Leaner and Faster: Two-Stage Model Compression for Lightweight Text-Image Retrieval

Siyu Ren   **Kenny Q. Zhu**[*]
Shanghai Jiao Tong University
Shanghai, China
roy0702@sjtu.edu.cn, kzhu@cs.sjtu.edu.cn

Abstract

Current text-image approaches (e.g., CLIP) typically adopt dual-encoder architecture using pre-trained vision-language representation. However, these models still pose non-trivial memory requirements and substantial incremental indexing time, which makes them less practical on mobile devices. In this paper, we present an effective two-stage framework to compress large pre-trained dual-encoder for lightweight text-image retrieval. The resulting model is smaller (39% of the original), faster (1.6x/2.9x for processing image/text respectively), yet performs on par with or better than the original full model on Flickr30K and MSCOCO benchmarks. We also open-source an accompanying realistic mobile image search application.

1 Introduction

Text-image retrieval is the task aiming at retrieving a list of relevant images from a large set of images given a textual query specified by the user. Recently, large-scale vision-language pre-training (VLP) has spawned models (Tan and Bansal, 2019; Li et al., 2020; Radford et al., 2021) that established state-of-the-art results in various vision-language tasks (Antol et al., 2015; Suhr et al., 2019), including text-image retrieval. Existing VLP models for text-image retrieval can be divided into two categories: cross-encoder architecture and dual-encoder architecture. Cross-encoder models show better retrieval accuracy by allowing fine-grained cross-modal attention among image and text. However, they are prohibitively slow to apply to the entire image pool because each image has to go through the deep Transformer again whenever a new text query comes in. Moreover, most cross-encoder models rely on external object detection models (Ren et al., 2015) to extract visual

---

[*] The corresponding author.

features, which further increase memory consumption. On the other hand, dual-encoder models are more scalable in that they allow pre-computing image representations as reusable vectors independent of the text queries. These image vectors can be indexed and efficiently retrieved at runtime using Approximate Nearest Neighbor (ANN) search (Johnson et al., 2017). As long as the image pool remains unchanged, the image encoder is not required.

However, a more practical scenario calls for dynamic indexing of new images into the pool (e.g., private photo collections on mobile devices), which requires both the image encoder and the text encoder to be resident in memory. This makes the above approach less practical on mobile devices with limited memory and processing power. Unfortunately, little attention has been paid to fulfill this need. In this paper, we show that a large dual-encoder model can be compressed into a much smaller and faster counterpart while retaining its retrieval accuracy using a novel two-stage compression framework. In the first stage, we make use of abundant non-paired texts/images to separately compress text or image encoder with an effective intra-modal contrastive knowledge distillation scheme. In the second stage, we sequentially fine-tune the distilled image or text encoder on paired text-image data with comprehensive learning objectives. Using CLIP (Radford et al., 2021) as the target model, our compressed models deliver comparable performance on MSCOCO and Flickr30K while being just 39% of the original size and 1.6x/2.9x times faster for processing image/text. Detailed ablation study shows the effectiveness of each component in the compression framework and their synergistic effects.

Our contributions are three-folds: 1) an effective compression framework tailored for lightweight text-image retrieval; 2) a leaner and faster model with competitive accuracy; 3) open-sourced models and text-to-image search mobile applications
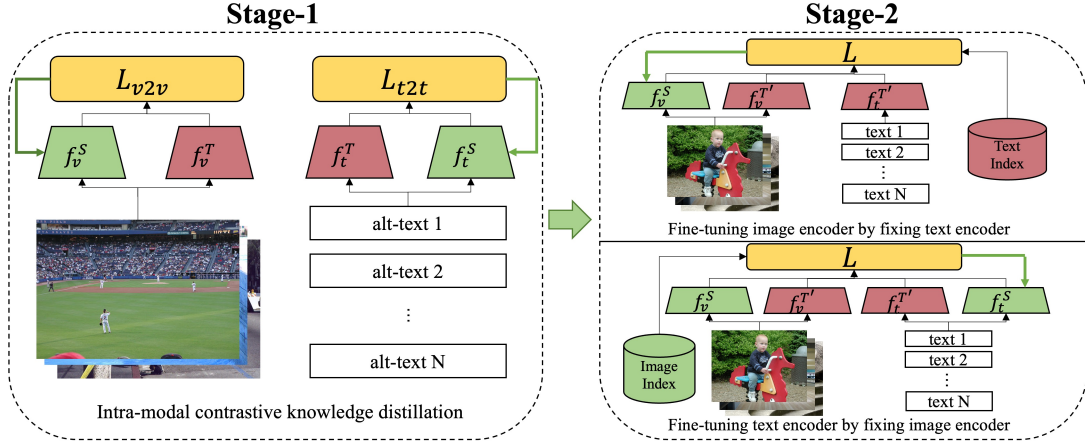
4085

Figure 1: In stage-1 (Section 3.2.1), we perform *intra-modal* contrastive knowledge distillation. In stage-2 (Section 3.2.2), we sequentially fine-tune f_v^S and f_t^S with knowledge distillation (KD) and corpus-level hard negative (HN) mining via pre-computed index. The total loss L is the sum of L_{t2v}, L_{v2t}, L_{KD}, and L_{HN}. The thin black arrows represent the input/output flows and the solid green arrows indicate the gradient flows.

on both iOS and Android at https://github.com/DRSY/MoTIS.

2 Related Work

Cross-encoder. Cross-encoder architecture (Tan and Bansal, 2019; Chen et al., 2019; Li et al., 2020) adopts a single Transformer network (Vaswani et al., 2017) which is able to process inputs from different modalities, e.g., images and texts. Benefitting from the self-attention mechanism, the hidden states of images and texts interact with each other at the patch/token-level, therefore yielding state-of-the-art retrieval accuracy. Though effective, these models suffer from huge memory consumption and inference latency, making them inpractical in time-sensitive real-world scenarios.

Dual-encoder. In contrast to cross-encoder, dual-encoder architecture (Radford et al., 2021; Jia et al., 2021) trains two seperate encoders for vision and language modalities. The exact choices of encoder architecture may be different. For example, CLIP utilizes Transformers for both visual and text encoders, while ALIGN (Jia et al., 2021) uses pre-trained BERT as text encoder and EfficientNet as visual encoder. In dual encoder, interactions between different modalities take place only at the final encoder layer, resulting in slightly worse performance compared to cross-encoders. Nevertheless, this late-interaction scheme of dual-encoder allows for efficient similarity computation, thus rendering it suitable for prividing real-time searching.

3 Approach

3.1 Background on Dual-Encoder

Dual-encoder architecture employs two separate neural networks to encode inputs from different modalities and map them to a shared space.

We denote the image encoder as f_v and the text encoder as f_t in the context of text-image retrieval. To train f_v and f_t, it is common to adopt an objective that pushes the embeddings of matched text-image pairs closer while pushing those of non-matched text-image pairs apart. Specifically, **C**ontrastive **L**anguage-**I**mage **P**retraining (CLIP) (Radford et al., 2021) optimizes an InfoNCE (van den Oord et al., 2018) loss:

$$\mathcal{L}_{t2v} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{f_t(x_i)^\top f_v(y_i)/\tau}}{\sum_{j=1}^{N}e^{f_t(x_i)^\top f_v(y_j)/\tau}} \quad (1)$$

Here, $f_t(x_i)$ and $f_v(y_j)$ are the L2-normalized embeddings of text in the i-th pair and image in the j-th pair. N is the mini-batch size and τ is the temperature to scale the logits. The final objective is the sum of L_{t2v} and its symmetric version L_{v2t}.

3.2 Two-Stage Model Compression

Despite good retrieval accuracy, models like CLIP still pose non-trivial memory footprint and inference time, which is undesirable for low-resource devices such as smart phones.

To tackle this issue, we propose a two-stage compression framework to make large dual-encoder model smaller and faster while retaining its accuracy. A schematic overview is illustrated in Figure

1. The first stage is *task-agnostic*, where we leverage massively available non-paired texts/images to separately compress the text/image encoder using an intra-modal contrastive knowledge distillation scheme. The second stage is *task-specific*, where we sequentially fine-tune the distilled image and text encoder using a combination of multiple techniques. We denote the image and text encoder of the large dual-encoder as f_v^T and f_t^T and those of the compressed model as f_v^S and f_t^S.

3.2.1 Stage-1

The extremely large scale of text-image pairs (e.g., 400 million used to train CLIP) makes it possible to make up for the noise in data and train over-parametrized large dual-encoder (i.e., f_v^T and f_t^T) from scratch to learn aligned visual and language representations. However, it is difficult to train small model (i.e., f_v^S and f_t^S) with lower capacity using the same *inter-modal* learning scheme.

To circumvent this issue, we propose to exploit massively available non-paired data from the web and optimize an *intra-modal* contrastive objective that aligns the output embeddings of f^S and pre-trained f^T, which can be seen as a form of knowledge distillation (Hinton et al., 2015). Here we take visual modality as an example. Given a collection of images $\{y_i\}_{i=1}^N$, we feed them to both f_v^S and f_v^T to produce two sets of image embeddings $\{f_v^S(y_i)\}_{i=1}^N$ and $\{f_v^T(y_i)\}_{i=1}^N$. Then we optimize the following contrastive objective for updating f_v^S:

$$\mathcal{L}_{v2v} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{f_v^S(y_i)^\top f_v^T(y_i)/\tau}}{\sum_{j=1}^N e^{f_v^S(y_i)^\top f_v^T(y_j)/\tau}} \quad (2)$$

The same formulation is symmetrically applied to language modality to obtain L_{t2t} for updating f_t^S:

$$\mathcal{L}_{t2t} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{f_t^S(x_i)^\top f_t^T(x_i)/\tau}}{\sum_{j=1}^N e^{f_t^S(x_i)^\top f_t^T(x_j)/\tau}} \quad (3)$$

Essentially, f_v^S/f_t^S is trained to recover the representation power of f_v^T/f_t^T in a decoupled manner.

3.2.2 Stage-2

After training f_v^S and f_t^S using general-domain data, it is necessary to adapt the learned representations to downstream tasks using in-domain data. First, we fine-tune f_v^T and f_t^T on paired text-image data $D = \{(x_i, y_i)\}_{i=1}^N$ using standard InfoNCE loss (Section 3.1). In the experiments, we found that jointly fine-tuning image and text encoder results in retrieval performance even worse than no fine-tuning at all. Therefore, we choose to sequentially fine-tune f_v^T/f_t^T by fixing the other one. The resulting fine-tuned encoders are denoted as $f_v^{T'}$ and $f_t^{T'}$ and are henceforth kept fixed. Next, for training f_v^S and f_t^S, we propose several techniques essential to successful compression:

Knowledge Distillation (KD). In addition to the standard InfoNCE loss, we design two kinds of knowledge distillation objectives to learn from $f_v^{T'}$ and $f_t^{T'}$. One is the Kullback-Leibler divergence between image-text matching distribution predicted by $f_v^{T'}$ and $f_t^{T'}$ and the one predicted by f_v^S and f_t^S. This resembles previous response-based knowledge distillation (Hinton et al., 2015). The other is the same contrastive objective defined in Section 3.2.1. It indirectly encourages the alingment between visual and language representations.

Sequential Finetuning (SF). Similar to how we get $f_v^{T'}$ and $f_t^{T'}$, we also fine-tune f_v^S and f_t^S in a sequential manner. Concretely, we first let the compressed model share the same text encoder with the target dual-encoder and only fine-tune its image encoder. After that, we then fix the image encoder and fine-tune its text encoder in the same way.

Hard Negative Mining (HN). Prior works on contrastive representation learning (Chen et al., 2020; Gao et al., 2021) typically exploit in-batch negative samples. Though efficient, image-text pairs in a batch are randomly sampled and are likely to be trivially unrelated. Models trained in such a way may fail in cases where candidates are similar. To achieve more accurate retrieval, we mine hard negatives from the entire corpus. In our sequential fine-tuning setting, we first use $f_t^{T'}$ to compute embeddings of all texts in the corpus and index them with Faiss (Johnson et al., 2017). During training f_v^S, for each image y_i we use $f_v^S(y_i)$ as query to the index and obtain its top-k texts as negative samples. Afterward, we use the trained f_v^S to compute embeddings of all images in the corpus and build the index. During training f_t^S, for each text x_i we use $f_t^S(x_i)$ as query to the index and get its top-k images as negative samples.

The complete training objective of stage-2 is defined as $\mathcal{L} = \mathcal{L}_{t2v} + \mathcal{L}_{v2t} + \mathcal{L}_{KD} + \mathcal{L}_{HN}$.

4 Experiment

4.1 Setup

Dataset. We use Conceptual Caption (Sharma et al., 2018) for stage-1 compression. It consists of 3M noisy image alt-text pairs. However, we do not

Image	Text	MSCOCO (1K)			MSCOCO (5K)			Flickr 30K		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
f_v^T	f_t^T	46.9	77.3	87.3	28.0	52.9	64.5	55.2	80.3	87.8
$f_v^{T'}$	$f_t^{T'}$	61.0	87.9	94.7	40.9	67.6	77.9	58.0	82.3	89.1
f_v^S	f_t^{TinyBERT}	41.4	76.7	88.1	21.3	47.2	61.0	30.2	59.1	71.2
f_v^S	$f_t^{S_4}$	62.0	88.0	94.4	42.0	69.2	79.0	55.0	81.3	88.4
f_v^S	$f_t^{S_6}$	**62.7**	**88.2**	**94.5**	**42.6**	**69.6**	**79.4**	**57.0**	**82.1**	**88.8**

Table 1: Comparisons of text-image retrieval results on MSCOCO (1K and 5K) and Flickr 30K.

Image	MSCOCO (5K)			Δ
	R@1	R@5	R@10	R@1
f_v^S	**36.7**	**64.6**	**75.3**	-
w/o stage-1	32.6	59.6	70.7	-4.1
stage-1$_{\text{MSE}}$	22.6	46.7	58.5	-14.1
stage-1$_{\text{InfoNCE}}$	31.7	58.5	69.6	-5.0
w/o SF	30.9	57.6	70.8	-5.8
w/o KD	35.8	63.1	74.2	-0.9
w/o HN	34.4	62.0	73.7	-2.3
w/o KD+HN	32.6	60.3	71.9	-4.1

Table 2: Ablation on design choices in both stages.

Image	Text	Disk Space MB	QPS$_v$ #	QPS$_t$ #
$f_v^{T'}$	$f_t^{T'}$	578	1.00x	1.00x
f_v^S	$f_t^{S_6}$	255	**1.51x**	1.98x
f_v^S	$f_t^{S_4}$	**230**	**1.51x**	**2.77x**

Table 3: Comparisons of disk space and QPS.

use the image-text alignment information but only treat it as a reservoir of general-domain images and texts. In stage-2, we use MSCOCO (Lin et al., 2014) and Flickr30K (Plummer et al., 2015) as the benchmarks. For MSCOCO, there are 113,287 images for training, 5,000 images for validation, and both 5K and 1K for testing. For Flickr30K, there are 28,783 images for training, 1,000 images for validation, and 1k for testing.

Evaluation Metrics. Following previous work, we use recall R@K (K=1,5,10) as the main metric of task performance. We also report the disk space (MB) and how many image/text queries can be encoded per second (QPS$_v$ for image and QPS$_t$ for text) to evaluate model's memory footprints and inference speed.

Target Model. We use the open-sourced ViT-B/32 CLIP as the target dual-encoder model to compress. The image encoder f_v^T is a 12-layer Vision Trans-

former (Dosovitskiy et al., 2020) with 768 hidden dimension and 12 attention heads. The text encoder f_t^T is a 12-layer Transformer with 512 hidden dimention and 8 attention heads. Note that this is the largest publically available version according to OpenAI's official repository.

Compression Configuration. For image encoder f_v^S, we use a ViT-S/16 with 384 hidden dimension. We initialize it with weights pretrained on ImageNet-21K (Ridnik et al., 2021)for faster convergence and better performance. For text encoder f_t^S, we experiment with both 6-layer and 4-layer Transformer (marked as $f_t^{S_6}$ and $f_t^{S_4}$), of which the weights are initialized from corresponding layers in f_t^T. We also compare with a baseline compression method that directly fine-tunes pretrained ViT-S/16 and 4-layer TinyBERT (Jiao et al., 2019) f_t^{TinyBERT} using InfoNCE loss throughout both stages.

Implementation Detail. In stage-1, we train 1 epoch using AdamW (Loshchilov and Hutter, 2017) with a batch size of 84 for both images and texts, learning rate of 3e-4, and weight decay of 0.1. In stage-2, we use the same optimization setting except that we train with batch size 96 for 5 epochs. We employ a cosine learning rate scheduler with 10,000 warm-up steps for both stages. All reported results are calculated on the test set using checkpoints with the highest validation performance.

4.2 Results

Main Results. Table 1 summarizes the main results. As can be observed, the pre-trained CLIP model can already deliver moderately good retrieval performance. The performance is further improved after fine-tuning. Fine-tuning pre-trained ViT-S/16 and TinyBERT underperforms the zero-shot CLIP, showing that training with *inter-modal* InfoNCE is not effective without extremely large-scale paired data. On most evaluation metrics, models compressed by our proposed two-stage pipeline

perform on par with or better than the fine-tuned target model. We also found that the capacity of text encoder has limited affect on the performance.

Ablation Study. We perform extensive ablations to study the importance of each proposed technique. Due to the computational budget, we only conduct ablation on the image encoder and fix the text encoder as $f_t^{T'}$. We evaluate *w/o stage-1*, *stage-1$_{MSE}$* (mean-square-error between f_v^S and f_v^T), and *stage-1$_{InfoNCE}$* (identical to the loss in Section 3.1) for stage-1 ablation. We also study the effectiveness of KD/SF/HN by removing them separately or together. We made several observations based on Table 2: 1) SF makes fine-tuning stable and is essential for convergence. 2) both KD and HN improve retrieval accuracy and are complementary to each other. 3) intra-modal contrastive distillation helps when image-text pairs are noisy and outperforms inter-modal infoNCE loss.

Efficiency. In Table 3, we compare the disk space and QSP used by models on a RTX 2080Ti of 12GB memroy. The compressed image encoder f_v^S takes 85MB disk space (39% of f_v^T) meanwhile being 1.51x times faster. Our compressed text encoder can achieve up to x2.77 inference speed-up and 40% size reduction (from 243MB to 146MB). We further benchmark models' memory and run-time performance on a real iPhone X with 1,000 images in the gallery for testing. It takes 870MB and 295MB for loading CLIP and our compressed model into main memory respectively. After indexing, the response time for a single text query is 0.4s for CLIP while it is only 0.1s for our compressed model. Although the results are hardward-dependent, our compressed model still shows an evident improvement in efficiency.

5 Conclusion

In this paper, we present a two-stage framework for lightweight text-image retrieval. Experiments on two benchmarks show the effectiveness of each component in the framework and the best performance is achieved when combining them together. It holds the merit of reducing model size and accelerating inference time, making memory/response-sensitive applications more practical.

Acknowledgement

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: visual question answering. *CoRR*, abs/1505.00468.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. UNITER: learning universal image-text representations. *CoRR*, abs/1909.11740.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *CoRR*, abs/2104.08821.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. *CoRR*, abs/2102.05918.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling BERT for natural language understanding. *CoRR*, abs/1909.10351.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. *CoRR*, abs/2004.06165.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2014. Microsoft coco: Common objects in context. Cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *CoRR*, abs/1505.04870.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Tal Ridnik, Emanuel Ben Baruch, Asaf Noy, and Lihi Zelnik-Manor. 2021. Imagenet-21k pretraining for the masses. *CoRR*, abs/2104.10972.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. 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)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy. Association for Computational Linguistics.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.