Transfer Entropy as a Measure of Information Flow in VLMs and LLMs

Jessica E. Liang[1,2], **Jianbo Shi**[1]
[1]Department of Computer and Information Science, University of Pennsylvania
[2]Operations Research Center, Massachusetts Institute of Technology
jeliang@mit.edu jshi@engineering.upenn.edu

Abstract

Understanding how information propagates within vision-language models (VLMs) and large language models (LLMs) is central to interpretability and efficient deployment. Attention weights indicate where probability mass is allocated, but do not directly measure directional influence, redundancy, or functional importance. We introduce *transfer entropy* (TE) as an information-theoretic measure of directed information flow in Transformers, and develop tractable TE estimators for high-dimensional hidden states at the granularity of layers, tokens, and attention heads. We first analyze multimodal models, beginning with LLaVA-1.5-7B and then CLIP ViT-B/16. In LLaVA-1.5-7B, vision→text TE increases with depth, suggesting that multimodal fusion is localized in the upper decoder layers. In CLIP, TE reveals late-layer redundancy in the vision tower and broader elevated TE in the text tower. We then analyze unimodal models, including RoBERTa, T5, Llama-3.2-3B, and Qwen-2.5-7B. Across these LLMs, TE reveals consistent depth-dependent structure: RoBERTa exhibits a mid-layer peak, decoder-only LLMs concentrate task-dependent computation in broad mid-depth bands, and T5 shows distinct encoder and decoder dynamics. Beyond layerwise analysis, TE also exposes token- and head-level communication roles, distinguishing sinks, broadcasters, and benign components. These signals support TE-guided layer, token, and attention-head pruning across VLMs and LLMs, consistently outperforming attention- or saliency-based heuristics under matched budgets. Overall, TE provides a unified directed-information lens for diagnosing, interpreting, and compressing Transformer-based models.

1 Introduction

Vision–Language Models (VLMs) and Large Language Models (LLMs) have become central to modern AI, enabling strong performance in multimodal reasoning, visual understanding, and natural language processing. These models rely on Transformer architectures, whose attention layers integrate information across tokens, layers, and modalities. However, their internal information dynamics remain opaque. In VLMs, it is often unclear where visual information enters the language stream; in LLMs, it is unclear which layers, tokens, and heads perform essential computation versus redundant propagation. Attention weights show where probability mass is assigned, but they do not directly measure directional influence or functional importance.

We address this gap using *transfer entropy* (TE), an information-theoretic measure of directed information flow. In Transformers, TE estimates how much a layer, token, or attention head contributes to downstream representational updates beyond what is already explained by existing states. Unlike raw attention or static similarity measures, TE captures directed information transfer across depth, training time, and modality, allowing us to identify redundancy, bottlenecks, and functional specialization.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

We develop tractable TE estimators for high-dimensional hidden states and apply them to both VLMs and LLMs. For VLMs, we study LLaVA-1.5-7B and CLIP ViT-B/16. In LLaVA-1.5-7B, vision→text TE increases with depth, suggesting that multimodal fusion is concentrated in upper decoder layers. In CLIP, TE reveals late-layer redundancy in the vision tower and broader high-TE behavior in the text tower. For LLMs, we analyze RoBERTa, T5, Llama-3.2-3B, and Qwen-2.5-7B, finding consistent depth-dependent TE profiles across encoder-only, encoder–decoder, and decoder-only architectures.

Contributions. Our main contributions are:

- **Directed information-flow analysis.** We introduce a TE-based framework for measuring information flow in Transformers at the layer, token, and attention-head levels.
- **Tractable TE estimation.** We develop tractable TE estimators for high-dimensional hidden states in VLMs and LLMs, making information-theoretic analysis practical for modern Transformer models.
- **VLM information-flow characterization.** We show that LLaVA localizes vision→text fusion in upper decoder layers, while CLIP exhibits distinct redundancy patterns in its vision and text towers.
- **Token- and head-level role discovery.** We use incoming and outgoing TE to identify sinks, broadcasters, and benign components, providing interpretable maps beyond raw attention mass.
- **TE-guided structured pruning.** We use TE to guide layer, token, and head pruning in VLMs and LLMs, achieving stronger performance than attention- or saliency-based heuristics under matched budgets.
- **Robustness across settings.** We show that TE profiles are stable across datasets and LoRA fine-tuning, while also detecting optimization pathologies such as overly large learning rates.

Overall, TE provides a unified directed-information lens for interpreting and compressing large VLMs and LLMs.

2 Related Work

Information-theoretic and directed-flow analyses. Information-theoretic views, including mutual information (MI) and the information bottleneck, have long been used to study representation learning and training dynamics [86, 82, 77, 15, 27]. However, estimating MI in high dimensions is difficult, motivating kNN estimators [41], contrastive bounds such as CPC/InfoNCE [87], and variational estimators such as MINE [5, 69]. Transfer entropy (TE) was introduced by [78] as a nonparametric measure of directed information transfer, with extensions to local TE [55], neuroscience applications [88, 8], and connections to Granger causality under linear–Gaussian assumptions [4]. High-dimensional TE estimation typically relies on kNN, Gaussian, or variational approximations [41, 4, 56, 5, 65]. Our work adopts TE as a scalable, layerwise diagnostic for Transformers, focusing on directed information flow rather than undirected representation similarity.

Representation similarity and Transformer analysis. Layerwise Transformer analyses often compare representations using SVCCA [74], CKA [40], or cosine-based similarity measures. Attribution and attention-based tools, including attention rollout/flow and gradient-based attribution, probe token-level pathways [1, 10, 83], but they do not directly measure the *new* information contributed from one layer to the next during training. In contrast, our approach uses a directed conditional information measure, $\mathrm{I}(X_{\ell+1}^{t+1}; X_{\ell}^{t+1} \mid X_{\ell+1}^{t})$, to characterize layerwise information dynamics.

Structured pruning in LLMs and VLMs. Reducing inference cost in LLMs and VLMs commonly relies on quantization [25, 18], sparse or structured pruning [24, 60], and layer dropping or replacement [23, 13, 26]. Many methods rank components using magnitude, similarity, or activation-based proxies, while derivative-based criteria study stability and sensitivity [68, 96]. Our contribution is to use TE as a directed information-flow criterion for identifying layers, heads, and tokens that add limited unique information and are therefore more suitable for structured pruning.

Vision–language models and cross-modal analysis. Modern VLMs align visual and textual representations through contrastive pretraining or generative instruction tuning, including CLIP [72], ALIGN [34], Flamingo [2], BLIP/BLIP-2 [46, 45], PaLI/PaLI-X [12], and LLaVA [54]. Existing analyses of VLM internals typically rely on probing, linear evaluation, attention, or representation similarity to study cross-modal fusion. We instead use TE as a layerwise information-flow diagnostic for how visual and textual signals propagate through adjacent blocks during fine-tuning.

Token pruning, merging, and fusion. Many visual tokens contribute marginally and can be removed, merged, or compressed across depth. In vision transformers, DynamicViT introduces hierarchical token sparsification [76], followed by learned policies [38], constraint-aware distillation [47], pruning–squeezing pipelines [93], and criteria based on necessity, dense prediction, frequency, similarity, attribution, reinforcement learning, graph structure, and hardware constraints [28, 85, 43, 49, 95, 57, 91, 39, 20, 22, 32]. Token merging methods such as ToMe and its extensions reduce information loss by fusing similar tokens [6, 44, 37, 61, 7]. For VLMs, token reduction has been studied through instruction-guided or adaptive pruning [31, 98, 11, 94], cross-modal and multi-cue selection [51, 58, 19], and coverage-, diversity-, graph-, or sink-based criteria [48, 17, 3, 100, 50, 35, 81]. Our TE-based token pruning differs by explicitly modeling directed token influence.

Attention head roles and pruning. Prior work shows that many attention heads are redundant and can be removed with limited quality loss [63, 89], using criteria such as Taylor importance [64], Fisher scores [42], entropy [63], or hybrid gradient–entropy methods such as HIES [14]. SmartTrim [92] prunes VLM tokens and heads using gradient-based saliency. Complementary work studies functional head roles such as syntactic heads, sinks, broadcasters, and benign heads [16, 89, 63, 75, 71], as well as debates on whether attention is explanatory [33, 79]. Our head-level analysis combines TE with attention-mass patterns to obtain gradient-free role-aware pruning signals.

3 Introduction to Transfer Entropy

Transfer entropy (TE), originally introduced by Schreiber [78], is an information-theoretic measure that quantifies the *directed* flow of information between two stochastic processes. Unlike symmetric measures such as mutual information, TE captures the *asymmetric* influence of one process on another, making it a powerful tool for identifying directional dependencies and potential causal relationships.

In the context of neural network training, and in particular VLMs or LLMs, TE naturally extends to measuring how the representation at one layer influences the representation at another layer in the *next training step*, after controlling for the layer's own prior state. This enables us to track how updates in a lower layer propagate upwards in the network over time, isolating the component of change attributable to inter-layer communication rather than internal memory.

For two discrete-time stochastic processes $X(t)$ and $Y(t)$, the transfer entropy from X to Y is defined as [78]:

$$T_{X\to Y} \;=\; \sum_{t} p\big(y_{t+1}, y_t, x_t\big)\, \log \frac{p\big(y_{t+1} \mid y_t, x_t\big)}{p\big(y_{t+1} \mid y_t\big)}, \tag{1}$$

where $p(\cdot)$ denotes the joint or conditional probability mass/density functions of the relevant variables.

Equivalently, TE can be expressed as a conditional mutual information (CMI) [80]:

$$T_{X\to Y} \;=\; I\big(Y_{t+1}; X_t \mid Y_t\big), \tag{2}$$

where $I(\cdot;\cdot \mid \cdot)$ is the CMI between Y_{t+1} and X_t given Y_t. This form reveals that TE measures the additional predictive information about the *future* of Y contributed by the *present* of X, after removing the influence of Y's own present state.

Because TE is inherently time-directed and conditioned on the receiver's state, it is well suited to studying training dynamics in deep networks, where representations evolve step by step. In what follows, we instantiate TE at the level of adjacent layers during fine-tuning of large language models (LLMs), turning it into a per-step, layer-to-layer influence measure.

4 Transfer Entropy for Layerwise Information Flow

4.1 Problem Formulation

In our LLM/VLM setting, we interpret the two processes $X(t)$ and $Y(t)$ in TE as the activations of layer l and layer $l{+}1$, respectively, evolving over training. The corresponding instantaneous transfer entropy is

$$\mathrm{TE}_l(t) = I\big(X_{l+1}^{t+1};\, X_l^{t+1} \,\big|\, X_{l+1}^{t}\big), \tag{3}$$

which isolates the influence of updated inputs from layer l on the updated outputs of layer $l{+}1$, beyond what can be explained by the latter's own previous state. Tracking $\mathrm{TE}_l(t)$ across training steps yields a dynamic, directed view of inter-layer information flow during fine-tuning.

Computing $\mathrm{TE}_l(t)$ exactly is intractable in LLMs and VLMs. The definition in (3) requires the true joint and conditional probability densities $p(X_{l+1}^{t+1}, X_l^{t+1}, X_{l+1}^{t})$, $p(X_{l+1}^{t}, X_l^{t+1})$, and $p(X_{l+1}^{t+1} \mid X_{l+1}^{t})$, which are unknown for high-dimensional, continuous-valued layer activations. Estimating these densities without strong assumptions is impractical due to the curse of dimensionality: hidden states in modern LLMs and VLMs often have thousands of dimensions, far exceeding what can be reliably covered by finite probe batches. Moreover, the dependencies between X_{l+1}^{t+1}, X_l^{t+1}, and X_{l+1}^{t}) are highly nonlinear, further complicating density estimation. As a result, practical TE computation relies on approximations, such as k-nearest neighbor estimators [41], Gaussian models [4][56], or variational mutual information bounds [5][65], which yield tractable, sample-based estimates but cannot recover the exact TE value.

4.2 Approximation of TE

Local perturbation model and Jacobians. Fix a layer index l and two consecutive training steps t and $t{+}1$. We study the directed dependence from the state of layer l at step t to the state of layer $l{+}1$ at step $t{+}1$, conditioned on the previous state of layer $l{+}1$ at step t. Around the operating point $x_0 = (X_l^t, X_{l+1}^t)$, inject a small, zero-mean Gaussian perturbation $\delta x \sim \mathcal{N}(0, \sigma^2 I)$ that probes the local response of adjacent layers. The details on the role of the perturbation δx are provided in Appendix A. Let

$$J_l \;=\; \left.\frac{\partial X_l^{t+1}}{\partial x}\right|_{x=x_0}, \qquad J_{l+1} \;=\; \left.\frac{\partial X_{l+1}^{t+1}}{\partial x}\right|_{x=x_0}, \tag{4}$$

and introduce the shorthand random variables

$$u \;=\; X_{l+1}^{t+1}, \qquad y \;=\; X_{l+1}^{t}, \qquad v \;=\; X_l^{t+1}. \tag{5}$$

A first-order Taylor expansion yields the local linear responses

$$u \;\approx\; u_0 + J_{l+1}\,\delta x, \tag{6}$$

$$v \;\approx\; v_0 + J_l\,\delta x, \tag{7}$$

$$y \;\approx\; y_0, \tag{8}$$

where (u_0, v_0, y_0) are deterministic offsets at x_0 and do not affect covariances. Crucially, y is evaluated at time t and thus carries no δx–induced variability at this local scale.

Theorem 4.1 (Jacobian Linear–Gaussian Approximation of Transfer Entropy (with v))**.** *Let $z := (v, y)$ and define the covariance blocks induced by (6)-(8):*

$$\Sigma_{uu} \;=\; \sigma^2\, J_{l+1} J_{l+1}^\top, \tag{9}$$

$$\Sigma_{uy} \;=\; 0, \tag{10}$$

$$\Sigma_{uz} \;=\; \sigma^2\, J_{l+1} \begin{pmatrix} J_l \\ J_{l+1} \end{pmatrix}^\top, \tag{11}$$

$$\Sigma_{zz} \;=\; \sigma^2 \begin{pmatrix} J_l J_l^\top & 0 \\ 0 & 0 \end{pmatrix}. \tag{12}$$

Define the positive semidefinite matrix

$$A(z) \;=\; \Sigma_{uu}^{-1/2}\, \Sigma_{uz}\, \Sigma_{zz}^{-1}\, \Sigma_{zu}\, \Sigma_{uu}^{-1/2}, \tag{13}$$

where Σ_{zz}^{-1} is Moore-Penrose pseudoinverse because Σ_{zz} is singular. Then the layerwise transfer entropy admits the determinant form

$$\mathrm{I}\big(X_{l+1}^{t+1};\, X_l^{t+1} \mid X_{l+1}^t\big) \;\approx\; -\frac{1}{2}\,\ln\det\big(I - A(z)\big), \tag{14}$$

and, when $\|A(z)\|$ is small, the first-order trace approximation

$$\mathrm{I}\big(X_{l+1}^{t+1};\, X_l^{t+1} \mid X_{l+1}^t\big) \;\approx\; \frac{1}{2}\,\mathrm{tr}\big(A(z)\big). \tag{15}$$

The proof of Theorem 4.1 is provided in Appendix B.

Remark 4.2 (Interpretation and edge cases). If $J_l = 0$ (no local sensitivity of v to the probe), then $A(z)$ collapses and $\mathrm{I}(X_{l+1}^{t+1}; X_l^{t+1} \mid X_{l+1}^t) \approx 0$. If J_l aligns strongly with J_{l+1}, observing v explains a large portion of the variability in u beyond y, and TE increases accordingly.

Theorem 4.3 (Cosine–Similarity Approximation of Transfer Entropy (with v))**.** *Let a mini-batch contain embeddings $\{\mathbf{z}_{l,i}, \mathbf{z}_{l+1,i}\}_{i=1}^{B}$ for consecutive samples (or tokens) indexed by i. Form finite differences*

$$\Delta\mathbf{z}_{l,i} \;=\; \mathbf{z}_{l,i+1} - \mathbf{z}_{l,i}, \tag{16}$$

$$\Delta\mathbf{z}_{l+1,i} \;=\; \mathbf{z}_{l+1,i+1} - \mathbf{z}_{l+1,i}, \tag{17}$$

which serve as surrogates for $J_l\delta x_i$ and $J_{l+1}\delta x_i$, respectively. Under a rank-one covariance approximation for each i (augmented with a small ridge ϵI), the Jacobian linear–Gaussian TE in (15) admits the practical estimator

$$\mathrm{I}\big(X_{l+1}^{t+1};\, X_l^{t+1} \mid X_{l+1}^t\big) \approx \frac{1}{2(B-1)} \sum_{i=1}^{B-1} \cos^2\big(\Delta\mathbf{z}_{l+1,i},\, \Delta\mathbf{z}_{l,i}\big) \tag{18}$$

and averaging (18) over batches yields a stable epoch-level estimate.

The proof of Theorem 4.3 is provided in Appendix C.

TE versus Granger Causality. Transfer entropy is closely related to Granger causality, but the two notions arise from different formalisms. Granger causality defines a directed influence from a source process X to a target process Y if past values of X improve the prediction of Y beyond what is achievable from the past of Y alone, typically operationalized via linear autoregressive models and variance-based hypothesis tests. In contrast, TE quantifies the same notion of directed influence using conditional mutual information in (3), and is therefore model-agnostic and, in principle, applicable to nonlinear and non-Gaussian dependencies. In the linear–Gaussian setting, the two measures are mathematically equivalent: Granger causality reduces to a log-ratio of prediction-error variances, while TE reduces to the same quantity up to a constant factor, as shown in [4]. Our Jacobian-based linear–Gaussian approximation in Theorem 4.1 can thus be viewed as a multivariate, local Granger-style model expressed in information-theoretic terms. However, our cosine-similarity estimator in Theorem 4.3 avoids fitting explicit autoregressive models and directly operates on high-dimensional layer activations in LLMs and VLMs, making it more practical in the large-scale neural setting.

Appendix D discusses the validity and error control for our TE surrogates. Appendix E compares the computational complexities of exact TE and its approximations. To illustrate their closeness in practice, an illustrative example is provided in Appendix F.

4.3 Summary of TE Approximation Algorithm

We summarize the procedure for computing instantaneous, layerwise transfer entropy using the cosine–similarity approximation in Algorithm 1. The method evaluates a fixed probe batch before and after each training update, forms finite-difference activation vectors for adjacent layers, and estimates TE via squared cosine similarity under a rank-one covariance approximation.

Algorithm 1 Instantaneous Layerwise TE via Cosine–Similarity (single combined procedure)

Require: Model with layers $1 \dots L$; training steps $t = 1 \dots T$; fixed probe batch $\mathcal{B}$; ridge $\epsilon > 0$; window size W; (optional) pooling operator $\mathrm{Pool}(\cdot)$.

Ensure: Curves $\widehat{\mathrm{TE}}_\ell(t)$ and windowed averages $\overline{\mathrm{TE}}_\ell(t)$ for all $\ell = 1, \dots, L-1$ and $t = 1, \dots, T-1$.

for $t = 1$ to $T-1$ **do**
  Run forward on $\mathcal{B}$ with θ_t and cache h_t^ℓ for all ℓ.
  Apply one optimizer step to obtain θ_{t+1}.
  Re-run forward on $\mathcal{B}$ with θ_{t+1} and cache h_{t+1}^ℓ.
  for $\ell = 1$ to $L-1$ **do** ▷ last layer has no upper neighbor
  (A) Select layer-pair tensors
    $V \leftarrow h_{t+1}^\ell$ ▷ X_ℓ^{t+1}: post-update, lower layer
    $Y \leftarrow h_t^{\ell+1}$ ▷ $X_{\ell+1}^t$: pre-update, upper layer
    $U \leftarrow h_{t+1}^{\ell+1}$ ▷ $X_{\ell+1}^{t+1}$: post-update, upper layer
  (B) Optional pooling (token/sequence level)
    if pooling enabled **then**
      $V \leftarrow \mathrm{Pool}(V);\ Y \leftarrow \mathrm{Pool}(Y);\ U \leftarrow \mathrm{Pool}(U)$
    end if
  (C) Finite differences across mini-batch order
    Let B be the number of (pooled) items in the batch.
    for $i = 1$ to $B-1$ **do**
      $\Delta \mathbf{z}_{\ell,i} \leftarrow V_{i+1} - V_i$
      $\Delta \mathbf{z}_{\ell+1,i} \leftarrow U_{i+1} - U_i$
    end for
  (D) Cosine-squared aggregation with ridge
    $s \leftarrow 0$
    for $i = 1$ to $B-1$ **do**
      $n_i \leftarrow \|\Delta \mathbf{z}_{\ell+1,i}\|^2 \cdot \|\Delta \mathbf{z}_{\ell,i}\|^2 + \epsilon$
      $c_i \leftarrow \dfrac{\left(\Delta \mathbf{z}_{\ell+1,i}^\top \Delta \mathbf{z}_{\ell,i}\right)^2}{n_i}$
      $s \leftarrow s + c_i$
    end for
  (E) Instantaneous TE estimate (per (15))
    $\widehat{\mathrm{TE}}_\ell(t) \leftarrow \dfrac{1}{2(B-1)}\, s$
  end for
end for
(F) Temporal smoothing (optional): For all ℓ, set $\overline{\mathrm{TE}}_\ell(t) \leftarrow \dfrac{1}{W} \sum_{i=0}^{W-1} \widehat{\mathrm{TE}}_\ell(t-i)$ for $t = W, \dots, T-1$.
return $\{\widehat{\mathrm{TE}}_\ell(t)\}$ and (optionally) $\{\overline{\mathrm{TE}}_\ell(t)\}$ for all layers ℓ and steps t.

4.4 TE for Layer Pruning in LLM/VLM

Rationale. The layerwise instantaneous TE in (3) quantifies how much of the *new* information in the updated upper-layer state $X_{\ell+1}^{t+1}$ is attributable to the updated lower-layer state X_ℓ^{t+1}, beyond what was predictable from the upper layer's own prior $X_{\ell+1}^t$. When $\mathrm{TE}_\ell(t)$ is large, the update at layer ℓ already explains most of the change at layer $\ell{+}1$; equivalently, layer $\ell{+}1$ contributes little *unique* information. This suggests a pruning rule that targets layer $\ell{+}1$ when the corresponding TE_ℓ is persistently high.

Score and selection. Define a smoothed, data-aware pruning score for the $(\ell{+}1)$-th layer by temporally averaging the instantaneous TE estimates from Algorithm 1. Let $\mathcal{T}$ denote the set of training steps over which TE is aggregated (e.g., one epoch or a sliding window), then the smoothed

pruning score is given by

$$S_{\ell+1} \;=\; \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \widehat{\mathrm{TE}}_{\ell}(t) \tag{19}$$

Given a pruning budget K (or a threshold τ), we then rank layers by $S_{\ell+1}$ and prune the top-K layers (or all layers satisfying $S_{\ell+1} \geq \tau$).

Policy and guardrails. We apply the rule "*higher* $\mathrm{TE}_{\ell} \Rightarrow$ *prune layer* $\ell{+}1$" subject to structural constraints typical in LLM/VLM: do not prune embeddings, the first fusion (cross-attention) block, or the final prediction head; enforce a minimum depth per module (vision encoder, text encoder, fusion decoder). In pre-LN Transformers, pruning layer $\ell{+}1$ is implemented by replacing its block with an identity (skip) mapping of the same shape, preserving residual pathways and masks. A short post-pruning fine-tune stabilizes the network.

5 Experimental Results

5.1 Information Flow in LLaVA-1.5-7B

We analyze information-flow dynamics in LLaVA-1.5-7B [54, 53] on MSCOCO-2014 [52] and Flickr8k [62]. LLaVA combines a frozen CLIP ViT-L/14 vision encoder with a LLaMA-7B decoder. Visual embeddings are projected into the LLaMA token space and inserted as image tokens, allowing visual information to influence text generation through decoder self-attention.

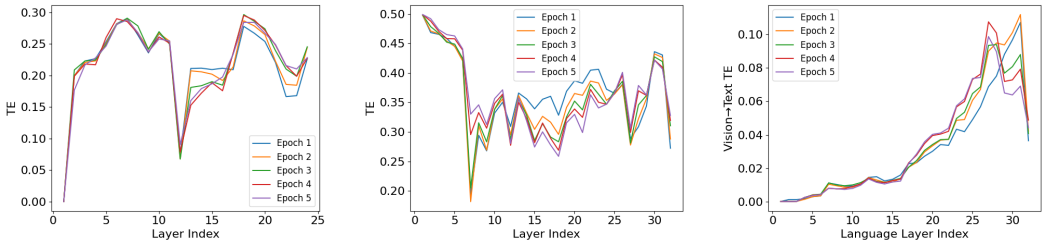
To quantify multimodal interaction, we compute a vision-to-text TE proxy inside the language tower. For each decoder layer ℓ, we pool image-token and text-token hidden states separately and define

$$\Delta_{\ell}^{\mathrm{vis}} = \mathbf{z}_{\ell,\mathrm{vis}}^{(2)} - \mathbf{z}_{\ell,\mathrm{vis}}^{(1)}, \qquad \Delta_{\ell}^{\mathrm{text}} = \mathbf{z}_{\ell,\mathrm{text}}^{(2)} - \mathbf{z}_{\ell,\mathrm{text}}^{(1)}. \tag{20}$$

The cross-modal score is

$$\mathrm{TE}_{\ell}^{\mathrm{vis}\to\mathrm{text}} \approx \tfrac{1}{2}\mathbb{E}\big[\cos^2(\Delta_{\ell}^{\mathrm{vis}}, \Delta_{\ell}^{\mathrm{text}})\big], \tag{21}$$

which measures how strongly perturbations in visual-token representations align with changes in text-token representations.

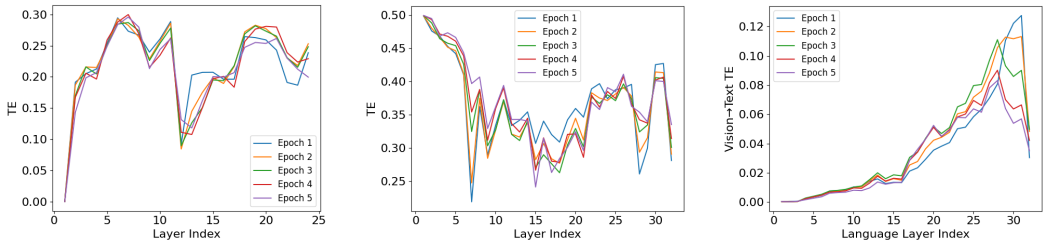


(a) (b) (c)

Figure 1: Information flow in **LLaVA-1.5-7B** using Flickr8k. TE vs. layer index during 5-epoch LoRA (rank 4) fine-tuning: (a) vision tower, (b) text tower, (c) vision→text.

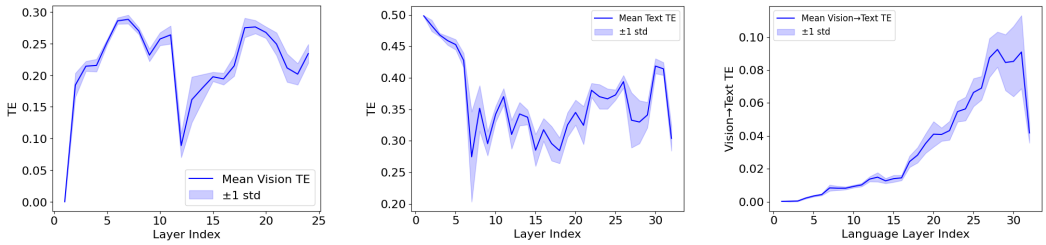
Figures 1 and 2 show the layer-wise TE curves during 5-epoch LoRA fine-tuning on Flickr8k and MSCOCO, and Figure 3 summarizes the mean and standard deviation across both datasets. Across datasets and epochs, LLaVA exhibits a stable three-part structure. First, the vision tower has a largely dataset-invariant TE profile, reflecting the frozen CLIP backbone. Second, the language tower maintains high TE across layers, consistent with iterative representation refinement in deep autoregressive transformers. Third, and most importantly, the vision→text TE is near zero in early decoder layers and increases steadily with depth, reaching its largest values in the upper language layers.

This upward vision→text trend helps explain why VLMs such as LLaVA require many decoder layers. Early layers mainly process textual and local token-level representations, while visual grounding is progressively injected and amplified only in deeper layers. Thus, the model needs sufficient depth for visual evidence to propagate through the language stack and affect final text representations. LoRA fine-tuning changes the strength of these flows but preserves the overall topology, suggesting that the roles of vision, text, and cross-modal layers remain structurally stable.



(a) (b) (c)

Figure 2: Information flow in **LLaVA-1.5-7B** using MSCOCO. TE vs. layer index during 5-epoch LoRA (rank 4) fine-tuning: (a) vision tower, (b) text tower, (c) vision→text.



(a) (b) (c)

Figure 3: Statistics of information flow in **LLaVA-1.5-7B**. Mean and standard deviation of the TE across 5-epoch LoRA (rank 4) fine-tuning on 2 datasets (MSCOCO and Flickr8k). (a) vision tower, (b) text tower, (c) vision→text.

Table 1: Layer-pruning performance on LLaVA-1.5-7B. The unpruned model obtains 0.6480. TE-based pruning consistently outperforms Streamline pruning and Short-LVLM across all pruning budgets.

# Layers Pruned	TE Pruning	Streamline Pruning	Short-LVLM
2	0.5370	0.4498	0.4548
4	0.3798	0.2937	0.2832
6	0.2867	0.1352	0.2248
8	0.2737	0.1220	0.1472

Layer pruning on LLaVA-1.5-7B. We further test whether TE can guide decoder-layer pruning in LLaVA-1.5-7B. We compare TE-based pruning with Streamline pruning [13] and Short-LVLM [59]; the unpruned model obtains 0.6480.

As shown in Table 1, TE-based pruning consistently outperforms both baselines across all pruning budgets. With 2 layers pruned, TE pruning achieves 0.5370, compared with 0.4498 for Streamline and 0.4548 for Short-LVLM. Even when pruning 8 layers, TE pruning retains 0.2737, while Streamline and Short-LVLM drop to 0.1220 and 0.1472. These results show that directed information flow is not only useful for interpreting cross-modal propagation, but also provides an actionable criterion for identifying less critical layers in multimodal transformers.

5.2 Additional Experiments on Information Flow in VLMs

We provide additional VLM experiments in Appendix G, showing that TE-based information flow is useful for both interpretation and structured compression in multimodal transformers.

First, Appendix G.1 studies layer-wise information flow in CLIP ViT-B/16 [72] after LoRA fine-tuning [30] on MSCOCO-2014 [52]. The vision tower exhibits a depth-dependent TE profile, with more redundancy concentrated in later transitions, while the text tower shows higher TE across

most layers. Using these profiles for layer pruning, TE-guided pruning consistently outperforms Short-LVLM [59] and Streamline pruning [13] under matched pruning budgets.

Second, Appendix G.2 extends information-flow analysis to tokens. By separating incoming and outgoing TE, we classify tokens as sinks, broadcasters, or benign tokens. This reveals modality-specific communication patterns: vision tokens develop localized sink/broadcaster clusters gradually across depth, whereas text tokens polarize into roles earlier and more globally. The same directional signal also provides an effective token-pruning criterion. On CLIP MSCOCO retrieval, pruning by $\mathrm{TE}_{\mathrm{out}} - \mathrm{TE}_{\mathrm{in}}$ outperforms SmartTrim [92] and Top-ViM [99]; on LLaVA-1.5-7B [54, 53] with VQAv2, TE-based vision-token pruning remains substantially stronger than Top-ViM [99] and SCOPE [17] under aggressive pruning.

Third, Appendix G.3 applies information-flow analysis to attention heads. We classify heads according to their interaction with CLS/EOT tokens and their TE scores, identifying sink, broadcaster, and benign heads. Role-aware pruning removes low-TE benign heads while preserving high-TE functional heads. On CLIP MSCOCO retrieval, TE-guided head pruning consistently outperforms SmartTrim [92], and an ablation that removes high-TE heads causes a large performance drop. This confirms that TE captures functional head importance beyond raw attention mass.

Overall, these appendix experiments support the central claim that directed information flow provides an actionable signal for VLM analysis and compression. Across layers, tokens, and heads, TE identifies components that are less critical to multimodal reasoning and yields stronger pruning performance than attention- or saliency-based baselines.

5.3 Additional Experiments on Information Flow in LLMs

We provide additional LLM experiments in Appendix H, showing that TE-based information flow is a stable diagnostic of layerwise computation and an effective signal for structured pruning across architectures, datasets, and training conditions.

First, Appendix H.1 studies $\text{RoBERTa}_{\text{base}}$ on the GLUE benchmark [90]. Across eight datasets and six epochs of full fine-tuning, the TE profiles exhibit a consistent non-monotonic pattern: information flow is low in the input and output layers but peaks in the middle layers. This suggests that intermediate layers are the main region of task adaptation and representational reuse. Using this profile for layer pruning, TE-based pruning removes four transformer layers while closely matching the full fine-tuning baseline, and it outperforms knowledge-entropy pruning [36], random LayerDrop [23], and SlimLLM [26]. A reverse ablation that prunes the lowest-TE layers causes large degradation, confirming that low-TE layers are more functionally critical.

Second, Appendix H.2 compares TE with Knowledge Entropy (KE). KE decreases monotonically with depth, reflecting progressive representation consolidation, whereas TE reveals dynamic layer-to-layer information transfer and exposes mid-layer adaptation that KE does not capture. This comparison shows that TE provides complementary and more actionable information for structured pruning.

Third, Appendix H.3 analyzes RoBERTa under LoRA fine-tuning. The TE curves remain highly stable across epochs, indicating that LoRA updates improve downstream performance without substantially changing the model's layerwise information-flow topology. This stability makes TE a repeatable pruning and diagnostic signal that does not need to be recomputed at every checkpoint.

Fourth, Appendix H.4 shows that TE can diagnose training instability. With an excessively large learning rate, TE collapses after the first epoch and validation performance degrades sharply, while a standard learning rate sustains mid-layer information flow and yields strong GLUE performance. Thus, TE also serves as an optimization diagnostic for detecting premature convergence or representational collapse.

Fifth, Appendix H.5 extends the analysis to T5-Base [73] on e-SNLI [9], CQA [84], ANLI [66], and SVAMP [67]. The decoder exhibits a clear middle-layer TE peak, while the encoder shows a more oscillatory pattern, reflecting encoder–decoder interaction. TE-guided decoder pruning is competitive with standard fine-tuning and outperforms random LayerDrop, demonstrating that the TE signal extends beyond encoder-only models.

Sixth, Appendix H.6 disentangles architecture and dataset effects by swapping datasets across RoBERTa and T5. The resulting mean and standard-deviation profiles show that the mean TE acts as an architecture-dependent fingerprint, while the TE variance identifies dataset-sensitive adaptation hotspots. This provides a principled way to separate stable architectural roles from task-specific plasticity.

Finally, Appendix H.7 evaluates larger decoder-only LLMs, including Llama-3.2-3B [21] and Qwen-2.5-7B [70]. Both models show stable high-TE mid-layer regions under LoRA fine-tuning on GLUE, with low-TE boundary layers near the input and output. This confirms that the main information-flow patterns observed in RoBERTa and T5 also appear in modern decoder-only LLMs.

Overall, the appendix experiments show that TE is robust across full fine-tuning, LoRA fine-tuning, encoder-only models, encoder–decoder models, and decoder-only LLMs. It captures layerwise information dynamics that are not visible from static entropy or raw performance alone, and it provides an actionable criterion for pruning, diagnosing training behavior, and identifying architecture- and dataset-dependent adaptation regions.

6 Conclusions and Future Work

We introduced a transfer-entropy (TE) perspective for analyzing directed information flow in Transformers, together with practical perturbation-based estimators at the granularity of *layers*, *tokens*, and *attention heads*. Unlike attention weights, which describe where probability mass is assigned, TE estimates which components drive representational updates. This yields interpretable maps of responsibility across depth, modality, tokens, and heads.

Across architectures, TE reveals stable and meaningful information-flow structure. RoBERTa shows a pronounced middle-layer TE peak; Llama-3.2-3B and Qwen-2.5-7B exhibit robust high-TE bands in mid-level decoder layers; T5 shows distinct encoder and decoder dynamics; CLIP displays late-layer redundancy in the vision tower and broader high-TE behavior in the text tower; and LLaVA-1.5-7B shows vision→text information flow concentrated in upper decoder layers. These patterns are robust under LoRA fine-tuning, predictive of pruning tolerance, and sensitive to optimization failures such as overly large learning rates.

Beyond layer-level analysis, TE also provides actionable token- and head-level signals. Token-level incoming and outgoing TE distinguish sinks, broadcasters, and benign tokens, revealing modality-specific communication roles: vision tokens develop localized spatial communication patterns, while text tokens polarize into functional roles earlier and more globally. Head-level TE similarly separates functionally important heads from low-impact benign heads, offering a clearer view of which heads drive layer updates beyond raw attention mass.

Using TE as a compression criterion, we demonstrated effective layer, token, and attention-head pruning across LLMs and VLMs. TE-guided pruning preserves performance better than competing heuristics under matched budgets in RoBERTa, T5, CLIP, and LLaVA. Ablations further show that removing high-TE components causes substantially larger degradation, confirming that TE captures functional importance rather than merely reflecting attention magnitude or static saliency. These results suggest that TE is not only useful for post-hoc interpretation, but also provides a practical signal for structured model compression.

Future work can extend TE in several directions. First, TE could be used as an online training diagnostic to detect representation collapse, guide optimizer or curriculum scheduling, and enable adaptive depth or early exiting. Second, role-aware token and head maps could guide KV-cache allocation, quantization, routing, and targeted placement of adapters or LoRA modules. Third, extending TE to cross-attention and cross-tower interactions could provide a more complete view of multimodal communication in VLMs. Finally, TE may help identify brittle or spurious information-flow patterns relevant to robustness, safety, and model steering.

Overall, TE provides a unified, data-driven lens for studying representation change in large models. By complementing attention- and similarity-based diagnostics with a directed measure of information transfer, TE supports compression, interpretability, and model steering. We provide details on computing infrastructure, datasets, and runtime in Appendix I; discuss limitations of our TE approximations in Appendix J; describe societal impacts and mitigation strategies in Appendix K; and disclose the use of LLMs in preparing this paper in Appendix L.

Acknowledgment

This material is based upon work supported by the U.S. National Science Foundation Graduate Research Fellowship under Grant No. 2637797. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[3] Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9392–9401, 2025.

[4] Lionel Barnett, Adam B. Barrett, and Anil K. Seth. Granger causality and transfer entropy are equivalent for gaussian variables. *Physical Review Letters*, 103(23):238701, 2009.

[5] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[6] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2023.

[7] Maxim Bonnaerens and Joni Dambre. Learned thresholds token merging and pruning for vision transformers. In *Workshop on Efficient Systems for Foundation Models @ ICML2023*, 2023.

[8] Terry Bossomaier, Lionel Barnett, Michael Harré, Joseph T Lizier, Terry Bossomaier, Lionel Barnett, Michael Harré, and Joseph T Lizier. *Transfer entropy*. Springer, 2016.

[9] Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[10] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 782–791, 2021.

[11] Hanning Chen, Yang Ni, Wenjun Huang, Yezi Liu, SungHeon Jeong, Fei Wen, Nathaniel D Bastian, Hugo Latapie, and Mohsen Imani. Vltp: Vision-language guided token pruning for task-oriented segmentation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 9353–9363. IEEE, 2025.

[12] Xi Chen, Xiao Wang, Soravit Changpinyo, Anthony J Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

[13] Xiaodong Chen, Yuxuan Hu, Jing Zhang, Yanling Wang, Cuiping Li, and Hong Chen. Streamlining redundant layers to compress large language models. In *International Conference on Learning Representations (ICLR)*, 2025.

[14] Minsik Choi, Hyegang Son, Changhoon Kim, and Young Geun Kim. Entropy meets importance: A unified head importance-entropy score for stable and efficient transformer pruning, 2025.

[15] Moulik Choraria, Xinbo Wu, Akhil Bhimaraju, Nitesh Sekhar, Yue Wu, Xu Zhang, Prateek Singhal, and Lav R. Varshney. Deepinsert: Early layer bypass for efficient and performant multimodal understanding. *arXiv preprint arXiv:2504.19327*, 2025.

[16] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does bert look at? an analysis of bert's attention. In *Proceedings of ACL*, 2019.

[17] Jinhong Deng, Wen Li, Joey Tianyi Zhou, and Yang He. SCOPE: Saliency-coverage oriented token pruning for efficient multimodel LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[18] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

[19] Mohamed Dhouib, Davide Buscaldi, Sonia Vanier, and Aymen Shabou. Pact: Pruning and clustering-based token reduction for faster visual language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14582–14592, 2025.

[20] Peiyan Dong, Mengshu Sun, Alec Lu, Yanyue Xie, Kenneth Liu, Zhenglun Kong, Xin Meng, Zhengang Li, Xue Lin, Zhenman Fang, et al. Heatvit: Hardware-efficient adaptive token pruning for vision transformers. In *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 442–455. IEEE, 2023.

[21] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[22] Nicholas John Eliopoulos, Purvish Jajal, James C Davis, Gaowen Liu, George K Thiravathukal, and Yung-Hsiang Lu. Pruning one more token is enough: Leveraging latency-workload non-linearities for vision transformers on the edge. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7153–7162. IEEE, 2025.

[23] Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*, 2019.

[24] Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pages 10323–10337. PMLR, 2023.

[25] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

[26] Jialong Guo, Xinghao Chen, Yehui Tang, and Yunhe Wang. SlimLLM: Accurate structured pruning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

[27] Max Hartman, Vidhata Jayaraman, Moulik Choraria, Akhil Bhimaraju, and Lav R. Varshney. Skip-it? theoretical conditions for layer skipping in vision-language models. *arXiv preprint arXiv:2509.25584*, 2025.

[28] Joakim Bruslund Haurum, Sergio Escalera, Graham W Taylor, and Thomas B Moeslund. Which tokens to use? investigating token reduction in vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 773–783, 2023.

[29] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*, 2023.

[30] 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*, 2022.

[31] Kai Huang, Hao Zou, Ye Xi, BoChen Wang, Zhen Xie, and Liang Yu. Ivtp: Instruction-guided visual token pruning for large vision-language models. In *European Conference on Computer Vision*, pages 214–230. Springer, 2024.

[32] Ryuto Ishibashi and Lin Meng. Automatic pruning rate adjustment for dynamic token reduction in vision transformer. *Applied Intelligence*, 55(5):342, 2025.

[33] Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *Proceedings of NAACL*, 2019.

[34] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.

[35] Yutao Jiang, Qiong Wu, Wenhao Lin, Wei Yu, and Yiyi Zhou. What kind of visual tokens do we need? training-free visual token pruning for multi-modal large language models from the perspective of graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 4075–4083, 2025.

[36] Jiyeon Kim, Hyunji Lee, Hyowon Cho, Joel Jang, Hyeonbin Hwang, Seungpil Won, Youbin Ahn, Dohaeng Lee, and Minjoon Seo. Knowledge entropy decay during language model pretraining hinders new knowledge acquisition. *ICLR*, 2025.

[37] Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1383–1392, 2024.

[38] Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. Learned token pruning for transformers. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 784–794, 2022.

[39] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *European conference on computer vision*, pages 620–640. Springer, 2022.

[40] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMlR, 2019.

[41] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69(6):066138, 2004.

[42] Woosuk Kwon, Sehoon Kim, Michael W. Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir Gholami. A fast post-training pruning framework for transformers. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[43] Dong-Jae Lee, Jiwan Hur, Jaehyun Choi, Jaemyung Yu, and Junmo Kim. Frequency-aware token reduction for efficient vision transformer. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[44] Sanghyeok Lee, Joonmyung Choi, and Hyunwoo J. Kim. Multi-criteria token fusion with one-step-ahead attention for efficient vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15741–15750, June 2024.

[45] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[46] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[47] Junyan Li, Li Lyna Zhang, Jiahang Xu, Yujing Wang, Shaoguang Yan, Yunqing Xia, Yuqing Yang, Ting Cao, Hao Sun, Weiwei Deng, et al. Constraint-aware and ranking-distilled token pruning for efficient transformer inference. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1280–1290, 2023.

[48] Kaiyuan Li, Xiaoyue Chen, Chen Gao, Yong Li, and Xinlei Chen. Balanced token pruning: Accelerating vision language models beyond local optimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[49] Sheng Li, Qitao Tan, Yue Dai, Zhenglun Kong, Tianyu Wang, Jun Liu, Ao Li, Ninghao Liu, Yufei Ding, Xulong Tang, and Geng Yuan. Mutual effort for efficiency: A similarity-based token pruning for vision transformers in self-supervised learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[50] Yangfu Li, Hongjian Zhan, Tianyi Chen, Qi Liu, Yu-Jie Xiong, and Yue Lu. Why $1 + 1 < 1$ in visual token pruning: Beyond naive integration via multi-objective balanced covering. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[51] Ruqi Liao, Chuqing Zhao, Jin Li, Weiqi Feng, Yi Lyu, Bingxian Chen, and Haochen Yang. Catp: Cross-attention token pruning for accuracy preserved multimodal model inference. In *2025 IEEE Conference on Artificial Intelligence (CAI)*, pages 1100–1104. IEEE, 2025.

[52] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. *European conference on computer vision (ECCV)*, pages 740–755, 2014.

[53] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024.

[54] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[55] Joseph T Lizier. *The local information dynamics of distributed computation in complex systems*. PhD thesis, University of Sydney, 2012.

[56] Joseph T. Lizier, Mikhail Prokopenko, and Albert Y. Zomaya. Local information transfer as a spatiotemporal filter for complex systems. *Physical Review E*, 77(2):026110, 2008.

[57] Chenglong Lu, Shen Liang, Xuewei Wang, and Wei Wang. Reinforcement learning-based token pruning in vision transformers: A markov game approach. *arXiv preprint arXiv:2503.23459*, 2025.

[58] Bozhi Luan, Wengang Zhou, Hao Feng, Zhe Wang, Xiaosong Li, and Houqiang Li. Multi-cue adaptive visual token pruning for large vision-language models. *arXiv preprint arXiv:2503.08019*, 2025.

[59] Ji Ma, Wei Suo, Peng Wang, and Yanning Zhang. Short-lvlm: Compressing and accelerating large vision-language models by pruning redundant layers. *ACM MM*, 2025.

[60] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720, 2023.

[61] Junzhu Mao, Yang Shen, Jinyang Guo, Yazhou Yao, Xiansheng Hua, and Hengtao Shen. Prune and merge: Efficient token compression for vision transformer with spatial information preserved. *IEEE Transactions on Multimedia*, 2025.

[62] Rubén Gaspar Marco, Sofia Strukova, Félix Gómez Mármol, and José A Ruipérez-Valiente. Annotated flickr dataset for identification of professional photographers. *Data in Brief*, 50:109511, 2023.

[63] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

[64] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11264–11272, 2019.

[65] Arnab Mondal, Arnab Bhattacharjee, Sudipto Mukherjee, Himanshu Asnani, Sreeram Kannan, and Prathosh AP. C-mi-gan: Estimation of conditional mutual information using minmax formulation. In *Conference on Uncertainty in Artificial Intelligence*, pages 849–858. PMLR, 2020.

[66] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*, 2019.

[67] Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094. Association for Computational Linguistics, June 2021.

[68] Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. *Advances in neural information processing systems*, 30, 2017.

[69] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.

[70] Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[71] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamala Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of ICML*, 2021.

[72] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[73] 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*, 21(140):1–67, 2020.

[74] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

[75] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *Proceedings of NeurIPS*, 2021.

[76] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

[77] Andrew Michael Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan Daniel Tracey, and David Daniel Cox. On the information bottleneck theory of deep learning. In *International Conference on Learning Representations*, 2018.

[78] Thomas Schreiber. Measuring information transfer. *Physical review letters*, 85(2):461, 2000.

[79] Sofia Serrano and Noah A. Smith. Is attention interpretable? In *Proceedings of ACL*, 2019.

[80] Payam Shahsavari Baboukani, Carina Graversen, Emina Alickovic, and Jan Østergaard. Estimating conditional transfer entropy in time series using mutual information and nonlinear prediction. *Entropy*, 22(10):1124, 2020.

[81] Seungjun Shin, Jaehoon Oh, and Dokwan Oh. Orthorank: Token selection via sink token orthogonality for efficient LLM inference. In *Forty-second International Conference on Machine Learning*, 2025.

[82] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[83] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[84] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of NAACL-HLT*, 2019.

[85] Quan Tang, Bowen Zhang, Jiajun Liu, Fagui Liu, and Yifan Liu. Dynamic token pruning in plain vision transformers for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 777–786, 2023.

[86] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Allerton Conference on Communication, Control, and Computing*, 1999.

[87] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint*, arXiv:1807.03748, 2018.

[88] Raul Vicente, Michael Wibral, Michael Lindner, and Gordon Pipa. Transfer entropy—a model-free measure of effective connectivity for the neurosciences. *Journal of computational neuroscience*, 30(1):45–67, 2011.

[89] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019.

[90] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[91] Hongjie Wang, Bhishma Dedhia, and Niraj K Jha. Zero-tprune: Zero-shot token pruning through leveraging of the attention graph in pre-trained transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16070–16079, 2024.

[92] Zekun Wang, Jingchang Chen, Wangchunshu Zhou, Haichao Zhu, Jiafeng Liang, Liping Shan, Ming Liu, Dongliang Xu, Qing Yang, and Bing Qin. SmartTrim: Adaptive tokens and attention pruning for efficient vision-language models. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14937–14953, Torino, Italia, May 2024. ELRA and ICCL.

[93] Siyuan Wei, Tianzhu Ye, Shen Zhang, Yao Tang, and Jiajun Liang. Joint token pruning and squeezing towards more aggressive compression of vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2092–2101, 2023.

[94] Cheng-En Wu, Jinhong Lin, Yu Hen Hu, and Pedro Morgado. Patch ranking: Token pruning as ranking prediction for efficient clip. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5842–5851. IEEE, 2025.

[95] Yaoyao Yan, Hui Yu, and Weizhi Xu. Attribution-driven adaptive token pruning for transformers. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[96] Greg Yang. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[97] Huzheng Yang, James Gee, and Jianbo Shi. Brain decodes deep nets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23030–23040, 2024.

[98] Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. Atp-llava: Adaptive token pruning for large vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24972–24982, June 2025.

[99] Zheng Zhan, Zhenglun Kong, Yifan Gong, Yushu Wu, Zichong Meng, Hangyu Zheng, Xuan Shen, Stratis Ioannidis, Wei Niu, Pu Zhao, and Yanzhi Wang. Exploring token pruning in vision state space models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[100] Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in MLLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

A Understanding the Perturbation

This section clarifies the role of the perturbation δx used in the Jacobian linear–Gaussian approximation of transfer entropy in Section 4.2. Importantly, the perturbation is *not* assumed to generate an additive temporal update of the form

$$\big(X_l^{t+1},\, X_{l+1}^{t+1}\big) \approx \big(X_l^t,\, X_{l+1}^t\big) + \delta x. \tag{22}$$

Such an assumption would collapse the Jacobians to the identity and render TE trivial. Instead, δx acts as a *local probe* for the transition map between steps t and $t{+}1$.

Perturbations as probes of the transition map. We model the update from step t to step $t{+}1$ as a deterministic map

$$F:\ x = (X_l^t,\, X_{l+1}^t) \ \longmapsto\ \big(X_l^{t+1}(x),\, X_{l+1}^{t+1}(x)\big). \tag{23}$$

The perturbation $\delta x \sim \mathcal{N}(0, \sigma^2 I)$ is applied in the *input space of* F around an operating point x_0. A first–order Taylor expansion yields

$$X_l^{t+1}(x_0 + \delta x) \approx v_0 + J_l\, \delta x, \tag{24}$$

$$X_{l+1}^{t+1}(x_0 + \delta x) \approx u_0 + J_{l+1}\, \delta x, \tag{25}$$

where J_l and J_{l+1} capture how infinitesimal changes to (X_l^t, X_{l+1}^t) propagate through the transition. These Jacobians encode the local sensitivity structure that underlies the TE approximation.

Conditioning on X_{l+1}^t and the role of y_0. The layerwise TE we approximate is

$$\mathrm{TE}_l(t) = I\big(X_{l+1}^{t+1}; X_l^{t+1} \,\big|\, X_{l+1}^t\big)\,, \tag{26}$$

so X_{l+1}^t serves as the conditioning variable. In the Jacobian approximation, however, we work *locally* around a fixed realization of the past state: for a given context y_0, we consider the TE

$$I\big(u; v \,\big|\, y = y_0\big) = I\big(X_{l+1}^{t+1}; X_l^{t+1} \,\big|\, X_{l+1}^t = y_0\big), \tag{27}$$

and approximate this quantity by linearizing the transition map at the operating point x_0 with $X_{l+1}^t = y_0$. In this local view, y is treated as a fixed context and only the innovation δx induces variability in (u, v). Concretely, the linearized model around (x_0, y_0) takes the form

$$u \approx u_0 + J_{l+1}\, \delta x, \tag{28}$$

$$v \approx v_0 + J_l\, \delta x, \tag{29}$$

$$y \approx y_0, \qquad \text{(held fixed)}. \tag{30}$$

The equality $X_{l+1}^t = y_0$ here should therefore be read as: *we are evaluating the conditional TE at a fixed realized value of the past state*, i.e., on the event $\{X_{l+1}^t = y_0\}$. Under this conditioning, y no longer carries any δx–induced variability, so the covariance between u and y in the local model satisfies

$$\Sigma_{uy} = 0, \tag{31}$$

as used in Theorem 4.1. This corresponds to computing the conditional covariance $\mathrm{Cov}(u, v \mid y = y_0)$ induced by local perturbations of the transition while keeping the past state X_{l+1}^t fixed.

B Proof of Theorem 4.1

Joint Gaussian structure. Because u and v are affine in the Gaussian perturbation δx, the pair (u, z) is jointly Gaussian with covariances given by (9)(12), and y is independent of δx to first order, hence

$$\Sigma_{uy} = 0 \tag{32}$$

Conditional entropy $\mathrm{H}(u \mid y)$. Since $\Sigma_{uy} = 0$, conditioning on y leaves the covariance of u unchanged at Σ_{uu}. Therefore

$$\mathrm{H}(u \mid y) \ =\ \frac{1}{2} \ln\Big((2\pi e)^{d_{l+1}} \det \Sigma_{uu}\Big). \tag{33}$$

Intuitively, the local probe δx injects variability only through J_{l+1}, which y cannot reduce at this scale.

Conditional entropy $\mathrm{H}(u \mid v, y)$**.** For jointly Gaussian (u, z),

$$\Sigma_{u|z} \;=\; \Sigma_{uu} - \Sigma_{uz}\Sigma_{zz}^{-1}\Sigma_{zu}. \tag{34}$$

Factoring out $\Sigma_{uu}^{1/2}$ yields the normalized Schur complement

$$\Sigma_{u|z} \;=\; \Sigma_{uu}^{1/2}\Big(I - \Sigma_{uu}^{-1/2}\Sigma_{uz}\Sigma_{zz}^{-1}\Sigma_{zu}\Sigma_{uu}^{-1/2}\Big)\Sigma_{uu}^{1/2} \;=\; \Sigma_{uu}^{1/2}\big(I - A(z)\big)\Sigma_{uu}^{1/2}. \tag{35}$$

Consequently,

$$\mathrm{H}(u \mid v, y) \;=\; \frac{1}{2}\ln\Big((2\pi e)^{d_{l+1}}\det\Sigma_{uu}\cdot\det\big(I - A(z)\big)\Big). \tag{36}$$

Subtracting and approximating. Starting from the Gaussian form of conditional entropy,

$$\mathrm{H}(u \mid y) = \tfrac{1}{2}\ln\big((2\pi e)^{d_u}\det\Sigma_{u|y}\big), \qquad \mathrm{H}(u \mid v, y) = \tfrac{1}{2}\ln\big((2\pi e)^{d_u}\det\Sigma_{u|v,y}\big), \tag{37}$$

subtracting $\mathrm{H}(u \mid v, y)$ from $\mathrm{H}(u \mid y)$ cancels the $(2\pi e)^{d_u}$ factor and leaves

$$\mathrm{I}\big(X_{l+1}^{t+1};\, X_l^{t+1} \mid X_{l+1}^{t}\big) \;=\; \tfrac{1}{2}\ln\frac{\det\Sigma_{u|y}}{\det\Sigma_{u|v,y}}. \tag{38}$$

By definition of $A(z)$, we can write

$$\Sigma_{u|v,y} = \big(I - A(z)\big)\,\Sigma_{u|y}, \tag{39}$$

so the determinants factor as

$$\begin{aligned}\frac{\det\Sigma_{u|y}}{\det\Sigma_{u|v,y}} &= \frac{\det\Sigma_{u|y}}{\det[(I - A(z))\,\Sigma_{u|y}]}\\ &= \frac{\det\Sigma_{u|y}}{\det(I - A(z))\,\det\Sigma_{u|y}}\\ &= \frac{1}{\det(I - A(z))}.\end{aligned} \tag{40}$$

Thus,

$$\mathrm{I}\big(X_{l+1}^{t+1};\, X_l^{t+1} \mid X_{l+1}^{t}\big) \;=\; -\tfrac{1}{2}\ln\det(I - A(z)), \tag{41}$$

which is (14). Since $A(z) \succeq 0$ (positive semidefinite), all its eigenvalues λ_i satisfy $0 \le \lambda_i \le 1$, hence $I - A(z)$ has eigenvalues in $(0, 1]$, ensuring $\det(I - A(z)) \in (0, 1]$ and making (14) nonnegative, as required for TE.

For the small-$A(z)$ regime, meaning that $\|A(z)\| \ll 1$ in operator norm, the matrix $A(z)$ acts as a small perturbation of the identity in $I - A(z)$. In this case, we can exploit the identity

$$\ln\det M \;=\; \mathrm{tr}\,\ln M, \tag{42}$$

which holds for any positive-definite matrix M. Applying this to $M = I - A(z)$ gives

$$\ln\det(I - A(z)) \;=\; \mathrm{tr}\,\ln(I - A(z)). \tag{43}$$

We then expand the matrix logarithm $\ln(I - A)$ via its Taylor–Mercator series, valid when $\rho(A) < 1$ (spectral radius less than 1) or, more conservatively, $\|A\| < 1$ in any sub-multiplicative norm:

$$\ln(I - A) \;=\; -A - \frac{1}{2}A^2 - \frac{1}{3}A^3 - \cdots. \tag{44}$$

Taking the trace of (44) gives

$$\mathrm{tr}\,\ln(I - A) = -\,\mathrm{tr}(A) - \frac{1}{2}\mathrm{tr}(A^2) - \frac{1}{3}\mathrm{tr}(A^3) - \cdots. \tag{45}$$

If A is small in norm, the higher-order terms $\mathrm{tr}(A^k)$ for $k \ge 2$ scale like $O(\|A\|^2)$ or smaller. Keeping only the leading term yields the first-order approximation:

$$\ln\det(I - A(z)) \;\approx\; -\,\mathrm{tr}\big(A(z)\big), \tag{46}$$

with an error of order $O(\|A(z)\|^2)$. Substituting this into (41) gives

$$\mathrm{I}\big(X_{l+1}^{t+1};\, X_l^{t+1} \mid X_{l+1}^t\big) \;\approx\; \tfrac{1}{2}\,\mathrm{tr}\big(A(z)\big), \tag{47}$$

which has the simple interpretation of being proportional to the total fractional reduction in the innovation variance of u due to conditioning on (v, y).

However, computing the corresponding covariance blocks exactly would require backpropagation through all hidden units and scales poorly with the hidden dimensionality of LLMs and VLMs.

C Proof of Theorem 4.3

We start from (15) and show that, under a finite-difference, rank-one approximation to the local Jacobians, the trace $\mathrm{tr}\big(A(z)\big)$ reduces to a squared cosine similarity.

From Jacobian to finite differences. Let $\mathbf{z}_l = h_l(x)$ and $\mathbf{z}_{l+1} = h_{l+1}(x)$ denote the pre-update activations of layers l and $l{+}1$ for some input x. For two nearby inputs x_i and x_{i+1}, a first-order Taylor expansion gives

$$\mathbf{z}_l(x_{i+1}) - \mathbf{z}_l(x_i) \;\approx\; J_l(x_i)\,\delta x_i, \qquad \mathbf{z}_{l+1}(x_{i+1}) - \mathbf{z}_{l+1}(x_i) \;\approx\; J_{l+1}(x_i)\,\delta x_i, \tag{48}$$

where $J_l(x_i)$ and $J_{l+1}(x_i)$ are the Jacobians of layers l and $l{+}1$ with respect to the input, and $\delta x_i = x_{i+1} - x_i$. In practice, we do not form J_l explicitly; instead, we use the observed finite differences

$$\Delta\mathbf{z}_{l,i} \;=\; \mathbf{z}_{l,i+1} - \mathbf{z}_{l,i} \;\approx\; J_l(x_i)\,\delta x_i, \tag{49}$$

$$\Delta\mathbf{z}_{l+1,i} \;=\; \mathbf{z}_{l+1,i+1} - \mathbf{z}_{l+1,i} \;\approx\; J_{l+1}(x_i)\,\delta x_i, \tag{50}$$

as surrogates for the Jacobian–input products $J_l\,\delta x_i$ and $J_{l+1}\,\delta x_i$. This viewpoint justifies interpreting the local covariance matrices in TE as being dominated by the outer products of these difference vectors.

Rank-one surrogates. For a fixed i, we therefore approximate the local covariances by

$$\Sigma_{uu}^{(i)} \;\approx\; \Delta\mathbf{z}_{l+1,i}\,\Delta\mathbf{z}_{l+1,i}^\top + \epsilon I, \tag{51}$$

$$\Sigma_{uz}^{(i)} \;\approx\; \Delta\mathbf{z}_{l+1,i}\begin{pmatrix}\Delta\mathbf{z}_{l,i}^\top & \Delta\mathbf{z}_{l+1,i}^\top\end{pmatrix}, \tag{52}$$

$$\Sigma_{zz}^{(i)} \;\approx\; \begin{pmatrix}\Delta\mathbf{z}_{l,i}\Delta\mathbf{z}_{l,i}^\top & \Delta\mathbf{z}_{l,i}\Delta\mathbf{z}_{l+1,i}^\top \\ \Delta\mathbf{z}_{l+1,i}\Delta\mathbf{z}_{l,i}^\top & \Delta\mathbf{z}_{l+1,i}\Delta\mathbf{z}_{l+1,i}^\top\end{pmatrix} + \epsilon I, \tag{53}$$

where ϵI provides a small ridge for numerical stability. As in standard rank-one whitening, $\Sigma_{uu}^{(i)\,-1/2}$ projects mainly onto the normalized direction of $\Delta\mathbf{z}_{l+1,i}$, and a small-coupling approximation renders $\Sigma_{zz}^{(i)\,-1}$ nearly block-diagonal with scalings $(\|\Delta\mathbf{z}_{l,i}\|^2+\epsilon)^{-1}$ and $(\|\Delta\mathbf{z}_{l+1,i}\|^2+\epsilon)^{-1}$ on each block.

Trace evaluation. Substituting the rank-one forms into

$$A^{(i)} = \Sigma_{uu}^{(i)\,-1/2}\,\Sigma_{uz}^{(i)}\,\Sigma_{zz}^{(i)\,-1}\,\Sigma_{zu}^{(i)}\,\Sigma_{uu}^{(i)\,-1/2} \tag{54}$$

lets us evaluate $\mathrm{tr}\big(A^{(i)}\big)$ explicitly.

First, note that $\Sigma_{uu}^{(i)} \approx \Delta\mathbf{z}_{l+1,i}\,\Delta\mathbf{z}_{l+1,i}^\top + \epsilon I$ is a rank-one perturbation of ϵI. Its inverse square root projects primarily onto the normalized direction $\widehat{\mathbf{u}} = \Delta\mathbf{z}_{l+1,i}/\|\Delta\mathbf{z}_{l+1,i}\|$, scaled by $(\|\Delta\mathbf{z}_{l+1,i}\|^2 + \epsilon)^{-1/2}$, plus an orthogonal component that does not contribute under the rank-one construction.

Next,

$$\Sigma_{uz}^{(i)} \approx \Delta\mathbf{z}_{l+1,i}\begin{pmatrix}\Delta\mathbf{z}_{l,i}^\top & \Delta\mathbf{z}_{l+1,i}^\top\end{pmatrix}, \tag{55}$$

and $\Sigma_{zz}^{(i)\,-1}$ is approximately block-diagonal with scalings $(\|\Delta\mathbf{z}_{l,i}\|^2+\epsilon)^{-1}$ and $(\|\Delta\mathbf{z}_{l+1,i}\|^2+\epsilon)^{-1}$ on the two diagonal blocks, since cross-terms are negligible under the weak-coupling assumption.

Multiplying (55) by $\Sigma_{zz}^{(i)\,-1}$ yields

$$\Sigma_{uz}^{(i)}\Sigma_{zz}^{(i)\,-1} \;\approx\; \Delta\mathbf{z}_{l+1,i}\left(\frac{\Delta\mathbf{z}_{l,i}^\top}{\|\Delta\mathbf{z}_{l,i}\|^2+\epsilon},\; \frac{\Delta\mathbf{z}_{l+1,i}^\top}{\|\Delta\mathbf{z}_{l+1,i}\|^2+\epsilon}\right). \tag{56}$$

Right-multiplying (56) by $\Sigma_{zu}^{(i)}$ (the transpose of $\Sigma_{uz}^{(i)}$) produces

$$\Sigma_{uz}^{(i)}\Sigma_{zz}^{(i)\,-1}\Sigma_{zu}^{(i)} \;\approx\; \frac{\Delta\mathbf{z}_{l+1,i}\,\Delta\mathbf{z}_{l,i}^{\top}\,\Delta\mathbf{z}_{l,i}\,\Delta\mathbf{z}_{l+1,i}^{\top}}{(\|\Delta\mathbf{z}_{l,i}\|^2+\epsilon)(\|\Delta\mathbf{z}_{l+1,i}\|^2+\epsilon)}. \tag{57}$$

This is again rank-one, aligned with $\widehat{\mathbf{u}}$, with scalar weight

$$\frac{\big(\Delta\mathbf{z}_{l+1,i}^{\top}\Delta\mathbf{z}_{l,i}\big)^2}{(\|\Delta\mathbf{z}_{l,i}\|^2+\epsilon)(\|\Delta\mathbf{z}_{l+1,i}\|^2+\epsilon)}. \tag{58}$$

Finally, sandwiching (57) with $\Sigma_{uu}^{(i)\,-1/2}$ from (54) divides out the $(\|\Delta\mathbf{z}_{l+1,i}\|^2+\epsilon)$ term in (58), leaving

$$\mathrm{tr}\big(A^{(i)}\big) \;\approx\; \frac{\big(\Delta\mathbf{z}_{l+1,i}^{\top}\Delta\mathbf{z}_{l,i}\big)^2}{\|\Delta\mathbf{z}_{l+1,i}\|^2\,\|\Delta\mathbf{z}_{l,i}\|^2+\epsilon}. \tag{59}$$

Recognizing the normalized dot product is a cosine similarity, (59) becomes

$$\mathrm{tr}\big(A^{(i)}\big) \;\approx\; \cos^2\big(\Delta\mathbf{z}_{l+1,i},\,\Delta\mathbf{z}_{l,i}\big)+O(\epsilon). \tag{60}$$

Averaging over $i=1,\dots,B-1$ and multiplying by $1/2$ (per (15)) yields (18), and averaging over batches further reduces estimator variance. The TE estimation is obtained

$$\mathrm{I}\big(X_{l+1}^{t+1};\,X_{l}^{t+1}\mid X_{l+1}^{t}\big) \;\approx\; \frac{1}{2(B-1)}\sum_{i=1}^{B-1}\cos^2\big(\Delta\mathbf{z}_{l+1,i},\,\Delta\mathbf{z}_{l,i}\big), \tag{61}$$

D Validity and Error Control for TE Surrogates

A central challenge in applying TE to LLM/VLM hidden states is that the exact conditional mutual information in (3) is intractable at scale. Our method therefore uses a sequence of approximations: (i) a *local linear–Gaussian* model around a probe operating point (Theorem 4.1), (ii) a *first-order* (trace) truncation of the resulting log-determinant TE (Theorem 4.1), and (iii) a *rank-one, finite-difference* surrogate that yields a bounded cosine proxy (Theorem 4.3). This section clarifies what each approximation estimates, provides explicit error bounds for the trace truncation, and describes empirical diagnostics we use to validate that the proxy preserves *ordering* and is robust enough for pruning decisions.

D.1 What our estimator approximates (and what it does not)

Theorem 4.1 characterizes TE under a local linear response model driven by a small isotropic probe perturbation $\delta x\sim\mathcal{N}(0,\sigma^2 I)$. The resulting quantity is a *local linear–Gaussian TE* that depends on the Jacobian-induced covariances at the operating point. The cosine estimator in Theorem 4.3 is a further simplification that replaces $J\delta x$ with *finite differences* across the probe batch and imposes a rank-one covariance approximation. Accordingly, throughout the paper we use $\widehat{\mathrm{TE}}_\ell(t)$ primarily as a *relative indicator* (for ranking/pruning), rather than claiming it recovers the exact $\mathrm{TE}_\ell(t)$ in nats for the full nonlinear model.

D.2 Trace truncation error for the linear–Gaussian TE

Under the linear–Gaussian model (Theorem 4.1), TE admits

$$\mathrm{TE}_{\mathrm{LG}}=-\tfrac{1}{2}\log\det(I-A),\qquad A\succeq 0,\;\;\rho(A)<1, \tag{62}$$

where $\rho(\cdot)$ is spectral radius. Using the standard series $-\log\det(I-A)=\sum_{k\ge1}\frac{1}{k}\mathrm{tr}(A^k)$, the trace surrogate corresponds to keeping only the $k=1$ term:

$$\mathrm{TE}_{\mathrm{trace}}=\tfrac{1}{2}\mathrm{tr}(A). \tag{63}$$

Let $\lambda_1,\dots,\lambda_m$ denote the eigenvalues of A (all in $[0,1)$). Then the truncation error has an exact series form

$$\mathrm{TE}_{\mathrm{LG}}-\mathrm{TE}_{\mathrm{trace}}=\tfrac{1}{2}\sum_{i=1}^{m}\sum_{k\ge2}\frac{\lambda_i^k}{k}\;\;\ge\;0, \tag{64}$$

i.e., the trace surrogate is *conservative* (it underestimates TE). Moreover, it admits a non-asymptotic bound:

$$0 \le \mathrm{TE}_{\mathrm{LG}} - \mathrm{TE}_{\mathrm{trace}} \le \frac{1}{4}\sum_{i=1}^{m} \frac{\lambda_i^2}{1-\lambda_i} \;\le\; \frac{1}{4}\cdot \frac{\|A\|_F^2}{1-\|A\|_2}, \tag{65}$$

where $\|\cdot\|_2$ and $\|\cdot\|_F$ are the spectral and Frobenius norms. Eq. (65) provides an explicit *a posteriori* criterion: when $\|A\|_2$ is bounded away from 1 and $\|A\|_F$ is small, the trace approximation is accurate. In Appendix F, we empirically verify this conservative behavior and trend preservation by comparing $\mathrm{TE}_{\mathrm{trace}}$ and the exact log-det TE under controlled low-rank sweeps.

Practical implication. Our pruning rule uses $\widehat{\mathrm{TE}}$ as a *ranking score*. Even when $\mathrm{TE}_{\mathrm{trace}}$ is biased low, Eq. (64) shows the bias increases smoothly with the eigenvalues of A, which preserves ordering in regimes where the spectrum is not near 1.

D.3 Local linearization error and perturbation-scale checks

The linear–Gaussian model itself is a local approximation. Write the post-update responses as

$$u = u_0 + J_{l+1}\delta x + r_u(\delta x), \qquad v = v_0 + J_l \delta x + r_v(\delta x), \tag{66}$$

where r_u, r_v are Taylor remainders. If the Jacobians are locally Lipschitz in a neighborhood of the probe point (equivalently, the relevant Hessians are bounded), then there exist constants C_u, C_v such that

$$\|r_u(\delta x)\| \le C_u \|\delta x\|^2, \qquad \|r_v(\delta x)\| \le C_v \|\delta x\|^2. \tag{67}$$

Thus, for sufficiently small σ, the remainder energy is $O(\sigma^4)$ whereas the linear term energy is $O(\sigma^2)$, implying the linear–Gaussian TE is stable to higher-order nonlinearities in the *small-perturbation* regime. In practice, we treat σ as a hyperparameter and verify stability by a *scale sweep*: we require the layerwise TE ranking to be consistent over a range of small σ (before numerical noise dominates at extremely small σ).

Takeaway. While our cosine estimator is not claimed to recover exact TE in the full nonlinear model, we provide (i) explicit error control for the log-det $\to$ trace truncation Eq. (65), (ii) a local linearization regime with perturbation-scale checks Eq. (67).

E Computational Complexity of TE and TE Approximation

Notation and cost model. Let N denote the number of items used by the estimator at a given step (e.g., $N =$ sequences in the probe batch), and let d_l, d_{l+1} be the hidden sizes at layers l and $l{+}1$. We write $d \coloneqq d_l{+}d_{l+1}$ and $d_u = d_{l+1}$. Let L be the number of layers whose TE is evaluated per step. One evaluation of the model's forward pass on the probe batch is denoted by C_{fwd} (wall-clock FLOPs/latency). Unless stated, complexities below are *per layer, per step*, not counting the two forward passes already needed to cache $\{y, v, u\}$ (Section 4).

E.1 Exact TE and density-based estimators (impractical at scale)

The definition $\mathrm{TE}_l(t) = \mathrm{I}(u; v \mid y)$ requires densities for $(u, v, y) \in \mathbb{R}^{d_u+d_l+d_u}$, i.e., a D-dimensional joint with $D = d_l{+}2d_{l+1}$. Nonparametric conditional MI estimators (e.g., kNN/Kraskov-type) need nearest-neighbor queries in D dimensions:

- **kNN CMI**: Naïve pairwise distances: $O(N^2 D)$ time and $O(ND)$ memory; space-partition trees degrade in high D, often reverting to near-quadratic time. Additionally, per-sample marginal neighbor counts (required by CMI) induce extra $O(N)$ book-keeping, keeping overall complexity $O(N^2 D)$ in practice.
- **Kernel density CMI**: Bandwidth selection and N^2 kernel accumulations yield $O(N^2 D)$ time (or $O(ND)$ with strongly compressed/random-feature approximations), with high variance in large D.

In modern VLMs where $d_l, d_{l+1} \sim 10^3$ and N is in the 10^3–10^4 range (tokens), these methods are computationally and statistically prohibitive.

E.2 Linear–Gaussian (Jacobian) approximation

Under the local linear–Gaussian model (Theorem 4.1), TE reduces to matrix covariances and a small matrix inverse on the concatenated variable $z = (v, y)$. Two cost components dominate:

(a) Forming covariances. Given the $N \times d$ design matrices of (v, y) and the $N \times d_u$ for u, empirical covariances can be computed via GEMMs in

$$O\big(Nd^2\big) \quad \text{for } \Sigma_{zz} \in \mathbb{R}^{d\times d}, \qquad O\big(Ndd_u\big) \quad \text{for } \Sigma_{uz} \in \mathbb{R}^{d_u\times d}.$$

Memory is $O(Nd)$ if streamed (Welford/online covariance) or $O(Nd)$–$O(Nd_u)$ if materialized.

(b) Small dense linear algebra. Inverting Σ_{zz} and forming $A(z)$ (refer to 13) costs

$$O(d^3) \text{ (Cholesky + solves)} \; + \; O(d^2 d_u) \text{ (products)}.$$

For d in the low thousands this is tractable on GPU, and independent of N once covariances are formed.

E.3 Variational MI bounds (MINE/InfoNCE-style)

Variational estimators optimize a critic T_ϕ to lower/upper-bound $\mathrm{I}(u; v \mid y)$: per update the complexity is $O(N\,P)$ for a critic with P parameters, plus the cost of negative sampling (often $O(N)$ or $O(N \log N)$; worst-case $O(N^2)$ if all-pairs are used). Over E training steps, time is $O(E\,N\,P)$ and memory tracks the critic activations. These methods are significantly heavier than the linear–Gaussian route and introduce extra variance/bias from optimization.

E.4 Cosine–similarity TE approximation (ours)

The estimator in Theorem 4.3 only needs finite differences and cosine similarities:

$$\text{Time:} \quad O\big(N(d_l + d_{l+1})\big), \qquad \text{Space:} \quad O(d_l + d_{l+1}) \text{ (streaming)}.$$

It adds negligible overhead beyond the two forward passes used to cache (y, v, u). Because it avoids neighbor searches, Jacobians, and critic training, it scales linearly in both N and dimensionality.

E.5 Wall-clock and practical considerations

Let $L_{\text{eval}} \le L$ be the number of layers at which TE is computed per step (possibly with stride). Excluding the two standard forward passes to cache (y, v, u), the additional per-step costs are

$$\begin{aligned}
k\text{NN CMI:} &\quad O\big(L_{\text{eval}}\, N^2(d_l{+}2d_{l+1})\big),\\
\text{Linear–Gaussian:} &\quad O\big(L_{\text{eval}}\,[N(d_l{+}d_{l+1})^2 + (d_l{+}d_{l+1})^3]\big),\\
\text{Variational } (E \text{ critic steps}): &\quad O\big(L_{\text{eval}}\, E\, N\, P\big),\\
\text{Cosine TE (ours):} &\quad O\big(L_{\text{eval}}\, N(d_l{+}d_{l+1})\big).
\end{aligned}$$

In multi-head attention where $d_l = d_{l+1} = d_{\text{model}}$, the cosine estimator scales as $O(L_{\text{eval}}\, N\, d_{\text{model}})$ and is typically dominated by the baseline forwards. In Table 2, we provide the comparison of asymptotic costs (per layer, per step), excluding the two forward passes used to cache (y, v, u) for different estimators.

Table 2: Asymptotic costs (per layer, per step), excluding the two forward passes used to cache (y, v, u).

Estimator	Time per layer	Space	Comments
kNN CMI	$O(N^2(d_l{+}2d_{l+1}))$	$O(N(d_l{+}2d_{l+1}))$	Degrades in high D
Linear–Gaussian (cov.)	$O\big(N(d_l{+}d_{l+1})^2 + (d_l{+}d_{l+1})^3\big)$	$O\big((d_l{+}d_{l+1})^2\big)$	Stable; closed form
Variational (MINE/InfoNCE)	$O(E\,N\,P)$	$O(P)$	Training overhead
Cosine TE (ours)	$O(N(d_l{+}d_{l+1}))$	$O(d_l{+}d_{l+1})$	Best cost/benefit

Practical guidance. For online diagnostics and pruning rules under tight budgets, the cosine–similarity estimator offers linear time and negligible memory overhead. When post-hoc analysis is feasible, the linear–Gaussian approximation provides calibrated, closed-form values at $O(Nd^2+d^3)$ cost. Nearest-neighbor or kernel CMI degrades sharply in high dimensions and is advisable only for very low-dimensional probes, while variational bounds, though flexible, incur the highest computational and optimization overhead and are best reserved for small-scale validation.

F TE Approximation versus Exact TE: Performance Comparison

Following Section 4, we linearize the two adjacent layers around a fixed probe batch and inject a small Gaussian perturbation, yielding $u \approx J_{l+1}\delta x + \varepsilon_u,\ v \approx J_l \delta x + \varepsilon_v$ with $\delta x \sim \mathcal{N}(0, \sigma^2 I)$ and isotropic noises $\varepsilon_u, \varepsilon_v$. Since $y = X_{l+1}^t$ is fixed at the operating point, we have $I(u; v \mid y) = I(u; v)$. We study a *latent rank sweep* by varying the number of driven directions $r \in \{1, \ldots, 8\}$ at a fixed, high alignment between the left singular subspaces of J_l and J_{l+1}. We define the latent rank r as the dimension of the driven subspace $\delta x \in \mathbb{R}^r$ that feeds both layers, equivalently $\mathrm{rank}(\Sigma_{uv})$, i.e., the number of nonzero canonical correlations between u and v that contribute to TE.

Let

$$A \;=\; \Sigma_{uu}^{-1/2}\,\Sigma_{uv}\,\Sigma_{vv}^{-1}\,\Sigma_{vu}\,\Sigma_{uu}^{-1/2}, \tag{68}$$

with covariance blocks $\Sigma_{uu}, \Sigma_{vv}, \Sigma_{uv}$ implied by J_{l+1}, J_l under the local model. We compare:

- **Exact TE**:
$$\mathrm{TE}_{\mathrm{exact}} \;=\; -\tfrac{1}{2}\ln\det\big(I - A\big). \tag{69}$$
- **Trace approximation (Theorem 4.1).** Using the first term in the series for $-\ln\det(I-A)$,
$$\mathrm{TE}_{\mathrm{trace}} \;=\; \tfrac{1}{2}\,\mathrm{tr}(A), \tag{70}$$
which omits higher-order powers of A.
- **Cosine–similarity approximation (Theorem 4.3)** Over the ordered probe batch, form finite differences $\Delta u_i = u_{i+1} - u_i$, $\Delta v_i = v_{i+1} - v_i$ and define
$$\widehat{\mathrm{TE}}_{\cos} \;=\; \overline{\cos^2\big(\Delta u_i, \Delta v_i\big)} \;\in [0,1], \tag{71}$$
a bounded geometric proxy that we *do not* map into nats for the main comparison.

To compare on a common $[0,1]$ scale, all TE-in-nats curves (Exact, Trace) are plotted using

$$\mathrm{norm}(\mathrm{TE}) \;=\; \frac{\log\big(1+\mathrm{TE}\big)}{\log\big(1+\mathrm{TE}_{\max}\big)}, \tag{72}$$

where $\mathrm{TE}_{\max}$ is the maximum $\mathrm{TE}_{\mathrm{exact}}$ within the sweep. The raw cosine proxy is already bounded and is shown directly as $\widehat{\mathrm{TE}}_{\cos} \in [0,1]$.

In Fig. 4, we summarize the normalized exact TE and the approximations from Theorems 4.1 and 4.3. Observe Fig. 4,

1. Monotone growth with rank. The exact TE increases steadily with r, reflecting the additive contribution of multiple driven directions shared by J_l and J_{l+1}.
2. Trace is conservative but trend-accurate. The first-order trace surrogate (70) systematically *underestimates* the exact TE across ranks while closely tracking its shape; the gap narrows as r increases, indicating moderate higher-order corrections.
3. Cosine-similarity preserves ordering. The raw cosine proxy rises smoothly with r and preserves the rank ordering, with a larger downward bias at small r that diminishes toward higher ranks.

Under the Jacobian Linear–Gaussian model, latent rank is a primary driver of layerwise transfer entropy. The cosine approximation computed from finite differences on cached activations provides a low-cost and stable *relative* indicator of TE changes across ranks, making it suitable for monitoring layerwise information flow during fine-tuning. For absolute quantification and theoretical alignment, the determinant form (69) is preferred; the first-order trace surrogate (70) provides a useful conservative approximation, while the cosine-similarity approximation serves as a low-cost relative indicator.

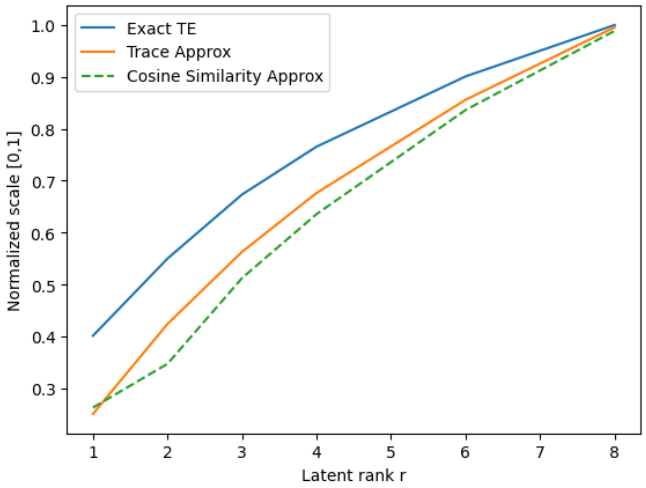


Figure 4: Normalized comparison between Exact TE, Trace (Theorem 4.1), and Cosine Similarity-based (Theorem 4.3) approximations as r increases.

G Additional Experimental Results for VLMs

G.1 Information Flow and Layer Pruning in CLIP ViT-B/16

We apply LoRA [30] fine-tuning to CLIP ViT-B/16 [72], whose vision and text encoders each consist of 12 transformer layers. LoRA fine-tuning ($r = 8$) is performed for two epochs on the MSCOCO2014 dataset [52]. Figure 5 plots TE_ℓ against layer index ℓ, where TE_ℓ denotes the transfer-entropy proxy between layer ℓ and its successor $\ell{+}1$ within either the vision or text stack.

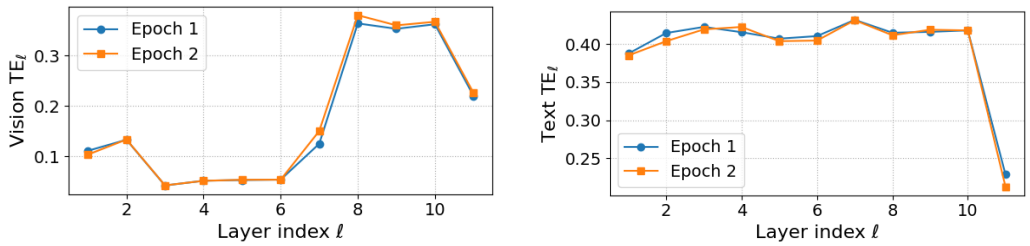


Figure 5: Layerwise TE profile of CLIP ViT-B/16 in LoRA ($r = 8$) fine tuning. (a) Vision TE_ℓ. (b) Text TE_ℓ.

As shown in Fig. 5a, across the 12-block ViT-B/16 vision backbone, the TE profile exhibits a clear depth-dependent structure. Early-to-mid transitions (ℓ=1–7) yield consistently low TE_ℓ, indicating that their successors are non-redundant and perform distinct transformations. By contrast, late transitions (ℓ=8–10) peak sharply, while the terminal transition (ℓ=11) decreases again. Under the successor-pruning rule, redundancy therefore concentrates in the successors of ℓ=8–10, identifying layers 9–11 as primary pruning targets. This pattern highlights a hierarchical importance: beginning–mid layers are essential for representation building, whereas latter layers are comparatively compressive and thus more easily removed. This observation aligns with recent findings that CLIP exhibits a hierarchical organization similar to that of the human brain [97]. In this view, early network layers correspond to early brain regions and are responsible for learning simpler concepts, while later layers correspond to higher-level brain regions and capture increasingly complex concepts. Such hierarchical alignment has been shown to facilitate fine-tuning on small datasets without catastrophic forgetting. In Fig. 5b, the text encoder displays a different behavior. The TE curve remains uniformly high across most of the depth (ℓ=1–10), with only a noticeable dip at the final transition (ℓ=11). Consequently, many successors (2–11) emerge as redundancy candidates, while the last layer

(successor of ℓ=11) should be preserved. Because high TE_ℓ values are broadly distributed rather than localized, pruning must be applied selectively and with spacing (e.g., choosing non-adjacent successors at prominent peaks) to avoid excessive loss of depth.

Table 3: Retrieval performance (MSCOCO) of CLIP ViT-B/16 under different pruning strategies.

Method	Pruned Layers	Image-to-Text Recall (%)			Text-to-Image Recall (%)		
		R@1	R@5	R@10	R@1	R@5	R@10
No Pruning	–	42.20	66.21	74.94	26.66	48.94	58.91
Our TE pruning	2V+2T	27.36	51.44	62.78	16.67	36.50	47.03
	4V+4T	15.95	34.81	45.51	9.53	24.08	33.11
	5V+5T	11.79	28.42	38.68	7.07	19.49	27.92
Short-LVLM	2V+2T	25.65	49.34	60.63	15.91	35.16	45.65
	4V+4T	13.10	30.33	40.28	7.95	20.85	29.29
	5V+5T	8.02	20.62	28.81	4.87	14.19	20.91
Streamline pruning	2V+2T	26.83	50.26	60.83	16.18	35.38	45.66
	4V+4T	12.02	28.49	38.18	7.57	20.03	28.37
	5V+5T	9.34	23.07	31.63	5.67	16.15	23.54

After LoRA (with rank 8) warm-up for two epochs, we applied TE-guided layer pruning followed by three epochs of LoRA fine-tuning. We also compared our approach with Short-LVLM [59] and streamline pruning [13]. The results are summarized in Table 3, which demonstrates that TE-based pruning consistently outperforms the other two approaches under matched pruning budgets. This indicates that TE avoids pruning functionally critical layers more effectively. Across all settings, text-to-image recall decreases more sharply than image-to-text, consistent with the known asymmetry of CLIP retrieval.

G.2 Information Flow in and out of Tokens in VLMs with Application to Token Pruning

G.2.1 Information Flow in and out of Tokens in VLMs

We consider CLIP with a text Transformer and a vision Transformer. For either encoder, let L be the number of layers, H the number of heads per layer, T the (padded) sequence length, and d the hidden size. For layer $l \in \{0, \dots, L-1\}$ and token index $j \in \{1, \dots, T\}$, let

$$h_j^{(l)} \in \mathbb{R}^d \quad \text{be the hidden state } \textit{before attention}, \text{ and}$$

$$\tilde{h}_i^{(l+1)} \in \mathbb{R}^d \quad \text{be the state } \textit{after attention and before the MLP}.$$

We denote by s the index of a suspected *attention sink* token (e.g., s=0 for `[CLS]` in the vision encoder; s=EOT position in the text encoder). All quantities are random variables induced by the data distribution (estimated over a batch or dataset).

Treating depth as discrete "time" $l \mapsto l{+}1$, we can define per-layer directional TE:

$$\mathrm{TE}_{s\to i}^{(l)} \;\coloneqq\; I\Big(\tilde{h}_i^{(l+1)};\, h_s^{(l)} \,\Big|\, h_i^{(l)}\Big), \tag{73}$$

$$\mathrm{TE}_{i\to s}^{(l)} \;\coloneqq\; I\Big(\tilde{h}_s^{(l+1)};\, h_i^{(l)} \,\Big|\, h_s^{(l)}\Big). \tag{74}$$

Aggregating over the *non-sink* tokens $\mathcal{I}_{\neg s} \coloneqq \{1, \dots, T\} \setminus \{s\}$ gives layer-l outflow and inflow scores:

$$\mathrm{TE_out}^{(l)}(s) \;=\; \frac{1}{|\mathcal{I}_{\neg s}|} \sum_{i \in \mathcal{I}_{\neg s}} \mathrm{TE}_{s\to i}^{(l)}, \tag{75}$$

$$\mathrm{TE_in}^{(l)}(s) \;=\; \frac{1}{|\mathcal{I}_{\neg s}|} \sum_{i \in \mathcal{I}_{\neg s}} \mathrm{TE}_{i\to s}^{(l)}. \tag{76}$$

If $\mathrm{TE_in}^{(l)}(s) \gg \mathrm{TE_out}^{(l)}(s)$, the sink primarily *collects* novel predictive information (collector). If $\mathrm{TE_out}^{(l)}(s) \gg \mathrm{TE_in}^{(l)}(s)$, it *broadcasts* information to others (broadcaster). If both are small while the sink attracts high attention mass, the sink effect is likely a softmax/norm artifact rather than substantive information flow.

Let $A^{(l,h)} \in [0,1]^{T\times T}$ be the attention matrix in layer l, head h (rows sum to one; rows = queries, columns = keys). The column-average attention to token j is

$$\alpha^{(l)}(j) \;=\; \frac{1}{HT}\sum_{h=1}^{H}\sum_{q=1}^{T} A_{qj}^{(l,h)}. \tag{77}$$

Typical attention sinks have very high $\alpha^{(l)}(s)$, yet may still exhibit small $\mathrm{TE_out}^{(l)}(s)$.

For layer l, let

$$t_{\text{out}}^{(l)}(j) \coloneqq \mathrm{TE_out}^{(l)}(j), \quad t_{\text{in}}^{(l)}(j) \coloneqq \mathrm{TE_in}^{(l)}(j), \quad \alpha^{(l)}(j) \coloneqq \frac{1}{HT}\sum_{h=1}^{H}\sum_{q=1}^{T} A_{qj}^{(l,h)}.$$

Optionally include the residual-stream norm $\nu^{(l)}(j) \coloneqq \|h_j^{(l)}\|_2$. To make thresholds robust, standardize each score across tokens in the layer using median and median absolute deviation (MAD):

$$\hat{x}^{(l)}(j) \;=\; \frac{x^{(l)}(j) - \mathrm{median}_k\, x^{(l)}(k)}{\mathrm{MAD}_k\, x^{(l)}(k)} \quad \text{for } x \in \{t_{\text{out}}, t_{\text{in}}, \alpha, \nu\}.$$

Define the directional gap $\widehat{\Delta}^{(l)}(j) \coloneqq \hat{t}_{\text{in}}^{(l)}(j) - \hat{t}_{\text{out}}^{(l)}(j)$.

Decision rules (per layer, data-driven). Choose layerwise thresholds $\tau_\alpha > 0$ (high attention), $\tau_{\text{out}} > 0$ (high outflow), $\tau_{\text{in}} > 0$ (high inflow), and a margin $\tau_\Delta > 0$. We choose $\tau_\alpha = 1.5$, $\tau_{\text{out}} = 1.0$, $\tau_{\text{in}} = 1.0$, and $\tau_\Delta = 0.5$ for *vision* tokens. For *text* tokens, we adopt looser thresholds to account for causal masking and shorter effective attention mass: $\tau_\alpha = 0.5$, $\tau_{\text{in}} = 0.3$, $\tau_{\text{out}} = 1.0$, and $\tau_\Delta = 0.2$.

Sink (collector): $\hat{\alpha}^{(l)}(j) \geq \tau_\alpha,\ \hat{t}_{\text{in}}^{(l)}(j) \geq \tau_{\text{in}},\ \widehat{\Delta}^{(l)}(j) \geq \tau_\Delta,$ and $\hat{t}_{\text{out}}^{(l)}(j) < \tau_{\text{out}}$. (78)

Broadcaster: $\hat{t}_{\text{out}}^{(l)}(j) \geq \tau_{\text{out}},\ \hat{t}_{\text{out}}^{(l)}(j) - \hat{t}_{\text{in}}^{(l)}(j) \geq \tau_\Delta,$ (no requirement on $\hat{\alpha}^{(l)}(j)$). (79)

Benign: all other tokens not meeting (78) or (79). (80)

We evaluate the vision and text encoders of CLIP ViT-B/16 based on MSCOCO-2014. Images are resized to 224×224 and normalized with CLIP's mean/std. The ViT encoder has $L = 12$ blocks and $T = 197$ tokens (CLS + 14×14 patches).

Figures 6–8 compare complementary TE and average attention across all layers of ViT-B/16 on vision tokens from the 14×14 patch grid. Figure 6 shows the outgoing transfer entropy ($\mathrm{TE}_{\mathrm{out}}$), highlighting tokens that act as broadcasters by distributing predictive information to many others. Figure 7 instead shows incoming transfer entropy ($\mathrm{TE}_{\mathrm{in}}$), identifying tokens that strongly depend on multiple sources, akin to sinks. Figure 8 displays the average attention weights, reflecting the raw allocation of queries to keys. While attention weights emphasize where probability mass is placed, the TE maps reveal deeper asymmetries in information flow: some patches exhibit high attention yet low TE, indicating redundancy, whereas others align across both views, suggesting true causal influence.

After LoRA (rank 8) fine tuning of CLIP ViT-B/16 using MSCOCO-2014 training dataset, we perform the token role classification based on the above algorithm. The token role classification results are plotted in Figures 9 (for vision tower) and 10 (for text tower), respectively. The vision and text token role plots reveal complementary patterns of specialization in CLIP's ViT-B/16. In the vision tower, most tokens in early layers remain benign, with only sparse sinks and broadcasters scattered across the grid. As depth increases, structured clusters of sinks and broadcasters begin to emerge, particularly from the middle layers onward. These clusters often span contiguous patches, suggesting that localized regions in the image start to assume distinct communication roles. Broadcasters spread information broadly across tokens, while sinks accumulate incoming attention mass, and

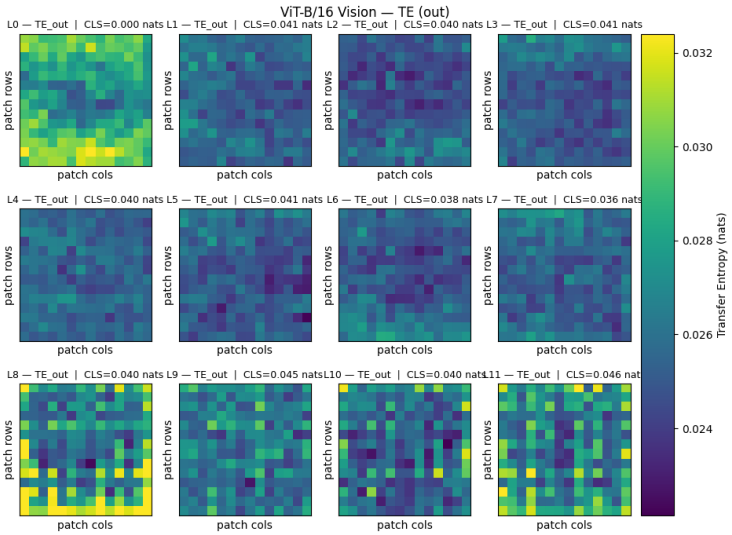


Figure 6: Transfer Entropy TE_out of ViT-B/16 (vision) tokens from the 14×14 patches. CLS status indicated in each title.

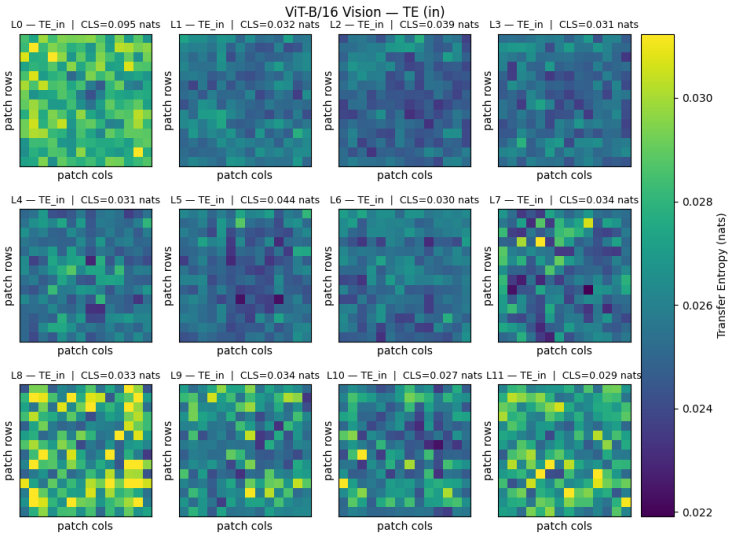


Figure 7: Transfer Entropy TE_in of ViT-B/16 (vision) tokens from the 14×14 patches. CLS status indicated in each title.

their structured distribution hints at higher-level object or region encoding. Overall, the vision tower gradually shifts from benign processing to more functionally specialized attention as representations mature.

In contrast, the text tower polarizes tokens into roles much more rapidly. By layer 2 or 3, large portions of the sequence are already dominated by sink tokens, while broadcasters appear in specific regions, especially in mid-to-late layers. Benign tokens shrink in proportion as depth increases, leaving most tokens clearly assigned as either sinks or broadcasters. The special tokens such as CLS/EOT remain consistently preserved across layers, acting as stable global anchors. This polarization reflects how linguistic tokens quickly settle into distinct roles: sinks that absorb contextual information and broadcasters that disseminate key semantic or syntactic cues across the sequence.

Comparing across towers, vision develops role structure more gradually and in a spatially localized manner, while text organizes roles earlier and more globally across the sequence. This difference reflects the modalities' distinct information structures—vision requires hierarchical feature building,

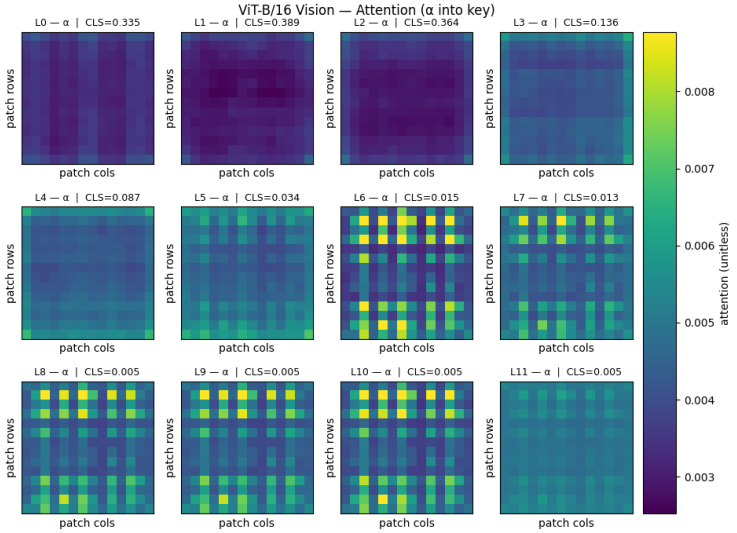


Figure 8: Average attention of ViT-B/16 (vision) tokens from the 14×14 patches. CLS status indicated in each title.

while text involves rapid semantic integration. From an interpretability perspective, these role maps highlight functional bottlenecks and communication pathways in multimodal transformers. They suggest applications such as diagnosing training pathologies through abnormal role distributions, or guiding targeted interventions that amplify or suppress specific roles to improve alignment in retrieval or generation tasks.

Classifying tokens into sink, broadcaster, and benign yields actionable signals for both analysis and deployment. Role maps provide layer-wise interpretability revealing which patches/words collect information versus disseminate it and support targeted diagnostics across datasets and corruptions. They enable efficiency tactics such as dynamic routing and early exiting (prioritizing broadcasters, cheap paths for persistently benign tokens), role-aware KV-cache allocation (retain high-impact states, compress benign ones), and adaptive attention reweighting to damp spurious broadcasters. Roles also inform gentle supervision and distillation—encouraging entity or CLS tokens to act as sinks, preserving broadcaster capacity for instruction tokens, and matching teacher–student role patterns rather than only logits. In multimodal settings, token roles yield attribution-style heatmaps that link words to image regions without gradients, and guide prompt editing by ensuring key tokens broadcast while constraints are absorbed by sinks. Finally, role-aware quantization of benign tokens offers compression with minimal accuracy risk, making token roles a versatile tool for interpretability, controllability, and efficiency.

G.2.2 Application to Token Pruning

TE-guided token pruning. Beyond analysis, the directional token-level TE scores can be used directly as a pruning criterion. Standard attention- or importance-based token pruning methods usually assign each token a scalar saliency score, but such scores do not explicitly distinguish whether a token mainly receives information from others or transmits information to others. In contrast, our formulation separately measures incoming and outgoing information flow. For each layer l and token j, we define the directional gap

$$g^{(l)}(j) = t^{(l)}_{\text{out}}(j) - t^{(l)}_{\text{in}}(j), \tag{81}$$

where $t^{(l)}_{\text{out}}(j)$ and $t^{(l)}_{\text{in}}(j)$ denote the token's outgoing and incoming transfer entropy scores, respectively. We then prune tokens with the largest positive gaps, i.e., tokens whose outgoing information flow is large relative to their incoming information flow. Empirically, our ablations show that tokens with larger $\text{TE}_{\text{out}} - \text{TE}_{\text{in}}$ are the best pruning candidates. This suggests that strongly broadcast-dominant tokens often transmit information that has already been propagated to other tokens, making them more removable than tokens that continue to collect novel contextual information.

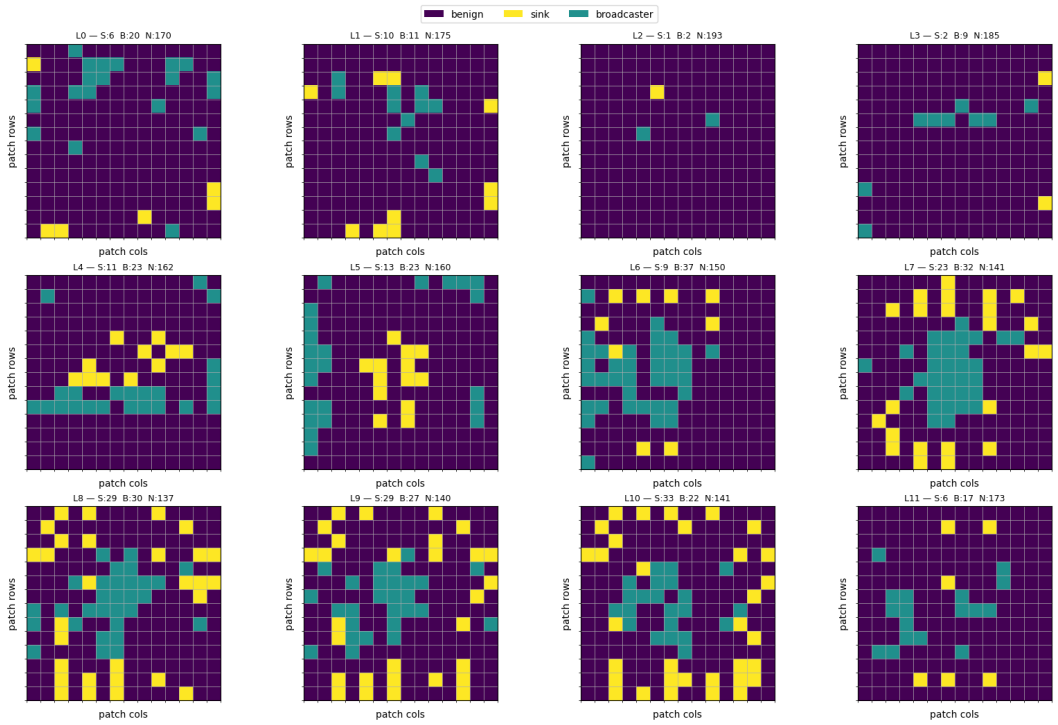


Figure 9: Vision token role classification per layer of CLIP ViT-B/16 based on MSCOCO training dataset.

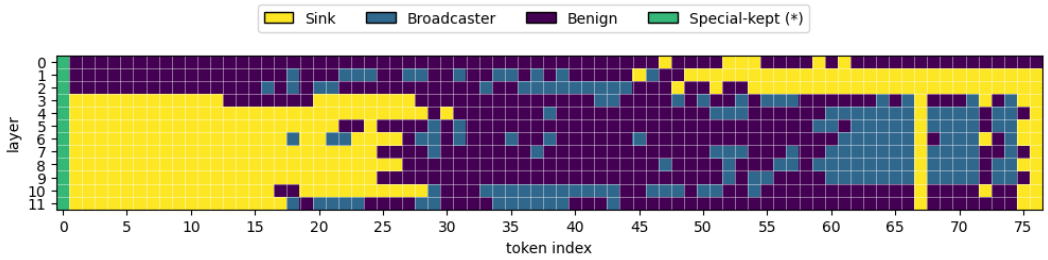


Figure 10: Text token role (length 77) classification per layer of CLIP ViT-B/16 based on MSCOCO train dataset.

CLIP retrieval under joint vision–text token pruning. We first evaluate TE-guided token pruning on CLIP ViT-B/16 using MSCOCO retrieval, where the same pruning ratio is applied to every layer in both the vision and text towers. Table 4 compares our method with SmartTrim [1] and Top-ViM [2]. TE-based pruning consistently achieves the strongest retrieval performance across all pruning ratios and all six retrieval metrics. At a moderate pruning ratio of 12.5%, TE pruning remains close to the unpruned model, obtaining 40.01 image-to-text R@1 and 25.69 text-to-image R@1, compared with 42.20 and 26.66 for no pruning. It also outperforms SmartTrim and Top-ViM by clear margins. The advantage becomes larger as the pruning ratio increases. At 25% pruning, TE achieves 39.14 image-to-text R@1, while SmartTrim and Top-ViM obtain only 32.45 and 25.11, respectively. Even at 50% pruning, TE retains 29.73 image-to-text R@1 and 18.78 text-to-image R@1, substantially improving over the attention- and importance-based baselines. These results indicate that directional information flow provides a more robust signal for deciding which tokens can be removed without severely damaging cross-modal retrieval.

LLaVA VQA under vision-token pruning. We further evaluate TE-guided pruning on LLaVA-1.5-7B using VQAv2, pruning tokens in the vision tower and reporting soft accuracy in Table 5. The unpruned model obtains 64.80% accuracy. With 20% token pruning, TE-based pruning maintains

Table 4: CLIP ViT-B/16 on MSCOCO retrieval under joint vision and text token pruning. We compare TE-based pruning with SmartTrim [92] and Top-ViM [99]. The same percentage of tokens is pruned in each layer of both the vision and text towers.

Pruning	Method	I→T R@1	I→T R@5	I→T R@10	T→I R@1	T→I R@5	T→I R@10
0%	No pruning	42.20	66.21	74.94	26.66	48.94	58.91
12.5%	TE	40.01	64.26	73.88	25.69	47.72	57.86
12.5%	SmartTrim	38.54	62.54	71.99	24.63	46.26	56.30
12.5%	Top-ViM	27.52	51.43	62.56	18.16	37.78	48.12
25%	TE	39.14	63.16	72.62	25.02	46.95	57.09
25%	SmartTrim	32.45	55.42	65.19	20.08	40.02	49.84
25%	Top-ViM	25.11	48.53	59.73	17.37	36.90	47.23
37.5%	TE	32.45	56.80	66.92	20.86	41.92	52.25
37.5%	SmartTrim	30.47	53.32	63.21	19.22	38.40	48.11
37.5%	Top-ViM	24.10	47.20	58.48	15.99	34.95	45.12
50%	TE	29.73	53.24	63.77	18.78	38.88	48.99
50%	SmartTrim	22.59	42.04	51.41	13.87	29.58	38.40
50%	Top-ViM	22.27	44.57	55.41	14.14	31.89	41.74

Table 5: LLaVA-1.5-7B on VQAv2 under vision-token pruning. We compare soft accuracy (%) of TE-based pruning with Top-ViM [99] and SCOPE [17]. Tokens are pruned in each layer of the vision tower.

Pruning	TE-based pruning	Top-ViM	SCOPE
0%	64.80	64.80	64.80
20%	64.67	57.73	62.03
30%	60.50	55.90	57.53
40%	59.80	42.07	47.07
50%	56.83	36.57	41.12
60%	56.07	34.27	36.27

64.67% accuracy, almost matching the full model, whereas Top-ViM and SCOPE drop to 57.73% and 62.03%, respectively. As the pruning ratio increases, the benefit of TE becomes more pronounced. At 40% pruning, TE achieves 59.80% accuracy, outperforming Top-ViM by 17.73 points and SCOPE by 12.73 points. Even under aggressive 60% pruning, TE retains 56.07% accuracy, while Top-ViM and SCOPE fall to 34.27% and 36.27%. Thus, TE-based pruning is not only effective for dual-encoder retrieval models such as CLIP, but also transfers to large generative VLMs where visual tokens must support downstream language generation.

Discussion. The token-pruning results support the central claim that transfer entropy captures inter-token dependence more directly than raw attention or scalar saliency. Attention weights reveal where probability mass is assigned, but they do not determine whether a token contributes novel predictive information to future representations. By comparing incoming and outgoing TE, our pruning rule distinguishes tokens that mainly collect information from tokens whose information has already been broadcast to the rest of the sequence. This distinction is especially important in VLMs, where redundant visual patches, special tokens, and text tokens can have very different communication roles. Across both CLIP on MSCOCO and LLaVA-1.5-7B on VQAv2, pruning according to $\mathrm{TE}_{\mathrm{out}} - \mathrm{TE}_{\mathrm{in}}$ yields substantially better accuracy–efficiency trade-offs than SmartTrim, Top-ViM, and SCOPE under matched pruning budgets. These findings suggest that TE is useful not only as an interpretability tool for diagnosing token roles, but also as a practical criterion for structured token pruning in multimodal transformers.

G.3 Information Flow among Attention Heads in CLIP with Application to Heads Pruning

We extend information flow analysis to *attention heads*. The aim is interpretability: to expose which heads aggregate information into the special token (CLS in vision; EOT in text), which spread information out from it, and which largely pass information without strong directional effect.

Per–head CLS/EOT mass. For layer $\ell \in \{0, \dots, L-1\}$ and head $h \in \{1, \dots, H\}$, let $P^{(\ell,h)} \in [0,1]^{T\times T}$ be attention probabilities (rows: queries, columns: keys) and let c index the special token (CLS/EOT). We summarize each head by its average attention *into* and *out of* the special token:

$$m_{\text{in}}^{(\ell,h)} := \mathbb{E}_q\left[P_{q\to c}^{(\ell,h)}\right], \qquad m_{\text{out}}^{(\ell,h)} := \mathbb{E}_k\left[P_{c\to k}^{(\ell,h)}\right], \tag{82}$$

where expectations are over tokens (and minibatch). Large m_{in} indicates funneling many queries into CLS/EOT (sink–like), whereas large m_{out} indicates broadcasting from CLS/EOT.

Layerwise TE–aware head labeling. Within each layer we standardize the per–head CLS/EOT attention masses $m_{\text{in}}^{(\ell,h)}$ and $m_{\text{out}}^{(\ell,h)}$ across heads using z–scores,

$$\tilde{m}_{\text{in}}^{(\ell,h)} = \text{zscore}\big(m_{\text{in}}^{(\ell,h)}\big), \qquad \tilde{m}_{\text{out}}^{(\ell,h)} = \text{zscore}\big(m_{\text{out}}^{(\ell,h)}\big),$$

and use the $\text{TE}_{\ell,h}$ as an impact proxy. Let $\text{pct}_p(\cdot)$ denote the p–th percentile computed *over heads in the same layer*. We form layerwise thresholds

$$\tau_{\text{in}} = \text{pct}_{p_{\text{in}}}\big(\tilde{m}_{\text{in}}^{(\ell,\cdot)}\big), \quad \tau_{\text{out}} = \text{pct}_{p_{\text{out}}}\big(\tilde{m}_{\text{out}}^{(\ell,\cdot)}\big), \quad \tau_{\text{TE}}^{\text{lo}} = \text{pct}_{p_{\text{TE}}^{\text{lo}}}\big(\text{TE}_{\ell,\cdot}\big), \quad \tau_{\text{TE}}^{\text{hi}} = \text{pct}_{p_{\text{TE}}^{\text{hi}}}\big(\text{TE}_{\ell,\cdot}\big).$$

In our experiments we use p_{in}=55, p_{out}=80, $p_{\text{TE}}^{\text{lo}}$=25, and $p_{\text{TE}}^{\text{hi}}$=60, and we *require* sufficient TE to confirm a non–benign role.

Decision rules (per head h in layer ℓ).

$$\textbf{Broadcaster:} \left(\tilde{m}_{\text{out}}^{(\ell,h)} \geq \tau_{\text{out}} \ \wedge \ \tilde{m}_{\text{in}}^{(\ell,h)} < \tau_{\text{in}}\right) \ \wedge \ \left(\text{TE}_{\ell,h} \geq \tau_{\text{TE}}^{\text{lo}}\right), \tag{83}$$

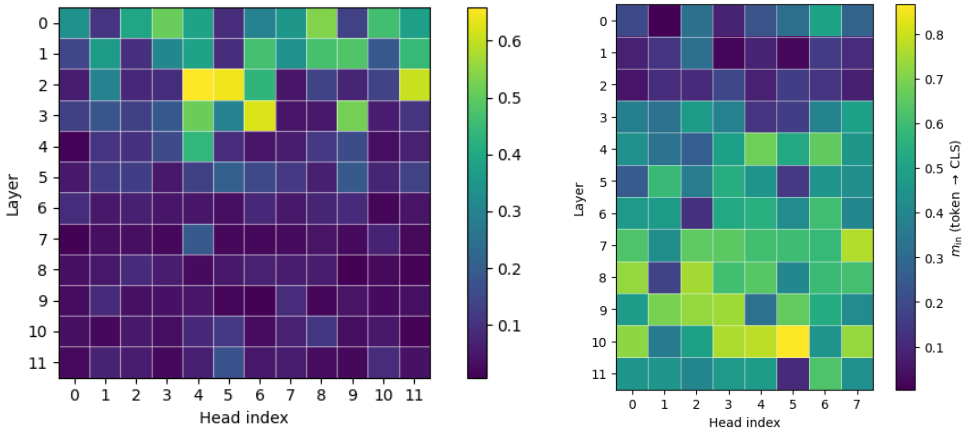
$$\textbf{Sink:} \left(\tilde{m}_{\text{in}}^{(\ell,h)} \geq \tau_{\text{in}} \ \wedge \ \tilde{m}_{\text{out}}^{(\ell,h)} < \tau_{\text{out}}\right) \ \wedge \ \left(\text{TE}_{\ell,h} \geq \tau_{\text{TE}}^{\text{lo}}\right), \tag{84}$$

$$\textbf{Ambiguous:} \text{ otherwise.} \tag{85}$$

Ambiguous heads are resolved using TE as a tie–breaker: if $\text{TE}_{\ell,h} \geq \tau_{\text{TE}}^{\text{hi}}$, assign

$$\text{role}(h) = \begin{cases} \text{sink}, & \text{if } \tilde{m}_{\text{in}}^{(\ell,h)} \geq \tilde{m}_{\text{out}}^{(\ell,h)}, \\ \text{broadcaster}, & \text{otherwise}, \end{cases}$$

else assign *benign*. Finally, as a safety demotion, any head provisionally labeled sink/broadcaster with $\text{TE}_{\ell,h} < \tau_{\text{TE}}^{\text{lo}}$ is relabeled *benign*. This scheme keeps labels driven primarily by CLS/EOT mass patterns while using TE to (i) require minimal functional impact for non–benign roles and (ii) stabilize borderline cases.



(a) (b)

Figure 11: m_{in} of each head in every layer for CLIP ViT-B/16. (a) Vision tower. (b) Text tower.

We trained CLIP ViT-B/16 with LoRA injection using rank r=8, scaling factor α=16, and no dropout. The dataset was MSCOCO-2014. Optimization was performed with AdamW at a learning rate of

10^{-4} and weight decay 0.0. The classification was done after two epochs of training. Heatmaps of m_{in} versus head index (layers on y–axis, heads on x–axis) was plotted in Fig. 11. They reveal that, in the *vision* tower, only a few heads exhibit strongly elevated m_{in} in early–to–mid layers, and this concentration softens in deeper layers—consistent with `CLS` being a modest collector while spatial interactions dominate. In contrast, the *text* tower shows steadily rising m_{in} across several heads with depth, indicating strong collection into `EOT`. Per–head TE heatmaps in Fig. 12 complement this view: vision contains sporadic, high–impact heads (isolated bright cells), whereas text displays more uniformly elevated TE in later layers, suggesting broader participation in the layer's net update. Applying the rules above yields head–role grids in Fig. 13 showing that vision layers retain a balanced mix with many benign heads interleaved with pockets of sinks/broadcasters, while text layers polarize more strongly by depth, with benign heads shrinking and clearer sink/broadcaster structure emerging.

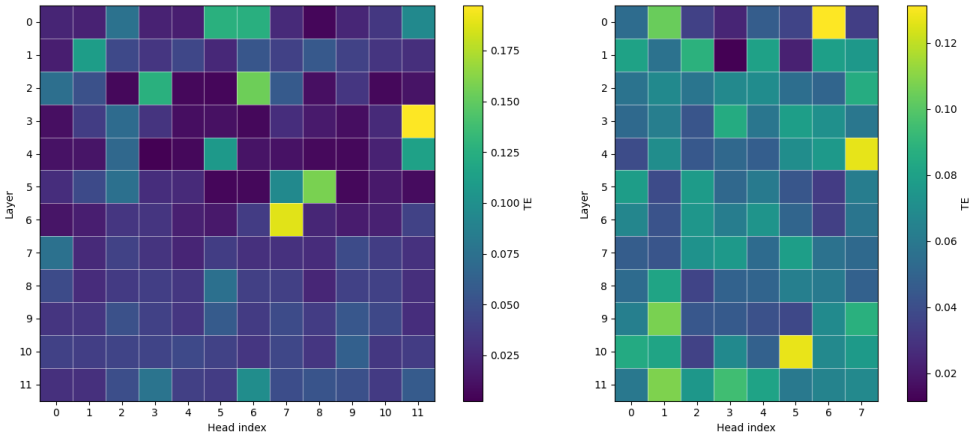


Figure 12: TE of each attention head in every layer. (a) Vision tower, (b) text tower.

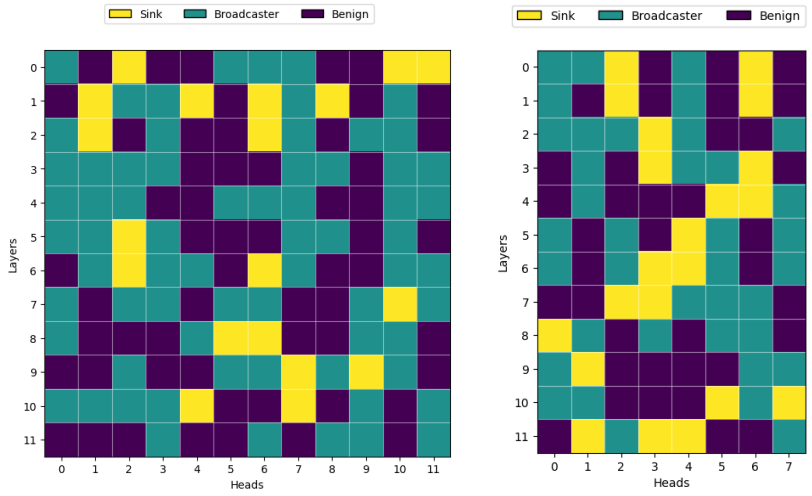


Figure 13: Attention heads classification in each layer for CLIP ViT-B/16. (a) Vision tower. (b) Text tower.

Head–level roles provide compact, layer–wise *maps of responsibility*. They support: (i) attribution of cross–modal interactions (which heads write into vs. read from `CLS`/`EOT`); (ii) diagnosis of training pathologies (e.g., layers dominated by sinks with little broadcasting); (iii) targeted qualitative analysis (visualizing only broadcaster heads when explaining an image–text match); and (iv) gentle interventions such as role–aware temperature scaling or dropout targeted at persistently benign heads

Table 6: MSCOCO retrieval performance of CLIP ViT-B/16 under attention-head pruning. Here, xV+yT denotes pruning x of the 12 attention heads per layer in the vision tower and y of the 8 attention heads per layer in the text tower. The first and last layers are kept unpruned.

Pruning per Layer	Method	I→T R@1	I→T R@5	I→T R@10	T→I R@1	T→I R@5	T→I R@10
None	No Pruning	42.20	66.21	74.94	26.66	48.94	58.91
2V + 0T	TE	39.00	63.53	73.18	24.84	47.13	57.34
2V + 0T	SmartTrim	37.25	61.65	71.42	23.69	45.61	55.86
3V + 0T	TE	37.56	61.98	71.89	24.16	46.04	56.27
3V + 0T	SmartTrim	36.25	60.48	70.38	22.95	44.62	54.84
3V + 0T	Ablation: High-TE	27.60	51.31	61.96	17.64	37.40	47.72
3V + 2T	TE	36.97	61.94	71.82	23.49	45.40	55.67
3V + 2T	SmartTrim	34.92	59.79	69.78	21.78	43.12	53.34
5V + 3T	TE	32.41	56.64	66.89	20.40	41.41	51.77
5V + 3T	SmartTrim	30.53	54.42	64.87	19.16	39.48	49.66

to improve robustness—without changing model capacity. Compared to raw attention, the joint view of $m_{\text{in}}/m_{\text{out}}$ with TE separates *where probability mass goes* from *which heads actually drive the layer update*, yielding a clearer interpretation of the communication structure inside CLIP's vision and text towers. Role-aware analysis turns attention heads into readable components: *sinks* aggregate evidence into CLS/EOT, *broadcasters* disseminate global context from it, and *benign* heads provide supportive routing. Visual and quantitative summaries make these behaviors observable, and they unlock practical levers for debugging, steering, and transferring CLIP.

TE-guided attention-head pruning. Beyond interpreting attention heads, the head-level information-flow score $\text{TE}_{\ell,h}$ can also be used as a pruning criterion. The role-classification results in Fig. 13 identify heads that act as sinks, broadcasters, or benign routing components, while the TE heatmaps in Fig. 12 measure the directed functional impact of each head on the layer update. We therefore prune heads in a role-aware manner: among heads classified as benign, we remove those with the lowest $\text{TE}_{\ell,h}$ values. A high $\text{TE}_{\ell,h}$ indicates that head h in layer ℓ has a strong directed impact on representation updates, whereas benign heads with low $\text{TE}_{\ell,h}$ are less functionally important and are therefore more suitable pruning candidates. To avoid damaging low-level and high-level aggregation, we keep the first and last layers unpruned.

MSCOCO retrieval under attention-head pruning. Table 6 reports MSCOCO retrieval performance of CLIP ViT-B/16 under matched attention-head pruning budgets. We compare TE-based pruning with SmartTrim [92]. Across all pruning configurations, TE-based pruning consistently outperforms SmartTrim on both image-to-text and text-to-image retrieval. For example, when pruning 2 vision heads per layer and no text heads, TE achieves 39.00 I→T R@1 and 24.84 T→I R@1, compared with 37.25 and 23.69 for SmartTrim. Under the stronger 3V+2T setting, TE obtains 36.97 I→T R@1 and 23.49 T→I R@1, again improving over SmartTrim's 34.92 and 21.78. Even under the aggressive 5V+3T setting, TE preserves better retrieval performance across all six metrics.

High-TE ablation. We also include an ablation that prunes high-TE heads instead of low-TE benign heads. As shown by the "Ablation: High-TE" row in Table 6, pruning high-TE heads causes a large performance drop. Under the 3V+0T budget, pruning low-TE benign heads gives 37.56 I→T R@1 and 24.16 T→I R@1, whereas pruning high-TE heads drops performance to 27.60 and 17.64, respectively. This ablation supports the interpretation that high-TE heads are functionally important and should be preserved, while low-TE benign heads are safer pruning candidates.

H Experimental Results for LLMs

H.1 Information Flow in LLMs with Application to Layer Pruning

H.1.1 Information Flow in LLMs

We evaluate TE for RoBERTa under full fine-tuning on the GLUE benchmark [90]. In Fig. 14, we summarize the TE versus the layer index (starting layer 2 to compute TE) in $\text{RoBERTa}_{\text{base}}$ fine tuning

for eight datasets on the GLUE benchmark. Across all datasets, the TE as a function of layer depth exhibits a highly consistent and distinctive profile. Specifically, the TE remains low in the input and output layers, but increases sharply through the early and middle layers, reaching a pronounced peak in the upper-middle layers of the model (typically layers 3 to 8). This non-monotonic "$\cap$-shaped" pattern is robust to both the choice of dataset and the epoch of fine-tuning. The peak TE values, observed in the middle layers, often surpass 0.2–0.3, while the TE in the final layer can be as low as 10^{-3}. Minor variations in the exact location and amplitude of the TE peak are observed across different datasets, reflecting some degree of task specificity, yet the qualitative trend remains universal. Furthermore, this profile is stable across training epochs, with early, middle, and late layers maintaining their respective TE characteristics throughout fine-tuning.

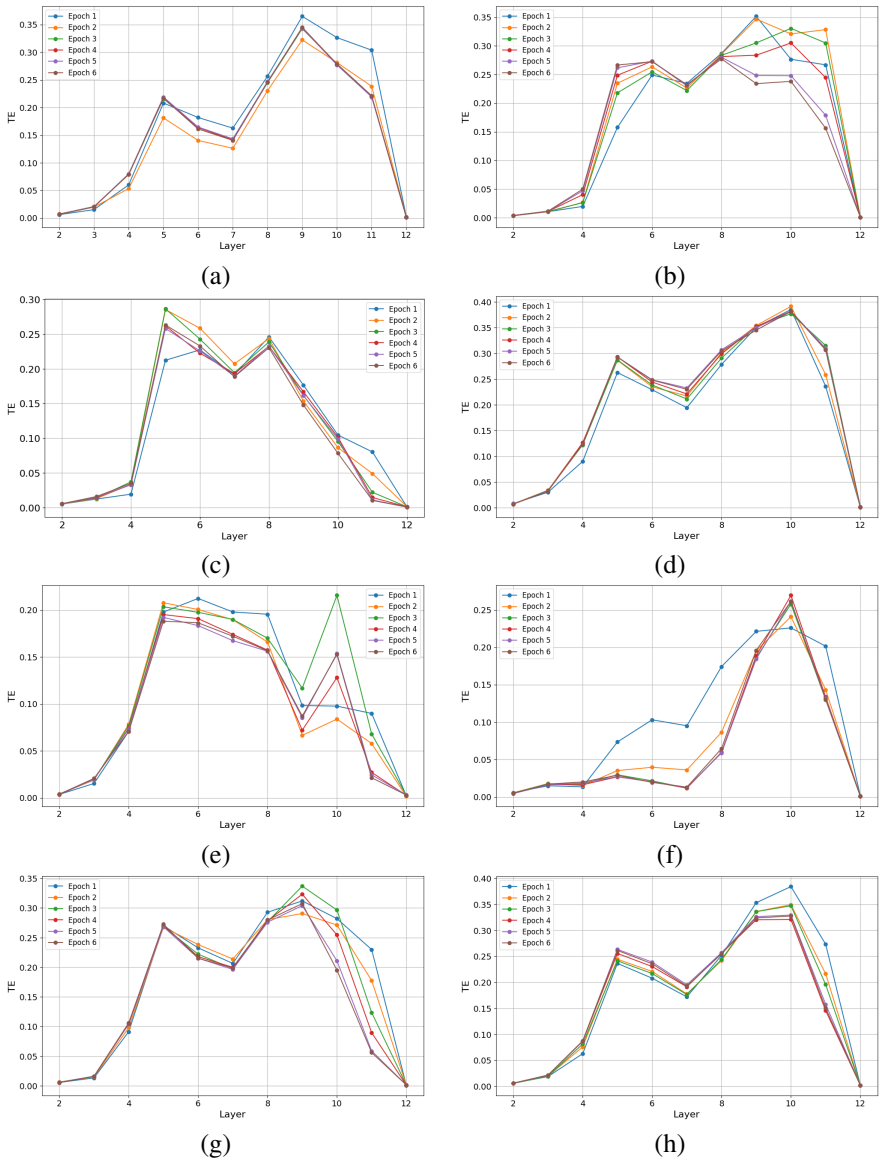


Figure 14: The TE versus layer index in RoBERTa$_{\text{base}}$ with full fine tuning for 6 epochs using eight datasets on the GLUE benchmark, (a) MNLI, (b) MRPC, (c) SST-2, (d) CoLA, (e) QNLI, (f) QQP, (g) RTE, (h) STS-B.

These results indicate a clear functional specialization across the layers of large language models. The low TE in the input layers suggests stable feature extraction, while the elevated TE in the middle layers marks a zone of increased representational uncertainty and flexibility, likely corresponding

to dynamic knowledge integration and task adaptation. Output layers, in contrast, return to low TE, reflecting highly deterministic, task-aligned decoding. The universality of this $\cap$-shaped TE profile implies that TE is an intrinsic property of transformer architectures during fine-tuning, largely independent of specific dataset or task.

Consequently, TE can serve as a principled metric for identifying which layers are most amenable to pruning or compression. Layers with high TE exhibit greater redundancy in their representations, often reflecting dynamic adaptation rather than essential information. As a result, pruning these layers with higher TE is less likely to harm the model's predictive accuracy or core capabilities, since their information is either noisy or can be compensated by neighboring layers. In contrast, low TE layers encode more stable and indispensable representations. This observation justifies a TE-guided pruning strategy, where layers with the highest TE are removed to achieve efficient model compression without significant degradation in performance.

H.1.2 Application to LLM Pruning

We prune layers using the TE. Intuitively, TE quantifies how much a layer's representation change is predictable from (or duplicative of) neighboring layers, i.e., how little novel, task-relevant information it contributes. Layers with higher TE thus encode more redundant or noisy variation and are stronger pruning candidates. We therefore rank layers by TE (higher is more prunable) and remove those at the top of the ranking subject to architectural constraints (e.g., preserving embeddings and the output head). This targeted criterion preferentially eliminates duplicative computation, reducing model size and inference latency while maintaining—often after a brief recovery fine-tune—accuracy and generalization.

Table 7 reports pruning results on $\text{RoBERTa}_{\text{base}}$ across the GLUE benchmark (MNLI, MRPC, SST-2, CoLA, QNLI, QQP, RTE, STS-B). The first row gives the fully fine-tuned baseline after 6 epochs, using the task-appropriate metric (accuracy for MNLI/QNLI/RTE/SST-2, Matthews correlation for CoLA, F1/accuracy for MRPC/QQP, and Pearson correlation for STS-B). For pruning, we remove four of the twelve transformer layers using four criteria: (1) pruning the four layers with the highest TE, (2) pruning the four layers with the highest knowledge entropy (KE) [36], (3) random structured LayerDrop following Fan et al. [23], and (4) SlimLLM layer pruning [26]. After each pruning/dropout operation, we fine-tune for 5 additional epochs. The post-pruning results are summarized in rows 2–5.

To assess whether TE provides a meaningful pruning signal, we also conduct a *reverse-TE pruning* experiment in which we prune the four layers with the *lowest* TE rather than the highest. Row 6 of Table 7 presents the corresponding results. As expected, pruning low-TE layers leads to substantial degradation, indicating that these layers introduce information that cannot be predicted from their predecessors. Conversely, high-TE layers are more redundant.

Table 7: GLUE benchmark accuracy (or correlation for STS-B) for $\text{RoBERTa}_{\text{base}}$ under full fine-tuning, TE pruning, KE pruning, layer dropout, and SlimLLM.

Method	MNLI	MRPC	SST-2	CoLA	QNLI	QQP	RTE	STS-B
No Pruning	0.7840	**0.8480**	**0.9220**	**0.8274**	**0.8731**	**0.8386**	**0.7509**	**0.9075**
Our TE Pruning	**0.7860**	**0.8480**	0.9186	0.8255	0.8548	0.8361	0.7437	0.9063
KE Pruning	0.3668	0.6838	0.7294	0.6913	0.5054	0.6318	0.5271	0.3125
Layer Dropout	0.6910	0.8235	0.9186	0.6913	0.8371	0.8003	0.7256	0.8938
SlimLLM	0.7660	0.8088	0.8979	0.8092	0.8548	0.8359	0.6787	0.9043
Low-TE Pruning	0.3470	0.6838	0.9094	0.6913	0.5094	0.6510	0.4729	0.2896

Our results show that TE pruning closely tracks the full fine-tuning baseline, indicating that TE is effective at identifying layers that contribute little novel, task-relevant information. In contrast, KE-based pruning causes substantial degradation across all tasks, suggesting it is not a reliable signal for structured layer removal in RoBERTa. LayerDrop yields intermediate performance—better than KE but consistently below the TE criterion, with especially large gaps on linguistically challenging tasks such as CoLA and QNLI. SlimLLM is competitive but generally trails our method and the no-pruning baseline, with parity on QNLI and near-parity on QQP/STS-B. Overall, these findings support the TE as a robust and principled pruning signal, enabling meaningful compression with minimal loss in downstream performance.

H.2 Comparison with Knowledge Entropy

While both our TE and the Knowledge Entropy (KE) of Kim et al. [36] aim to quantify the information dynamics within deep language models, they capture fundamentally different aspects of model behavior. The knowledge entropy measures the uncertainty or variability in how a language model utilizes its parametric knowledge, specifically focusing on how broadly the model integrates memory vectors (such as the values in feed-forward layers) via the distribution of memory coefficients. This concept reflects the model's reliance on a diverse versus narrow set of knowledge sources during pretraining, and its decline is shown to impede new knowledge acquisition and retention. In contrast, our TE formalism measures the conditional uncertainty in the representations produced by each layer during fine-tuning, explicitly modeling how the output of a layer at a given epoch depends on previous representations across both space (layers) and time (epochs). Thus, while knowledge entropy pertains to the diversity of internal memory utilization, TE characterizes the information dynamics of model representations as they evolve over training and through the network's depth, offering a broader space-time view of uncertainty propagation in LLMs.

In parallel with the TE analysis, we computed the knowledge entropy at each layer of $\text{RoBERTa}_{\text{base}}$ throughout full fine-tuning. As shown in Fig. 15, the knowledge entropy consistently exhibits a monotonic decrease from the input to the output layers, with only minor fluctuations across epochs. The initial layers maintain high knowledge entropy values (e.g., above 3.2), which steadily decline as information propagates deeper into the model, reaching their lowest levels (as low as 2.7) in the final output layers. This trend is remarkably robust, with the overall shape and absolute values of the knowledge entropy profile remaining nearly invariant across training epochs. The observed monotonic decay reflects the progressive refinement and compression of the input representation, with uncertainty and information content being gradually distilled as the model approaches its final prediction.

Contrasting this behavior with the layerwise TE profile reveals key differences in how these two measures characterize model dynamics. While knowledge entropy captures the static uncertainty of representations and reflects the gradual consolidation of knowledge along the depth of the network, the TE provides a dynamic perspective by quantifying the flow of information between adjacent layers. Notably, the TE displays a pronounced non-monotonic $\cap$-shaped curve, peaking in the middle layers and indicating zones of elevated representational flexibility and information exchange. In contrast, knowledge entropy decreases smoothly and does not capture these dynamic transitions. Thus, TE uncovers a functional specialization within the network, highlighting regions of heightened adaptation and integration that are obscured by the more static, cumulative view offered by knowledge entropy. This comparison underscores the value of TE as a complementary metric for understanding the layerwise information flow of transformer models during fine-tuning.

H.3 Information Flow in LoRA-Based Fine Tuning of LLMs

In Fig. 16, we plot the TE versus the layer index in $\text{RoBERTa}_{\text{base}}$ LoRA fine tuning for eight datasets on the GLUE benchmark. Across all six epochs of LoRA-based fine-tuning, we observe that the TE for each transformer layer remains highly stable. Specifically, the TE curves for each epoch are nearly indistinguishable, and the numerical values for each layer fluctuate only minimally from one epoch to the next. This trend holds for both lower and higher layers, indicating that the relative ordering and magnitude of TE across layers are well-preserved throughout the LoRA training process. These results suggest that LoRA adaptation, unlike standard full-parameter fine-tuning, does not substantially alter the internal information dynamics of the network as measured by TE, even as downstream task accuracy improves.

The small standard deviation of layerwise TE observed during LoRA fine-tuning demonstrates that the information-theoretic structure of the model is remarkably robust under LoRA updates. This stability has important practical implications: TE can serve as a reliable and repeatable metric for both model analysis and pruning decisions, even in the presence of continued LoRA-based adaptation. Furthermore, the insensitivity of TE to LoRA's low-rank parameter updates indicates that TE-based diagnostics and compression strategies are transferable across epochs, reducing the need for repeated metric estimation at every checkpoint. Overall, these findings reinforce the utility of TE as a principled, stable foundation for layer selection and model optimization in transformer-based architectures.

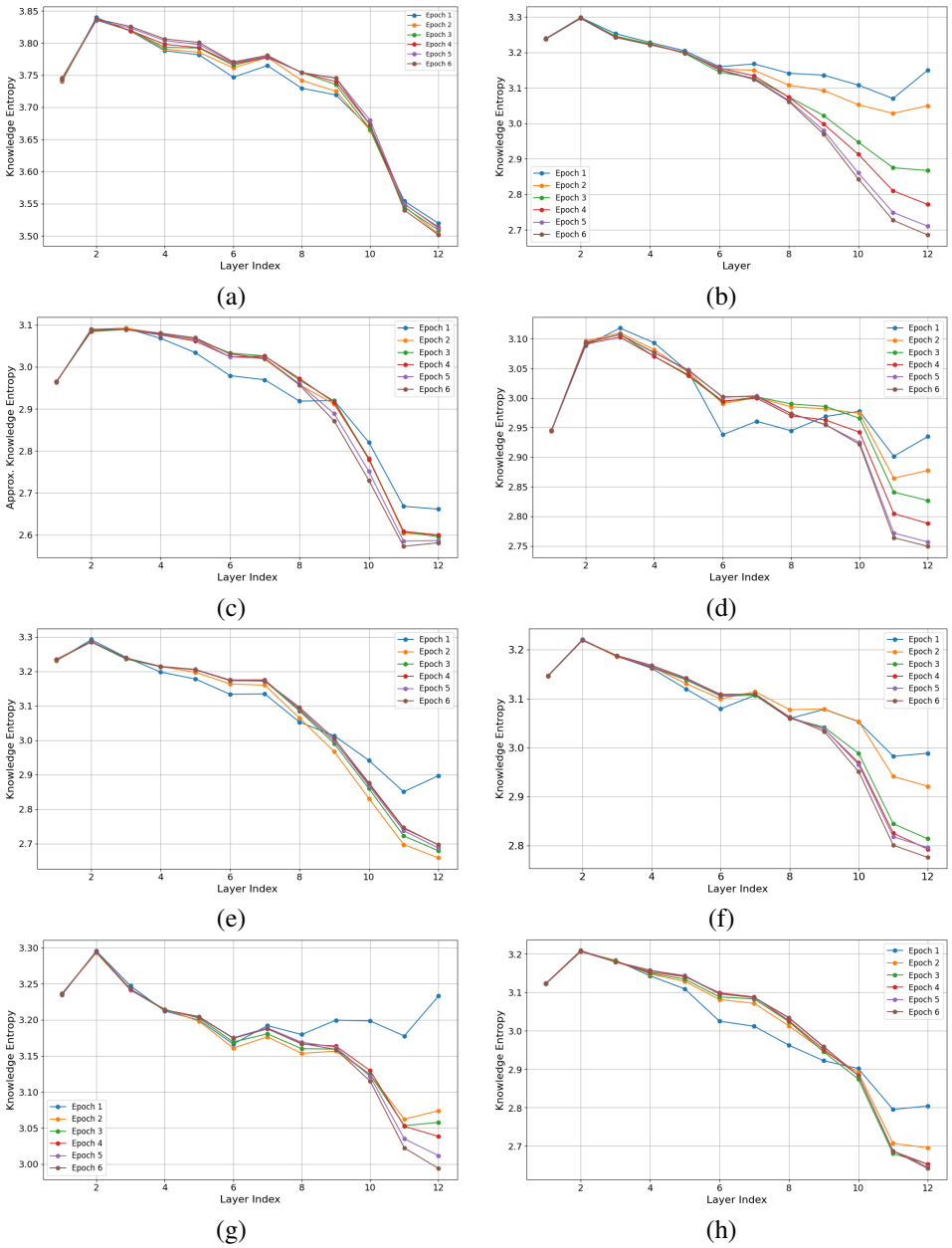


Figure 15: The knowledge entropy versus layer index in RoBERTa$_{\text{base}}$ with full fine tuning for 6 epochs using eight datasets on the GLUE benchmark, (a) MNLI, (b) MRPC, (c) SST-2, (d) CoLA, (e) QNLI, (f) QQP, (g) RTE, (h) STS-B.

H.4 Impact of Learning Rate on Information Flow

The experiments in Section H.1 use a learning rate of $\alpha = 2 \times 10^{-5}$. To isolate the effect of the learning rate on information dynamics during fine-tuning, we repeat the same setup with a substantially larger value, $\alpha = 0.02$, which is unusually high for Adam-based optimization. Unless noted otherwise, all other hyperparameters and data processing are identical to Section 4.1: RoBERTa$_{\text{base}}$, full fine-tuning for 6 epochs on the eight GLUE tasks.

Figure 17 shows the evolution of the *layerwise TE* under $\alpha = 0.02$. Across tasks, the first epoch exhibits uniformly high TEs—consistent with large representational updates as the model rapidly adapts. Beginning in epoch 2, however, the scores collapse across nearly all layers and remain low

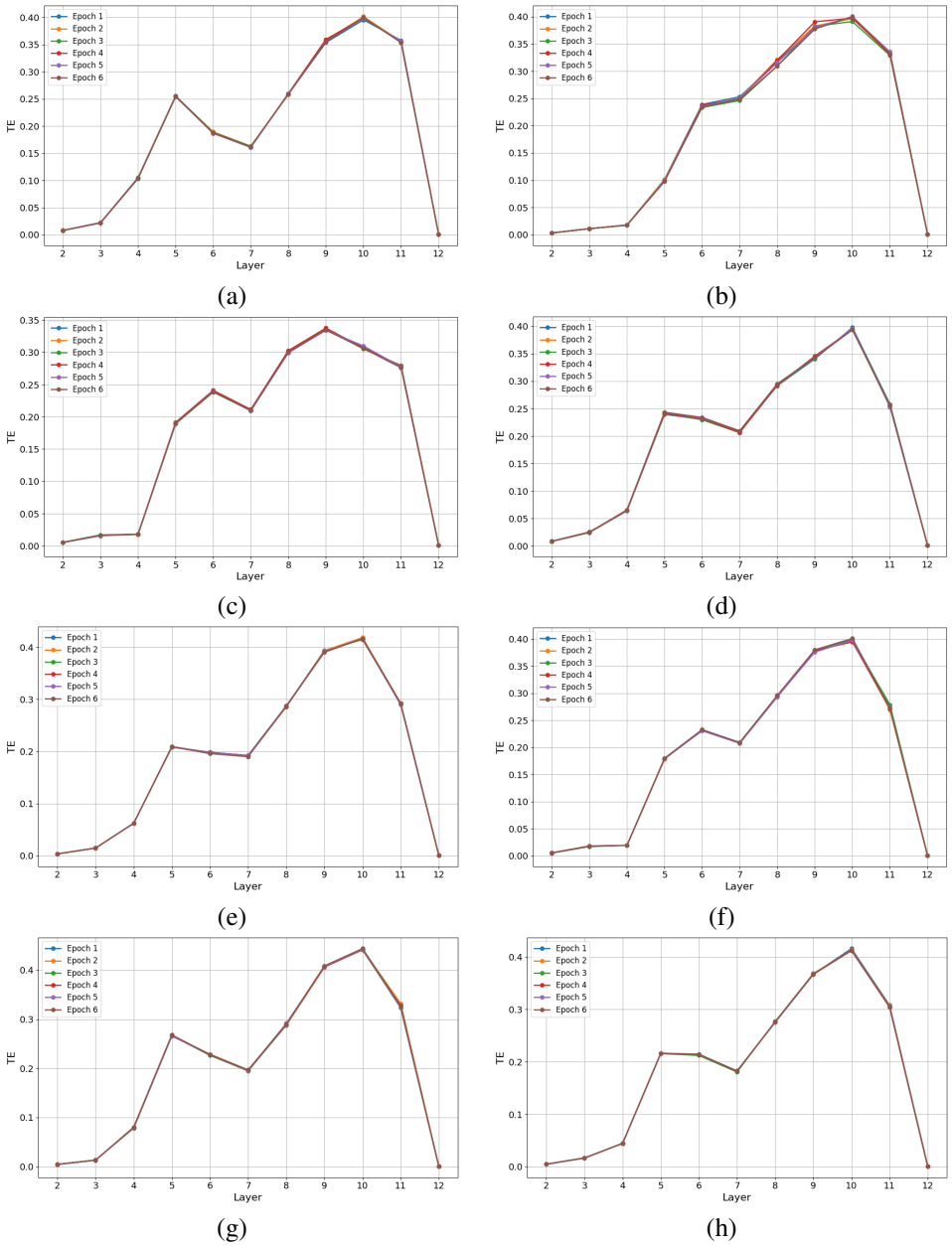


Figure 16: The TE versus layer index in RoBERTa$_{\text{base}}$ with LoRA ($r = 2$) tuning for 6 epochs using eight datasets on the GLUE benchmark, (a) MNLI, (b) MRPC, (c) SST-2, (d) CoLA, (e) QNLI, (f) QQP, (g) RTE, (h) STS-B.

thereafter. Validation accuracy plateaus at the same time, indicating that optimization has converged very quickly—often into a sharp local minimum—with little subsequent representation change. In other words, the model ceases to acquire new information, as reflected by both the suppressed TEs and the stagnant accuracy.

By contrast, with the smaller learning rate $\alpha = 2 \times 10^{-5}$ (Section 4.1), we observe a qualitatively different pattern: TEs in intermediate layers remain elevated for multiple epochs, indicating continued updates to internal representations and sustained information flow. This prolonged, non-trivial activity is accompanied by gradual accuracy improvements, consistent with steadier exploration of the solution space and better generalization.

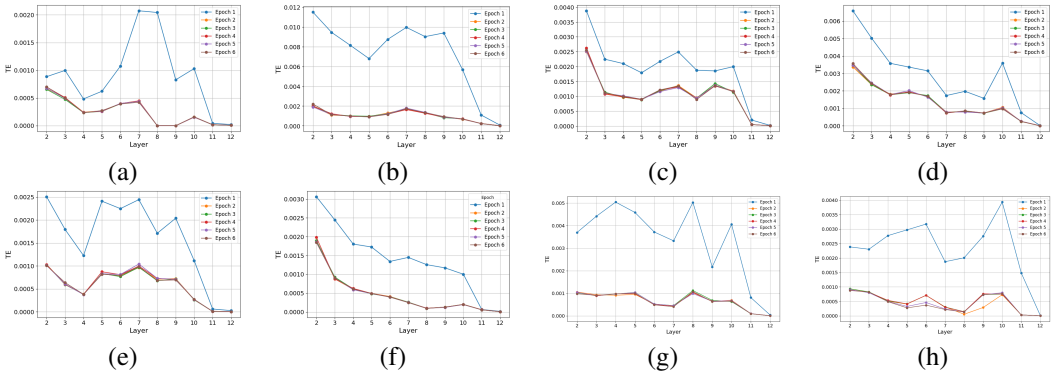


Figure 17: With learning rate $\alpha = 0.02$, *TE* versus layer index in RoBERTa$_{\text{base}}$ during 6-epoch full fine-tuning on GLUE: (a) MNLI, (b) MRPC, (c) SST-2, (d) CoLA, (e) QNLI, (f) QQP, (g) RTE, (h) STS-B.

Table 8 quantifies the effect on downstream performance. The standard learning rate yields consistently strong results (accuracy for MNLI/QNLI/RTE/SST-2/QQP/MRPC, Matthews correlation for CoLA, and Pearson correlation for STS-B), ranging from 0.75 (RTE) to 0.92 (SST-2). In contrast, the high learning rate causes large drops across all tasks—for example, MNLI $0.78 \rightarrow 0.34$, SST-2 $0.92 \rightarrow 0.49$, and QNLI $0.87 \rightarrow 0.51$—and fails to converge on STS-B (`NaN`).

Table 8: GLUE performance for RoBERTa$_{\text{base}}$ under full fine-tuning with two learning rates. Metrics are accuracy except CoLA (Matthews correlation) and STS-B (Pearson correlation).

Learning Rate	**MNLI**	**MRPC**	**SST-2**	**CoLA**	**QNLI**	**QQP**	**RTE**	**STS-B**
$\alpha = 2 \times 10^{-5}$	0.7840	0.8480	0.9220	0.8274	0.8731	0.8386	0.7509	0.9075
$\alpha = 0.02$	0.3400	0.6838	0.4908	0.6913	0.5054	0.6318	0.5271	`NaN`

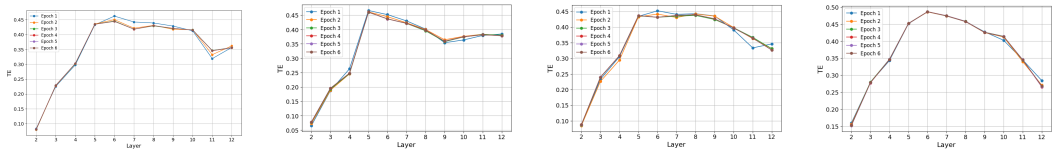
Takeaway. The learning rate strongly governs information dynamics: an excessively large α drives premature convergence and representational collapse (low TEs), stalling learning and harming generalization; a moderate α sustains information flow in intermediate layers and supports gradual, robust improvement. We therefore adopt $\alpha = 2 \times 10^{-5}$ for the remainder of our experiments.

H.5 Information Flow in T5-Base with Application to Layer Pruning

In addition to RoBERTa$_{\text{base}}$, we also study the Text-to-Text Transfer Transformer (T5) architecture, as presented by Raffel *et al.* [73], which unifies a broad spectrum of NLP tasks by expressing each as a text-to-text problem. T5 is available in multiple configurations, including T5-Small with roughly 60M parameters and T5-Base with about 220M parameters. For our investigation into TE, we focus on T5-Base.

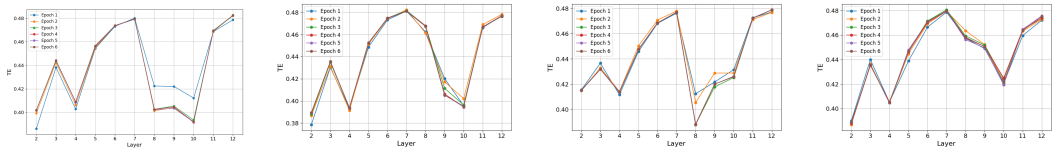
T5-Base is a transformer-based encoder–decoder architecture comprising 12 encoder layers and 12 decoder layers. Each encoder layer contains a multi-head self-attention sublayer and a position-wise feed-forward network, with residual connections and layer normalization. Decoder layers mirror this structure but include an additional multi-head cross-attention sublayer that attends to the encoder's final hidden states, enabling conditioning on the input representations. Concretely, each decoder layer has three attention modules: masked self-attention (to prevent attending to future tokens), cross-attention (encoder–decoder interaction), and a feed-forward sublayer. In T5-Base, all attention modules employ 12 heads with model dimension 768, enabling complex sequence transduction and flexible conditioning.

We evaluate using four established benchmarks—e-SNLI [9], ANLI [66], CommonsenseQA (CQA) [84], and SVAMP [67]—chosen to span language understanding and reasoning scenarios. All samples are preprocessed with the `T5` tokenizer for consistent training/evaluation. Following [29], we refer to e-SNLI, CQA, ANLI, and SVAMP as CoT (Chain-of-Thought) benchmarks.



(a) (b) (c) (d)

Figure 18: **Decoder (T5-Base).** TE vs. layer index during 6-epoch full fine-tuning on four datasets: (a) e-SNLI, (b) CQA, (c) ANLI, (d) SVAMP.



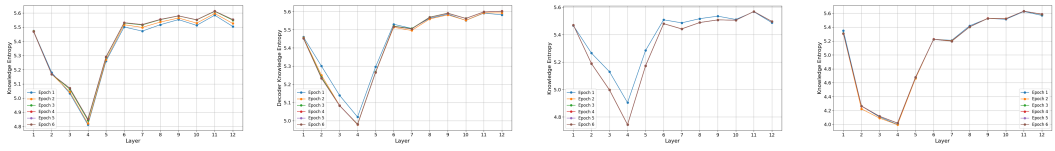
(a) (b) (c) (d)

Figure 19: **Encoder (T5-Base).** TE vs. layer index during 6-epoch full fine-tuning on four datasets: (a) e-SNLI, (b) CQA, (c) ANLI, (d) SVAMP.

We fine-tune T5-Base separately on each dataset. Figure 18 shows *TE* profiles across decoder layers over six epochs. The curves exhibit a characteristic $\cap$ shape, peaking at intermediate decoder layers across epochs. This indicates that the intermediate decoder layers are the locus of active information transformation and integration to facilitate output generation and loss minimization. In contrast, the final decoder layer's TE consistently decreases, reflecting the need for stable, deterministic outputs as decoding concludes. These observations highlight a functional specialization: intermediate layers contribute novel, task-relevant transformation, whereas the output layer consolidates representations into predictions.

Figure 19 shows TE as a function of encoder layer index. Unlike the simpler $\cap$-shaped or monotonic trends often seen in encoder-only models like RoBERTa-Base, T5-Base's encoder exhibits an oscillatory pattern (up $\rightarrow$ down $\rightarrow$ up $\rightarrow$ down $\rightarrow$ up). We attribute this non-monotonic behavior to encoder–decoder interaction: the presence of a decoder enables iterative information exchange and refinement across stages, repeatedly transforming and integrating representations in the encoder. This interplay fundamentally shapes layer-wise information dynamics in sequence-to-sequence models like T5-Base.

Figure 20 reports *knowledge entropy* (KE) across decoder layers after six epochs on each dataset. The KE profiles typically decrease initially, rise through intermediate layers, and then plateau (down $\rightarrow$ up $\rightarrow$ flat).

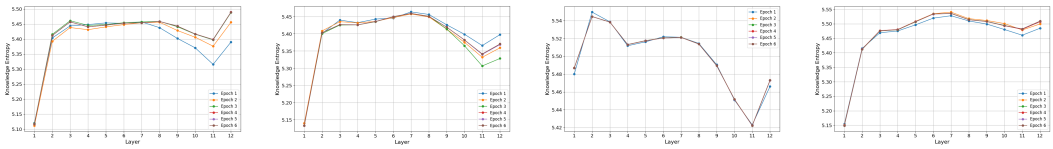


(a) (b) (c) (d)

Figure 20: **Decoder (T5-Base).** Knowledge entropy vs. layer index during 6-epoch full fine-tuning on four datasets: (a) e-SNLI, (b) CQA, (c) ANLI, (d) SVAMP.

Figure 21 shows KE across encoder layers. Notably, ANLI exhibits a distinct KE pattern relative to the other datasets, underscoring that KE depends on both architecture and data characteristics.

We next prune four of the twelve decoder layers in T5-Base using four strategies: TE–based pruning, KE-based pruning (KE-Pruning), random LayerDrop, SlimLLM [26], and a fine-tuning baseline without pruning. Table 9 compares accuracy on e-SNLI, CQA, ANLI, and SVAMP. Fine-tuning attains the highest accuracy on e-SNLI and CQA, while RS-Pruning performs best on ANLI and



(a) (b) (c) (d)

Figure 21: **Encoder (T5-Base).** Knowledge entropy vs. layer index during 6-epoch full fine-tuning on four datasets: (a) e-SNLI, (b) CQA, (c) ANLI, (d) SVAMP.

SVAMP, with a large margin on SVAMP (0.3400 vs. 0.2600 for fine-tuning). KE-Pruning yields intermediate results, and LayerDrop is lowest across datasets. These findings demonstrate the effectiveness of a principled, TE–driven pruning criterion over random dropout and show that targeted pruning can rival or surpass standard fine-tuning on challenging tasks.

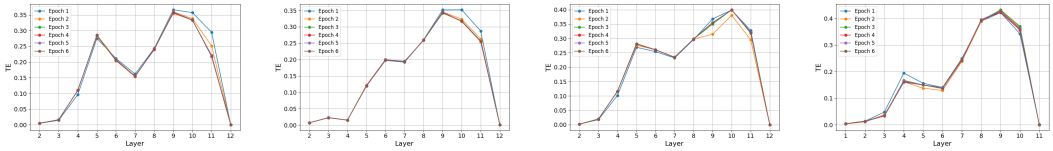
Table 9: T5-Base accuracy on four benchmarks.

Approach	e-SNLI	CQA	ANLI	SVAMP
Fine-Tuning	**0.5375**	**0.5807**	**0.3890**	0.2600
Our TE-Pruning	0.5350	0.5717	0.3690	**0.3400**
KE-Pruning	0.5310	0.5405	0.3180	0.3200
Layer Dropout	0.5250	0.4742	0.2900	0.3050
SlimLLM	0.5320	0.5667	0.3690	0.2700

H.6 Model Architecture vs Dataset: Impact on Information Flow

H.6.1 Dataset Impact on Information Flow

To systematically investigate the interplay between model architecture and data domain in shaping information dynamics, we analyze TE using two widely adopted pretrained language models: $\text{RoBERTa}_{\text{base}}$ and T5-Base. For each model, we fine-tune on a comprehensive suite of datasets representative of standard evaluation benchmarks. Specifically, $\text{RoBERTa}_{\text{base}}$ is evaluated on the GLUE benchmark, while T5-Base is tested on a diverse set of chain-of-thought (CoT) reasoning datasets. A natural question follows: Are the observed TE profiles an intrinsic property of the model architecture, or do they depend critically on the choice of dataset? In other words, if we exchange the datasets—fine-tuning $\text{RoBERTa}_{\text{base}}$ on CoT tasks and T5-Base on GLUE—will the resulting TE trajectories remain similar, or will they exhibit fundamentally different patterns?



(a) (b) (c) (d)

Figure 22: **$\text{RoBERTa}_{\text{base}}$ (encoder).** TE vs. layer index during 6-epoch full fine-tuning on four CoT datasets: (a) e-SNLI, (b) CQA, (c) ANLI, (d) SVAMP.

To answer this, we perform experiments for both models under swapped dataset conditions. For each model–dataset pair, we compute the layer-wise TE across all epochs of fine-tuning, thereby capturing the evolution of information dynamics through the model's hierarchy. The results for $\text{RoBERTa}_{\text{base}}$ on four CoT datasets are shown in Figure 22, while Figures 23 and 24 present the TE profiles for the T5-Base *encoder* and *decoder*, respectively, across the eight GLUE datasets. By comparing these figures, we disentangle the relative contributions of model structure and dataset complexity to the formation of layer-wise information dynamics. This analysis provides insights into the generality and dataset-sensitivity of the TE as a diagnostic for neural information processing in pretrained language models.

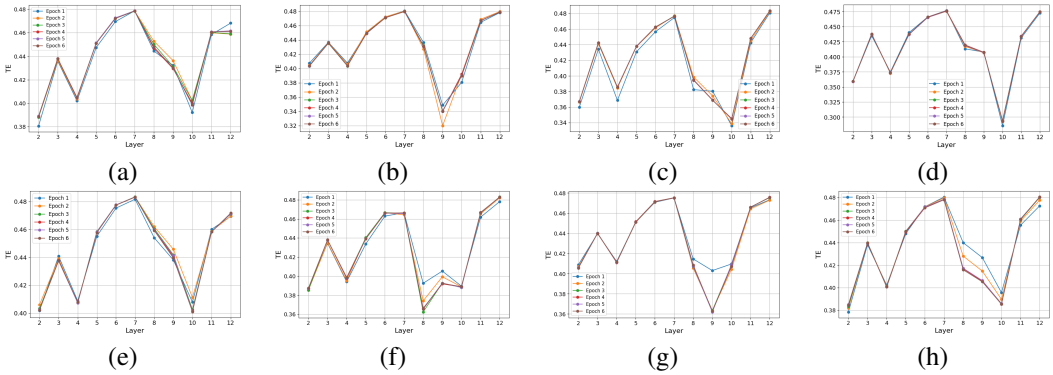


Figure 23: **T5-Base (encoder).** TE vs. layer index during 6-epoch full fine-tuning on the GLUE benchmark: (a) MNLI, (b) MRPC, (c) SST-2, (d) CoLA, (e) QNLI, (f) QQP, (g) RTE, (h) STS-B.

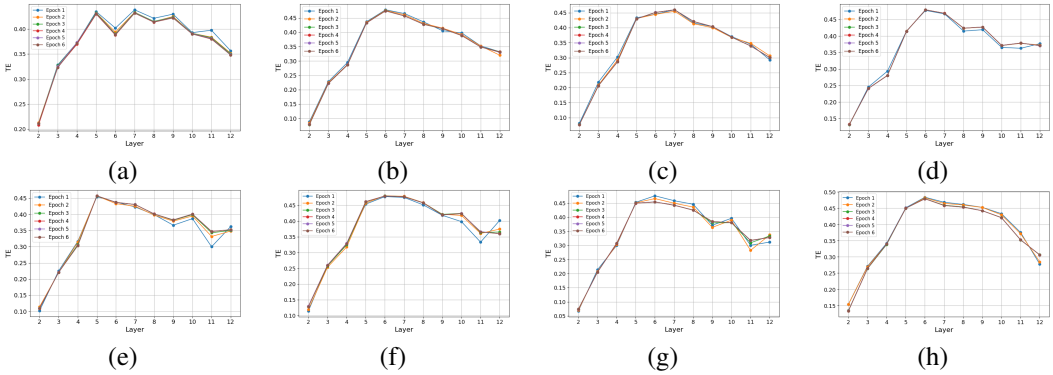


Figure 24: **T5-Base (decoder).** TE vs. layer index during 6-epoch full fine-tuning on the GLUE benchmark: (a) MNLI, (b) MRPC, (c) SST-2, (d) CoLA, (e) QNLI, (f) QQP, (g) RTE, (h) STS-B.

In Figure 25, we show the mean and standard deviation of the TE across the 12 encoder layers of $\text{RoBERTa}_{\text{base}}$, aggregated over 12 datasets and 6 epochs of fine-tuning. The mean profile is strongly non-monotonic: it starts low in the initial layers, rises to a broad peak spanning mid-to-upper layers (layers 4–9), then drops sharply at the output layer. Notably, the highest TEs occur at layers 7–9 (mean ≈ 0.24–0.29), indicating these layers sustain the most dynamic representational changes during adaptation. The standard deviation mirrors this pattern, with peak variability in the same layers, suggesting substantial cross-dataset and temporal diversity in information flow at these depths. In contrast, the input and output layers consistently exhibit low scores and minimal variability, reflecting more stable processing roles. These findings highlight the mid-to-upper layers of $\text{RoBERTa}_{\text{base}}$ as the most dynamic and dataset-sensitive during fine-tuning, underscoring their critical role in task adaptation, while the extremal layers serve as stable feature encoders/decoders.

In Figure 26a, we present the mean and standard deviation of the TE in the T5-Base encoder over 12 layers, aggregated from six epochs across 12 datasets. The profile is notably high and stable: the mean begins near 0.78 in the input layer, quickly rises above 0.94 in the middle layers, and peaks near 0.96 at layers 5–6. This elevated plateau reflects sustained representational activity and information flow, indicating strong capacity for flexible representation and complex feature extraction. The standard deviation remains very low (below 0.03) through the lower/middle layers, demonstrating consistency across datasets, then increases in the upper layers (especially layers 7–9, std ≈ 0.06–0.07), suggesting increased sensitivity to dataset-specific adaptation. Unlike models with sharp mid-depth peaks, T5-Base shows a flatter curve, with only minor decreases in the final layers—evidence that representational flexibility is distributed more evenly across the encoder while the upper layers retain greater adaptive capacity.

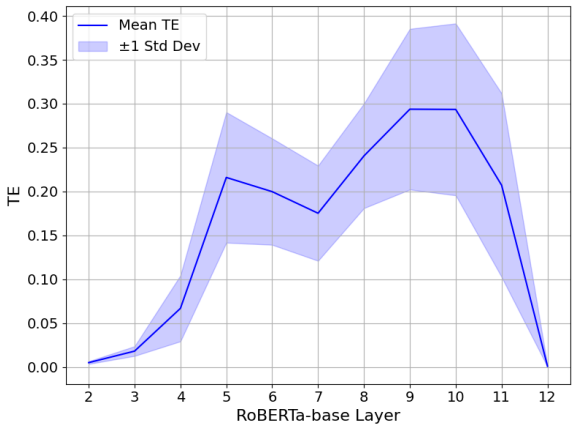
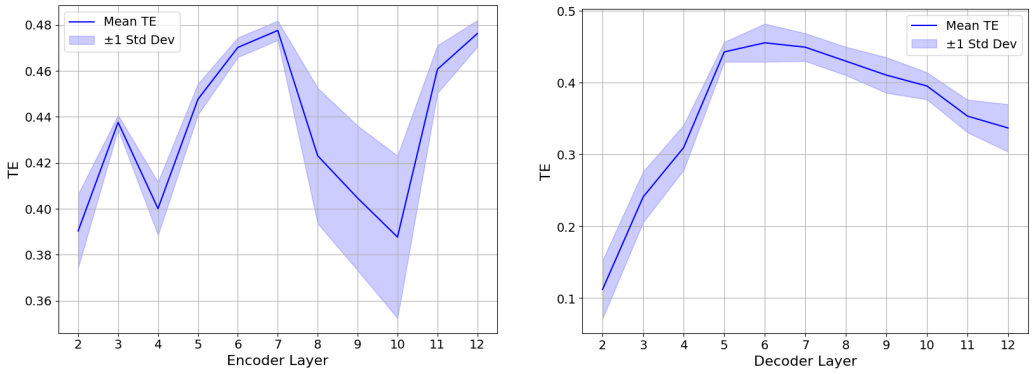


Figure 25: **RoBERTa$_{\text{base}}$ (encoder).** Mean and standard deviation of the TE across 12 layers with 6-epoch fine-tuning on 12 datasets.



(a) (b)

Figure 26: **T5-Base.** Mean and standard deviation of the TE with 6-epoch fine-tuning on 12 datasets. (a) Encoder, (b) Decoder.

In Figure 26b, we show the mean and standard deviation of the TE in the T5-Base decoder, again aggregated over six epochs and twelve datasets. The decoder exhibits a distinct dynamic: the mean starts low (≈ 0.22), rises rapidly through lower/middle layers to a peak of ≈ 0.91 at layer 5, then gradually declines to about 0.67 in the final layer. This pattern suggests that the early and middle decoder layers introduce and maintain the richest representational transformations, whereas upper layers consolidate and refine representations for task-specific outputs. The standard deviation is largest in the initial layer (≈ 0.08), reflecting high cross-dataset variability when forming initial representations, then decreases and stabilizes in the middle layers before rising again toward the output. Altogether, this TE profile underscores the specialization of decoder segments for transformation and output integration during sequence generation.

H.6.2 Interpretation of Mean and Standard Deviation of TE

Analyzing TE profiles across layers provides insight into how architectural design and dataset-specific adaptation shape information dynamics in LLMs. We interpret the *mean* TE at layer ℓ as reflecting the architecture's intrinsic propensity for representational *reuse or duplication* at that depth (e.g., driven by depth, attention topology, and residual connectivity). Because the mean is aggregated over datasets and epochs, it captures a stable, architecture-driven pattern of where the model tends to compress, relay, or re-express information versus where it tends to introduce new transformations. In this view, *low* mean TE suggests layers that contribute more novel, task-relevant change, whereas *high* mean TE indicates layers whose outputs are more predictable from neighboring layers.

In contrast, the *std* of the TE quantifies how strongly a layer's behavior is modulated by dataset-specific factors. High std marks *adaptation hotspots* whose representational redundancy varies with domain shift, label structure, or semantic variability; low std of TE indicates a more data-agnostic regime where behavior is governed primarily by architectural priors rather than input diversity. Thus, std of TE highlights the layers most responsive to fine-tuning, where the model flexibly adjusts redundancy in service of task demands.

Taken together, this dual perspective—*mean* as an architectural fingerprint and *std* as dataset-induced plasticity—enables a nuanced dissection of functional roles across depth. Practically, layers with *high mean* and *low std* TE are promising candidates for structured pruning or parameter sharing, while layers with *low mean* and/or *high std* warrant preservation or targeted adaptation (e.g., adapters/LoRA), informing both analysis and design of efficient LLMs.

H.7 Information Flow in Larger LLMs

We evaluate two open-weight decoder-only Transformer LLMs, Llama-3.2-3B and Qwen-2.5-7B. Llama-3.2-3B [21] is a 3B-parameter autoregressive model with 28 Transformer decoder layers, designed for efficient reasoning and long-context inference. Qwen-2.5-7B [70] is a larger 7B-parameter decoder-only model comprising 28 layers, trained on a diverse multilingual and code-rich corpus to provide strong general-purpose performance.

We evaluate Llama-3.2-3B on the GLUE benchmark (8 datasets) [90] using a two-stage procedure that combines LoRA fine-tuning with TE-based layer analysis. Each dataset is tokenized with the Llama tokenizer and padded to a maximum sequence length of 256. The base model is initialized for sequence classification and augmented with rank-4 LoRA on all attention and MLP projection matrices. We fine-tune for two epochs using AdamW with linear warmup. To measure information flow across depth, we register lightweight forward hooks on consecutive decoder blocks and compute a cosine-similarity TE proxy that captures the alignment between successive activation changes across training batches. After each epoch, we obtain a 27-edge TE profile (Layers 2–28), indicating where the model concentrates task-specific computation.

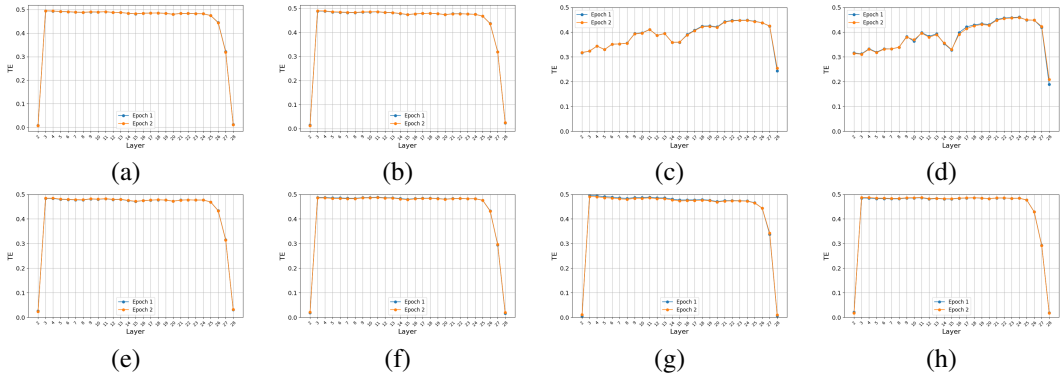


Figure 27: Information flow in **Llama-3.2-3B**. TE vs. layer index during 2-epoch LoRA (rank 4) fine-tuning on the GLUE benchmark: (a) MNLI, (b) MRPC, (c) SST-2, (d) CoLA, (e) QNLI, (f) QQP, (g) RTE, (h) STS-B.

For Llama-3.2-3B, the resulting TE curves (Fig. 27) exhibit a clear three-zone structure: TE is near zero in the first decoder block, rises rapidly to a plateau around ≈ 0.45 across most middle layers, and then drops sharply in the final layers. The close overlap between epochs and the small cross-task variance (Fig. 28) show that short LoRA fine-tuning preserves this pattern, with a stable band of mid-level layers carrying the bulk of task-dependent computation while boundary layers behave more like input/output adapters.

We apply the same LoRA fine-tuning and TE analysis pipeline to Qwen-2.5-7B across all GLUE tasks. The model is trained for two epochs with rank-4 LoRA using the Qwen tokenizer and identical optimization settings. After each epoch, we compute layer-wise TE using the same hook-based cosine-similarity metric, yielding a 27-edge TE profile (Layers 2–28) that reveals how Qwen-2.5-7B distributes computation during adaptation.

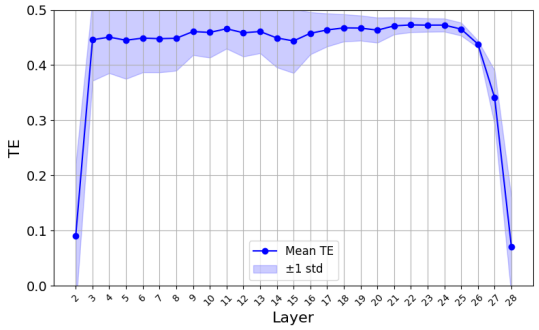


Figure 28: Statistics of information flow in **Llama-3.2-3B.** Mean and standard deviation of the TE across 28 layers with 2-epoch LoRA (rank 4) fine-tuning on 8 datasets in GLUE benchmark.

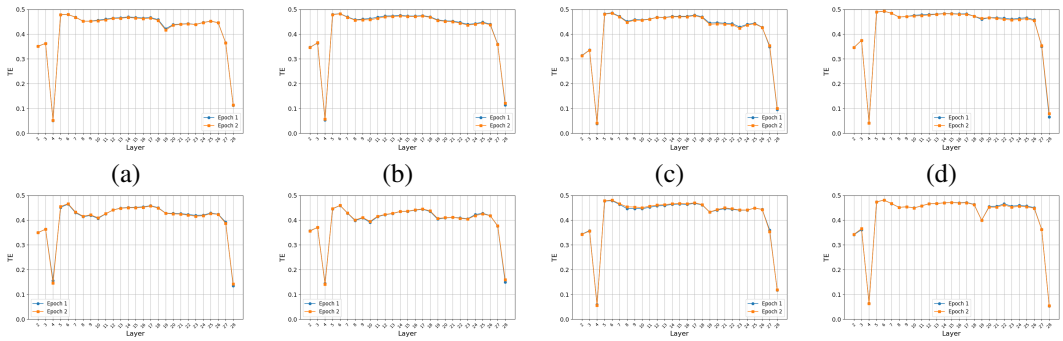


Figure 29: Information flow in **Qwen-2.5-7B.** TE vs. layer index during 2-epoch LoRA (rank 4) fine-tuning on the GLUE benchmark: (a) MNLI, (b) MRPC, (c) SST-2, (d) CoLA, (e) QNLI, (f) QQP, (g) RTE, (h) STS-B.

For Qwen-2.5-7B, the TE curves (Fig. 29) follow a broadly similar shape but with richer internal structure. TE rises steeply in early layers, exhibits a pronounced dip from Layer 3 to Layer 4 across all tasks, then recovers to a stable high plateau (≈ 0.40–0.48) before decreasing near the model top. The mean and variance summary (Fig. 30) shows that this early dip and top-layer decline are the main sources of cross-task variability, while the mid-depth plateau forms a robust high-TE backbone that dominates computation during GLUE fine-tuning.

I Computing Infrastructure, Datasets, and Runtime

In LLMs, RoBERTa-base, T5-base, Llama-3.2-3B, and Qwen-2.5-7B are all pretrained models. For fine-tuning using GLUE and CoT datasets, we use a subset of each of the 12 datasets. Specifically, we randomly select up to 5,000 training examples per dataset (using all available data if the training set contains fewer than 5,000 samples). For evaluation, we randomly sample 1,000 examples from each test set.

For VLMs, both CLIP ViT-B/16 and LLaVA-7B are pre-trained. MSCOCO-2014 [52] was used for both models. The MSCOCO-2014 dataset consists of 82,783 training images and 40,504 validation images. The dataset is organized into two image folders and accompanying annotation files, which provide the class labels and bounding box locations for objects contained in each image. In addition to object detection and segmentation, each image is paired with five human-written captions, making MSCOCO a widely used benchmark for image–text tasks such as retrieval and captioning. In the LLaVA-7B experiment, Flickr8k was also used. Flickr8k [62] is a widely used benchmark for vision-language modeling and captioning. It contains 8,000 natural images collected from Flickr, each annotated with five human-written captions that describe salient objects, actions, and scene

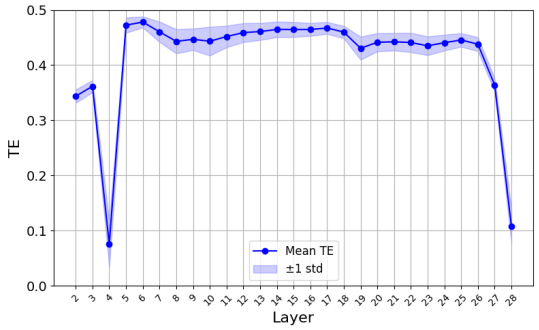


Figure 30: Statistics of information flow in **Qwen-2.5-7B.** Mean and standard deviation of the TE across 28 layers with 2-epoch LoRA (rank 4) fine-tuning on 8 datasets in GLUE benchmark.

context. Despite its modest size, Flickr8k captures diverse real-world scenarios and provides a clean testbed for analyzing image–text alignment, cross-modal information flow, and multimodal fusion dynamics in VLMs.

Table 10 summarizes the runtimes of key experiments conducted on a Colab A100 GPU. These results highlight that both the choice of model and the specific method have a significant impact on runtime.

J Limitations

Although transfer entropy provides a principled and interpretable measure of directed information flow, our approach has several limitations. Conceptually, exact TE requires estimating conditional densities over very large hidden states, which is infeasible for modern LMs and VLMs. Our approach therefore relies on (i) a local linear-Gaussian approximation via Jacobians (Theorem 4.1) and (ii) a further rank-one approximation that leads to the cosine-similarity estimator (Theorem 4.3). These approximations introduce several potential sources of bias: for example, finite-batch effects and local nonlinearity can lead to noisy or slightly inflated TE values when random fluctuations align across layers, while the rank-one assumption can underestimate TE when information propagates through multiple independent directions.

Practically, we mitigate these issues by (a) averaging the estimator over many mini-batches and training steps, (b) working with pooled representations (rather than individual tokens) to reduce variance, and (c) focusing on relative patterns (shape of TE vs. depth, changes across epochs, and comparisons across architectures) rather than interpreting absolute TE values as exact information in bits. Accordingly, we interpret our TE curves primarily as stable, qualitative probes of layer-to-layer information flow rather than precise, fully nonparametric TE estimates. Appendix F compares the cosine-based approximation with exact TE in a controlled low-dimensional setting.

K Societal Impacts and Mitigation Strategies

This work studies transfer entropy (TE) as a diagnostic and compression tool for LLMs and VLMs. Its main positive impact is improved transparency and efficiency: TE can help identify which layers, heads, or cross-modal pathways contribute directed information flow, and TE-guided pruning may reduce inference cost, energy use, and deployment barriers.

The main risks come from misuse and over-interpretation. More efficient models can lower the cost of both beneficial and harmful deployments, including spam, misinformation, or surveillance. In addition, TE scores are based on practical approximations and should not be treated as definitive causal explanations. A low-TE component may still matter for rare, safety-critical, or out-of-distribution inputs. To mitigate these risks, TE-based pruning should be paired with standard safety evaluations, red-teaming, robustness and fairness checks, and conservative post-pruning validation.

Table 10: Runtime details for key experiments.

Experiments	Runtime
RoBERTa-base using GLUE datasets	2h 50m
RoBERTa-base using CoT datasets	1h 30m
T5-base using GLUE Datasets	3h 20m
T5-base using CoT datasets	1h 50m
KE for RoBERTa using GLUE datasets	2h 35m
KE for T5 using CoT datasets	1h 20m
LayerDropout for RoBERTa using GLUE	1h 50m
LayerDropout for T5-base using CoT	1h 10m
SlimLLM for RoBERTa using GLUE datasets	2h 5m
SlimLLM for T5 using CoT datasets	1h 12m
RoBERTa-base FT with $\alpha = 0.02$	2h 50m
Llama-3.2-3B information flow using GLUE	3h 25m
Qwen-2.5-7B information flow using GLUE	4h 35m
LLaVA-1.5-7B using MSCOCO	5h 20m
LLaVA-1.5-7B using Flick8k	3h 10m
CLIP ViT-B/16: TE Layer Pruning	6h 20m
CLIP ViT-B/16: Streamline Layer Pruning	6h 20m
CLIP ViT-B/16: Short-LVLM Layer Pruning	6h 35m
CLIP ViT-B/16: Attention head classification	3h 35m
LLaVA-1.5-7B using VQAv2 TE (no pruning)	11h 30m
LLaVA-1.5-7B using VQAv2 TE (2 layers pruned)	11h 05m
LLaVA-1.5-7B using VQAv2 TE (4 layers pruned)	10h 41m
LLaVA-1.5-7B using VQAv2 TE (6 layers pruned)	10h 16m
LLaVA-1.5-7B using VQAv2 TE (8 layers pruned)	9h 51m
LLaVA-1.5-7B using VQAv2 Streamline (no pruning)	11h 30m
LLaVA-1.5-7B using VQAv2 Streamline (2 layers pruned)	11h 05m
LLaVA-1.5-7B using VQAv2 Streamline (4 layers pruned)	10h 41m
LLaVA-1.5-7B using VQAv2 Streamline (6 layers pruned)	10h 16m
LLaVA-1.5-7B using VQAv2 Streamline (8 layers pruned)	9h 51m
LLaVA-1.5-7B using VQAv2 Short-LVLM (no pruning)	11h 57m
LLaVA-1.5-7B using VQAv2 Short-LVLM (2 layers pruned)	11h 31m
LLaVA-1.5-7B using VQAv2 Short-LVLM (4 layers pruned)	11h 06m
LLaVA-1.5-7B using VQAv2 Short-LVLM (6 layers pruned)	10h 40m
LLaVA-1.5-7B using VQAv2 Short-LVLM (8 layers pruned)	10h 14m
CLIP ViT-B/16 using MSCOCO TE (12.5% token pruning)	5h 10m
CLIP ViT-B/16 using MSCOCO TE (25% token pruning)	4h 5m
CLIP ViT-B/16 using MSCOCO TE (37.5% token pruning)	3h 10m
CLIP ViT-B/16 using MSCOCO TE (50% token pruning)	2h 30m
LLaVA-1.5-7B using VQAv2 (20% token pruning)	8h 40m
LLaVA-1.5-7B using VQAv2 (30% token pruning)	7h 20m
LLaVA-1.5-7B using VQAv2 (40% token pruning)	6h 10m
LLaVA-1.5-7B using VQAv2 (50% token pruning)	5h 15m
LLaVA-1.5-7B using VQAv2 (60% token pruning)	4h 30m

Finally, computing TE may require access to model activations or attention patterns. When applied to sensitive data, practitioners should minimize logging, aggregate statistics when possible, discard intermediate activations after use, and restrict access to diagnostic traces. Overall, TE can support more interpretable and efficient models, but it should be deployed with careful validation, privacy protection, and safety monitoring.

L Use of Large Language Models

During the preparation of this paper, we used ChatGPT 5 to assist with writing and experimentation. On the writing side, it helped with grammar and spelling checks, and polished several paragraphs. For experiments, it helped diagnose and fix bugs in our Python implementation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: It's at the end of the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The assumptions and proof are provided in the main paper and Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The details of the experiment settings are provided in the main paper and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code of this paper is provided in the supplementary material. No new dataset from this paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The training and test details are provided in the main paper and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Many experiments on information flow are performed on multiple datasets and epochs, their uncertainties are plotted with mean and std.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The information is provided at the end of Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The NeurIPS code of Ethics is strictly conformed.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: It's included at the end of Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: It's at the end of the Appendix.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and models are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Code is provided by this paper.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No human subjects is used in the experiment.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subject is used.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: It's provided at the end of the Appendix.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.