mengsay Loem[1]
[1]Sansan 株式会社
{yugen.sato,mengsay.loem}@sansan.com

概要

視覚言語モデル（Vision Language Model; VLM）は視覚質問応答や文字認識など多様なタスクで高性能を示す一方，内部機構として，視覚情報と言語知識がどの層・どの計算モジュールを介して出力に寄与するかは十分に理解されていない．大規模言語モデル（Large Language Model; LLM）では，因果追跡により予測に因果的に効く層・モジュールを定量化できることが示されており，この知見は VLM の解析や効率的な微調整設計にも有用である可能性がある．本研究ではタスクを知識依存／知覚依存に分類し，因果追跡の分析により Transformer 系 VLM の層・モジュール（Attention/MLP）単位の重要度を推定して両者の寄与構造を比較する．さらに，重要度上位モジュールのみに LoRA を付与して微調整し，因果的重要度に基づくモジュール選択が効率的な学習設計の指針となり得るかを検証する．

1　はじめに

視覚言語モデル（Vision Language Model; VLM）は，視覚質問応答，キャプション生成，文字認識など幅広いタスクで高い性能を示している [1, 2, 3]．一方，実運用では計算資源やデータ規模の制約からパラメータ効率の良い微調整が重要となり，特に LoRA[4] が広く用いられているが, LoRA の付与先（層・モジュール）の選定指針は十分に整理されておらず全層・全モジュールに一様に付与する設定も多い [5, 6, 7] ため，学習可能パラメータ数や計算量の観点で非効率になり得る．

この問題に対し，LLM の内部解析は有力な手掛かりを与える. MLP が知識の保持・想起に関与しうることが指摘され [8]，ROME[9] や MEMIT[10] は因果追跡により予測に効くモジュールを特定し，局所編集への有用性を示した．また，Attention は情報のコピーやルーティングを担う回路として解釈できることが報告されている [9, 11]．VLM でも同様に予測に効く回路が層・モジュール単位で偏在するのであれば，その重要度に基づいて LoRA の付与先を絞ることで更新対象を削減しつつ性能を維持・改善できる可能性がある．

ただし，VLM では，タスクが要求する情報源が一様ではない．知識依存タスクでは視覚入力は手掛かりを与え，回答形成は言語知識の想起・統合に大きく依存すると考えられる．一方，文字認識や文書理解のような知覚依存タスクでは視覚トークンからの情報抽出とテキストトークンへの結合が支配的になり得る．したがって，出力に寄与する層・モジュールの分担は，知識依存／知覚依存で異なる可能性がある．VLM に対する因果追跡ベースの解析 [12, 13, 14] は提案されつつあるものの，(i) 知識依存／知覚依存という軸で同一の因果指標により寄与構造を直接比較し，(ii) その差異を LoRA の適用先選択として実際に検証する試みは限定的である．

本研究では，知識依存タスクと知覚依存タスクを定義し，ROME に基づく因果追跡（図 1）により Transformer 系 VLM の層・モジュール重要度（Attention/MLP）を推定して比較する．さらに，推定した重要度に基づき Top-k モジュールのみに LoRA を付与して微調整することで, (1) タスク群間で寄与構造に差があるか，(2) その差異（因果的重要度）が効率的な LoRA 配置設計に結びつくか，を検証する．

本研究で得られた主な知見は以下である．

- 知識依存タスクと知覚依存タスクでは，因果追跡で推定される重要層・重要モジュールの分布に差が観測される．
- 因果追跡に基づく Top-k 選択はランダム選択と同等となる場合があり，因果的重要度がそのまま LoRA 配置の最適指針になるとは限らない．

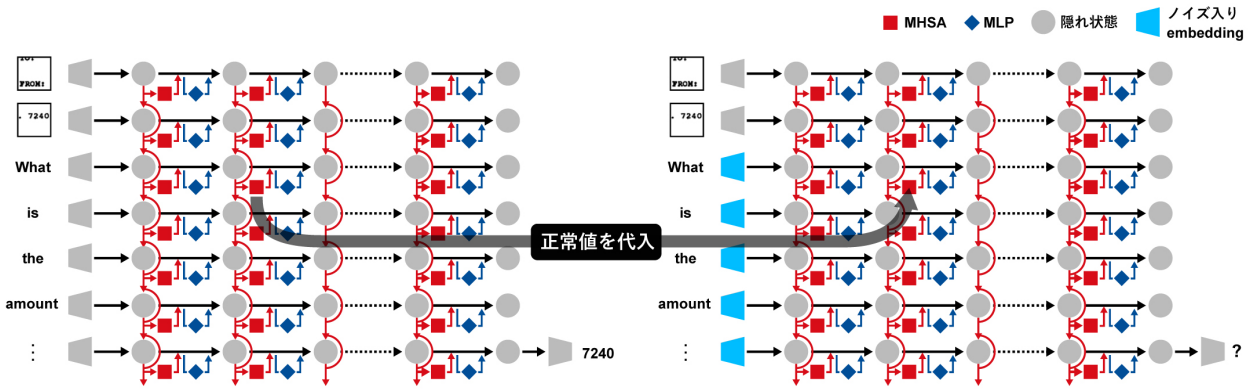

図 1　因果追跡の概略 [9]．clean 実行 (左) と corrupted 実行 (右) を用意し，corrupted 側の特定層・モジュールの活性を clean で置換して出力変化を測定する．MHSA は Multi-Head Self-Attention を指す．

2　手法

2.1　タスクの定義

本研究では, 知識依存タスクを画像に加え常識や外部知識を要するタスク，知覚依存タスクを画像中の文字情報抽出など画像内で完結するタスクとしてそれぞれ定義する．知識依存タスクとして OK-VQA[15]，A-OKVQA[16]，ScienceQA[17] を用いる．知覚依存タスクとして DocVQA[18], InfographicVQA[19]，TextVQA[20] を用いる．タスク群内の偏りを抑えるため, 各データセットから同数のサンプルを抽出する（抽出数 N は 3 節で記す）．以降，知識依存タスクを知識系，知覚依存タスクを知覚系と表記する．

2.2　ROME に基づく因果追跡

本研究では，ROME[9] の因果追跡に従い，Transformer 系 VLM における層・モジュール単位の重要度を推定する．画像 I と質問 Q を入力として，モデルは回答系列を生成する．図 1 に因果追跡の概要を示す．通常の入力埋め込みを用いた順伝播を *clean* 実行 と呼ぶ．一方，入力埋め込みの一部に攪乱（corruption；本研究では後述のノイズ付与）を加えた順伝播を *corrupted* 実行 と呼ぶ．ROME では，corrupted 実行の途中で，層 l のモジュール $m \in \{\text{attention}, \text{mlp}\}$ の出力（活性）を clean 実行で得られた対応する出力に置換する操作を *patch* と呼び，patch による出力の回復量から当該モジュールの因果的寄与を測定する．効果量の計算には teacher forcing を用いる．正解系列を $\text{gold} = (\text{gold}_1, \ldots, \text{gold}_L)$ とし，評価対象とするトークン位置の集合を $\mathcal{T} \subseteq \{1, \ldots, L\}$ とする．実行条件 $s \in \{\text{clean}, \text{corrupted}, \text{patch}(l, m)\}$ における gold のスコアを，評価位置での次トークン確率の幾何平均として

$$p_s(\text{gold}) := \exp\left(\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \log p_s(\text{gold}_t \mid I, Q, \text{gold}_{<t})\right) \tag{1}$$

と定義する．

まず，corruption によって gold スコアがどれだけ低下したかを全体効果（Total Effect; TE）として

$$\text{TE} = p_{\text{clean}}(\text{gold}) - p_{\text{corr}}(\text{gold}) \tag{2}$$

で定義する．次に，corrupted 実行において層 l・モジュール m を patch したときの回復量を間接効果（Indirect Effect; IE）として

$$\text{IE}(l, m) = p_{\text{patch}(l,m)}(\text{gold}) - p_{\text{corr}}(\text{gold}) \tag{3}$$

で定義する．さらに，層・モジュール間で比較可能な指標として Restoration Rate（RR）を

$$\text{RR}(l, m) = \frac{\text{IE}(l, m)}{\text{TE} + \epsilon} \tag{4}$$

で定義する（ϵ は分母安定化の小定数）．RR は，corruption で失われたスコア低下分（TE）のうち，patch により回復できた割合（IE）を表す．各サンプルでの RR を平均した $\text{RR}_{\text{mean}}(l, m)$ により重要度ランキングを得る．

2.3　ノイズによる corruption 設計

本研究では，入力 (I, Q) のうち, 質問テキスト Q に対応する埋め込みにへの corruption を採用する．埋め込み系列を $E \in \mathbb{R}^{B\times T\times d}$（$B$：バッチサイズ，$T$：系列長，$d$：隠れ次元）とし，その全要素の標準偏差を $\sigma_E = \text{Std}(E)$ とする．ノイズ強度のハイパーパラ

メータ s [1] を用いて，平均 0・分散 $(\sigma_E s)^2$ のガウスノイズ

$$N \sim \mathcal{N}\left(0, (\sigma_E s)^2 \mathbf{I}\right), \qquad N \in \mathbb{R}^{B\times T\times d} \tag{5}$$

を生成する．さらに，質問トークン位置を表すマスク $M \in \{0,1\}^{B\times T}$ を用い，質問に対応する位置にのみノイズを付与して corrupted 埋め込み $\tilde{E}$ を

$$\tilde{E} = E + (N \odot M), \tag{6}$$

と定義する．この corruption は，入力系列を保ったまま質問表現のみを局所的に攪乱し，その影響が出力確率へ伝播する経路を層・モジュール単位で比較することを目的とする．知識系タスクでは質問が想起すべき知識を指定する手掛かりとして働く一方，知覚系タスクでは質問は主に抽出対象の指示となり，回答は画像内情報から得られることが多い．そこで，本研究では，画像表現ではなく質問表現への corruption に統一し，どの層・モジュールが corruption の影響をどの程度回復するかを因果追跡で評価する．

2.4 LoRA 適用先選択

因果追跡により得られたモジュール重要度 $\mathrm{RR}_{\mathrm{mean}}(l,m)$（$l$: 層，$m \in \{\mathrm{attention}, \mathrm{mlp}\}$）に基づき，LoRA を付与するモジュール集合を選択する．本研究では，更新対象を限定した設定として $k = 10$ を用い，タスク群間での重要度分布の差が LoRA 配置設計に結びつくかを検証する．

タスク群ごとの Top-k: shared　知識系タスク群 $t = \mathrm{knowledge}$ と知覚系タスク群 $t = \mathrm{perception}$ それぞれについて，$\mathrm{RR}_{\mathrm{mean}}{}^{(t)}(l,m)$ の上位 k 個のモジュールを選択し，LoRA を付与する．両タスク群で同一モジュールが選択される場合（重複）は許容する．本設定は，因果追跡で重要と推定されたモジュールに更新容量を集中させることで，少ない学習可能パラメータでも有効に適応できるという仮説を検証するためのものである．

タスク差分に基づく Top-k: task-diff　タスク群ごとに重要度のスケールが異なる可能性を考慮し，最大値で正規化した重要度を

$$\widehat{\mathrm{RR}}^{(t)}(l,m) = \frac{\mathrm{RR}_{\mathrm{mean}}{}^{(t)}(l,m)}{\max_{l',m'} \mathrm{RR}_{\mathrm{mean}}{}^{(t)}(l',m') + \epsilon} \tag{7}$$

と定義する．ϵ は分母安定化の小定数で，本研究では e^{-8} とした．次に，タスク群間差分

$$\Delta(l,m) = \widehat{\mathrm{RR}}^{(\mathrm{knowledge})}(l,m) - \widehat{\mathrm{RR}}^{(\mathrm{perception})}(l,m) \tag{8}$$

を計算し，$\Delta(l,m)$ が大きい順に上位 k 個を Top-k-task-diff-knowledge，$-\Delta(l,m)$ が大きい順に上位 k 個を Top-k-task-diff-perception として選択する．本設定は，タスク群間で相対的に寄与が偏るモジュールへ LoRA を割り当てることで，タスク特性に応じた配置設計の有効性を検証することを目的とする．

1) TE が極端に小さくならないよう $s = 2.0$ とした．

3 実験

3.1 設定

Qwen3-VL-4B-Instruct[21] と Gemma-3-4b-it[22] を用いる．学習設定の詳細は付録 D に示す．LoRA のハイパーパラメータは全条件で共通とし，$r = 64$，$\alpha = 16$，dropout = 0.05 とした．各データセットについて，因果追跡（重要度推定）に 1,000 サンプル，LoRA 学習に 3,000 サンプル，評価に 1,000 サンプルを用いる．本研究では, 正誤判定の煩雑化を防ぐため各データの答えとモデル出力の完全一致の正解率で評価を行う．

3.2 LoRA 適用先の比較

以下の条件を比較する[2]．

- **Baseline**: ベースモデルによるゼロショット[3]．
- **LLM-all**: LLM の全モジュールに LoRA を付与．
- **Random-k**: k 個のモジュールを一様ランダムに選び LoRA を付与．
- **Top-k-shared**: タスク群ごとに $\mathrm{RR}_{\mathrm{mean}}$ 上位 k 個を選び LoRA を付与（タスク間の重複を許容）．
- **Top-k-task-diff**: タスク群間の重要度差分が大きい上位 k 個を選び LoRA を付与．

Random-k には{42，43，44}の 3 つの seed 値を用い，最終的にその平均をとる．

4 結果と考察

4.1 因果追跡の重要度分布

図 4 に RR ヒートマップを示す．知識系と知覚系で層方向のパターンが異なり, 特に知覚系では Attention 側のピークが目立つ層が観測された．図 3 に，正規化した RR のタスク差分 $\Delta(l,m)$ を示す．知識系タスクは知覚系タスクに比べ後段 MLP の寄与が大きく, 反対に知覚系タスクでは Attention の寄与

2) Top-k-shared はタスク群別 Top-k，Top-k-task-diff は正規化重要度の差分 Top-k に基づく．詳細は 2 節参照．

3) プロンプトは付録 C.1 に記載

表 1　LoRA 学習による正解率の比較 ($k = 10$)

	Qwen3-VL-4B			Gemma-3-4B		
LoRA 適用先	知識系	知覚系	学習パラメータ数	知識系	知覚系	学習パラメータ数
Baseline	0.417	0.765	–	0.433	0.539	–
LLM-all	0.614	0.818	$\simeq 132M$	0.550	0.588	$\simeq 120M$
Random-k	0.599	0.798	$\simeq 18M$	0.547	0.566	$\simeq 16M$
Top-k-shared	0.586	0.783	$\simeq 16M$	0.526	0.544	$\simeq 15M$
Top-k-task-diff	0.596	0.797	$\simeq 15M$	0.542	0.559	$\simeq 15M$

が大きい結果となり, タスク間で明確に異なる寄与構造が確認された．(データセット別の結果: 付録 E)

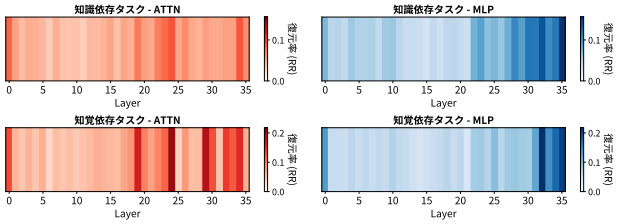

図 2　RR ヒートマップ (Qwen3-VL)．上段：知識系タスク, 下段：知覚系タスク．左：Attention, 右：MLP.

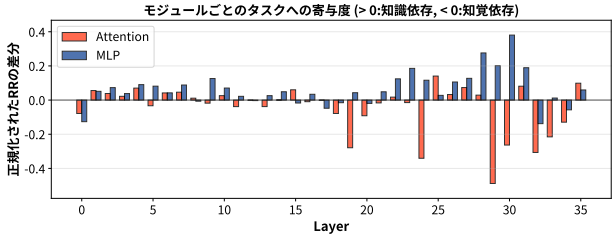

図 3　RR のタスク差分（知識系 – 知覚系）(Qwen3-VL). 正は知識系タスク, 負は知覚系タスクへの寄与度が高い.

4.2　LoRA 適用先選択の有用性

表 1 より，LoRA を用いた全条件は両モデル・両タスク群で Baseline を上回り，LoRA による微調整が有効に機能していることを確認できる．LLM-all は最大の改善を示す一方，学習可能パラメータ数は約 120–132M と大きい．これに対し，更新対象を大幅に削減した Random-k（約 16–18M）でも一貫した改善が得られた．

一方，因果追跡に基づく Top-k 選択は，Qwen3-VL と Gemma の双方で Random-k を安定して上回らず，Top-k-shared は下回り，Top-k-task-diff も概ね同等に留まった．以上より，因果的重要度に基づく単純な Top-k 選択が LoRA 配置の最適指針になるとは限らないことが示唆される．

4.3　Top-k 選択戦略による結果の違い

Top-k-shared は，各タスク群で因果追跡により重要と推定されたモジュールをそのまま LoRA 適用先とするため，直感的には有望な戦略である．しかし，本研究の因果追跡は質問埋め込みへの corruption に基づくため，推定される重要度は質問情報が攪乱された状況からの復元に寄与する回路を強調する可能性がある．この指標はタスク間で寄与構造を比較する上では有用である一方，LoRA による性能改善に直結する学習で変えるべき重みの候補と一致しない場合がある．

Top-k-task-diff はタスク群間の重要度差分に基づき，タスク固有の回路を抽出できることが期待される．一方で，差分が大きいことは寄与の相対的な偏りを示すに留まり，そのモジュールが学習で性能改善に結びつくボトルネックであるとは限らない．実際，Qwen3-VL と Gemma の双方で Top-k-task-diff は Random-k と概ね同水準であった．以上より，差分 Top-k はメカニズム差の手がかりを与える一方で，LoRA 配置の最適指針としては十分でない場合があることが示唆される．

5　おわりに

本研究では，知識依存タスクと知覚依存タスクを対比し，質問埋め込みへの corruption に基づく ROME の因果追跡により，Transformer 系 VLM の層・モジュールの重要度を推定した．その結果，知識依存では後段 MLP の寄与が相対的に大きく，知覚依存では Attention の寄与が相対的に大きいなど，タスク特性に応じて寄与構造が異なることを確認した．さらに，推定した重要度に基づいて LoRA の適用先を選択し微調整を行い，全モジュール LoRA およびランダム選択 LoRA と比較した．少数モジュールへの LoRA でも一定の改善は得られた一方，因果追跡に基づく Top-k 選択はランダム選択を安定して上回らず，因果的重要度がそのまま LoRA 配置の最適指針になるとは限らないことが示唆された．

今後は，視覚トークン側の corruption 導入による頑健性検証に加え，より良いモジュール選択戦略などを通じて，タスク差の解析と効率的な LoRA 設計の接続を強める．

参考文献

[1] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In **Advances in Neural Information Processing Systems (NeurIPS 2023)**, 2023.

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023.

[3] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

[4] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In **International Conference on Learning Representations (ICLR 2022)**, 2022.

[5] NVIDIA. NVIDIA NeMo documentation: PEFT guide (e.g., `match_all_linear`). https://docs.nvidia.com/nemo/automodel/latest/guides/llm/peft.html. Accessed 2026-01-08.

[6] Hugging Face. PEFT documentation: LoRA configuration reference (e.g., `target_modules`). https://huggingface.co/docs/peft/en/package_reference/lora. Accessed 2026-01-08.

[7] Hugging Face. PEFT documentation: Quantization / QLoRA (e.g., `target_modules=all-linear`). https://huggingface.co/docs/peft/en/developer_guides/quantization. Accessed 2026-01-08.

[8] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)**, pp. 5484–5495. Association for Computational Linguistics, 2021.

[9] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In **Advances in Neural Information Processing Systems (NeurIPS 2022)**, 2022.

[10] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass editing memory in a transformer. In **International Conference on Learning Representations (ICLR 2023)**, 2023.

[11] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. https://transformer-circuits.pub/2021/framework/index.html, 2021. Accessed 2026-01-08.

[12] Vedant Palit, Rohan Pandey, Aryaman Arora, and Paul Pu Liang. Towards vision-language mechanistic interpretability: A causal tracing tool for BLIP, 2023. Presented at ICCV Workshops (CLVL), 2023.

[13] Sambit Basu, Joe Grayson, Michael Morrison, Besmira Nushi, Soheil Feizi, and Daniela Massiceti. Understanding information storage and transfer in multi-modal large language models. In **Advances in Neural Information Processing Systems (NeurIPS 2024)**, 2024.

[14] Qiming Li, Zekai Ye, Xiaocheng Feng, Weihong Zhong, Weitao Ma, and Xiachong Feng. Causal tracing of object representations in large vision language models: Mechanistic interpretability and hallucination mitigation, 2025.

[15] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. OK-VQA: A visual question answering benchmark requiring external knowledge. In **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)**, 2019.

[16] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-OKVQA: A benchmark for visual question answering using world knowledge, 2022.

[17] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In **Advances in Neural Information Processing Systems (NeurIPS 2022)**, 2022.

[18] Minesh Mathew, Dimosthenis Karatzas, R. Manmatha, and C. V. Jawahar. DocVQA: A dataset for VQA on document images, 2020.

[19] Minesh Mathew, Viraj Bagal, Rubèn Pérez Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. InfographicVQA, 2021.

[20] Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)**, pp. 8317–8326, 2019.

[21] Qwen Team. Qwen3 technical report, 2025.

[22] Gemma Team. Gemma 3. https://goo.gle/Gemma3Report, 2025. Accessed 2026-01-08.

A　Limitations

本設計は質問埋め込みへの corruption に基づくため，文字認識やレイアウト解析など視覚情報抽出そのものを直接破壊せず，知覚依存タスクにおける視覚回路を十分に同定できない可能性がある．今後は視覚トークン側への corruption も導入や選択するモジュール数 k を変動させた検証，ノイズ設計の設定拡張など，結果の頑健性を検証する必要がある．また，本研究で利用したモデルのサイズはいずれも 4B 級に限定される．より頑健な結果のために，さらに大規模なモデルでの検証が必要である．

B　LoRA 適用先モジュール

本節では，Qwen3-VL-4B における因果追跡に基づく LoRA 適用先選択で実際に採用されたモジュールを示す．

B.1　タスク群ごとの Top-k: shared

表 2 に，shared 方式で選択された Top-k モジュールを示す．両タスクの Top-k は一部重複しており，重複の存在はタスク間で共通に寄与するモジュールが含まれる可能性を示唆する．

表 2　shared 方式で選択された Top-10 モジュール

知識系	L23-mlp，L24-attn，L28-mlp，L29-mlp，L30-mlp，L31-mlp，L32-mlp，L33-mlp，L34-mlp，L35-mlp
知覚系	L0-attn，L19-attn，L24-attn，L29-attn，L32-attn，L32-mlp，L33-mlp，L34-attn，L34-mlp，L35-mlp

B.2　タスク差分に基づく Top-k: task-diff

表 3　task-diff 方式で選択された Top-10 モジュール（差分 Top-k）．

知識系	L30-mlp，L28-mlp，L29-mlp，L31-mlp，L23-mlp，L25-attn，L27-mlp，L9-mlp，L22-attn，L24-mlp
知覚系	L29-attn，L24-attn，L32-attn，L19-attn，L30-attn，L33-attn，L32-mlp，L34-attn，L0-mlp，L20-attn

C　プロンプト

C.1　因果追跡，ベースライン評価用

Prompt example

```
You are a visual question answering system.
Answer the question *only* with the final
answer word or phrase.
Do not add any explanation.

Question: {QUESTION}
Answer:
```

D　LoRA 学習設定

本研究の LoRA 微調整は以下の設定で実施した．

- 学習: 3 epoch，バッチサイズ 4，勾配蓄積 2（実効バッチサイズ 8）．
- 最適化: 学習率 2×10^{-4}, スケジューラ: linear.
- 計算効率：gradient checkpointing を有効化．

E　データセットごとの復元率 RR

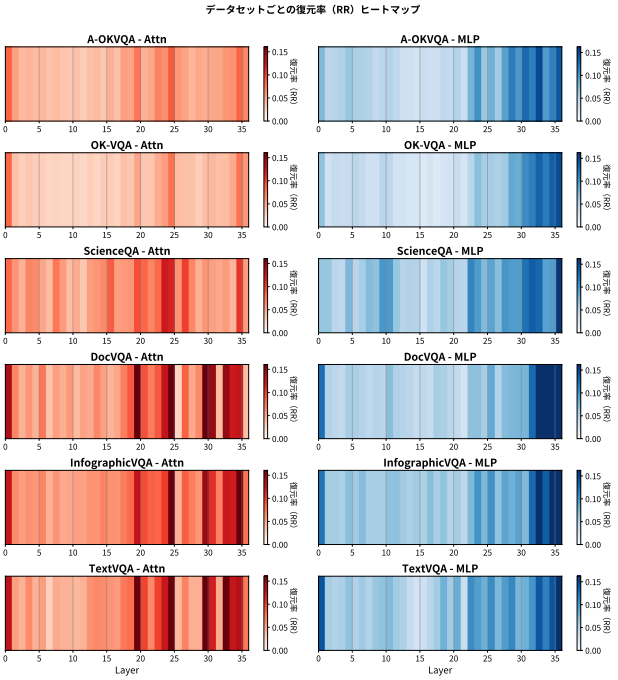

図 4　RR ヒートマップ (Qwen3-VL)．左：Attention, 右：MLP．上三段：知識系, 下三段：知覚系