# MULTIFLOW: Shifting Towards Task-Agnostic Vision-Language Pruning

Matteo Farina[1] Massimiliano Mancini[1] Elia Cunegatti[1]
Gaowen Liu[2] Giovanni Iacca[1] Elisa Ricci[1,3]
[1]University of Trento [2]Cisco Research [3]Fondazione Bruno Kessler

## Abstract

*While excellent in transfer learning, Vision-Language models (VLMs) come with high computational costs due to their large number of parameters. To address this issue, removing parameters via model pruning is a viable solution. However, existing techniques for VLMs are task-specific, and thus require pruning the network from scratch for each new task of interest. In this work, we explore a new direction: Task-Agnostic Vision-Language Pruning (TA-VLP). Given a pretrained VLM, the goal is to find a unique pruned counterpart transferable to multiple unknown downstream tasks. In this challenging setting, the transferable representations already encoded in the pretrained model are a key aspect to preserve. Thus, we propose Multimodal Flow Pruning (*MULTIFLOW*), a first, gradient-free, pruning framework for TA-VLP where: (i) the importance of a parameter is expressed in terms of its magnitude and its information flow, by incorporating the saliency of the neurons it connects; and (ii) pruning is driven by the emergent (multimodal) distribution of the VLM parameters after pretraining. We benchmark eight state-of-the-art pruning algorithms in the context of TA-VLP, experimenting with two VLMs, three vision-language tasks, and three pruning ratios. Our experimental results show that* MULTIFLOW *outperforms recent sophisticated, combinatorial competitors in the vast majority of the cases, paving the way towards addressing TA-VLP. The code is publicly available at* https://github.com/FarinaMatteo/multiflow.

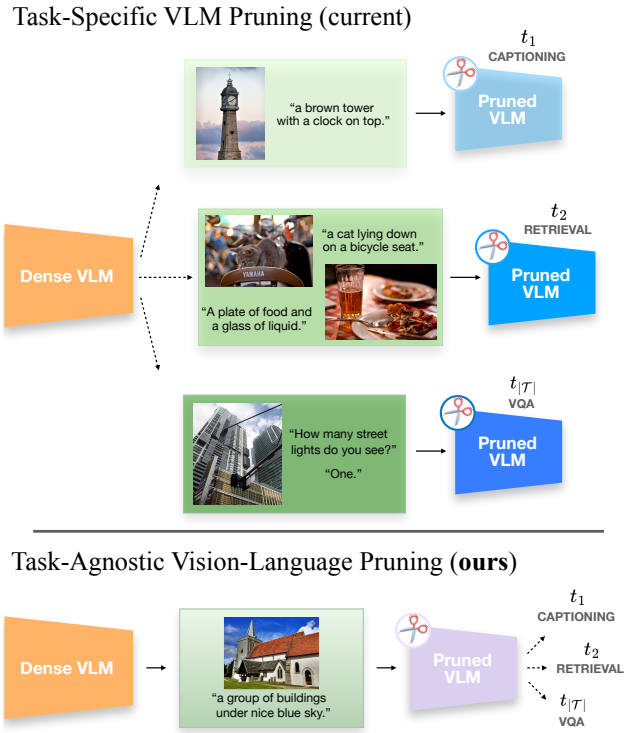


Figure 1. The conceptual difference between existing VLM pruning methods [58, 65] and our proposed Task-Agnostic Vision-Language Pruning. While existing pruning methods use task-specific knowledge, hence requiring pruning the dense model from scratch for different tasks, we propose to shift the perspective and formalize TA-VLP, which only requires pruning once.

## 1. Introduction

Large-scale vision-language models (VLMs) [38–40, 53] show remarkable transfer learning capabilities and achieve state-of-the-art results in multiple vision-language tasks after fine-tuning with task-specific data and little architectural changes. However, these practical advantages come at the price of a huge number of parameters, *e.g.*, in the order of hundreds of millions [39], hindering deployment in memory-constrained devices. A solution to this problem is to reduce the network size via pruning [23], a compression technique whose aim is to explicitly remove network parameters. In the context of VLMs, existing approaches perform pruning for specific downstream tasks [58, 65], where the obtained pruned models achieve good transfer learning performance once fine-tuned. Critically, this requires re-pruning the model from scratch if the downstream task changes. This is not only time-consuming, but fur-

Corresponding author: m.farina@unitn.it.

ther assumes that the original VLM parameters can easily be stored. To overcome these issues, we tackle the problem from a different perspective, investigating the possibility of pruning a VLM while maintaining its general transfer learning capabilities. Towards this goal, we propose Task-Agnostic Vision-Language Pruning (TA-VLP), where the aim is to prune a VLM *once* and obtain a sparse model transferable to multiple *unknown* tasks when fine-tuned (see Fig. 1). While appealing, finding an optimal solution to TA-VLP is challenging as we cannot use any task-specific priors nor feedback during pruning, and different downstream tasks may focus on different visual/linguistic cues (*e.g.*, local for visual question answering, global for captioning). Nevertheless, we can still rely on one anchor: the pretrained VLM. In fact, pretraining uses a generic objective, such as vision-language alignment, which applied to large-scale data enables learning generic and transferable representations. These representations depend on the network parameters and on how the (multimodal) activations propagate through the network. Intuitively, if we assume the pretrained model to be transferable, its pruned counterpart should preserve the learned activation patterns.

Following this principle, we propose Multimodal Flow Pruning (MULTIFLOW), a first method for TA-VLP. MULTIFLOW models each layer as a bipartite graph, where nodes are activations and edges are parameters. Exploiting calibration data, the saliency of a parameter is modeled by combining its magnitude with the average signal emitted/aggregated by the input/output nodes it connects. However, directly pruning using these scores may lead to biases w.r.t. the depth of a layer, and ignores that activation patterns and magnitudes may differ among modalities. To overcome this issue, we disentangle modalities and guide pruning with the emergent distribution of the magnitude of the parameters. Our experiments on XVLM [71] and BLIP [39], with three vision-language tasks and pruning ratios, show that MULTIFLOW consistently matches or surpasses existing methods while requiring no gradient information.

**Contributions.** To summarize, our contributions are:

- We formalize Task-Agnostic Vision-Language Pruning, whose aim is to prune a VLM *once* while maintaining transferability to *unknown* downstream tasks.
- We propose Multimodal Flow Pruning, a first specific method for TA-VLP, where the importance of a parameter depends on the aggregated importance of the nodes it connects and its magnitude, exploiting multimodal priors to guide the distribution of each layer and avoid biases.
- We benchmark existing methods and MULTIFLOW on TA-VLP, with multiple vision-language tasks, VLMs, and pruning ratios, demonstrating the effectiveness of MULTIFLOW. These results and the proposed benchmark also highlight a large gap w.r.t. the performance of the dense model, paving the way for future research on this topic.

## 2. Related Work

**Post-training pruning.** Several works remove parameters after training [16, 26–28, 35, 36], aiming to reduce inference time [47, 48] and storage requirements [21, 72]. While some of these techniques are data-free and mostly rely on weight magnitude, others are data-driven and exploit first- or second-order information. Another line of data-driven algorithms relies on combinatorial optimization [4, 59, 69] or iterative procedures, like iterative magnitude pruning (IMP) [18, 55], which alternates training until convergence, magnitude pruning, and weight rewinding. While we also exploit the weight magnitude in our pruning criterion, we additionally consider how information propagates through the target VLMs, fostering transferability to *unknown* tasks.

**Pruning at initialization.** Since our goal is to prune the model *before* fine-tuning it on downstream tasks, our work is also closely related to pruning at initialization (PaI). PaI methods rely on a *saliency* function that evaluates each connection, removing those with the lowest saliency scores to meet a target sparsity level [1, 8, 37]. Notably, the same saliency scores can be re-used for different sparsity targets without any additional computational overhead. The saliency function can also be applied iteratively, trading-off efficiency for performance [52, 61, 68]. PaI methods can be categorized based on their input as well, with data-dependent algorithms [1, 8, 37, 52] employing samples from the target dataset, and data-free techniques using synthetic inputs [61, 68]. Our work shares the same rationale of these techniques, *i.e.*, the saliency of network connections is central to our pruning strategy. However, while existing PaI methods mostly focus on task-specific objectives with backward gradient propagation, MULTIFLOW is gradient-free, as it is only based on the forward function of the model.

**Pruning in Vision and/or Language.** Several works explore pruning in the context of vision and language, with works sparsifying large pretrained models in NLP, *e.g.*, BERT [29, 33, 34, 70], GPT [20] and LLaMA [60], and vision transformers [17], *e.g.*, via model [29, 36, 67, 73] or token [5, 41, 45, 54, 62] pruning. To the best of our knowledge, UPop [58] and EfficientVLM [65] are the only pruning algorithms specifically developed and benchmarked on multiple vision-language tasks. The former progressively prunes the target VLM during fine-tuning, while the latter is a distill-then-prune framework. However, both methods require task-specific knowledge *by design*. In this work, we take instead a different direction, investigating how we can extract task-agnostic subnetworks from VLMs.

**Unimodal Task-Agnostic Pruning.** In the context of Vision- or Language-only, some works already explored the ability of pruned models to generalize to multiple tasks [66]. For instance, [10, 12] assess that the *train-prune-retrain* paradigm of IMP can also be successfully applied during

unimodal pretraining. In principle, a task-agnostic subnetwork can also emerge when pretraining by optimizing the pruning masks as trainable parameters (a common procedure in continual learning [44] or neural architecture search [9]), as shown in [43]. However, complete access to the pretraining phase is often out of reach due to its large computational demand. Exploratory works also study the effects of pruning on BERT transfer [24], highlighting that finetuning recovers dense performance when less than half of the parameters are pruned, or on self-supervised CNNs [6].

Our work has different rationales. Motivated by [23], which showcases the additional difficulty in finding task-agnostic pruned models with Vision and Language w.r.t. unimodal scenarios, we design the first algorithm for TA-VLP. We do not assume access to large amounts of data nor the pretraining phase, striving for *fast* and *efficient* pruning. Hence, we avoid the burden of both computing and storing gradients for large VLMs with a gradient-free algorithm. Neither IMP nor Mask Training meet these principles.

## 3. Task-Agnostic Vision-Language Pruning

In this section, we formally define the *Task-Agnostic Vision-Language Pruning* (TA-VLP) problem, discussing its challenges and its relation with prior work on pruning VLMs.

**Preliminaries.** Let $f$ denote a VLM, and let $\Theta \in \mathbb{R}^n$ be its corresponding parameters after pretraining on a large-scale dataset $\mathcal{D}_p$ of image-text pairs. Given data for a specific vision-language task $t$, we can fine-tune $\Theta$ to improve performance on $t$ itself. While the standard practice is to directly update $\Theta$, the latter is extremely high-dimensional and cannot always be stored. To circumvent this issue, pruning algorithms for VLMs [58] prune $\Theta$ explicitly for $t$. Formally, given task data $\mathcal{D}_t$, they aim at a binary mask $\mathbf{m}_t \in \mathbb{B}^n$, by maximizing a task-dependent criterion $\mathcal{C}_t$:

$$\mathbf{m}_t = \arg\max_{\mathbf{m}} \mathcal{C}_t\Big(f(\Theta \odot \mathbf{m}), \mathcal{D}_t\Big) \quad \text{s.t. } ||\mathbf{m}||_0 = k \tag{1}$$

where $k$ denotes the sparsity constraint (*i.e.*, the number of parameters to preserve). This mask should maximize the performance $\mathrm{p}_t$ of model $f$ on task $t$, when $f$ is trained using a given algorithm $\mathcal{A}_t$, *i.e.*, $\mathrm{p}_t(\mathcal{A}_t, f, \Theta \odot \mathbf{m}_t)$. However, with this setup one needs to re-prune the model from scratch for every new task, which requires both time and storage of the original $\Theta$. To overcome these issues, we shift the perspective towards task-agnostic model pruning.

**Task-agnostic VLM Pruning.** The goal of TA-VLP is to prune a VLM once while preserving trainability for arbitrary downstream tasks, without re-compressing the model from scratch. Formally, we aim for a task-agnostic mask $\mathbf{m}_a$ that maximizes the performance of $f$ on a series of *unknown* downstream tasks $\mathcal{T}$:

$$\sum_{t \in \mathcal{T}} \mathrm{p}_t(\mathcal{A}_t, f, \Theta \odot \mathbf{m}_a). \tag{2}$$

As it is unfeasible to collect data for *unknown* target tasks, a TA-VLP algorithm should produce $\mathbf{m}_a$ from a generic dataset $\mathcal{D}_g$ and a generic criterion $\mathcal{C}_g$, *i.e.*:

$$\mathbf{m}_a = \arg\max_{\mathbf{m}} \mathcal{C}_g\Big(f(\Theta \odot \mathbf{m}), \mathcal{D}_g\Big) \quad \text{s.t. } ||\mathbf{m}||_0 = k. \tag{3}$$

In this work, we always assume that $\mathcal{D}_g$ is much smaller than the pretraining dataset $\mathcal{D}_p$, *i.e.*, $|\mathcal{D}_g| << |\mathcal{D}_p|$, as the aim of TA-VLP should not be to re-train a smaller model from scratch, but rather to efficiently prune an existing VLM. TA-VLP entails several challenges, as different tasks may exhibit stronger sensitivity to visual, textual, or fused knowledge. Intuitively, the solution to TA-VLP requires finding the optimal trade-off among modalities and encoded knowledge, respecting the priors of the pretrained VLM. In the following, we describe how we tackle TA-VLP by considering the multimodal information flow within VLMs.

## 4. MULTIFLOW: Multimodal Flow Pruning

In this section we introduce ***Multi**modal **Flow** Pruning*, a first algorithm for (unstructured) Task-Agnostic Vision-Language Pruning. We first discuss how we model the information flow within the VLM, and then how we exploit (multimodal) pretraining priors when pruning.

### 4.1. Modeling the Information Flow

As we lack task priors, we have one anchor when performing TA-VLP: the pretrained VLM. In fact, if we assume that the pretrained VLM encodes transferable representations, preserving how these representations emerge should maintain also the transferability to downstream tasks. We exploit this principle and tackle the problem from the perspective of the information flowing through the network, framing network pruning through the lens of message passing.

Without loss of generality, let us focus on a linear layer. We can represent a dense linear projection as a directed, weighted and complete bipartite graph $G = (L \cup R, E)$, where $L$ and $R$ are disjoint sets of nodes and $E$ is the set of edges connecting them. Note that an edge $e_{lr} \in E$ connecting nodes $l \in L$ and $r \in R$ corresponds to a parameter $\theta_{lr} \in \Theta$. In this context, we model the importance of $\theta_{lr}$ as the information passing through its corresponding edge $e_{lr}$. This depends on three different values: (i) the weight of the edge, (ii) the saliency of the input node $l$, and (iii) the saliency of the output node $r$. In the following, we describe how we measure and combine these components.

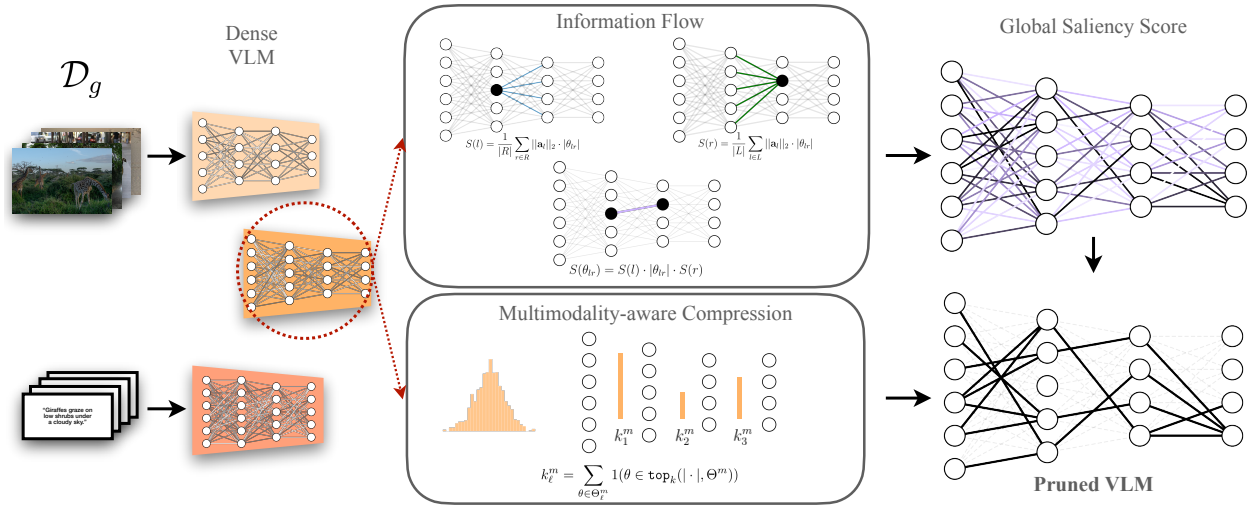


Figure 2. **MULTIFLOW**. Orange trapezoids represent groups of parameters processing different modalities (i) To compute the **information flow** score for a parameter $\theta_{lr}$, MULTIFLOW combines the importance of the input neuron $l$ and that of output neuron $r$, aggregating them via the local hop from $l$ to $r$ through $\theta_{lr}$ (ii) A **global saliency score** is obtained by computing (i) for all edges, and a **global modality-aware distribution** that exploits the emergent properties of large-scale pretraining guides layer-wise pruning.

**Importance of an edge.** This component is directly estimated from the absolute value of its corresponding weight, *i.e.*, $I(e_{lr}) = |\theta_{lr}|$, a simple and effective importance estimation technique well-known in the literature [18, 26]. Note that this measure inherently exploits pretraining information, as weight magnitudes can be viewed as long-term accumulators of movement in the parameter space as an emergent property from pretraining [56].

**Saliency of a node.** In each layer, input and output nodes have distinct roles: the former account for forwarding information, while the latter account for aggregating it. Given a vector $\mathbf{a}_l \in \mathbb{R}^N$ collecting $N$ activations to an input node $l$ of a given layer, we can anchor on the distinct roles of the neurons to assign ad-hoc saliency criteria. Following this intuition, we frame the saliency of $l \in L$ as the average strength of the signal it *emits* towards *all* output neurons:

$$S(l) = \frac{1}{|R|} \sum_{r \in R} ||\mathbf{a}_l||_2 \cdot |\theta_{lr}|. \tag{4}$$

On the other hand, we treat the saliency of an output node $r \in R$ as the average strength of the signals it *receives*:

$$S(r) = \frac{1}{|L|} \sum_{l \in L} ||\mathbf{a}_l||_2 \cdot |\theta_{lr}|. \tag{5}$$

Note that we use magnitude and norms to avoid potential sign misalignments between the two: those would create misleading importance scores as, *e.g.*, a negative weight with large magnitude may greatly influence the output and vice versa. To ensure that Eqs. (4) and (5) take into account both the modulation of the edges and the activation patterns grounded in the VLM forward process, we estimate the activation norms over the available calibration data $\mathcal{D}_g$, similarly to concurrent work on LLM pruning [60]. Notably, this only requires forwarding $\mathcal{D}_g$ through the model, which is much faster than also computing gradients.

**Final score.** Collating previous concepts, we define the saliency of each parameter $\theta_{lr}$ connecting nodes $l$ and $r$ via the edge $e_{lr}$, as the saliency of the path from $l$ to $r$:

$$S(\theta_{lr}) = S(l) \cdot I(e_{lr}) \cdot S(r) = S(l) \cdot |\theta_{lr}| \cdot S(r). \tag{6}$$

Note that Eq. (6) equally balances the contribution of each graph part, involving both the information captured by the edge weights and its relation to the most salient nodes.

## 4.2. Multimodality-aware compression

Properly defining a saliency criterion ultimately enables ranking the parameters of a model and, consequently, network pruning. Given a target sparsity constraint $k$, the most straightforward solution would be to instantiate a binary mask preserving the top-k parameters according to the scores in Eq. (6). However, this would ignore potential sources of bias. For instance, as deeper layers accumulate magnitude from preceding ones, this may cause a large discrepancy in the scores, with the risk that the pruning criterion penalizes early layers and induces layer collapse [61]. This also applies to the multimodal nature of the model: as different layers may receive inputs from different modalities, by assuming that information equally flows among them, we may overlook their respective distribution, hence biasing our pruning mask on one of them. We provide more insights on both these phenomena in Sec. 6.

To avoid biasing the model towards a specific modality and/or network level, we re-weight the importance of each parameter based on the prior distribution given by the pretrained VLM parameters. In fact, we found that the magnitude of the weights *detached* from the input/output information flow is a good indicator of the overall distribution that the pruned network should maintain layer-wise. This estimation tends to be more accurate if we keep into account which modality the layer processes, and its relevance increases with the disentanglement among modalities. Formally, given a layer $\ell$ processing information from modality $m$ (*e.g.*, visual or textual), with its corresponding parameters $\Theta_\ell^m$, we define its active parameter count $k_\ell^m$ as:

$$k_\ell^m = \sum_{\theta \in \Theta_\ell^m} \mathbb{1}(\theta \in \texttt{top}_k(|\cdot|, \Theta^m)) \tag{7}$$

where $\mathbb{1}$ is the indicator function and $\Theta^m \subset \Theta$ is the subset of parameters processing a specific modality $m$ as input. $\texttt{top}_k(|\cdot|, \Theta^m)$ is the set of top-k elements in $\Theta^m$ if we rank elements according to their magnitude. Given this prior distribution $k_\ell^m$, the final mask for the layer is:

$$\mathbf{m}_{lr}^{\ell,m} = \begin{cases} 1 \text{ if } \theta_{lr} \in \texttt{top}_{k_\ell^m}(S, \Theta_\ell^m) \\ 0 \text{ otherwise} \end{cases} \tag{8}$$

using $S$ as criterion for the $\texttt{top}_k$. With Eq. (8), the compressed network will take into account: (i) the information flow (as estimated via Eq. (6)), (ii) the bias of the information at different levels of the network, and (iii) the peculiar flow of each modality (Eq. (7)). All this information, combined, allows the pruned model to maintain priors of the original model parameters and preserve core connections. The overall procedure of MULTIFLOW is depicted by Fig. 2.

## 5. Experiments

In this section, we benchmark well-established and recent pruning algorithms in TA-VLP. We experiment with three different downstream tasks: Image-Text Retrieval, Image-Captioning, and Visual Question Answering [13, 25, 42]. We additionally report Vision-only experiments in Sec. A.2.

**Architectures.** To study how distinct model designs impact TA-VLP, we experiment with two significantly different VLMs: $\text{BLIP}_{\text{BASE}}$ [39], which uses a multimodal mixture of encoder-decoder networks, and $\text{XVLM}_{\text{CLIP}}$[71], where a vision-encoder and a text-encoder process information in parallel streams, with a final fusion encoder merging their output. While BLIP's design is unique, many modern VLMs share the principles of modality separation (*e.g.*, [53]) and fusion (*e.g.*, [38, 40]) at the core of XVLM. For both, their vision encoders use 16×16-sized patches.

**Baselines.** We compare to classical and recent approaches applicable to TA-VLP, divided between data-free and data-driven methods. For data-free baselines, we include **One-shot Magnitude Pruning** (**OMP**, [26]), shown to outperform several state-of-the-art algorithms after training [19] and comparable to IMP for pretrained transformers [29]. We then include **LAMP** [36], which extends OMP by unifying layer-wise calibration and global pruning. For data-driven methods, we choose two PaIs and two post-training pruning methods. Among PaIs, we select **SNIP** [37], which retains connections based on their estimated impact on the loss function, and **ITERSNIP** [52] which gradually prunes by applying SNIP iteratively. For post-training pruning methods, we test **CHITA**, a recent state-of-the-art algorithm that relies on a low-rank decomposition of the Hessian matrix of the loss function [4], together with its iterative variant **CHITA++** [4]. Additional baselines can be found in Sec. A.

**Experimental setup.** We test all pruning methods at 63% and 75% global sparsity for all tasks, choosing them to exceed trivial sparsities (*i.e.*, $\leq$ 50% [24]) and test in the neighborhood of the *essential sparsity* (*i.e.*, the limit after which the performance drop always overcomes the sparsity gain [29], $\sim$ 70%). We study the extreme 90% sparsity in the next section. Data-dependent methods use the same set of calibration data on a *per-run* basis. To ensure no task-specific data is used for pruning, we construct $\mathcal{D}_g$ from CC3M [57] and VisualGenome [31], and discard image-question pairs from the latter since they would collide with a portion of the data used for VQA finetuning. We sample $B = 3000$ batches from both datasets with a batch size of $b = 32$, totaling around $\sim$ 5% of the standard 4M pretraining set for VLMs ($\mathcal{D}_p$) [50]. In this way, a critical requirement of TA-VLP is satisfied, *i.e.*, $|\mathcal{D}_g| << |\mathcal{D}_p|$.

For first- and second-order methods, we use general-purpose pretraining losses, as defined in the original papers, excluding objectives that require fine-grained annotations not always available (*i.e.*, visual grounding in XVLM). Note that these methods already have some form of task prior and are, thus, expected to outperform the others: for both XVLM and BLIP, the pretraining loss contains either all or a subset of the finetuning losses. For all downstream tasks, we fine-tune the pruned models with the same setup of the original papers and average the results over 3 runs with different seeds. We report additional details in the Appendix.

### 5.1. Image-Text Retrieval (ITR)

**Setup.** We evaluate all methods and architectures for Image-Text Retrieval (ITR) on MSCOCO [30], analyzing both Text-to-Image retrieval, where the model should pick a matching image from a pool of target ones given a textual query, and the specular task of Image-to-Text retrieval.

**Results.** Results are summarized in Tab. 1, where we report the established Image-Recall (IR) and Text-Recall (TR) metrics at different levels. MULTIFLOW outperforms prior methods at all image-grounded and text-grounded metrics,

| Method | Sparsity | BLIP$_{\text{BASE}}$ | | | | XVLM$_{\text{CLIP}}$ | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Image-to-Text [%] | | Text-to-Image [%] | | Image-to-Text [%] | | Text-to-Image [%] | |
| | | R@1 | R@5 | R@1 | R@5 | R@1 | R@5 | R@1 | R@5 |
| DENSE | 0% | 80.72 | 95.08 | 62.99 | 85.27 | 78.18 | 94.48 | 60.94 | 84.41 |
| RANDOM | | 0.02 | 0.12 | 0.02 | 0.10 | 21.48 | 46.66 | 14.89 | 37.61 |
| SNIP | | 68.06±0.36 | 89.63±0.06 | 51.85±0.12 | 78.61±0.03 | 70.19±0.15 | 91.27±0.17 | 53.48±0.11 | 80.22±0.04 |
| ITERSNIP* | | 0.03±0.01 | 0.14±0.04 | 0.02±0.00 | 0.10±0.01 | 67.12±1.33 | 89.18±0.94 | 50.78±0.89 | 78.63±0.60 |
| OMP | 63% | 75.39±0.24 | 92.95±0.18 | 58.71±0.22 | 82.82±0.08 | 76.02±0.64 | 93.35±0.20 | 58.96±0.02 | 83.26±0.12 |
| LAMP | | 70.36±0.12 | 90.49±0.20 | 53.83±0.21 | 79.62±0.16 | 75.32±0.21 | 93.27±0.13 | 58.38±0.31 | 83.17±0.09 |
| CHITA | | 74.36±0.13 | 92.06±0.25 | 57.44±0.16 | 82.20±0.17 | 76.05±0.12 | 93.69±0.04 | 58.98±0.08 | 83.39±0.02 |
| CHITA++ | | 75.00±0.29 | 92.59±0.14 | 58.01±0.09 | 82.29±0.19 | 76.59±0.20 | 93.70±0.27 | 59.31±0.11 | 83.34±0.05 |
| MULTIFLOW | | **76.31±0.09** | **93.27±0.10** | **59.02±0.09** | **83.08±0.10** | **77.35±0.51** | **93.75±0.04** | **60.21±0.16** | **83.99±0.05** |
| RANDOM | | 0.06 | 0.08 | 0.05 | 0.14 | 12.18 | 30.98 | 8.63 | 24.35 |
| SNIP | | 51.33±0.49 | 79.51±0.79 | 37.62±0.81 | 67.08±0.65 | 57.83±0.59 | 84.78±0.52 | 43.10±0.53 | 72.67±0.41 |
| ITERSNIP* | | 0.03±0.01 | 0.12±0.07 | 0.04±0.03 | 0.14±0.07 | 46.16±0.84 | 76.13±0.89 | 33.52±0.51 | 64.14±0.56 |
| OMP | 75% | 63.37±0.35 | 85.97±0.58 | 48.28±0.35 | 75.47±0.17 | 70.27±0.28 | 90.91±0.35 | 53.85±0.12 | 80.22±0.22 |
| LAMP* | | 2.10±3.57 | 5.93±10.03 | 1.46±2.49 | 4.75±8.06 | 69.38±0.28 | 90.97±0.38 | 53.15±0.43 | 79.96±0.30 |
| CHITA* | | 0.99±1.63 | 1.49±2.29 | 1.02±1.71 | 1.59±2.55 | 70.15±0.52 | 91.01±0.13 | 54.05±0.09 | 80.36±0.03 |
| CHITA++ | | 64.62±0.26 | 87.07±0.12 | 48.72±0.14 | 76.25±0.13 | 70.33±0.04 | 91.27±0.13 | 54.32±0.17 | 80.61±0.16 |
| MULTIFLOW | | **65.73±0.60** | **87.97±0.52** | **49.85±0.59** | **77.18±0.45** | **73.87±0.13** | **92.91±0.23** | **56.94±0.10** | **82.29±0.07** |

Table 1. Results for Image-Text Retrieval on COCO at 63% and 75% sparsity. The dense BLIP and XVLM upper bounds are reported on top. For further context, we include the random baseline as the lower bound. The **best performer** is bold; the second best is underlined. The superscript * denotes algorithms that perform comparably to the random baseline when pruning BLIP.

at all sparsity levels, and for all models. With XVLM, the gap between MULTIFLOW and the second best increases with the sparsity: at 63% sparsity, MULTIFLOW outperforms CHITA++ by +0.76% and +0.90% on TR@1 and IR@1 respectively, up to +3.54% and +2.62% when the overall sparsity increases to 75%. When pruning BLIP, MULTIFLOW remains the best performer while OMP becomes the second best method, outperforming CHITA++ at the lower sparsity. In general, performance drops with BLIP are much larger than those of XVLM, with 3 out of 7 methods (denoted by *) performing comparably to the random baseline. This hints at the following: *given a fixed task, different VLMs expose different "prunabilities"*. Beyond the comparison between MULTIFLOW and the state-of-the-art, these experiments unveil another important finding: *general purpose knowledge is retained by most methods*. This observation is signaled by the TR@5 and IR@5 metrics, where the performance drop is contained regardless of the algorithm.

## 5.2. Image Captioning (IC)

**Setup.** Image Captioning (IC) is the task of generating a text given a source image (and a prompt in VLMs). We employ COCO Captions to evaluate the pruned models for this task [13]. Following [39, 71], during fine-tuning both BLIP and XVLM optimize a language-modeling loss and use the prompt `"a picture of"`. For quantitative evaluation, we use the standard BLEU@4 [51] and CIDEr [64] scores. We report METEOR [3] and SPICE [2] in the Appendix.

**Results.** The outcome of this experiment is reported in Tab. 2. Notably, a pattern observed in ITR is confirmed also for IC: with XVLM, the gap between MULTIFLOW and the second best performing algorithm increases with the sparsity (+0.50% BLEU@4 and +2.46 CIDEr at 63% becoming +1.06% and +4.47 at 75% sparsity). When pruning BLIP, MULTIFLOW and CHITA++ perform equally well, with an edge for MULTIFLOW at 63% sparsity and the opposite at 75%. Importantly, we make a key observation: for both ITR and IC, *there is no fixed second best*. Different algorithms overcome each other when changing either the target model or sparsity. This further underlines the importance of designing *ad-hoc* methods for TA-VLP.

## 5.3. Visual Question Answering (VQA)

**Setup.** VQA requires the model to answer a question by analyzing the content of an associated image. The training set for the task collates the train and validation splits from the VQA2.0 dataset [25], and the image-question pairs from Visual Genome [31], as customary in the field. Given that generative capabilities are implicitly embodied by IC, with VQA we focus on the *analytical capabilities* of the pruned models. Thus, we evaluate closed-set VQA, and let pruned models choose from a predefined set of 3129 answers [38].

**Results.** We evaluate all methods against the official evaluation website and provide results in Tab. 2. VQA evaluation confirms the patterns: MULTIFLOW is the best performer in 3 out of 4 instances. Within these 4 instances, we emphasize the further absence of a fixed second-best method: (1) OMP is the second best with BLIP at 63% sparsity; (2) at the same level, CHITA and CHITA++ perform on par as the second best with XVLM; (3) at 75% MULTIFLOW lies less than one standard deviation below the best performer CHITA++ and (4) the latter underperforms the proposed method by −0.81% with XVLM. These results disclose a complementary finding to Sec. 5.1: not only VLMs, but also *vision-language tasks expose different prunabilities*. In the next section, we further investigate both phenomena.

| Method | Sparsity | $\text{BLIP}_{\text{BASE}}$ | | | $\text{XVLM}_{\text{CLIP}}$ | | |
|---|---|---|---|---|---|---|---|
| | | VQA | Image Captioning | | VQA | Image Captioning | |
| | | *test-dev* | BLEU@4 | CIDEr | *test-dev* | BLEU@4 | CIDEr |
| DENSE | 0% | 76.31 | 39.10 | 131.14 | 76.92 | 38.99 | 130.43 |
| RANDOM | 63% | 54.22 | 15.88 | 38.54 | 61.48 | 23.73 | 69.00 |
| SNIP | | 71.66±0.07 | 36.52±0.01 | 117.46±0.26 | 72.62±1.45 | 37.30±0.39 | 122.98±0.82 |
| ITERSNIP | | 63.26±0.01 | 26.57±0.10 | 75.80±0.26 | 72.92±0.14 | 36.52±0.16 | 120.12±0.52 |
| OMP | | 73.59±0.04 | 37.36±0.07 | 124.86±0.27 | 75.16±0.94 | 37.97±0.09 | 126.13±0.12 |
| LAMP | | 72.16±0.01 | 35.96±0.02 | 119.23±0.03 | 75.37±0.03 | 37.83±0.18 | 125.74±0.29 |
| CHITA | | 73.34±0.06 | 37.26±0.06 | 124.76±0.18 | 75.82±0.04 | 37.82±0.20 | 125.75±0.57 |
| CHITA++ | | 73.52±0.03 | 37.52±0.05 | **125.54±0.10** | 75.82±0.08 | 37.71±0.11 | 125.44±0.54 |
| MULTIFLOW | | **73.74±0.08** | **37.74±0.17** | 125.40±0.29 | **76.02±0.03** | **38.47±0.01** | **128.59±0.13** |
| RANDOM | 75% | 51.78 | 13.37 | 30.16 | 58.68 | 20.51 | 56.35 |
| SNIP | | 67.07±0.13 | 33.5±0.21 | 101.29±0.62 | 69.77±0.03 | 34.29±0.27 | 110.22±1.41 |
| ITERSNIP | | 52.28±0.10 | 14.35±0.23 | 29.88±0.28 | 67.11±0.08 | 31.15±0.32 | 98.24±1.74 |
| OMP | | 69.41±0.07 | 34.92±0.05 | 113.77±0.24 | 73.48±0.07 | 35.74±0.09 | 118.37±0.29 |
| LAMP | | 63.33±0.78 | 28.2±0.08 | 88.25±0.12 | 73.22±0.02 | 36.34±0.01 | 120.13±0.13 |
| CHITA | | 69.08±0.02 | 35.13±0.07 | 114.11±0.14 | 73.44±0.02 | 35.98±0.39 | 119.12±1.05 |
| CHITA++ | | **70.13±0.05** | **35.77±0.07** | **116.97±0.14** | 74.00±0.01 | 36.07±0.04 | 119.48±0.16 |
| MULTIFLOW | | 70.09±0.03 | 35.73±0.10 | 116.31±0.13 | **74.81±0.06** | **37.40±0.06** | **124.60±0.08** |

Table 2. Results on VQA2.0 and COCO Captions at 63% and 75% sparsity. The dense BLIP and XVLM upper bounds are reported on top. For further context, we include the random baseline as the lower bound. The **best performer** is bold; the second best is underlined.

# 6. Additional Analyses

## 6.1. Extreme sparsity and different prunability

In Sec. 5, we report results at 63% and 75% sparsity, revealing that different VLMs and vision-language tasks exhibit inherently different prunabilities. Here, we verify if these observations remain valid at 90% global sparsity, an extreme compression level where performance reliable for real-world applications has not yet been reached in VLM pruning. All of these experiments are depicted in Fig. 3.

**VLMs are not equally prunable**. Results in Fig. 3 convey a strong message: even if identically pruned and similar in total parameter counts, every pruned BLIP model underperforms the corresponding pruned XVLM although starting from a generally better or comparable dense performance. No method can produce meaningful results when pruning BLIP at 90% sparsity regardless of the task, with all methods failing in ITR (*i.e.*, $R@1 \leq 1\%$) and being almost on par with the random baseline in VQA and IC. We hypothesize that parameters integrating different modalities are a key aspect to preserve, and that the explicit disentanglement among vision, text, and fusion modalities within XVLM makes it less sensitive to parameter removal. Here, MULTIFLOW outperforms CHITA++, the average second best, by +6.93, +10.14, and +24.88 in VQA-acc, BLEU@4, and TR@1, respectively, while also being $41\times$ faster on average, (see Tab. 7 in Appendix C). We believe this is a great leap forward in pruning VLMs to extreme sparsities.

**Algorithms rank differently across settings.** The observation that *no fixed second best* is present at 63% and 75% sparsities also remains valid at 90%. For example, LAMP outperforms SNIP in ITR, while the opposite happens in IC. This hints at the fact that pruning algorithms preserve different types of encoded knowledge and further underlines the effectiveness of the task-agnostic design of MULTIFLOW which steadily maintains good performance across tasks, outperforming all competitors in all tasks for XVLM.

**VL tasks can be ranked by difficulty.** The *prune-then-transfer* paradigm leads to extreme differences in how performance drops according to the downstream task. With XVLM, the gaps between the dense upper bound and the best performer MULTIFLOW are $\downarrow$ TR@1 $\sim$ 40% and $\downarrow$ IR@1 $\sim$ 43% for ITR, but jump to only $\downarrow$ *test-dev*-acc $\sim$ 12% and $\downarrow BLEU@4 \sim 20\%$ for VQA and IC. While this may highlight different sensitivities among metrics, it also enables ranking the tasks as VQA $<$ IC $<$ ITR in terms of their difficulty, hinting that a good algorithm for TA-VLP shall emphasize image-text alignment.

## 6.2. Ablations and sanity checks on MULTIFLOW

In this section we perform two ablation studies of MULTIFLOW, testing the performance of the inverted mask, and ablating the impact of the imposed distribution.

**Inversion.** Inspired by [19], we check if the saliency function at the core of a score-based pruning algorithm effectively extracts the most important weights, by inverting the mask and maintaining the parameters with the lowest scores. Tab 3 shows that the score in MULTIFLOW successfully does so: inverting the pruning mask even extracts subnetworks that perform far *worse* than the random baseline.

**Imposed distribution.** Here we test the impact of the imposed multimodal prior described in Sec. 4.2 on MULTIFLOW, replacing it with two variants (i) not imposing a prior distribution on the layer-wise pruning ratios (*i.e.*, the mask is computed taking the top-k global scores), (ii) determining the layer-wise pruning ratios according to the weight

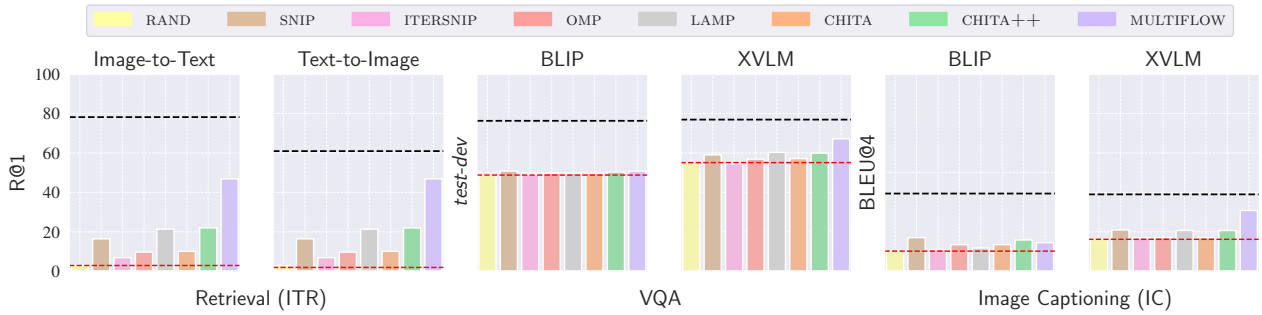


Figure 3. Experiments at 90% sparsity. ITR with XVLM (left) - VQA with both BLIP and XVLM (center) - IC with both BLIP and XVLM (right). The **random** and **dense** baselines are also reported. All experiments follow the same configuration as those of Tabs. 1 and 2.

| Ablation | Sparsity | Text R@1 | Image R@1 |
|---|---|---|---|
| RANDOM | | 21.48 | 14.89 |
| MULTIFLOW w/ Inversion | | 7.15 | 5.16 |
| MULTIFLOW w/o distribution | 63% | 0.01±0.01 | 0.02±0.0 |
| MULTIFLOW w/o *multimodality* | | 76.66±0.09 | 59.96±0.20 |
| MULTIFLOW | | 77.35 | 60.21 |
| RANDOM | | 12.18 | 8.63 |
| MULTIFLOW w/ Inversion | | 1.91 | 0.88 |
| MULTIFLOW w/o distribution | 75% | 0.02±0.0 | 0.01±0.01 |
| MULTIFLOW w/o *multimodality* | | 72.50±0.05 | 55.75±0.12 |
| MULTIFLOW | | 73.87 | 56.94 |
| RANDOM | | 2.92 | 1.93 |
| MULTIFLOW w/ Inversion | | 0.43 | 0.20 |
| MULTIFLOW w/o distribution | 90% | 0.02±0.0 | 0.03±0.01 |
| MULTIFLOW w/o *multimodality* | | 35.55±0.26 | 25.00±0.06 |
| MULTIFLOW | | 46.87 | 34.77 |

Table 3. Ablation study on ITR and MSCOCO (XVLM). Both the inversion and the imposed distribution studies are reported.

magnitude. These distributions are displayed in Fig. 4 for different parts of the model (*i.e.*, vision, text, fusion), and with corresponding performance on ITR in Tab. 3. As we can see from Fig. 4, without imposing any prior distribution (blue line), the model would either heavily prune early layers (*e.g.*, vision) or full modalities (*e.g.*, text), resulting in performance even worse than random due to model collapse (Tab. 3). The magnitude-based distribution (*i.e.*, w/o multimodality) already recovers this effect, with a more distributed pruning across layers and modalities. However, including multimodal priors leads to the best results, especially at 90% sparsity (*e.g.*, +11% on TR@1, +9% on IR@1). This experiment discloses two final findings: (i) considering potential biases in activation patterns among layers and modalities is fundamental for VLM pruning and (ii) in high-sparsity regimes, a small shift in the layer-wise distribution can correspond to a large performance gap.

## 7. Conclusions

In this work, we formalized and addressed *Task-Agnostic Vision-Language Pruning* (TA-VLP). We proposed Multimodal Flow Pruning (MULTIFLOW), an approach that preserves the information flow within the original VLM by exploiting a (multimodal) prior on the weight magnitude for layer-wise pruning, and by incorporating the saliencies of input/output nodes into the scoring criterion for network parameters. We also benchmarked 8 pruning methods for TA-VLP, using two VLMs, three vision-language tasks, and different pruning ratios, showing that MULTIFLOW outperforms existing state-of-the-art methods in the vast majority of the cases. While our results highlight a large gap between the dense model and its pruned counterpart, this work is a step toward finding task-agnostic pruned VLMs.

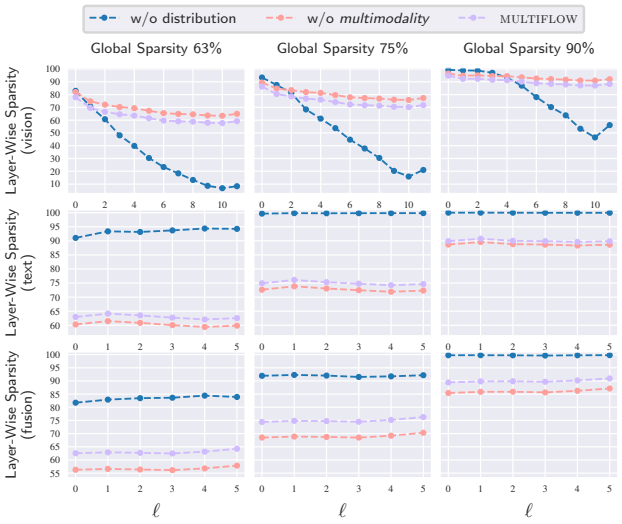


Figure 4. Comparison of the sparsities obtained at each layer $\ell$ of each modality by (i) pruning with the `topk` global scores of MULTIFLOW (denoted by *w/o distribution*), (ii) OMP (*w/o multimodality*) and (iii) MULTIFLOW. The figure displays XVLM.

**Acknowledgements.** We acknowledge the CINECA award under the ISCRA initiative for the availability of high-performance computing resources and support. E.R. and M.M. are supported by the MUR PNRR project FAIR - Future AI Research (PE00000013), funded by NextGeneration EU. E.R. is also supported by the EU projects AI4TRUST (No.101070190) and ELIAS (No.01120237). M.F. is supported by the PRIN project LEGO-AI (Prot.2020TA3K9N) and the PAT project "AI@TN". This work was also partially sponsored by a Cisco Research Grant.

## References


[1] Alizadeh, Milad and Tailor, Shyam A. and Zintgraf, Luisa M and van Amersfoort, Joost and Farquhar, Sebastian and Lane, Nicholas Donald and Gal, Yarin. Prospect Pruning: Finding Trainable Weights at Initialization using Meta-Gradients. In *International Conference on Learning Representations*, 2022. 2

[2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: Semantic Propositional Image Caption Evaluation. In *European Conference on Computer Vision*. Springer, 2016. 6, 3, 5

[3] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic evaluation measures for Machine Translation and/or Summarization*, 2005. 6, 3, 5

[4] Riade Benbaki, Wenyu Chen, Xiang Meng, Hussein Hazimeh, Natalia Ponomareva, Zhe Zhao, and Rahul Mazumder. Fast as CHITA: Neural Network Pruning with Combinatorial Optimization. In *International Conference on Machine Learning*, 2023. 2, 5

[5] Maxim Bonnaerens and Joni Dambre. Learned Thresholds Token Merging and Pruning for Vision Transformers. *Transactions on Machine Learning Research*, 2023. 2

[6] Mathilde Caron, Ari Morcos, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Pruning convolutional neural networks with self-supervision. *arXiv preprint arXiv:2001.03554*, 2020. 3

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1

[8] Chaoqi Wang and Guodong Zhang and Roger Grosse. Picking Winning Tickets Before Training by Preserving Gradient Flow. In *International Conference on Learning Representations*, 2020. 2

[9] Arnav Chavan, Zhiqiang Shen, Zhuang Liu, Zechun Liu, Kwang-Ting Cheng, and Eric P Xing. Vision transformer slimming: Multi-dimension searching in continuous optimization space. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 3

[10] Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. The lottery ticket hypothesis for pre-trained BERT networks. *Advances in Neural Information Processing Systems*, 2020. 2

[11] Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. *Advances in Neural Information Processing Systems*, 2021. 2

[12] Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Michael Carbin, and Zhangyang Wang. The lottery tickets hypothesis for supervised and self-supervised pre-training in computer vision models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16306–16316, 2021. 2

[13] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 5, 6

[14] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision*, 2020. 4

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 1

[16] Xin Dong, Shangyu Chen, and Sinno Pan. Learning to prune deep neural networks via layer-wise optimal brain surgeon. *Advances in Neural Information Processing Systems*, 2017. 2

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021. 2, 1

[18] Jonathan Frankle and Michael Carbin. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations*, 2019. 2, 4

[19] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Pruning Neural Networks at Initialization: Why Are We Missing the Mark? In *International Conference on Learning Representations*, 2021. 5, 7

[20] Elias Frantar and Dan Alistarh. SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot. In *International Conference on Machine Learning*, 2023. 2

[21] Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*, 2019. 2

[22] Trevor Gale, Matei Zaharia, Cliff Young, and Erich Elsen. Sparse gpu kernels for deep learning. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2020. 4

[23] Zhe Gan, Yen-Chun Chen, Linjie Li, Tianlong Chen, Yu Cheng, Shuohang Wang, Jingjing Liu, Lijuan Wang, and Zicheng Liu. Playing lottery tickets with vision and language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 1, 3

[24] Mitchell Gordon, Kevin Duh, and Nicholas Andrews. Compressing BERT: Studying the Effects of Weight Pruning on Transfer Learning. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, 2020. 3, 5

[25] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5, 6

[26] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *Advances in Neural Information Processing Systems*, 2015. 2, 4, 5

[27] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in Neural Information Processing Systems*, 2015.

[28] Babak Hassibi and David Stork. Second order derivatives for network pruning: Optimal brain surgeon. *Advances in Neural Information Processing Systems*, 1992. 2

[29] Ajay Jaiswal, Shiwei Liu, Tianlong Chen, and Zhangyang Wang. The emergence of essential sparsity in large pre-trained models: The weights that matter. *Advances in Neural Information Processing Systems*, 2023. 2, 5

[30] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 5

[31] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 2017. 5, 6

[32] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1

[33] Eldar Kurtic, Daniel Campos, Tuan Nguyen, Elias Frantar, Mark Kurtz, Benjamin Fineran, Michael Goin, and Dan Alistarh. The optimal BERT surgeon: Scalable and accurate second-order pruning for large language models. *Conference on Empirical Methods in Natural Language Processing*, 2022. 2

[34] Woosuk Kwon, Sehoon Kim, Michael W. Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir Gholami. A Fast Post-Training Pruning Framework for Transformers. In *Advances in Neural Information Processing Systems*, 2022. 2

[35] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in Neural Information Processing Systems*, 1989. 2

[36] Jaeho Lee, Sejun Park, Sangwoo Mo, Sungsoo Ahn, and Jinwoo Shin. Layer-adaptive Sparsity for the Magnitude-based Pruning. In *International Conference on Learning Representations*, 2021. 2, 5

[37] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. SNIP: Single-Shot Network Pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019. 2, 5

[38] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in Neural Information Processing Systems*, 2021. 1, 5, 6

[39] 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*, 2022. 1, 2, 5, 6, 3

[40] 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*, 2023. 1, 5, 4

[41] Youwei Liang, Chongjian GE, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. EViT: Expediting Vision Transformers via Token Reorganizations. In *International Conference on Learning Representations*, 2022. 2

[42] 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. In *European Conference on Computer Vision*. Springer, 2014. 5

[43] Yuanxin Liu, Fandong Meng, Zheng Lin, Peng Fu, Yanan Cao, Weiping Wang, and Jie Zhou. Learning to win lottery tickets in bert transfer via task-agnostic mask training. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022. 3, 1

[44] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *European Conference on Computer Vision*, 2018. 3

[45] Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. AdaViT: Adaptive Vision Transformers for Efficient Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2

[46] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, 2019. 4

[47] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning Convolutional Neural Networks for Resource Efficient Inference. In *International Conference on Learning Representations*, 2017. 2, 1

[48] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 4

[49] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics & Image Processing*, 2008. 1

[50] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2Text: Describing images using 1 million captioned photographs. *Advances in Neural Information Processing Systems*, 2011. 5

[51] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002. 6, 4

[52] Pau de Jorge and Amartya Sanyal and Harkirat Singh Behl and Philip H. S. Torr and Grégory Rogez and Puneet Kumar Dokania. Progressive Skeletonization: Trimming more fat from a network at initialization. In *International Conference on Learning Representations*, 2021. 2, 5

[53] 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*, 2021. 1, 5, 2

[54] 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*, 2021. 2

[55] Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing Rewinding and Fine-tuning in Neural Network Pruning. In *International Conference on Learning Representations*, 2020. 2

[56] Victor Sanh, Thomas Wolf, and Alexander Rush. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in Neural Information Processing Systems*, 2020. 4

[57] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018. 5

[58] Dachuan Shi, Chaofan Tao, Ying Jin, Zhendong Yang, Chun Yuan, and Jiaqi Wang. UPop: Unified and progressive pruning for compressing vision-language transformers. In *International Conference on Machine Learning*, 2023. 1, 2, 3

[59] Sidak Pal Singh and Dan Alistarh. WoodFisher: Efficient second-order approximation for neural network compression. *Advances in Neural Information Processing Systems*, 2020. 2

[60] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A Simple and Effective Pruning Approach for Large Language Models. *International Conference on Learning Representations*, 2024. 2, 4

[61] Tanaka, Hidenori and Kunin, Daniel and Yamins, Daniel L and Ganguli, Surya. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Advances in Neural Information Processing Systems*, 2020. 2, 4

[62] Yehui Tang, Kai Han, Yunhe Wang, Chang Xu, Jianyuan Guo, Chao Xu, and Dacheng Tao. Patch slimming for efficient vision transformers. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2

[63] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. *Advances in Neural Information Processing Systems*, 2024. 4

[64] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based Image Description Evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 6, 4

[65] Tiannan Wang, Wangchunshu Zhou, Yan Zeng, and Xinsong Zhang. EfficientVLM: Fast and accurate vision-language models via knowledge distillation and modal-adaptive pruning. In *Findings of the Association for Computational Linguistics: ACL 2023*, 2023. 1, 2

[66] Runxin Xu, Fuli Luo, Chengyu Wang, Baobao Chang, Jun Huang, Songfang Huang, and Fei Huang. From dense to sparse: Contrastive pruning for better pre-trained language model compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 2

[67] Huanrui Yang, Hongxu Yin, Maying Shen, Pavlo Molchanov, Hai Li, and Jan Kautz. Global Vision Transformer Pruning With Hessian-Aware Saliency. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 2

[68] Yite Wang and Dawei Li and Ruoyu Sun. NTK-SAP: Improving neural network pruning by aligning training dynamics. In *International Conference on Learning Representations*, 2023. 2

[69] Xin Yu, Thiago Serra, Srikumar Ramalingam, and Shandian Zhe. The combinatorial brain surgeon: pruning weights that cancel one another in neural networks. In *International Conference on Machine Learning*, 2022. 2

[70] Ofir Zafrir, Ariel Larey, Guy Boudoukh, Haihao Shen, and Moshe Wasserblat. Prune once for all: Sparse pre-trained language models. *Advances in Neural Information Processing Systems*, 2021. 2

[71] Yan Zeng, Xinsong Zhang, and Hang Li. Multi-Grained Vision Language Pre-Training: Aligning Texts with Visual Concepts. In *International Conference on Machine Learning*, 2022. 2, 5, 6, 3

[72] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *International Conference on Learning Representations, Workshop Track Proceedings*, 2018. 2

[73] Mingjian Zhu, Yehui Tang, and Kai Han. Vision transformer pruning. *arXiv preprint arXiv:2104.08500*, 2021. 2