Activation Transport Operators

Andrzej Szablewski*
University of Cambridge
as3623@cam.ac.uk

Marek Masiak*
University of Oxford
marek.masiak@dtc.ox.ac.uk

Abstract

The residual stream mediates communication between transformer decoder layers via linear reads and writes of non-linear computations. While sparse-dictionary learning-based methods locate features in the residual stream, and activation patching methods discover circuits within the model, the mechanism by which features *flow* through the residual stream remains understudied. Understanding this dynamic can better inform jailbreaking protections, enable early detection of model mistakes, and their correction. In this work, we propose Activation Transport Operators (ATO), linear maps from upstream to downstream residuals k layers later, evaluated in feature space using downstream SAE decoder projections. We empirically demonstrate that these operators can determine whether a feature has been *linearly transported* from a previous layer or *synthesised* from non-linear layer computation. We develop the notion of *transport efficiency*, for which we provide an upper bound, and use it to estimate the size of the residual stream subspace that corresponds to linear transport. We empirically demonstrate the linear transport, report transport efficiency and the size of the residual stream's subspace involved in linear transport. This compute-light (no finetuning, < 50 GPU-h) method offers practical tools for safety, debugging, and a clearer picture of where computation in LLMs behaves linearly. Our code is available at `https://github.com/marek357/activation-transport-operators`.

1 Introduction

Transformer layers modify token-wise residual stream states through a sequence of attention and MLP updates Elhage et al. [2021]. Much of what can be read from these vectors is linear—decoders, probes, and logit-lens all apply affine maps—yet what gets written into the stream is the result of nonlinear mechanisms (LayerNorm, softmax attention, gating in MLPs) Razzhigaev et al. [2024]. Many interpretability tools focus either on locating where a behaviour "lives" or decoding what a representation "means" but they rarely study explicit operators that predict and reconstruct how specific features move from one site in the network to another.

On the intervention side, variants of activation and path patching reliably identify layers, heads, and positions that are causally important for a behaviour Goldowsky-Dill et al. [2023], Kramár et al. [2024]. Ferrando and Voita [2024] present Information Flow Routes, which push further by constructing global, causally validated flow graphs for predictions, yet—like patching—it characterizes influential paths without yielding an explicit map that predicts downstream hidden states. On the decoding side, logit and tuned lenses nostalgebraist [2020], Belrose et al. [2025], provide affine readouts from intermediate residuals into vocabulary space, and sparse autoencoders (SAEs) recover monosemantic features at scale Cunningham et al. [2023]. Furthermore, in their recent study, Lawson et al. [2025] use multi-layer SAEs to study layer similarity, suggesting some evidence of a split between feature

*Equal contribution

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Mechanistic Interpretability.

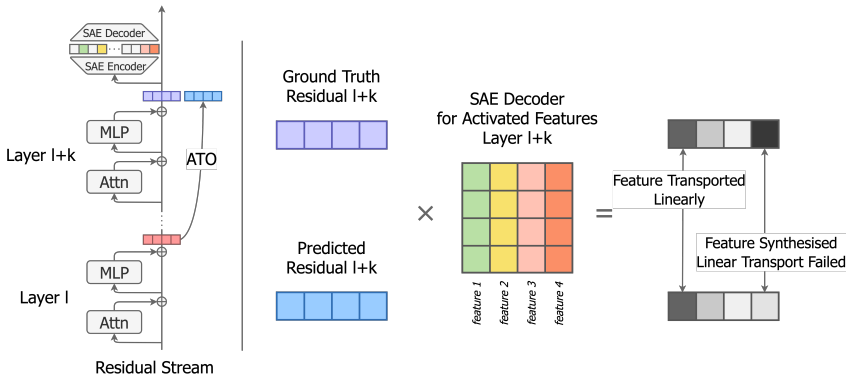

Figure 1: ATO predicts downstream residual stream vector. Using an SAE, we identify activated features. True and predicted residuals are projected onto SAE decoder vectors and compared.

transport and non-linear feature recomputation. Meanwhile, activation steering methods demonstrate powerful control via learned activation edits but focus on exogenous behaviour shaping rather than explaining endogenous feature flow Rodriguez et al. [2024].

This work aims to bridge attribution and representation analysis by introducing Activation Transport Operators (ATOs)—explicit, regularised linear maps that predict downstream residual vectors from upstream residuals. ATOs are learned from paired activations collected during ordinary forward passes. Crucially, ATOs are not a claim that the network is globally linear, but they serve as a test for local linear preservation of a specific feature between two sites in the stream (Figure 1). High predictive and causal scores indicate linear transport, while failure indicates downstream feature synthesis or nonlinear recomputation.

Our core contributions are as follows: 1) we formally define Activation Transport Operators and empirically study our method using available LLMs and SAEs, evaluating it with per-feature predictive fidelity and causal ablation, and 2) we introduce the notion of transport efficiency, and show its link to the size of the communication subspace of the residual stream.

2 Methodology

We study *downstream features* in a decoder-only transformer through the lens of the residual stream. Let $v_{l,i} \in \mathbb{R}^d$ denote the *upstream* residual vector at layer l and token position i. For a feature f identified at layer $l{+}k$ by its downstream SAE decoder direction $d_f^{(l+k)} \in \mathbb{R}^d$, the feature is "observed" at $(l{+}k, j)$. Our objective is to test whether the downstream activation aligned with f can be *linearly attributed* to earlier residual states. To this end, we learn an affine, rank-constrained transport operator:

$$T_r : \mathbb{R}^{d_{\text{model}}} \to \mathbb{R}^{d_{\text{model}}}, \qquad \hat{v}_{l+k,j} \;=\; T_r\, v_{l,i} + b,$$

where we rank-constrain the transport operator by computing the singular value decomposition: $T_r = U_r S_r V_r^\top$ with rank $r \le d_{\text{model}}$ (and $b \in \mathbb{R}^{d_{\text{model}}}$). Location pairs $(l, i) \to (l{+}k, j)$ are sampled using explicit policies, which we refer to as j*-policies*. In this work, we use a single j-policy: *same-token* ($j{=}i$), which maps upstream to downstream for the same position in a sequence. However, in future work, we plan to explore more complex policies, such as *attention-reader* Top-K, *delimiter-pair*, and *copy-target*. The operator is fitted on many such pairs with ridge, lasso, or elasticnet regularisation. Importantly, evaluation is done in *feature space* rather than on raw residuals. We compare the downstream decoder projections:

$$a_{\text{true}} \;=\; (d_f^{(l+k)})^\top v_{l+k,j}, \qquad a_{\text{pred}} \;=\; (d_f^{(l+k)})^\top \hat{v}_{l+k,j} \;=\; (d_f^{(l+k)})^\top (T_r\, v_{l,i} + b) \tag{1}$$

using regression metrics (specifically, R^2 and MSE). High agreement indicates that the component of the downstream state relevant to f is *transported* through a low-dimensional linear channel. On the

other hand, poor agreement (despite reasonable upstream sources and policies) suggests the activation is *synthesised locally* by later non-linear computations.

We causally validate the transport operators by ablating the upstream site (l, i) (i.e., zeroing or projecting out the upstream gate) and injecting the reconstructed vector $\hat{v}_{l+k,j}$ at the target. Restoration of the feature projection and associated behaviour (e.g., structured-format correctness or continuation accuracy) provides direct evidence of linear transport along the learned operator. Additionally, we compare the results with the *zero intervention*, which involves completely ablating the downstream residual vector by setting it to zero [Mohebbi et al., 2023, Olsson et al., 2022]. We include this comparison to quantify the maximum corruption we can introduce to the residual stream, thereby measuring the model error (e.g. perplexity increase) if the residual stream contains no information at layer $l+k$. We expect this to be significantly larger than the error induced by transport operators.

Transport efficiency To better understand the process of feature transport, we seek to find the upper bound for R^2 of our rank-r transport operator. Hence, we define the R^2_{ceiling} as the maximal R^2 value achievable by any linear predictor at rank r. In this analysis, we shift our focus to the task of predicting downstream residual stream vectors, stacked in matrix $Y \in \mathbb{R}^{N \times d_{\text{model}}}$ from upstream residual stream vectors, stacked in matrix $X \in \mathbb{R}^{N \times d_{\text{model}}}$. Assuming zero-mean, the ceiling for transport efficiency at rank r is given by: $R^2_{\text{ceiling}}(r, Y) = \frac{1}{d_{\text{model}}} \sum_{i=1}^{r} \rho_i^2$, where ρ_i^2 are the squared canonical correlations. In Appendix A we rigorously derive this upper bound. Therefore, we can define the transport efficiency as: $\text{Eff} = \tilde{R^2}(r, \hat{Y}_T)/R^2_{\text{ceiling}}(r, Y) \in [0, 1]$, where $\tilde{R^2}(r, \hat{Y}_T)$ is the R^2 metric of rank-r-ATO-predicted downstream residual vectors *in whitened Y space*. We need to transform the ATO predictions to the whitened Y space to allow for apples–to–apples comparison of explained variance. Transport efficiency plateaus when increasing ATO's rank does not enhance the relative predictive ability of the operator. This can be observed in Figure 4 with $k = 10$.

Estimating the dimensionality of Linear Transport Subspace (LTS) We use the notion of effective dimensionality [Del Giudice, 2020] to define the dimensionality of the subspace of the residual stream with linear transport: $\text{d}_{\text{eff}} = (\sum_i \rho_i^2)^2 / \sum_i (\rho_i^2)^2$.

Experimental setup We discuss the setup and experimental details in Appendix B.

3 Results

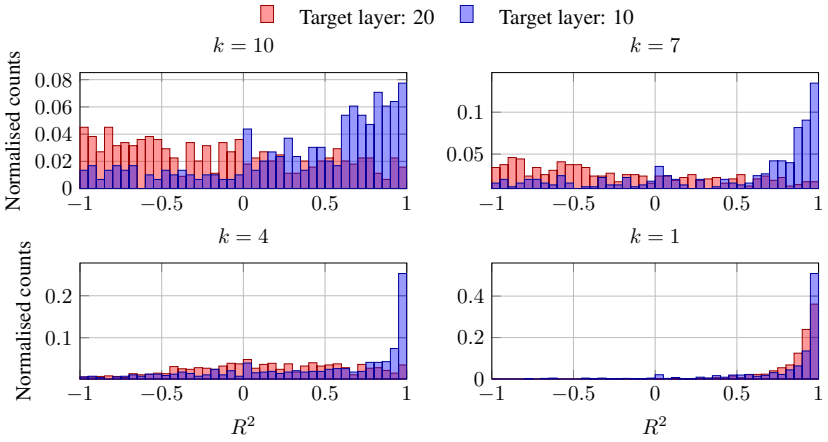

Figure 2: Per-feature R^2 of operators depend on both the target layer depth and the leap size k.

Most linear transport occurs in nearby layers and deteriorates over large distances Comparing the per-feature R^2 between full-rank operators shows that those trained for small leaps ($k = 1, k = 4$ for target layer 10) successfully transport a significant number of features ($R^2 > 0.95$). While this number is deteriorating with the growing leap size k, we also find that feature transport is generally less common in the later layers of the transformer, even with small ks (shown as per-plot distribution

shifts in Figure 2). This suggests that information management in the residual stream may have two regimes. In early layers, the stream has the capacity to accept new features without the need to evict existing ones, hence we observe more transport. Once the residual stream fills up with information, later layers in the model prioritise newly synthesised or non-linearly transformed features, deleting old information from the stream, further supporting the idea introduced by Elhage et al. [2021].

However, we also observe an inverse trend with significantly larger leaps in deeper layers. For example, in layer 21 in Figure 3, the transport reaches its minimum at $k = 10$. Counterintuitively, as the distance between the source and target layer further increases, the R^2 metric improves. We find this phenomenon intriguing and will analyse it in detail in further work.

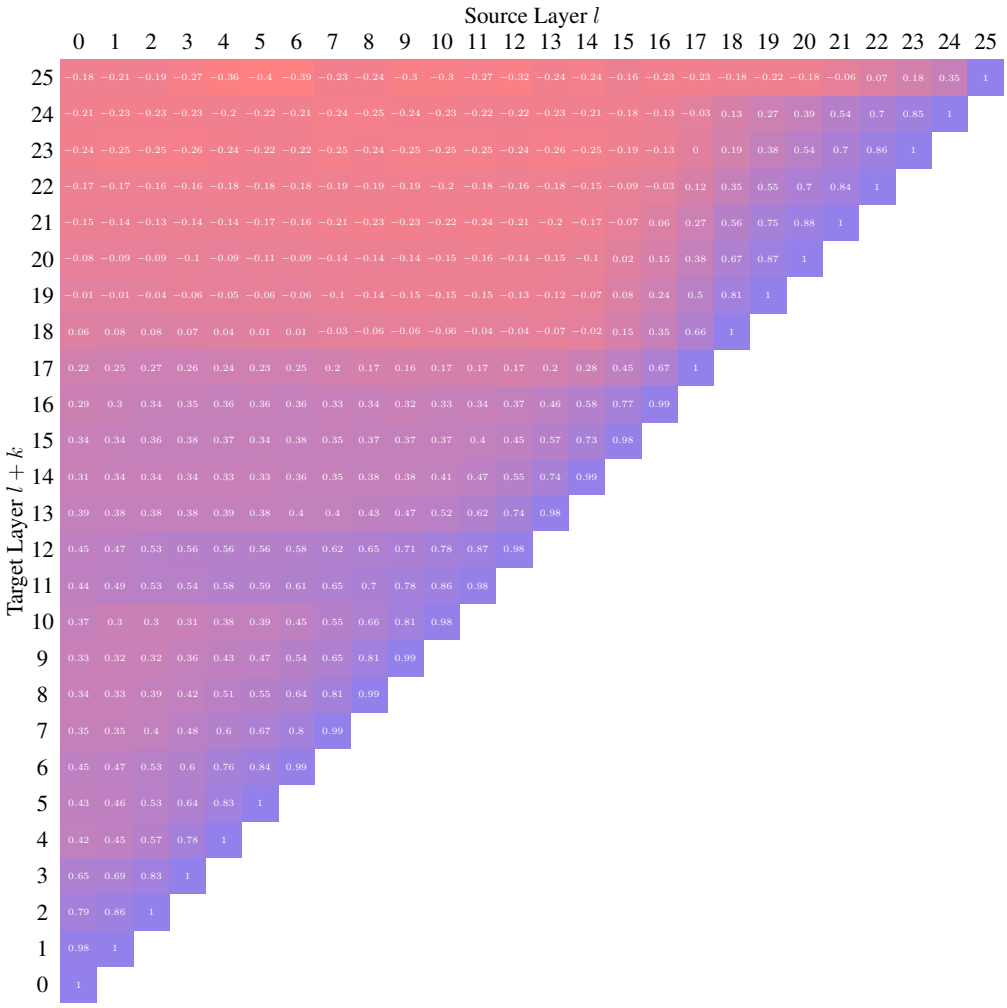

Figure 3: Average per-feature R^2 for all source-target combinations. Note that the sets of chosen SAE features are different across target layers, hence values in the same column may not be directly comparable. Constant leap sizes k are represented by the diagonals.

Transport efficiency and LTS size depend on the transport distance Figure 4 shows that transport efficiency over longer leaps (k=7, 10) saturates early and at lower values, indicating a smaller linear transport subspace (i.e. $d_{eff} = 1453$, and $d_{eff} = 1291$, respectively). On the other hand, in the adjacent-layer case (k=1), we observe almost linear improvement of transport efficiency with ATO rank, approaching $R^2_{ceiling}$ near full rank. Such result is consistent with a larger set of linearly transported directions, size of which is estimated at $d_{eff} = 2198$. The dimensionality of the LTS should guide ATO rank selection: choosing r above the LTS size yields no population gain beyond the CCA ceiling: extra rank mainly fits noise, which may inflate training R^2 but will not generalise.

Using ATOs yields only marginal perplexity increase We compare perplexity for the unedited, ATO-patched and zero-intervened models. ATOs raise perplexity only slightly, with the effect growing with leap size k. The zero-intervened model is significantly worse (similar to using ATO with a null vector), and provides an upper bound on degradation. However, even at k=10 the increase is 7.1% of max degradation, and for k<5, it stays below 1.2%. Trends in Figure 5 hold beyond the ablations of 5 out of 256 sequence positions; applying ATOs to all positions yields at most a 13.5% increase at k=10 (with upper-bound perplexity of 12.4529). Thus, ATOs substantially recover language-modelling ability otherwise lost under zero-intervention, supporting their use for targeted diagnostics and edits.

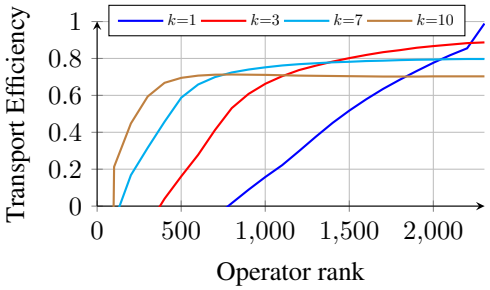

Figure 4: Transport efficiency for the target layer 10 with different leap (k) values.

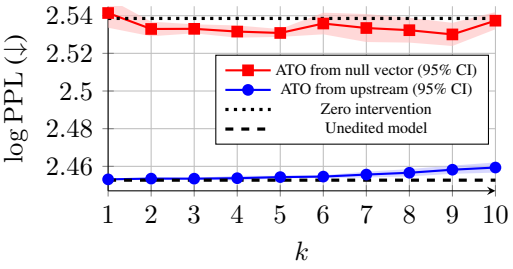

Figure 5: Log-perplexity for unedited and ablated models. Ablated five positions per sequence.

Limitations Our study has several limitations. First, we used a single, trivial same-token j-policy, which biases results toward local transport and may miss attention-mediated cross-token routing—exploring IFR-guided or data-driven j selection is left for future work. Second, we evaluated only a single model, therefore, we cannot claim that linear transport is pervasive across architectures or depths without broader replication. Third, our linear operators do not distinguish between features that are transported from earlier layers and those that arise as their linear combinations. Hence, we underestimate the number of synthesised features. Finally, in this work, we do not present feature-targeted editing built with our operators, which we aim to tackle in a follow-up work. In principle, leveraging feature-specific transport between layers could allow low-compute inference-time corrections of the generated text.

4 Conclusions

We introduced *Activation Transport Operators* (ATOs): explicit, regularised linear maps that predict a downstream residual vector from upstream residuals and are evaluated in SAE feature space. High predictive and causal scores indicate linear transport of a feature, while failure suggests downstream synthesis or nonlinear recomputation. Empirically, we find that transport is strongest over short layer distances and weakens with depth and leap size, suggesting an early-layer regime where the residual stream behaves as a shared linear channel followed by later layers that prioritise synthesis and recomposition. Our transport efficiency metric quantifies how close an operator gets to the best possible linear prediction, while the efficiency analysis implies that the dimensionality of the Linear Transport Subspace is tightly linked to the optimal rank of ATO. Taken together, ATOs provide a simple, testable method for mapping feature flow. We expect richer j-policies and multi-source operators to reveal attention-mediated routing and to enable feature-targeted, low-compute edits during inference.

References

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens, 2025. URL `https://arxiv.org/abs/2303.08112`.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023. URL `https://arxiv.org/abs/2309.08600`.

Marco Del Giudice. Effective dimensionality: A tutorial. *Multivariate Behavioral Research*, 56 (3):527–542, March 2020. ISSN 1532-7906. doi: 10.1080/00273171.2020.1743631. URL http://dx.doi.org/10.1080/00273171.2020.1743631.

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. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Javier Ferrando and Elena Voita. Information flow routes: Automatically interpreting language models at scale, 2024. URL https://arxiv.org/abs/2403.00824.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching, 2023. URL https://arxiv.org/abs/2304.05969.

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. Atp*: An efficient and scalable method for localizing llm behaviour to components, 2024. URL https://arxiv.org/abs/2403.00745.

Tim Lawson, Lucy Farnik, Conor Houghton, and Laurence Aitchison. Residual stream analysis with multi-layer saes, 2025. URL https://arxiv.org/abs/2409.04185.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2, 2024. URL https://arxiv.org/abs/2408.05147.

Hosein Mohebbi, Willem Zuidema, Grzegorz Chrupała, and Afra Alishahi. Quantifying context mixing in transformers, 2023. URL https://arxiv.org/abs/2301.12971.

nostalgebraist. interpreting GPT: the logit lens. *LessWrong*, August 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. Accessed 2025-08-23.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, and et al. In-context learning and induction heads, Mar 2022. URL https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Anton Razzhigaev, Matvey Mikhalchuk, Elizaveta Goncharova, Nikolai Gerasimenko, Ivan Oseledets, Denis Dimitrov, and Andrey Kuznetsov. Your transformer is secretly linear, 2024. URL https://arxiv.org/abs/2405.12250.

Pau Rodriguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, Marco Cuturi, and Xavier Suau. Controlling language and diffusion models by transporting activations, 2024. URL https://arxiv.org/abs/2410.23054.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://cerebras.ai/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, 2023. URL https://huggingface.co/datasets/cerebras/SlimPajama-627B.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger,

Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL `https://arxiv.org/abs/2408.00118`.

A Transport efficiency

Assuming zero-mean, we define the following covariance matrices:

$$\Sigma_{XX} = \tfrac{1}{N} X^\top X, \quad \Sigma_{YY} = \tfrac{1}{N} Y^\top Y, \quad \Sigma_{YX} = \tfrac{1}{N} Y^\top X, \quad \Sigma_{XY} = \Sigma_{YX}^\top.$$

We employ canonical cross-correlation analysis (CCA) to find directions $a \in \mathbb{R}^{d_{\text{model}}}$ (in downstream residual stream) and $b \in \mathbb{R}^{d_{\text{model}}}$ (in upstream residual stream) maximizing the correlation between the scalar canonical variates, $u = Ya$, and $v = Xb$, subject to $\mathrm{Var}(u) = \mathrm{Var}(v) = 1$. Hence, we use the whitening trick to meet the unit variance condition: $\tilde{Y} = Y\,\Sigma_{YY}^{-1/2}$, and $\tilde{X} = X\,\Sigma_{XX}^{-1/2}$. Now the covariances of the modified matrices are identities: $\frac{1}{N}\tilde{Y}^\top\tilde{Y} = I_{d_{\text{model}}}$, $\frac{1}{N}\tilde{X}^\top\tilde{X} = I_{d_{\text{model}}}$. The whitened cross-covariance is given by $C = \frac{1}{N}\tilde{Y}^\top\tilde{X} = \Sigma_{YY}^{-1/2}\,\Sigma_{YX}\,\Sigma_{XX}^{-1/2} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$.

Let the singular value decomposition breakdown of the whitened cross-covariance matrix be $C = U\,\mathrm{diag}(\rho_1, \rho_2, \ldots)\,V^\top$, with singular values $\rho_1 \geq \rho_2 \geq \cdots \geq 0$. By definition, these ρ_i are the canonical correlations. In other words, in this normalised space, CCA decomposes the relationship between X and Y into orthogonal channels, with each channel strength ρ_i, which quantifies how well that specific Y direction can be predicted from X. For completeness, the corresponding canonical directions are $a_i = \Sigma_{YY}^{-1/2} U_{:i}$ and $b_i = \Sigma_{XX}^{-1/2} V_{:i}$.

Furthermore, we analyse the matrix $K = CC^\top$. This matrix has the following singular value decomposition: $K = U\,\mathrm{diag}(\rho_i)\,V^\top V\,\mathrm{diag}(\rho_i)\,U^\top = U\,\mathrm{diag}(\rho_i^2)\,U^\top$. Importantly, whitening Y, implies the optimal linear predictor with rank constraint r captures at most the top-r canonical modes. Therefore, the fraction of explained variance is: $R^2_{\text{ceiling}}(r,\, Y) = \frac{1}{d_{\text{model}}} \sum_{i=1}^{r} \rho_i^2$.

B Experimental setup

We conduct experiments using Gemma 2 2B model with hidden dimension $d_{\text{model}} = 2304$, and a suite of pre-trained sparse autoencoders Gemma Scope Team et al. [2024], Lieberum et al. [2024]. We use SAEs trained on the post-layer residual stream with the canonical L0 sparsity target and 16,384-dimensional latent space. For training and evaluation of the transport operators, we collect post-layer residual stream hidden states computed over 250,000 tokens from the uniformly subsampled SlimPajama dataset Soboleva et al. [2023], available under Apache 2.0 license. We subsequently split the dataset into 60% train, 20% validation and 20% test splits. For each layer, we identify $\sim 5\%$ high-quality SAE features, which we use in the operator evaluation by processing 120,000 dataset tokens and applying heuristics preferring features with high semantic coherence (low token entropy), centred probability mass in the unembedding space projections, as well as most significant causal effects. Furthermore, we filter out highly redundant and dead features.

To study the dynamics of feature transport throughout the model, we investigate target decoder layers 10 and 20 and compare the reconstruction of the same set of features per target layer, offset by $k = \{1, \ldots, 9\}$. Additionally, we ablate over all target layers andthe leap size to create the heatmap shown in Figure 3. We implement transport operators as L_2-regularised ridge regression models, trained using 5-fold cross-validation with grid search over regularisation parameter α, and choose a model with the highest R^2 score. To evaluate the models, we measure the reconstructions of transport operators with regards to the selected SAE features. To address the inherent sparsity of SAE features, we ensure predicting only activated latents. Furthermore, we analyse only those, which activated at least ten times in the test dataset and achieved $R^2 > -1$.

In the transport efficiency study, we evaluate transport operators by computing whitened R^2 of the rank-r-ATO-predicted downstream residuals, for all values r starting with 1 and incremented by 50 until d_{model}.

In the causal validation, we compare the unedited and ablated models by computing perplexity over a held-out subset over 100 sequences of 256 tokens. We experiment with 3 configurations of distinct token positions, to which the modification is applied: only one position, five positions, and all positions in a sequence. In the first two cases, we randomly choose positions from throughout the sequence and average the resulting perplexity over 3 sets of positions for robustness. We perform all computation in single precision (`float32`) using M1 Pro and M2 Max hardware.